MOTOROLA

AAL-5 SAR Application Guide

C-WARE SOFTWARE TOOLSET, VERSION 2.3

CSTAA5-UG/D
Rev 01

Copyright © 2002 Motorola, Inc. All rights reserved. No part of this documentation may be reproduced in any form or by any means or used to make any derivative work (such as translation, transformation, or adaptation) without written permission from Motorola.

Motorola reserves the right to revise this documentation and to make changes in content from time to time without obligation on the part of Motorola to provide notification of such revision or change.

Motorola provides this documentation without warranty, term, or condition of any kind, either implied or expressed, including, but not limited to, the implied warranties, terms or conditions of merchantability, satisfactory quality, and fitness for a particular purpose. Motorola may make improvements or changes in the product(s) and/or the program(s) described in this documentation at any time.

C-5, C-Port, and C-Ware are all trademarks of C-Port, a Motorola Company. Motorola and the stylized Motorola logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

CONTENTS

ABOUT THIS GUIDE

Guide Overview

This document describes the design and features of the C-Ware AAL-5 SAR application (application identifier **oc3SarQ**).

This guide is intended for users of the C-Ware Software Toolset (CST) who want to build any application provided in the CST or who want to develop new C-Ware-based applications targeted to a C-Port network processor device.

This guide contains one chapter that covers the following major topics:

- Overview
- System Configuration
- Application Feature Overview
- Application Control and Data Flow
- Resource Utilization
- Supplied Application Files
- Alternate Application Configurations

DF Documents

Electronic documents are provided as PDF files. Open and view them using the Adobe® Acrobat® Reader application, version 3.0 or later. If necessary, download the Acrobat Reader from the Adobe Systems, Inc. web site:

http://www.adobe.com/prodindex/acrobat/readstep.html

PDF files offer several ways for moving among the document's pages, as follows:

- To move quickly from section to section within the document, use the *Acrobat bookmarks* that appear on the left side of the Acrobat Reader window. The bookmarks provide an expandable 'outline' view of the document's contents. To display the document's Acrobat bookmarks, press the 'Display both bookmarks and page' button on the Acrobat Reader tool bar.
- To move to the referenced page of an entry in the document's Contents or Index, click on the entry itself, each of which is "hot linked."
- To follow a cross-reference to a heading, figure, or table, click the blue text.
- To move to the beginning or end of the document, to move page by page within the document, or to navigate among the pages you displayed by clicking on hyperlinks, use the Acrobat Reader navigation buttons shown in this figure:

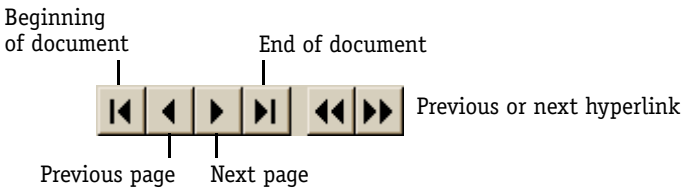


Table 1 summarizes how to navigate within an electronic document.

Table 1 Navigating Within a PDF Document

TO NAVIGATE THIS WAY	CLICK THIS
Move from section to section within the document.	A bookmark on the left side of the Acrobat Reader window
Move to an entry in the document's Contents or Index.	The entry itself
Follow a cross-reference (highlighted in blue text).	The cross-reference text
Move page by page.	The appropriate Acrobat Reader navigation buttons
Move to the beginning or end of the document.	The appropriate Acrobat Reader navigation buttons
Move backward or forward among a series of hyperlinks you have selected.	The appropriate Acrobat Reader navigation buttons

Guide Conventions

The following visual elements are used throughout this guide, where applicable:

This icon and text designates information of special note.

Warning: *This icon and text indicate a potentially dangerous procedure. Instructions contained in the warnings must be followed.*

Warning: *This icon and text indicate a procedure where the reader must take precautions regarding laser light.*

This icon and text indicate the possibility of electrostatic discharge (ESD) in a procedure that requires the reader to take the proper ESD precautions.

n History

Table 2 provides details about changes made for each revision of this guide.

Table 2 *Build System Conventions Guide* Revision History

REVISION DATE	CST REVISION	CDS REVISION	CHANGES
Novermber 1, 2002	2.2	2.0	Added documentation of Q-5 and host simulation support.
September 14, 2001	2.0	2.0	New document.

ıted Product umentation

Table 3 lists the documentation for the C-Ware library of reference applications.

Table 3 C-Ware Application Library Documentation Set

DOCUMENT NAME	PURPOSE	DOCUMENT ID
AAL-5 Fabric Port SAR to Gigabit Ethernet Switch Application Guide	Describes the key characteristics of the **gbeOc12SarFp** applications.	CSTAA5F2G-UG
AAL-5 SAR Application Guide	Describes the key characteristics of the **oc3SarQ** application.	CSTAA5-UG
AAL-5 SAR to Gigabit Ethernet Switch Application Guide	Describes the key characteristics of the **gbeOc12Sar** application.	CSTAA52G-UG
AAL-5 SAR to Gigabit Ethernet Switch with Q-5 TMC Application Guide	Describes the key characteristics of the **gbeOc12SarQ** application, which integrates the Q-5 TMC.	CSTAA52GQ-UG
ATM Cell Switch Application Guide	Describes the key characteristics of the **atmCellSwitchQ** application.	CSTAATMCS-UG
Differentiated Services Domain to IP Domain POS OC-12 Switch Application Guide	Describes the key characteristics of the **diffServPosOc12Q** application.	CSTDFSVQ-UG
FibreChannel to Gigabit Ethernet IP Gateway Application Guide	Describes the key characteristics of the **gbeFc** application.	CSTAFC2G-UG
Frame Relay to ATM to 10/100 Ethernet Switch Router Application Guide	Describes the key characteristics of the **switchRouter** application.	CSTAFRAE-UG
Gigabit Ethernet Switch Application Guide	Describes the key characteristics of the **gbeSwitch** application.	CSTAGBE-UG
Multi-PHY Switch Application Guide	Describes the key characteristics of the **mphySwitch** application.	CSTAMPHYS-UG
Packet Over SONET Switch Application Guide	Describes the key characteristics of the **posOc48Sc** application.	CSTAPOS-UG
Packet Over SONET to Ethernet Switch Application Guide	Describes the key characteristics of the **enetOc3Switch** application.	CSTAPOS2E-UG
Packet Over SONET to Gigabit Ethernet Switch Application Guide	Describes the key characteristics of the **posGbeSwitch** application.	CSTAPOS2G-UG
Voice Over IP to Voice Over ATM Media Gateway Application Guide	Describes the key characteristics of the **voIpToVoAtmSwitch** application.	CSTAVOIP-UG
Fabric Processor Configuration Component Guide	Describes the key characteristics of the **fabrics** application component.	CSTCFPC-UG
GMII Gigabit Ethernet Autonegotiation Component Guide	Describes the key characteristics of the **gmiiAutoNeg** application component.	CSTCGEAN-UG
ICMP Support Component Guide	Describes the key characteristics of the **ip** application component.	CSTCICMP-UG
MPC750 SBC Host Stack Support Component Guide	Describes the key characteristics of the **stackSupport** application component.	CSTCMHSS-UG

C-Ware Application Library Documentation Set (continued)

NT NAME	PURPOSE	DOCUMENT ID
PHY Configuration Component Guide	Describes the key characteristics of the **phy** application component.	CSTCPHYC-UG
QMU Configuration and RC Support Component Guide	Describes the key characteristics of the **queueUtils** application component.	CSTCQRCS-UG
SONET Monitoring Component Guide	Describes the key characteristics of the **sonet** application component.	CSTCSMC-UG

AAL-5 SAR Application Guide

Overview

This document is a functional and design specification for the **oc3SarQ** application in the CST.

This document goes into detail about the following topics:

- System Configuration
- Application Feature Overview
- Application Control and Data Flow
- Resource Utilization
- Supplied Application Files
- Alternate Application Configurations

For information on how to run the Oc3SarQ application see the Readme file in the apps\Oc3SarQ\doc directory.

Prerequisite Reading

Readers of this document are assumed to have read or be familiar with the topics in the following documents in the CST:

- *C-Ware Software Toolset Getting Started Guide* — How to get started with the CST.
- *Build System Conventions* — Description of how the build system works.
- *Q-5 TMC Functional Guide* — Describes detailed functionality of Q-5.
- *Q-5 TMC API User's Guide* — Includes a brief Q-5 functional overview as well as the Q-5 TMC API reference documentation.

System Configuration

This application does not run on the C-Ware Development System (CDS). For this release, the **oc3SarQ** application runs only in software simulation.

ation Feature Overview

The **oc3SarQ** application in the CST demonstrates a two-port OC-3c AAL5 SAR (segmentation and reassembly). CST2.2 and later releases include dynamic Q-5 configuration capabilities, which demonstrates configuration of the Q-5 using the Q-5 TMC API as well as communication between HOST and XP.

Feature Overview and Standards Support

This application supports the following functions:

- AAL5 support
- Round robin traffic scheduling Virtual Channel (VC) support
- Up to four thousand receive and transmit connections per port (Hardware only)
- Support for OC-3c rates
- Direct indexed 12 bit Virtual Path Identifier (VPI)/ Virtual Channel Identifier (VCI) combination TLU lookup
- Segmentation occurs on CPs 4 and 5 and reassembly occurs on CPs 6 and 7

ATM Protocol Model Review

ATM switches and edge devices use ATM, a connection-oriented technology, to forward cells over LANs and WANs. The fixed-size of the ATM cell allows for more predictable and faster traffic flow. Table 4 lists the ATM protocol stack and the functions that are performed at the different layers.

The AAL layer is only used at ATM connection endpoints.

The Physical Medium Dependent (PMD) sublayer provides the bit transmission clocking over the physical medium. The Transmission Convergence (TC) sublayer converts between the bit stream clocked to the physical medium and ATM cells.

A Header Error Check (HEC) occurs at the Transmission Convergence sublayer. This check corrects any single-bit error in the header and can also detect many patterns of multiple-bit errors. If errors are detected in the received header, then the cell is discarded.

The ATM Layer (layer 2 of the ATM model) performs functions similar to that of the Data Link layer (layer 2) of the OSI Reference Model. However, the ATM layer also has characteristics of the Network layer (OSI layer 3), such as hierarchical address space and a complex routing protocol.

Table 4 ATM Switching and ATM SAR Implementation Architecture

OSI Layer	ATM Layer		Functions Performed
Network	AAL	Convergence Sublayer	• Verify the Cyclic Redundancy Check (CRC) of the Protocol Data Unit (PDU) (AAL5 only). • Add and remove the 48 byte padding on the PDU (AAL5 only).
		SAR Sublayer	• Segment PDUs into cells. • Reassemble cells into PDUs.
Datalink	ATM		• Control traffic. • Generate and remove cell header. • Analyze and translate VPI/VCI. • Multiplex and demultiplex cells.
Physical	Physical	Transmission Convergence Sublayer	• Generate Header Error Check (HEC) on transmit and use HEC to detect and correct errors on receive. • Delineate cell on receive. • Adapt frame (map cells into time division multiplex formats) on transmit. • Decouple cell rate and transfer rate (speed-matching), which is required by synchronous media (SONET, DS3). • Generate and remove transmission frame. • Report Physical Layer status.
		Physical Medium Dependent Sublayer	• Clock bit transmission over physical medium. • Support the SONET STS-3c interface (on OC-3).

Typical point-to-point ATM links or interfaces are:

- **User Network Interface (UNI)** — The connection between the user equipment and the ATM equipment, such as a leased line between the customer site and the carrier access point.
- **Network Node Interface (NNI)** — The connection between ATM devices.

ATM UNI signaling is used between an ATM end-system and an ATM switch across an ATM UNI. ATM NNI signaling is used across NNI links.

ATM SAR Functionality Review

ATM Segmentation and Reassembly (SAR) involves segmenting data into 53Byte ATM cells for transmission and reassembling ATM cells into packets for upper layer processing. New for CST2.1 is that those SAR functions are now preformed by Q-5 traffic management coprocessor.The ATM Adaptation Layer (AAL) supports the convergence of different types traffic onto ATM. SARs, as they are commonly called, perform all or a subset of required AAL functionality and also allow upper layers to transport traffic which does not need AAL service to the ATM layer directly (Operations Administration and Maintenance (OAM) cells, Resource Management cells (RM-cells), and so on).

AAL Types

Table 5 lists the four AAL application types.

Table 5 AAL Types and Associated Protocols

AAL Type	Service Category				
	CBR	rt-VBR	nrt-VBR	ABR	UBR
AAL1	Circuit emulation, N-ISDN, Voice over ATM				
AAL2		Variable bit rate voice and video			
AAL3/4			ATM/SMDS		
AAL5	LANE 2.0, PNNI 1.0	Video on demand, LANE 2.0, PNNI 1.0	Frame Relay, ATM/SMDS, LANE 2.0, PNNI 1.0	LANE 2.0, PNNI 1.0	IP over ATM, LANE 1.0/2.0, PNNI 1.0, MPOA

Most SARs provide AAL5 service and allow raw cell traffic to be operated on by a management processor. There are also specialized SARs available that provide AAL1 support.

AAL5 Segmentation and Reassembly PDU Format

During an AAL5 SAR receive operation, ATM cells are segmented and then reassembled into a PDU as shown in Figure 1. During an AAL5 transmit operation, an upper layer Common Part Convergence Sublayer-Service Data Unit (CPCS-SDU) is padded with the number of bytes required to make an even number of ATM cells and the trailer (consisting of the Common Part Convergence Sublayer-User to User (CPCS-UU), CPI, Length, and CRC-32 fields) is added. Then the PDU is segmented into cells and transmitted.

Figure 1 CPCS-PDU Format for AAL5

Number of Bytes	1 - 65527	0 - 47	1	1	2	4
Field Name	**CPCS-PDU Payload (CPCS-SDU)**	**PAD**	**CPCS-UU**	**CPI**	**Length**	**CRC-32**

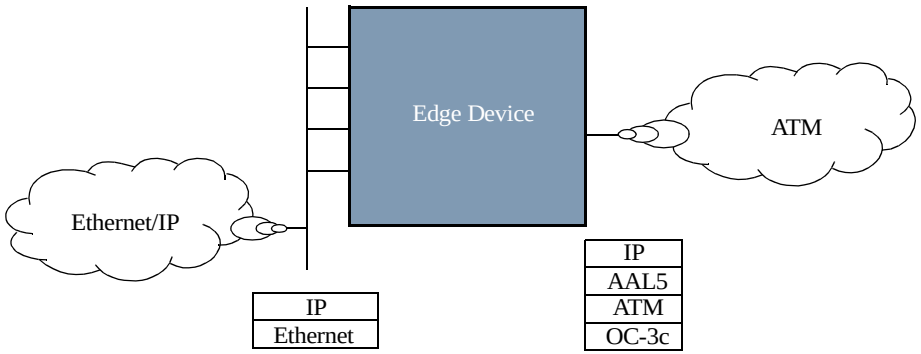
The AAL5 PDU trailer is described below:

- Common Part Convergence Sublayer User-to-User (CPCS-UU) — Allows a higher-layer protocol at the transmitter to send information to the receiver without being examined by the AAL5 layer.
- Common Part Indicator (CPI) — Interprets length field, a value of 0 means length in bytes. Non-zero values are undefined.
- Length — Length of the PDU (currently within the encapsulated PDU), allows the receiver to detect lost cells.
- Cyclic Redundancy Check (CRC) — 32-bit CRC of entire encapsulated PDU.

ATM SAR Example

SAR support is necessary in both ATM switches and edge devices in the network. An example of an edge device that requires SAR support is shown Figure 2.

Figure 2 AAL Support in an IP Switched Edge Device



tion Components Used

The CST provides a number of application components that are used across applications. This application uses the following application components provided in the CST:

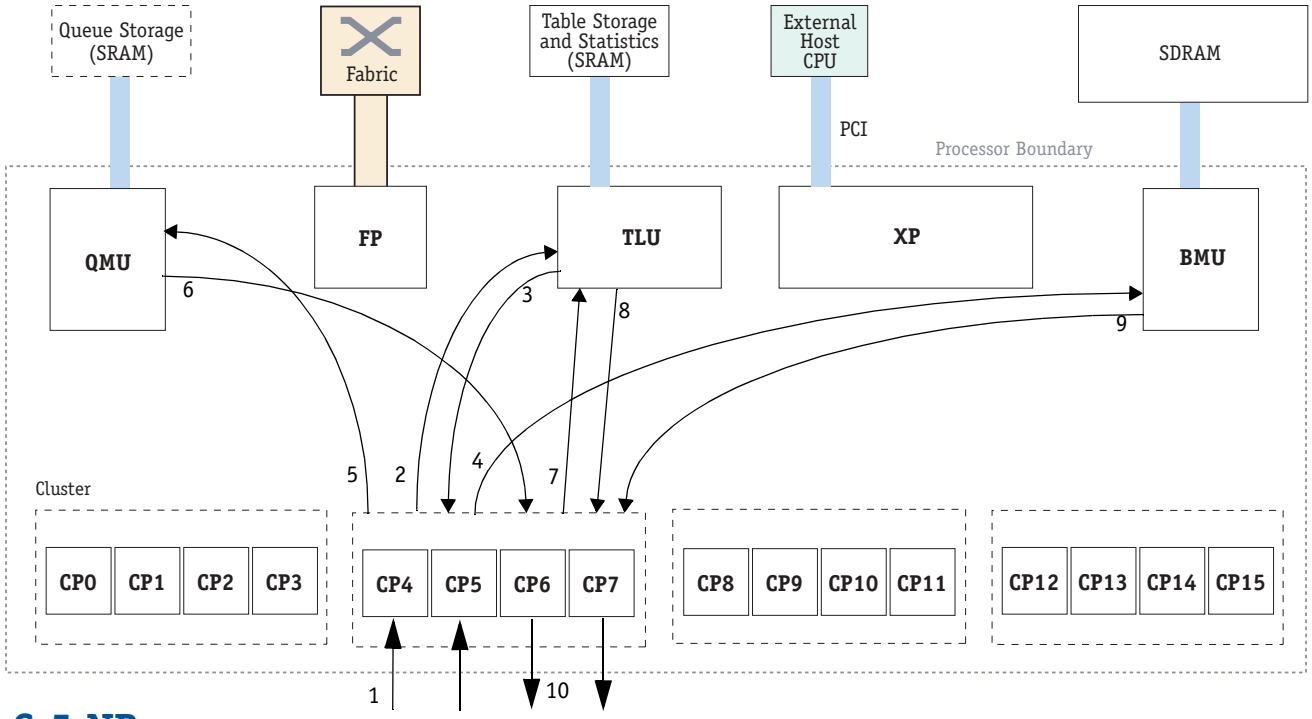
- **phy** (OC-3c Rx and Tx Bit processors)
- **sonet**
- **commSvc**

See the documentation in the **apps\components\<componentName>\doc** *directory for the documentation on the software components that this application uses.*

lication Control and Data Flow

Figure 3 shows the data flow through the C-5e processor during a typical transfer.

Figure 3 Data Flow Scenario



C-5 NP

1. RX SDP receives ATM cells
2. RX SDP launches lookup of ATM VC index, accumulates CRC using TLU XOR command
3. TLU sends VC index lookup response to RX CPRC
4. SDP/CPRC DMA's payload to buffer memory
5. CPRC enqueues descriptor to QMU (Q-5)
6. TX CPRC dequeues descriptor
7. CPRC launches lookup of VC index, SDP accumulates CRC using TLU XOR command
8. TLU sends TX VC index lookup response to CPRC
9. CPRC/SDP DMA's payload to buffer memory
10. TX SDP transmits ATM cells

rce Utilization

To support this application's features, the processors within the C-5 are performing a variety of tasks. The sections below enumerate what functionality is taking place and how the different processors are being used. For the **oc3SarQ** application, the C-5 components support:

- Processing of the data from the physical interface
- Identification of various cell types (OAM, RM)

XPRC

The XPRC is a general-purpose processor that provides the management, control, and exception processing functions. The XPRC controls C-5 boot up, configuration, and initialization of all of the system components.

The XP application is split into two sections. The first section performs service initialization, configures system resources, and loads the Channel Processors. The second section completes any necessary initialization and starts the Channel Processors. A hand-shaking mechanism is implemented to synchronize communication between the host and XP. This is also where the host configures the Q-5, after which the XP enters the main loop. This partitioning scheme uses the available IMEM resource to its fullest. The main application code is overlaid on the initialization section, reducing the amount of IMEM used during run-time. For more information see the C-Ware Application Design Guide document.

In the case of the **oc3SarQ** application, the XP is responsible for the addition of table entries to the TLU via the Ring Bus, passing of control and data information to and from the CPRC, and performing SONET defect soaking and monitoring.

CP

The C-5's 16 Channel Processors (CPs) are the components most closely associated with processing data from a physical interface. Each CP performs several functions that aid in the processing of ATM cells during Segmentation and Reassembly. This application uses the CPs for the following:

- Support for 2 OC-3 Rx interfaces and 2 OC-3 Tx interfaces
- Support for segmentation and reassembly
- Support for OAM and RM cells

CPRC

The Channel Processor RISC Core (CPRC) is a general-purpose processor core that supports a MIPS™ 1-like instruction subset. The CPRC program uses the C-Ware Communication Programming Interface (CPI) that provides a high-level abstraction of the C-5's function calls for ease-of-programming the C-5.

The CPRC program is used for the following functions in the OC-3c AAL-5 SAR application:

- Processing the lookup results from the TLU.
- Constructing a 'descriptor' for forwarding to the Q-5.
- Characterizing the frame and making forwarding decisions based on the TLU lookup results and frame parsing.
- Controlling the reassembly and DMA of the cells into a frame buffer, checking for length errors, checking the results of the final CRC32 TLU XOR command indicating whether the CRC passed.
- Controlling the segmentation and DMA of the cells from a frame buffer, providing the trailer information to the txSDP and obtaining the accumulated CRC-32 from the TLU, calculating the pad.
- Draining the single queue per transmit CPRC.
- Initiating receive program on the RxSDP
- Initiating transmit program on TxSDP

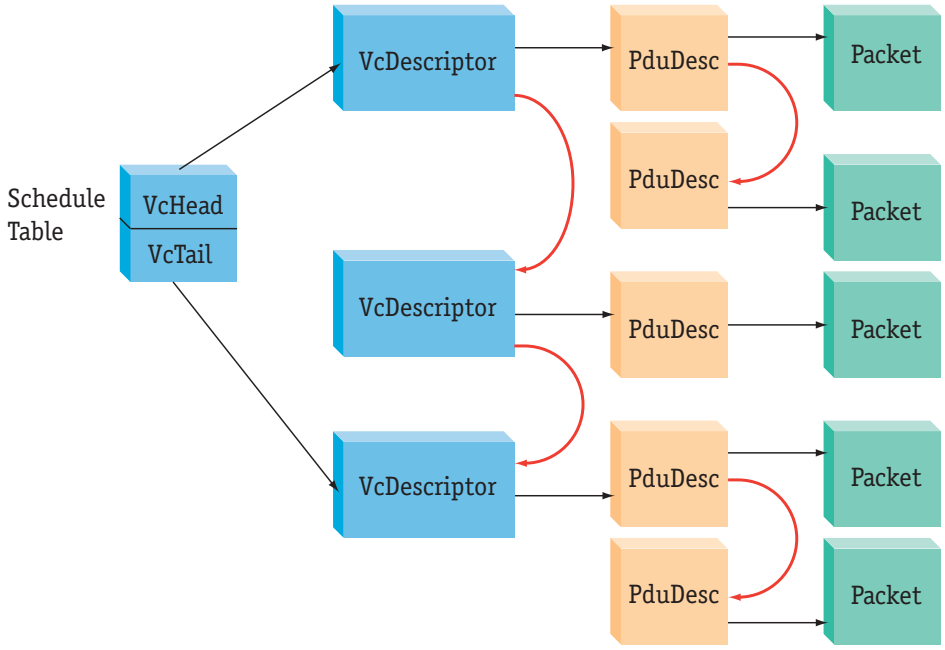
Transmit CPRC ATM Scheduling

The ATM application supports 64 simultaneous segmentations in round robin fashion for UBR support. To achieve true per-VC ATM scheduling at OC-3 rates, requires an external queue scheduler (Q5).

On transmit, descriptors which point to packets are dequeued from its transmit queue. The descriptor contains information about which VC this packet is on, the VC Index. This information is used to queue the packet to the appropriate VC queue.

Figure 4 shows how UBR traffic is queued on each currently active VC. VcDescriptors are maintained and managed in DMEM by the CPRC. The VcDescriptor has information about the connection. Presently, this consists only of the VC Index and a NextPtr as all connections are UBR. To support full traffic shaping in later releases, the Schedule Table will contain many entries and be indexed by the current time-slot.

Figure 4 UBR per-VC Packet Queuing with Round-Robin Support



CPRC/SDP Interface

The CPRC interfaces to the TxByte processor via a set of special purpose registers called Merge Space registers. The Merge Space registers are used to communicate information such as whether the cell to be transmitted is an end-of-message cell (EOM), how many bytes to pad and what the trailer information is to be added.

The CPRC and the RxByte processor share a set of registers to do table lookup processing over the Ring Bus. Lookup requests are referenced using a request tag, that refers to one of four Ring Bus registers assigned for initiating requests. Slots 0 and 1 are accessible to the SDP, while slots 0 through 3 are accessible to the CPRC. Each slot contains a 4Byte control register and two 4Byte data registers. These data registers contain information to be delivered to the TLU such as the type of lookup being requested and the key information to use in determining a match. Depending on the type of lookup being issued and the length of the key, the number of slots needed for any single lookup can be 1, 2, or 4.

The following request tags are used by the rxSDP in the OC3c AAL5 SAR application:

- Tag 0: VC lookup — Uses only slots 0 to support a 16 bit indexed VC lookup - read command.
- Tag 0: XOR VC Lookup— Uses both slots 0 and 1 to support a 16 bit key indexed XOR for the CRC32.

The following request tags are used by the receive CPRC in the OC3c AAL5 SAR application:

- Tags 2: VC Writeback— Uses both slots 2 and 3 to launch a write command.

Note that because the same request tag is used for multiple lookups, the application needs to check the AVAIL bit in the Ring Bus transmit control register to make sure the register is available for use by the program.

The following request tags are used by the txSDP in the OC3c AAL5 SAR application.

- Tag 0: XOR VC Lookup— Uses both slots 0 and 1 to support a 28 bit key indexed XOR for the CRC32.

The following request tags are used by the receive CPRC in the OC3c AAL5 SAR application:

- Tag 2: XOR VC Lookup on LAST— Uses both slots 0 and 1 to support a 16 bit key indexed XOR for the CRC32 on the LAST cell.
- Tag 2: VC Lookup— Uses slot 2 to launch a VC read command
- Tag 2: VC Writeback— Uses both slots 2 and 3 to launch a VC write command

The CPRC also has access to a set of Ring Bus receive control registers for checking lookup responses referred to as response tags. These response tags map to one of eight response slots available to CPRC, each of which contains a 4Byte control and 8Byte data portion. One response slot is therefore necessary for every eight bytes of data being returned from a lookup. When a lookup is initiated by either the SDP or the CPRC, the slot on which the response should land is indicated in the request. The status of the response is then checked by the CPRC using the appropriate response tag and the data returned is read upon receipt of a successful lookup indication.

The following response slots are assigned for receive processing in the OC3c AAL5 SAR application:

- Response Tag 0: VC Lookup Response — Uses slots 0,1,2 and 3 to support the return of 32 bytes of data.
- Response Tag 4: CRC XOR on LAST response — Uses slot 4 to support the return of 8 bytes of data.
- Response Tag 5: IP DA response — Uses slots 5 and 6 to support the return of 16 bytes of data.

The following response slots are assigned for transmit processing in the OC3c AAL5 SAR application:

- Response Tag 0: VC Lookup Response — Uses slots 0 and 1 to support the return of 28 bytes of data.
- Response Tag 4: CRC XOR on LAST response — Uses slot 4 to support the return of 8 bytes of data.

RxSDP

The Receive Serial Data Processor (RxSDP) performs bit- and byte-level interpretation and parsing of the incoming ATM cell stream encapsulated in OC-3c SONET frames. In addition to providing general receive processing, the RxSDP supports the following ATM and AAL5 -specific Switching functions:

- accessing the SPE from the SONET frame (including SONET frame synchronization pointer processing and frame error detection)
- Cell Synchronization
- OAM CRC-10 Validation
- ATM Cell Header parsing
- VC Lookup initiation
- Header Identification (OAM, EOM (End of Message), RM, idle/unassigned)
- AAL5 partial CRC lookup initiation
- AAL5 Trailer written to Extract Space

Initial RxSDP Processing

The initial processing of the incoming data stream involves:

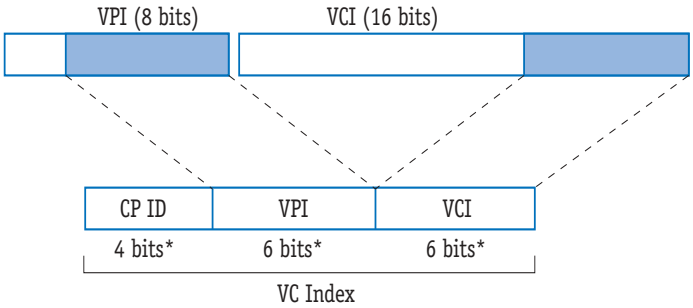
- Serial to Parallel Conversion — The OC3c bit-wide data received on the line is converted from serial to parallel format for processing by the RxBit microsequencer.
- SONET Frame Synchronization— .The rxBit microcode determines SONET frame synchronization on the A1 and A2 bytes of the SONET frame and asserts a 'frame_sync' pulse to the RxSonetFramer block. It maintains a state machine requiring two good frames to go in frame and 4 to go out of frame.
- SONET Overhead and Pointer Processing— The RxSDP implements the OC3c SONET frame processing in the rxSONET hardware block. This block interprets the SPE payload pointer, presents SONET defect and overhead information in registers accessible to the RC.
- Cell Synchronization— The RxSDP implements the cell delineation algorithm as specified in ATM Forum UNI 3.1 including the HUNT, PRESYNC and SYNC state machine. This processing is performed in the RxSync microsequencer. While in SYNC, the payload of idle/unassigned cells and bad HEC cells are dropped and an indication is provided to the RxByte processor.
- OAM CRC-10 Validation - All cells are checked for CRC10 errors by the RxSync microsequencer and forwarded with an indication of failure or success.

RxByte Processor

The SDP Receive Byte (RxByte) Processor is a microsequencer within the RxSDP. It is responsible for the byte-level processing of ATM cells from the physical interface. Specific ATM and AAL5 functionality includes the following:

- ATM Cell HeaderParsing — RxByte drops all idle/unassigned cells transparent to the RC. Additionally, it provides the VPI, VCI, PTI and CLP fields to the RC in extract space.
- VC Lookup Initiation— RxByte determines a 16 bit key composed of the port, VPI and VCI fields and launches a VC lookup to the TLU. The SDP calculates the VC Index by concatenating the six bits of the VPI with the six bits of the VCI and then adding the result to a CP-unique base. An example is shown in Figure 5.

Figure 5 VC Index Calculation



* The actual number of VPI and VCI bits is configurable.

The microcode issues a VC lookup to the TLU. Meanwhile the SDP is pulling in the cell data off the incoming port and storing it in SDP-internal FIFO memory and then sending it to SDRAM.

Table 6 Reassembly Extract Space Field Descriptions

Field Name	Size	Offset	Description
cellHeader	4	0	Outgoing ATM header to apply to cells
VccIndex	2	4	16bit entity containing the VccIndex used for the TLE request launched by the SDP
pduHdrStatus	1	6	GOOD_HEC_CELL 0x00 IDLE_UA_CELL 0x01 BAD_HEC_CELL 0x80
congestionDrops	1	8	16bit count of the number of PDUs dropped since last extract ownership (i.e. the number dropped by the RxSDP due to congestion on the part of the CPRC
crc10Indicator	1	9	Non-zero Indicates CRC10 error. Only valid for OAM/RM cells. Don't care for all other.
camValue	1	10	Value used to cam for -F4 OAM cells -F5 OAM/RM cells
encodedPti	1	11	Unused data
aal5Eom	1	12	End of Message indicator
aal5Uui	1	13	AAL5 User-User Indication
aal5Cpi	1	14	AAL5 Common Part Indicator (always 0)
aal5Length	2	15	AAL5 PDU length

Field Name	Size	Offset	Description
aal5Crc32	4	17	AAL5 CRC32
pduPayldStatus	1	21	Non-zero Indicates Error detected in PHY transport while payload being transferred
llc_1483	4	22	LLC from RFC 1483
snap_1483	4	26	OUI from RFC 1483
ethertype_1483	2	30	Ether type from RFC 1483

- Header Identification— RxByte identifies the following types of cells: OAM, EOM (End of Message), RM and provides this information to the RC via extract space. For field definitions refer to Table 3 and Table 4.
- AAL5 partial CRC32 Lookup Initiation— RxByte performs a partial CRC calculation on the payload of all assigned cells received and launches a TLU XOR command to accumulate the CRC in the TLU. For EOM cells, the RxByte processor launches a TLU XOR command with the LAST indicator set. A response is returned to the RC indicating that the CRC was valid or invalid.
- AAL5 Trailer Access— The RxByte processor provides the AAL5 trailer information to the RC in extract space (CPI, UUI, length, CRC32).
- Error and Status Reporting — The RxByte processor program reports errors and frame status to the CPRC via extraction space. This includes any physical layer error, CRC, HEC, header validation errors, and so on.

Table 7 IP Extract Space Field Descriptions

Field Name	Size	Offset	Description
ipControl	4	32	Not used in this application
version	1	36	IP version (should be 4)
IHL	1	37	IP header length
TOS	1	38	IP Type of Service field
pad	1	39	Reserved
length	2	40	IP total length
indentifier	2	42	IP identifier field
flags	1	44	IP flags
pad1	1	45	Reserved
fragOffset	2	46	IP fragmentation offset

Field Name	Size	Offset	Description
TTL	1	48	IP time to live
protocol	1	49	IP protocol field
checksum	2	50	IP checksum
sourceAddr	4	52	IP source address
destAddr	4	56	IP destination address
pad2[4]	4	60	Reserved

TxSDP

The Transmit Serial Data Processor (TxSDP) performs bit- and byte-level transmission of the outgoing OC3c SONET encapsulated ATM cell stream. In addition to providing general transmit processing, the TxSDP specifically supports the following AAL5 functions:

- Addition of the HEC on the ATM Cell Header
- SONET Frame and pointer generation
- Calculation of partial CRCs for the AAL5 Trailer
- Performs TLU XOR command for AAL5 CRC32 Accumulation.
- Addition of AAL5 Trailer and Padding based upon information from the RC
- CRC10 Addition for OAM Cells

TxByte Processor

The SDP Transmit Byte (TxByte) Processor is a serial processor within the TxSDP. It is responsible for the following functionality in the OC3c AAL5 SAR application:

- Transmits bytes — The TxByte processor reads bytes from CP DMEM and then loads the bytes into the large FIFO in order to stage transmission to the TxBit processor.
- Cell Header Generation - the ATM cell header is read from extract space and a valid HEC is generated.
- CRC32 generation — The AAL5 CRC is calculated and appended to the end of the AAL5 frame. on the last cell of the frame For cells which are not the last cell of the frame, the txSDP launches an XOR command with the partial CRC for the current cell.

Table 8 Segmentation Merge Space Field Descriptions

Field Name	Size	Offset	Description
cellHeader	4	0	Outgoing ATM header to apply to cells
VccIndex	2	4	16bit entity containing the VccIndex used for the TLE request launched by the SDP.
PayloadLength	1	6	Unused data
oamCell	1	7	Unused data
aal5Eom	1	8	End of Message indicator
pad[3]	1	9	Unused data
aal5Uui	1	10	AAL5 User-User Indication
aal5Cpi	1	11	AAL5 Common Part Indicator (always 0)
aal5Length	2	12	AAL5 PDU length
aal5Crc32	4	14	AAL5 CRC32
reserved[12]	12	18	Unused data

Additional TxSDP Processing

In addition to the processing performed by the TxByte Processor, the TxSDP performs the following OC3c SONET functions to support bit-level transmit processing of ATM cells in SONET frames:

- SONET Framing and Pointer Generation— The TxSONET block generates SONET frames and encapsulates ATM cells in these frames from the txByte processor.

TLU

The Table Lookup Unit (TLU) provides lookup table management. The TLU stores its table information in external SRAM that is managed by the TLU's SRAM controller.

The AAL5 SAR application creates 4 different tables in the TLU for data path forwarding support:

- A receive VC Lookup Table containing state information on the Virtual Connection
- An IP Routing Table for layer 3 IP forwarding
- A CRC Constant table used by the TLU to calculate the CRC32 for both receive and transmit
- A transmit VC Lookup Table containing information on the Virtual Connection.

Table 9 ATM VC Table Field Descriptions

Field Name	Size	Offset	Description
vcType	1	0	#define SAR_VC_TYPE_1577 0 #define SAR_VC_TYPE_NONE 1
rxPort	1	1	Receive OC3 port id which is also CP id
rxVpi	2	2	Receive port VPI
rxVci	2	4	Receive port VCI
txPort	1	6	Transmit OC3 port id which is also CP id
txVpi	2	7	Transmit port VPI
txVci	2	9	Transmit port VCI
txQ	4	11	Transmit Queue Id

Table 10 IP Forwarding Table Field Descriptions

Field Name	Size	Offset	Description
maskBits	1	0	Number of significant bits in the key
fabricId	1	1	Not used in this application
type	1	2	ipRouteTypeInvalid ipRouteTypeInternal ipRouteTypeLocal ipRouteTypeRemote
pad1	1	3	Reserved
appData1	4	4	Application specific data. In this application the data is outgoing VPI/VCI.
queueId	4	8	The egress queue id.
pad2	4	12	Reserved

BMU

The Buffer Management Unit (BMU) manages the C-5's data buffers for payload management. This application allocates one buffer pool per CP. The size of the buffers in each buffer pool is 16,384 bytes (the buffer sizes must be a power of 2).

The CPRC sets up packet transactions to SDRAM for packet payload that fire under software control. The BMU ensures that packet data that is copied into CP DMEM via the RxByte processor is correctly moved via a DMA transaction to the correct SDRAM buffer and offset for each port.

The AAL5 SAR Application performs BMU transfers differently from most other applications because it is required to transfer data which crosses 64 byte boundaries in

the BMU. These transfers require 2 DMAs to be initiated. Additionally, the buffer handle is not known before the scope is presented to the rxSDP and thus, the DMA must be monitored closely and setup on the fly by the RC.

QMU

The Queue Management Unit (QMU) manages off-chip descriptor queues that are used to provide QoS support to the **oc3SarQ** application. These descriptors are mapped one-to-one with VOPs in Q-5 configuration.

The descriptors are stored in external SRAM. The application allocates a 32 queues per ATM interface.

Table 11 ATM Cell Forwarding Descriptor

Field Name	Size	Description
bufHandle	4	Buffer handle of PDU being forwarded
vcIndex	2	Used to extract transmit VPI/VCI information
length	2	Length of PDU

FP

This application does not use the Fabric Port.

HOST

In CST2.2 a host component is added to simulate dynamic interactions between the Q-5 TMC and the C5e NP. There two new components added to the simulation environment to mimic the hardware environment. Besides **cxesim** which simulates C5e hardware, **pcisrv** is added to simulate PCI bus, and **pcihost** host is added to simulate a process running on the host (Solaris, Windows NT or Windows 2000).

Host Processor Interaction

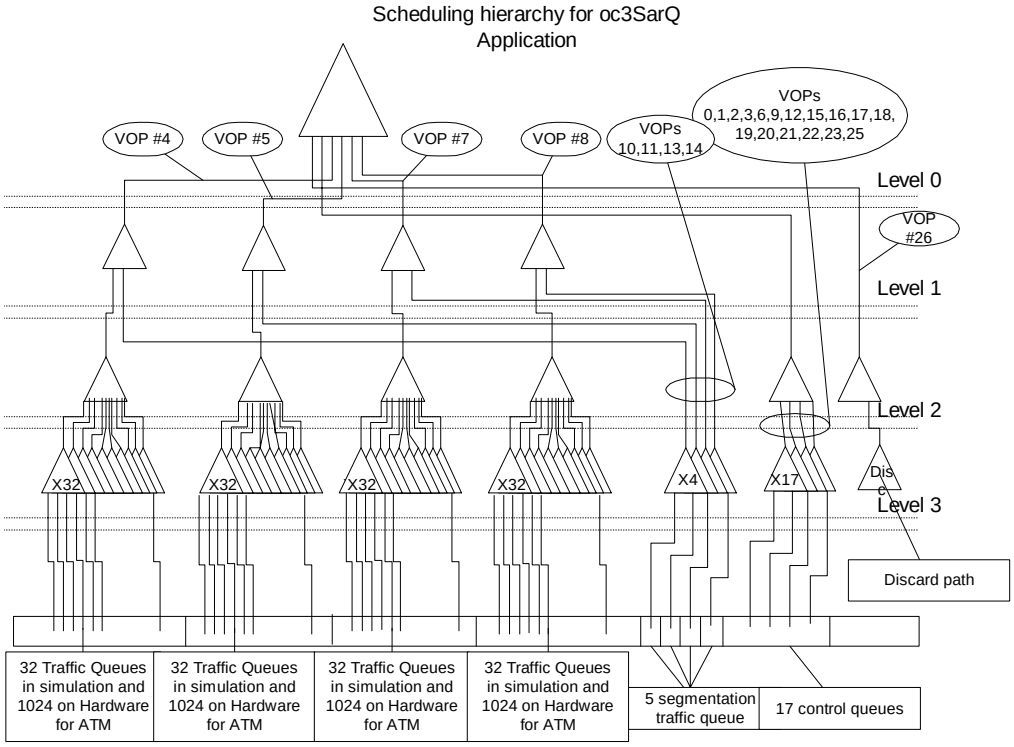
In addition to what is included in the C-5 driver, starting with CST 2.2 this application includes a host component. This component is strictly used for configuration on the Q-5. Because the application runs only in simulation, the hosts supported are Solaris and Windows NT or 2000. This application also make use of offline table building in place of host table functionality.

Q-5 Configuration

With the implementation of the Q-5 TMC API, the **oc3SarQ** configuration is changed to take into account the requirements of Q-5 API. For example, the flow of configuring queues on the XP must match the flow of configuring VOPs on the host. The simulation environment also uses the scaled version of actual hardware configuration. This is mainly

to reduce simulation time that is significantly increased with addition of the host. The host configuration features the following:

- 32 Traffic Queues per ATM port (4*32 = 128 TQs) in simulation
- 1024 Traffic Queues per ATM port (4*1024 = 4096 TQs) on hardware
- 128 Level 3 schedulers for ATM traffic (32 schedulers per port)
- 17 control queue Level 3 schedulers and 4 for segmentation and reassembly
- 6 Level2 schedulers and 4 Level1 schedulers
- 26 VOPs that map to 26 configured queues onthe XP (this is one-to-one correspondence)



Offline Table Building

A host processor maintains many of the tables that are used in the forwarding path. In order to simulate this, a Windows NT application is provided that builds tables, and adds and deletes entries while logging the activity to a file. A Perl script is supplied that can be used to translate the log file into an array of TLU SRAM entries defined in a **tle_writes.h** header file. The AAL5 SAR application uses the array defined in the header file to initialize the TLU as though a host processor had created the tables and inserted some entries. For additional information refer to **Table Building** section at the end of this document.

ed Application Files

Below is a list of the files that are a part of this application and a brief description of their contents.

Interfaces

Files that are used across processors are as follows:

File name	Location	Description
ipIf.h	.../c5/inc/	Defines IP interfaces between processors.
ipTableIf.h	.../c5/inc/	Defines structure of IP table.
q5Config.h	.../c5/inc/	Defines configuration needed by the Q-5.
sarIf.h	.../c5/inc/	Defines SAR interfaces between processors.

XPRC

Files that run on the XPRC are as follows:

File name	Location	Description
sarTableXp.h	.../c5/xprc/inc/	Contains definitions for creating the all the SAR tables.
tle_restore.h	.../c5/xprc/inc/	Offline table building services initialization data support.
sarTableXp.c	.../c5/xprc/src/	Contains SAR table building code.
xpMain.c	.../c5/xprc/src/	Main program and entry point for this application.
xpMainInit.c	.../c5/xprc/src/	First initialization phase and entry point for this application.

CPRC

Files that run on the CPRC are as follows:

File name	Location	Description
sarCpMain.h	.../c5/cprc/inc/	Contains definitions for SAR global variables.
sarRxCp.h	.../c5/cprc/inc/	Contains definitions for SAR receive processing.
sarRxInitCp.h	.../c5/cprc/inc/	Contains definitions for SAR RX initialization code.
sarTxCp.h	.../c5/cprc/inc/	Contains definitions for SAR transmit processing.
sarTxInitCp.h	.../c5/cprc/inc/	Contains definitions for SAR TX initialization code.
cpMain.c	.../c5/cprc/src/	SAR main program on the CPRC
sarRxCp.c	.../c5/cprc/src/	SAR receive routine.
sarRxInitCp.c	.../c5/cprc/src/	SAR RXinitialization routine.
sarTxCp.c	.../c5/cprc/src/	SAR transmit routine.
sarTxInitCp.c	.../c5/cprc/src/	SAR TX initialization routine.

SDP

Files that run on the SDP are as follows:

File name	Location	Description
ipFilterParse.h	.../c5/sdp/inc/	Implements RxByte IP parsing.
rxByte.c	.../c5/sdp/src/	SAR RxByte cell handling ucode.
rxSync.c	.../c5/sdp/src/	SAR RxSync cell syncronization ucode.
txByte.c	.../c5/sdp/src/	SAR TxByte cell transmission ucode.

Some files that run on the SDP come from the common application components area and aren't located with the rest of the files for this application.

Host

Files that run on the host or offline for this application are as follows:

File name	Location	Description
offline.c	.../offline/	Offline table build used to support C-5e.
tables.c	.../offline/	Application-defined table creation routines used to support C-5e.
tables.h	.../offline/	Definitions for table support for C-5e.

Files that run on the host to configure Q-5 for this application are as follows:

File name	Location	Description
tmcConfig.c	.../host/np/src	Configuration file that contains crucial calls to Q-5 TMC API.
tmHostMain	.../host/np/src	The host main entry point.
tmcConfig.h	.../host/np/inc	Configuration defines necessary for configuration of Q-5.

Files common to all host-XP communications

File name	Location	Description
tmXpComm.c	.../components/commSvc/chip/xprc/src	Communication initialization routine for the XP.
tmXpComm.h	.../components/commSvc/chip/inc	Function definitions for the initialization routines.
tmcHostXpComm.h	.../components/commSvc/inc	Defines needed by both XP and the host.
tmcHostXpComm.c	.../components/commSvc/host/np/src	Handshaking routines for the host.

Binaries

Binary files used for this application are as follows:

File name	Location	Description
oc3SarQ.dsc	.../run/	Package descriptor file for this application that describes what MIPS and micro-sequencer images run on which processors for this application.
oc3SarQ.pkg	.../run/bin/$VARIANT/	Package file for this application.
oc3SarQ.sdp	.../run/bin/$VARIANT/	Application's SDP image.
oc3SarQXp.dcp	.../run/bin/$VARIANT/	XPRC main MIPS ELF file.
oc3SarQXp.map	.../run/bin/$VARIANT/	XPRC main memory map.
oc3SarQXpInit.dcp	.../run/bin/$VARIANT/	XPRC init phase MIPS ELF file.
oc3SarQXpInit.map	.../run/bin/$VARIANT/	XPRC init phase memory map.
oc3SarQCp.dcp	.../run/bin/$VARIANT/	CPRC SAR MIPS ELF file.
oc3SarQCp.map	.../run/bin/$VARIANT/	CPRC SAR memory map.
pcihost or pcihost.exe	..../run/bin/$VARIANT/	Host executable that communicates with C5e via "pciserv"

Simulation files

Simulation files needed for this application are as follows:

File name	Location	Description
config	.../run/	Software simulator configuration file used for this application.
MC.config	.../run/	BMU configuration file.
sim.in	.../run/	Software simulator input file used for this application.
Tlu.State	.../run/	TLU SRAM configuration file.

ate Application Configurations

This application can only run in a host-based configuration. It can also run either with or without the Fabric Port enabled. SONET mode can also be change to SDH per each build of the application by setting SDH flag.

Table Building

New starting with CST 2.1 “offline” component was introduced with hash try key table building capabilities.

Since now the table building is done offline the files generated are **tle_restore.h** and **Tlu.State**. **tle_restore.h** is incorporated into XPRC build, and **Tlu.State** is loaded during simulator initialization.