



**MOTOROLA**

**ATOM1 :**  
**MC68360 ATM Microcode**  
**User's Manual**

**Document Reference:**

Version	Comments	Release date
0.1	First Release	18th Aug., 1994
0.2	Updated User's Manual	1st Nov., 1994
1.0	Updated for CRC-32 and buffer scattering	21st. March, 1996

Motorola reserves the right to make changes without further notice to any product herein to improve reliability, function, or design. Motorola does not assume any liability arising out of the application or use of any product, circuit, or software described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such intended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and the Motorola logo\* are registered trademarks of Motorola.

\*  **MOTOROLA**

# TABLE OF CONTENTS

1.	INTRODUCTION .....	5
1.1.	Key Features of ATOM1 .....	5
1.2.	MC68360 SCCs .....	6
1.3.	Conventions .....	6
2.	ATOM1 OPERATION .....	7
2.1.	Number of Connections .....	7
2.2.	Cell Delineation .....	7
2.3.	AAL Frame Check Sequences .....	8
2.4.	PDH & SDH Physical Layer Signalling & OAM Functions .....	8
2.5.	Traffic Control .....	8
2.6.	Physical Interface .....	8
2.7.	Buffer Descriptor and Buffer Structures .....	8
3.	REGISTERS .....	9
3.1.	General Registers .....	9
3.1.1.	RISC Controller Configuration Register (RCCR) .....	9
3.1.2.	CP Control Registers .....	9
3.2.	SCC Registers .....	10
3.2.1.	General SCC Mode Register (GSMR) .....	10
3.2.2.	Protocol Specific Mode Register (PSMR) .....	10
3.2.3.	SCC Event Register (SCCE) .....	11
3.2.4.	SCC Mask Register (SCCM) .....	11
3.2.5.	SCC Status Register (SCCS) .....	12
3.3.	Serial Interface .....	12
3.3.1.	Serial Interface Registers .....	12
3.3.2.	Serial Interface RAM .....	12
3.4.	Parallel Port Registers .....	13
4.	ATOM1 SCC PARAMETER RAM .....	14
4.0.1.	BD Queue Pointers (RBASE and TBASE) .....	15
4.0.2.	SCC Function Code Registers (RFCR and TFCR) .....	15
4.0.3.	Transparent Mode Receive Maximum Buffer Length (MRBLR) .....	15
4.0.4.	SCC Internal State Parameters (RSTATE and TSTATE) .....	15
4.0.5.	BD and Buffer Pointers (R_PTR, RBD_PTR, T_PTR and TBD_PTR) .....	16
4.0.6.	Transmit and Receive Buffer Queue Start Pointers (TX_BUFF1, RX_BUFF1) ..	16
4.0.7.	Buffer Counters (R_CNT and T_CNT) .....	16
4.0.8.	Receiver Delineation Counters (ALPHA and DELTA) .....	16
4.0.9.	Version Number (AVERSION) .....	16
4.0.10.	CAM Port Selection (CAM_PORT) .....	16
4.0.11.	Empty Cell Data (EHEAD and EPAYLOAD) .....	16
4.0.12.	Status Information (ASTATUS) .....	17
4.0.13.	Non-Matching Header Storage and Counter (NMA_HEAD and NMA_CNT) ....	17
4.0.14.	HEC Error Counter (HEC_ERR) .....	17
4.0.15.	Incoming Header Mask and Look-up Table (HEADMASK and HEADn) .....	18
4.0.16.	Temporary Data Storage .....	18

5.	ATOM1 COMMANDS .....	18
5.1.	Restart Reception Command .....	19
5.2.	CRC-32 Calculator .....	19
6.	BUFFER DESCRIPTORS & BUFFERS .....	21
6.1.	ATOM1 Buffer Descriptor and Buffer Queues .....	21
6.2.	Receive Buffer Descriptor and Data Buffer .....	22
6.3.	Transmit Buffer Descriptor and Data Buffer .....	23
6.3.1.	Transmit Bandwidth Reservation .....	24
7.	ATOM1 CAM INTERFACE .....	25
8.	LOADING & CONFIGURING ATOM1 .....	27
9.	PERFORMANCE .....	27
9.1.	Command Execution Latency .....	29
9.2.	CRC-32 Calculation Execution .....	29
10.	ATOM1 DISK STRUCTURE .....	30
11.	MC68360 MASK SET APPLICABILITY .....	30
12.	ORDERING INFORMATION .....	31
13.	REFERENCES .....	31
Appendix 1.	STATE MACHINE DETAILS .....	32
	ATOM1 Transmitter .....	32
	ATM Cell Payload Scrambling .....	33
	ATOM1 Receiver .....	33
	HEC Delineation Mechanism .....	33
Appendix 2.	ATOM1 DEMONSTRATION SOFTWARE .....	36

## LIST OF TABLES

Table 1.	CP Control Registers .....	9
Table 2.	MC68360 Serial Interface Programming Example .....	12
Table 3.	Serial Interface RAM Programming Example .....	12
Table 4.	ATOM1 TDM A Port Pin Requirements .....	13
Table 5.	MC68360 Port Register Programming .....	13
Table 6.	ATOM1 SCC Parameter RAM Memory Map .....	14
Table 7.	ATOM1 CRC-32 Calculator Parameter RAM Memory Map .....	20
Table 8.	ATOM1 Performance .....	28
Table 9.	ATOM1 and other Protocol Performance .....	28
Table 10.	CRC-32 Calculator Performance .....	29

## LIST OF FIGURES

Figure 1.	Typical ATOM1 Application System .....	6
Figure 2.	ATOM1 RCCR .....	9
Figure 3.	ATOM1 PSMR .....	10
Figure 4.	ATOM1 SCCE .....	11
Figure 5.	ASTATUS .....	17
Figure 6.	HEADMASK .....	18
Figure 7.	HEADn .....	18
Figure 8.	ATOM1 Transmit Buffer and BD Queue Example .....	21
Figure 9.	ATOM1 Receive Buffer Descriptor .....	22
Figure 10.	ATOM1 Transmit Buffer Descriptor .....	23
Figure 11.	Transmit Buffer .....	24
Figure 12.	Mixed Voice and Data Transmission Example .....	25
Figure 13.	ATOM1 CAM Interface .....	26
Figure 14.	ATOM1 Disk Structure .....	30
Figure 15.	ATOM1 Transmitter Procedure .....	32
Figure 16.	ATM Cell Payload Scrambling Mechanism .....	33
Figure 17.	ATOM1 Receiver Procedure .....	34
Figure 18.	Cell Delineation State Diagram .....	35

# ATOM1

## Asynchronous Transfer Mode Microcode for the QUICC

### 1. INTRODUCTION

ATOM1 is an MC68360 RAM-based microcode that provides physical layer ATM functions by converting one or more of the QUICC's serial communication controllers (SCCs) into an ATM cell transmitter and receiver. ATOM1 provides the user with cell streaming facilities (cell reception and transmission) and event indications. The user can add AAL functions to ATOM1 using software.

The primary application of ATOM1 is intended to be G.804 plesiochronous digital hierarchy (PDH) and synchronous digital hierarchy (SDH) E1 and DS1 ATM equipment. Such equipment is used for signalling and low rate data transfer and may be part of the telecommunications infrastructure or terminal equipment.

Figure 1 shows an ATM communications module built around the QUICC running an ATOM1 microcode. This module could be a board within a larger system or part of a larger ATM board. The ATOM1 microcode runs on the QUICC's RISC communications processor (CP) and is stored in the dual port RAM. The user interface to ATOM1 is through buffer descriptors located in the QUICC's dual port RAM and data buffers located in external memory. The QUICC's CPU32+ controls the module and memory using the on-chip memory controller and other integration features.

#### 1.1. Key Features of ATOM1

The key features of ATOM1 are:

- Cell transmission and reception for all AAL protocols.
- Handles serial data rates up to 8 Mbps with a 25 MHz QUICC and up to 11 Mbps with a 33 MHz QUICC.
- Any or all of the QUICC's SCCs can simultaneously handle ATM cells.
- Transmit and receive data buffers located in main memory.
- On transmission, ATOM1 constructs the cell header and appends the user defined payload.
- Bandwidth reservation option in the transmit queue to allow mixing of data and isochronous services.
- On reception, ATOM1 verifies incoming cell headers and strips the HEC before passing the cell to the user.
- Automatic empty cell transmission when there are no pending data transfers.
- Receiver filters and discards empty cells and those with non-matching addresses.
- CAM support on reception for handling many connections.
- User controlled cell scrambling option.
- Incoming cells with incorrect HECs are received and marked as such.
- A CRC-32 calculator is available to reduce processor performance requirements when

handling AAL5 cells.

- Standard QUICC features are available when ATOM1 is running.

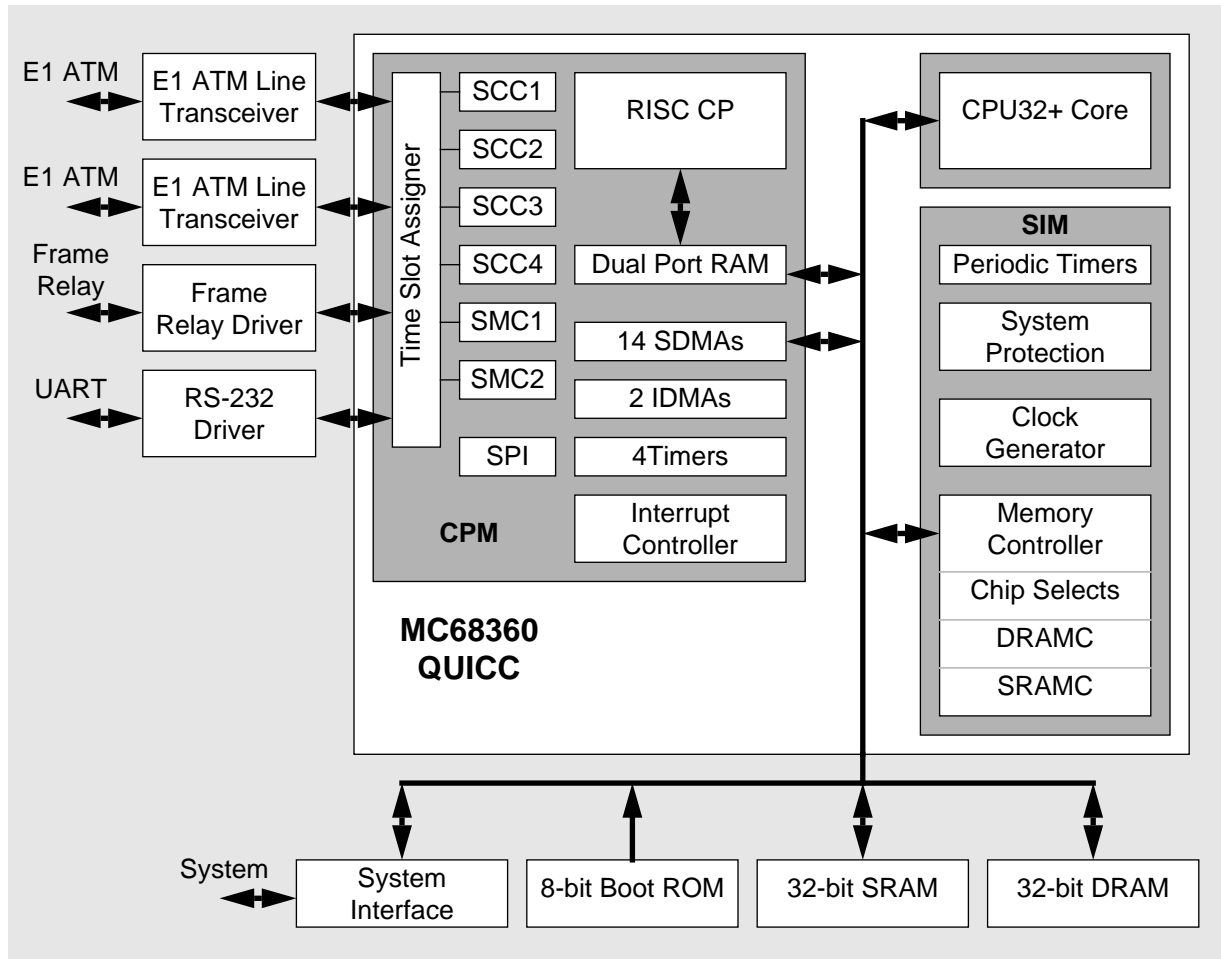


Figure 1. Typical ATOM1 Application System

## 1.2. MC68360 SCCs

ATOM1 can operate on any MC68360 SCC and can simultaneously operate on any two, three or all four SCCs. To maintain the performance of ATOM1, it is recommended that the highest speed connection operates on SCC1 and the lowest speed connection on SCC4.

ATOM1 uses the SCC in transparent mode with microcode performing ATM cell functions. When ATOM1 is loaded and RAM microcode is enabled, transparent mode operation is available on any SCC other than ones running ATOM1.

## 1.3. Conventions

The reader is assumed to be familiar with the QUICC and ATM. For further information about the QUICC, please refer to the MC68360 User's Manual. Full details of the ATM protocols can be found in the relevant ITU-T recommendations and the ATM Forum UNI document listed at the end of this manual.

In this manual, the term “user” is used to describe the CPU software that drives ATOM1. In the MC68360 this may be the CPU32+ or an external CPU.

## **2. ATOM1 OPERATION**

ATOM1 transmits and receives 53-byte ATM cells as defined in the ATM Forum UNI specification (ATOM1 also handles NNI cells). Cells are transmitted from a single queue and the user has full control over each cell's GFC, VPI, VCI, PT, CLP and payload fields. ATOM1 generates and inserts the cell HEC field on transmission and checks and strips it on reception. An option is included to apply the HEC coset pattern (binary 01010101) to transmitted and received cells.

ATOM1 performs cell rate adaptation to the capacity of the transmission link by the inserting empty cells when no cells are queued by the user. On reception, ATOM1 ignores empty cells. The user can define the empty cell to be either an unassigned or idle cell.

Received cells with HEC errors are received and marked as such. The HEC is passed to the user to allow user software to implement HEC correction algorithms as required. ATOM1 does not attempt to correct any HEC errors.

ATOM1 provides an option to scramble data on transmission and receive scrambled data using the  $X^{43}+1$  scrambling algorithm. The first cell to be transmitted by ATOM1 after initialisation will not be correctly scrambled because there is no valid data in the 43-bit delay line. ATOM1 always transmits an empty cell first therefore avoiding data corruption. On reception, the descrambling algorithm self-synchronises before the HEC delineation process is complete and cell reception begins.

### **2.1. Number of Connections**

ATOM1 transmits one cell at a time on the virtual connection specified for that cell. The user can transmit ATM cells with the whole range of the cell's address field (VCI and VPI combined).

ATOM1 receives cells after matching the cell header with a header look-up table or an external CAM. The internal look-up table has up to 16 entries and cell headers can have bits masked before the comparison.

The user must maintain enough open channels to receive OAM and signalling cells. User software must handle OAM and signalling cells.

### **2.2. Cell Delineation**

The user must provide synchronisation signals to the QUICC and octet-align incoming cells to the synchronisation signals. ATOM1 provide SDH/PDH oriented cell delineation using the HEC mechanism defined in I.432 on an octet basis. The synchronisation signals need not delimit the start of cell but must be octet aligned with incoming cells. When using E1 and T1 ATM links, the cells are always octet aligned (see ITU G.804) and synchronisation signals are provided by the E1 and T1 interface devices.

When reception commences, ATOM1 takes a short while to acquire correct cell delineation. Once ATOM1 has locked to the incoming cell stream, it remains locked unless there are excessive errors. An interrupt is generated whenever the cell lock status changes and a status bit will indicate the current delineation status.

Further details of the cell delineation mechanism are given in Appendix A.

### **2.3. AAL Frame Check Sequences**

To increase software throughput of AAL5 frames, a CRC-32 calculator is included. This is not an integral part of the cell transmission and reception process in order that CRC-32 sequences may be calculated across long AAL5 frames made up of interleaved ATM cells. The CRC-32 calculator is a command which ATOM1 interprets by working through a data buffer generating the required CRC-32 sequence.

### **2.4. PDH & SDH Physical Layer Signalling & OAM Functions**

PDH and SDH framing physical layer signalling and OAM functions are not provided by ATOM1 but the user can handle these by routing the PDH/SDH OAM, signalling and overhead timeslots to one of the QUICC's SMCs or another SCC.

### **2.5. Traffic Control**

ATOM1 transmits cells from its single transmission queue in the order that they are presented. It is the user's responsibility to apply traffic management techniques to avoid congestion and breaking the UPC rules if the connection data rate is less than the transmit bit rate. Failure to apply these techniques may result in bursty traffic on open connections.

When ATOM1's transmit queue is empty, ATOM1 will automatically transmit empty cells.

### **2.6. Physical Interface**

The ATOM1 microcode is independent of the physical interface standard used. Generally, one of the QUICC's TDM ports will be used with ATOM1 to allow easy connection to an E1 or T1 line interface device. Other serial interfaces are not precluded and may be selected when programming the MC68360 serial interface. This User's Manual describes the TDM interface using the Time Slot Assigner (TSA).

### **2.7. Buffer Descriptor and Buffer Structures**

ATOM1 uses an optimized version of the standard MC68360 data buffer and buffer descriptor (BD) structures. The ATOM1 structures are better suited to handling numerous small data buffers as is required in ATM. The BDs are just two bytes long instead of the normal eight byte BDs and buffers are fixed length. Data buffers are scattered in receive and transmit buffer memory areas, each 16K bytes, located anywhere in the processor's memory map.



### 3. REGISTERS

ATOM1 requires several general MC68360 and SCC registers to be configured for operation and reports status in SCC registers.

#### 3.1. General Registers

All the MC68360 general registers (SIM, memory controller, chip select, SDMA, etc.) must be configured as described in the MC68360 User's Manual with the exception of the CP Control Registers and RISC Controller Configuration Register. Interrupts are generated by ATOM1 and the MC68360 interrupt controller must be configured by the user (see the MC68360 User's Manual, section 7.15 for details).

##### 3.1.1. RISC Controller Configuration Register (RCCR)

The MC68360's RCCR controls microcode operation. To run ATOM1, the RCCR should be set as shown in Figure 2. Further details are given in the MC68360 User's Manual, section 7.1.1.

	15	14	13		8	7	6	5	4	3	2	1	0
	TIME	-		TIMEP		EX1M	EX0M	EX1P	EIEE	SCD	LRAM	ERAM	
ATOM1	0	0		0		0	0	00	0	0	1	1	

**Figure 2. ATOM1 RCCR**

Setting RCCR bit 0 enables RAM microcode operation and this should not be done until ATOM1 and the MC68360 are initialized.

Note: the RISC timers are unavailable when ATOM1 is running and the TIME and TIMEP bits must be cleared in the RCCR.

##### 3.1.2. CP Control Registers

The user must configure CP Control registers as shown in Table 1. Note, the value programmed into CPC4 depends on the QUICC device version used.

Address	Name	Description	User Writes (hex)
REGB+\$5CC	CPCR1	CP Control Register 1	8056
REGB+\$5CE	CPCR2	CP Control Register 2	8076
REGB+\$5D0	CPCR3	CP Control Register 3	8036
REGB+\$5D2	CPCR4	CP Control Register 4	8E18 rev B1 MC68360 8818 rev B2, B3, B4, C0, C1 and E MC68360s

**Table 1: CP Control Registers**

## 3.2. SCC Registers

ATOM1 operates using an SCC and is enabled in the SCC's General SCC Mode Register (GSMR). The clock source is configured in the serial interface registers (see section 3.3.1).

### 3.2.1. General SCC Mode Register (GSMR)

ATOM1 requires the following bits to be set in the GSMR: TRX, TTX, CDP, CTSP, CDS and CTSS. To enable ATOM1's transmit and receive functions, the ENT and ENR bits must be set when the initialisation sequence has been completed. Full GSMR details are given in the MC68360 User's Manual, section 7.10.2.

Additionally, the REVD bit in the GSMR (bit 45) changes function to become the CAM control bit when ATOM1 is running:

CAM	0	Use the internal address table for address matching.
	1	Use the external CAM for address matching.

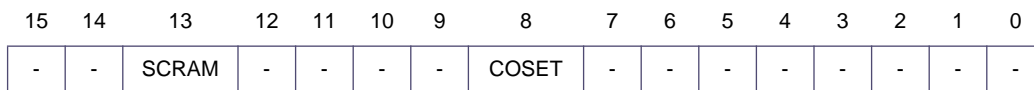
To enable an SCC to run ATOM1, the SCC must be in transparent mode and have MRBLR = 0. When MRBLR is programmed with a non-zero value, the SCC operates in transparent mode. Mixed mode operation of an SCC (where the transmitter runs ATOM1 and the receiver is in transparent mode or vice versa) running ATOM1 is not allowed.

When ATOM1 is loaded and running, the maximum serial performance of an SCC in transparent mode is reduced very slightly.

### 3.2.2. Protocol Specific Mode Register (PSMR)

The SCC Protocol Specific Mode Register (PSMR) controls various SCC functions that are protocol specific. Before enabling ATOM1, the user should initialize the PSMR. The PSMR should only be written when the transmitter and receiver are disabled, otherwise erratic behaviour could result.

For ATOM1, the PSMR is used to control the scrambling and HEC coset functions of both the transmitter and receiver (see Figure 3).



**Figure 3. ATOM1 PSMR**

SCRAM	0	Disable cell payload scrambling during transmission and reception.
	1	Enable cell payload scrambling during transmission and reception.
COSET	0	Do not apply the HEC coset rules.
	1	Apply the HEC coset to all transmitted and received cells.

### 3.2.3. SCC Event Register (SCCE)

The SCC event register (SCCE) contains status bits for events recognized by ATOM1 and the SCC circuitry. Whenever a bit is set in the SCCE, an interrupt to the user is generated unless the corresponding bit in the SCC Mask register (SCCM) is set. All bits in the SCCE are sticky bits: they are set by the SCC hardware or ATOM1 and must be reset by the user writing a one.

Bits 15-8 are set by the SCC hardware and some unused bits may be set. The user should mask interrupts from all unused bits in the SCCM. Bits 7-0 are set by ATOM1 to indicate events that it has detected (see Figure 4).

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	GLR	GLT	DCC	-	-	NMA	SYNC	FIFO	HEC	OAM	BSY	TX	RX

**Figure 4. ATOM1 SCCE**

- RX** A cell has been received and a receive buffer used. This bit is set on completion of reception.
- TX** A cell has been transmitted and a transmit buffer used.
- BSY** A cell was received and discarded due to lack of receive buffers.
- OAM** A control cell (OAM or reserved for future use as indicated by the PT field) been received and a receive buffer used. This bit is set on completion of reception. When this bit is set, the RX bit is also set but about 10 system clocks later.
- HEC** A cell has been received with a HEC error. This bit is also set when the SYNC bit is set. The HEC bit is set as soon as a HEC error is detected - it does not wait until the complete cell is received.
- FIFO** A transmit FIFO underrun or receive FIFO overrun error has occurred. See ASTATUS for the exact status.
- SYNC** ATOM1 has lost or gained cell delineation.
- NMA** ATOM1 received a cell with a non-matching address.
- DCC** The carrier sense status as generated by the DPLL has changed. See description of the Transparent Mode SCCE in the MC68360 User's Manual, section 7.10.21.11.
- GLT** A clock glitch was detected by the SCC on the transmit clock. See description of the Transparent Mode SCCE in the MC68360 User's Manual, section 7.10.21.11.
- GLR** A clock glitch was detected by the SCC on the receive clock. See description of the Transparent Mode SCCE in the MC68360 User's Manual, section 7.10.21.11.

### 3.2.4. SCC Mask Register (SCCM)

The SCCM has the same bit definitions as the SCCE and allows the user to mask interrupts. Further details are given in the MC68360 User's Manual, section 7.10.8.2.

### 3.2.5. SCC Status Register (SCCS)

The function of the SCCS register is unchanged when ATOM1 uses an SCC.

## 3.3. Serial Interface

ATOM1 may use the MC68360's serial interface (SI) in any mode although the time division multiplex mode to allows easy connection to E1 and T1 line interface devices. This section of the User's Manual concentrates on the TDM interface although use of other serial interfaces is not precluded.

It is the user's responsibility to ensure that incoming cells are octet aligned with the SI handshake or synchronisation pulses. ATOM1 will then apply the cell HEC delineation mechanism correctly and synchronise with the incoming cell stream.

ATOM1 can use either of the MC68360 TDM ports. Programming information is given here for use of TDM A and can easily be converted for TDM B. Further information about the SI is given in the MC68360 User's Manual, section 7.8.

### 3.3.1. Serial Interface Registers

The SI registers SIMODE, SIGMR and SICR should be programmed for the required interface standard and the user must configure the SI RAM pointer (SIRP) as required in the system. An example of programming the SI registers for connection to a line interface device with common transmit and receive clocks is shown in Table 2. In this example, ATOM1 is running on SCC1.

Note: the timeslot assigner TSA must not be enabled in SIGMR until the SI RAM and other registers have been configured.

Register		User Writes (hex)
SIMODE	Serial Interface Mode Register	0000 0058
SIGMR	Serial Interface Global Mode Register	04
SICR	Serial Interface Clock Route Register	0000 0040

**Table 2: MC68360 Serial Interface Programming Example**

### 3.3.2. Serial Interface RAM

ATOM1 can use the MC68360 TSA with independent receiver and transmitter routing RAM tables. An example of programming these two tables for back-to-back ATM cell transmission and reception is given in Table 3. TDM A is used and data is routed to SCC1. Other SCCs and TDM B may be used. For debug purposes, all four SI strobes are asserted during various timeslots: L1ST1 during cell header transmission, L1ST2 on HEC transmission, L1ST3 during cell header reception and L1ST4 on HEC reception.

SI RAM Entry	User Writes (hex)	Strobe Asserted	Remarks
Rx A 0	104E	L1ST3	Receive four bytes of cell header
Rx A 1	2042	L1ST4	Receive one byte HEC field
Rx A 2	007E	-	Receive first sixteen bytes of cell payload

**Table 3: Serial Interface RAM Programming Example**

SI RAM Entry	User Writes (hex)	Strobe Asserted	Remarks
Rx A 3	007E	-	Receive second sixteen bytes of cell payload
Rx A 4	007F	-	Receive final sixteen bytes of cell payload
Tx A 0	044E	L1ST1	Transmit four bytes of cell header
Tx A 1	0842	L1ST2	Transmit one byte HEC field
Tx A 2	007E	-	Transmit first sixteen bytes of cell payload
Tx A 3	007E	-	Transmit second sixteen bytes of cell payload
Tx A 4	007F	-	Transmit final sixteen bytes of cell payload

**Table 3: Serial Interface RAM Programming Example**

### 3.4. Parallel Port Registers

For ATOM1 operation over TDM A, the MC68360 parallel port pins should be set up to interface to external signals as shown in Table 4. The TSA strobes and SDMA Acknowledge pins are shown and common receive and transmit clocks are used.

Signal	Pin	Direction
L1TXDA	PA6	Output
L1RXDA	PA7	Input
L1RSYNCA	PC11	Input
L1RCLKA	PA8	Input
L1ST1	PB12	Output
L1ST2	PB13	Output
L1ST3	PB14	Output
L1ST4	PB15	Output
SDACK1*	PC10	Output
SDACK2*	PC8	Output

**Table 4: ATOM1 TDM A Port Pin Requirements**

To achieve this, Table 5 shows an example of how the port registers may be programmed.

Register	User Writes (hex)
PIPC PIP Configuration Register	0000
PADIR Port A Data Direction Register	00C0
PAPAR Port A Pin Assignment Register	01C0
PAODR Port A Open Drain Register	0000
PBDIR Port B Data Direction Register	0000 0000
PBPAR Port B Pin Assignment Register	0000 F000
PBODR Port B Open Drain Register	0000
PCDIR Port C Data Direction Register	0500
PCPAR Port C Pin Assignment Register	0D00
PCSO Port C Special Option Register	0000
PCINT Port C Interrupt Control Register	0000

**Table 5: MC68360 Port Register Programming**

## 4. ATOM1 SCC PARAMETER RAM

When ATOM1 operates on a particular SCC, the SCC parameter RAM memory map is that shown in Table 6. The user must initialize the parameter RAM with the values given after reset or when re-initializing ATOM1.

Address	Name	Width	Description	User Writes (hex)
SCC Base + 00	RBASE	Word	Receive BD Base Address	UD
SCC Base + 02	TBASE	Word	Transmit BD Base Address	UD
SCC Base + 04	RFCR	Byte	Receive Function Code	UD
SCC Base + 05	TFRCR	Byte	Transmit Function Code	UD
SCC Base + 06	MRBLR	Word	Transparent Rx Max Buffer Length	0000
SCC Base + 08	RSTATE	Long	Receiver Internal State	use init rx/tx command
SCC Base + 0C	R_PTR	Long	Receiver Internal Data Pointer	-
SCC Base + 10	RBD_PTR	Word	Receive BD Pointer	use init rx/tx command
SCC Base + 12	R_CNT	Word	Receive Internal Byte Counter	-
SCC Base + 14	RTEMP	Long	Receiver Temporary Data Storage	-
SCC Base + 18	TSTATE	Long	Transmitter Internal State	use init rx/tx command
SCC Base + 1C	T_PTR	Long	Transmit Internal Data Pointer	-
SCC Base + 20	TBD_PTR	Word	Transmit BD Pointer	use init rx/tx command
SCC Base + 22	T_CNT	Word	Transmit Internal Byte Counter	-
SCC Base + 24	TTEMP	Long	Transmitter Temporary Data Storage	-
SCC Base + 28	ALPHA	Word	Receiver Delineation Alpha Counter	-
SCC Base + 2A	DELTA	Word	Receiver Delineation Delta Counter	-
SCC Base + 2C	AVERSION	Word	ATOM1 Version Number	-
SCC Base + 2E	CAM_PORT	Word	CAM Port B Mask	-
SCC Base + 30	RX_BUFF1	Long	Receive Buffer Area Start Address	UD
SCC Base + 34	TX_BUFF1	Long	Transmit Buffer Area Start Address	UD
SCC Base + 38	EHEAD	Long	Empty Cell Header (little endian byte format)	0000 0000 or 0100 0000
SCC Base + 3C	EPAYLOAD	Long	Empty Cell Payload (little endian byte format)	6A6A 6A6A
SCC Base + 40	RSTUFF	Long	Receive Data Stuffing Location (53 to 52 byte conversion)	-
SCC Base + 44	SHUFFLESTATE	Word	Receiver Data Shuffling State	0000
SCC Base + 46	RHECTEMP	Word	Temporary HEC Storage	-
SCC Base + 48	THECTEMP	Word	Temporary HEC storage area	-
SCC Base + 4A	ASTATUS	Word	ATOM1 status word	0000
SCC Base + 4C	NMA_HEAD	Long	Non-Matching Address Cell Header	-
SCC Base + 50	NMA_CNT	Word	Non-Matching Address Cell Counter	0000
SCC Base + 52	HEC_ERR	Word	HEC Error Counter	0000

**Table 6: ATOM1 SCC Parameter RAM Memory Map**

Address	Name	Width	Description	User Writes (hex)
SCC Base + 54	RSCRAM	Long	Receiver Scrambling Storage	-
SCC Base + 58	RSCRAM1	Long		-
SCC Base + 5C	TSCRAM	Long	Transmitter Scrambling Storage	-
SCC Base + 60	TSCRAM1	Long		-
SCC Base + 64	HEADMASK	Long	Cell Header Comparison Mask	UD
SCC Base + 68	HEAD1	Long	Cell Header Look-Up Table Entry 1	UD
SCC Base + 6C	HEAD2	Long	Cell Header Look-Up Table Entry 2	UD
SCC Base + 70	HEAD3	Long	Cell Header Look-Up Table Entry 3	UD
SCC Base + 74	HEAD4	Long	Cell Header Look-Up Table Entry 4	UD
SCC Base + 78	HEAD5	Long	Cell Header Look-Up Table Entry 5	UD
SCC Base + 7C	HEAD6	Long	Cell Header Look-Up Table Entry 6	UD
SCC Base + 80	HEAD7	Long	Cell Header Look-Up Table Entry 7	UD
SCC Base + 84	HEAD8	Long	Cell Header Look-Up Table Entry 8	UD
SCC Base + 88	HEAD9	Long	Cell Header Look-Up Table Entry 9	UD
SCC Base + 8C	HEAD10	Long	Cell Header Look-Up Table Entry 10	UD
SCC Base + 90	HEAD11	Long	Cell Header Look-Up Table Entry 11	UD
SCC Base + 94	HEAD12	Long	Cell Header Look-Up Table Entry 12	UD
SCC Base + 98	HEAD13	Long	Cell Header Look-Up Table Entry 13	UD
SCC Base + 9C	HEAD14	Long	Cell Header Look-Up Table Entry 14	UD
SCC Base + A0	HEAD15	Long	Cell Header Look-Up Table Entry 15	UD
SCC Base + A4	HEAD16	Long	Cell Header Look-Up Table Entry 16	UD

UD = User Defined

**Table 6: ATOM1 SCC Parameter RAM Memory Map**

#### **4.0.1. BD Queue Pointers (RBASE and TBASE)**

The user configures RBASE and TBASE to define the starting location of the receive and transmit BD queues in the dual-port RAM. Further information is given in the MC68360 User's Manual, section 7.10.7.1.

#### **4.0.2. SCC Function Code Registers (RFCR and TFCR)**

The user configures RFCR and TFCR to define the SDMA function code pin settings during SDMA transfers. Further information is given in the MC68360 User's Manual, section 7.10.7.2.

#### **4.0.3. Transparent Mode Receive Maximum Buffer Length (MRBLR)**

To run ATOM1 on an SCC, the user must program that SCC's MRBLR to zero. When MRBLR is programmed with a non-zero value, the SCC operates in transparent mode as described in the MC68360 User's Manual, section 7.10.21.

#### **4.0.4. SCC Internal State Parameters (RSTATE and TSTATE)**

ATOM1 uses RSTATE and TSTATE to store internal state variables and flags during operation. The user must not write to these locations.

#### **4.0.5. BD and Buffer Pointers (R\_PTR, RBD\_PTR, T\_PTR and TBD\_PTR)**

During cell transmission and reception, ATOM1 uses R\_PTR, RBD\_PTR, T\_PTR and TBD\_PTR as pointers to the current BD and data locations. RBD\_PTR and TBD\_PTR point to the current or next BD to use and need to be initialized with the Init Rx/Tx command to point to the locations given in RBASE and TBASE. R\_PTR, and T\_PTR point to the next data location in memory. When ATOM1 is operating, these locations should not be written by the user. Further information is given in the MC68360 User's Manual, sections 7.10.7.4 and 7.10.7.5.

#### **4.0.6. Transmit and Receive Buffer Queue Start Pointers (TX\_BUFF1, RX\_BUFF1)**

The user configures TX\_BUFF1 and RX\_BUFF1 to define the starting location of the transmit and receive buffer areas in main memory. This is the starting address of the 16K byte areas where transmit and receive buffers reside. TX\_BUFF1 and RX\_BUFF1 must be long-word aligned addresses (bits 0 and 1 must be zero).

TX\_BUFF1 and RX\_BUFF1 can be dynamically changed provided that care is taken to synchronise the buffer queues held in user software and the queues that the QUICC's CP sees. The CPM reads TX\_BUFF1 and RX\_BUFF1 every time that it opens a data buffer so it is advised that the transmit and receive queues are allowed to empty before TX\_BUFF1 and RX\_BUFF1 are changed.

#### **4.0.7. Buffer Counters (R\_CNT and T\_CNT)**

During ATM cell transmission and reception, ATOM1 uses R\_CNT and T\_CNT to count the number of bytes transmitter or received. The user must not write to these locations.

#### **4.0.8. Receiver Delineation Counters (ALPHA and DELTA)**

ATOM1 applies the HEC delineation mechanism described in I.432 with Alpha=7 and Delta=6 to locate and maintain cell synchronisation. During reception, ATOM1 updates the ALPHA and DELTA parameter locations and the user must not write to these locations. There is no need to initialise these locations.

#### **4.0.9. Version Number (AVERSION)**

During operation, ATOM1 writes a version number to AVERSION. The current version number stored in AVERSION is \$0001.

#### **4.0.10. CAM Port Selection (CAM\_PORT)**

When the CAM interface is enabled, the user should set one or more bits in CAM\_PORT to select which of the QUICC's Port B pins is the match input or inputs for this SCC. Bit 0 of CAM\_PORT corresponds to PB0, bit 1 to PB1, etc.

#### **4.0.11. Empty Cell Data (EHEAD and EPAYLOAD)**

ATOM1 transmits and receives empty cells using EHEAD and EPAYLOAD. On transmission, an empty cell is made up by transmitting EHEAD once, calculating and transmitting a HEC and transmitting EPAYLOAD twelve times. On reception, the incoming header is compared with EHEAD to check for empty cells and EPAYLOAD is not used.

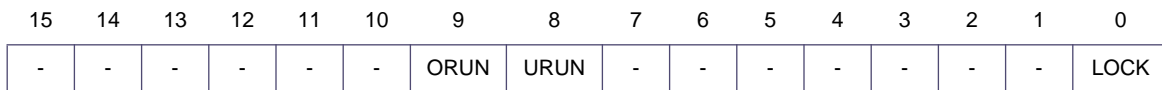


The user is free to select whatever empty cell header and payload values are required: the ATM Forum UNI Specification states that unassigned cells should be transmitted while the ITU mandates use of idle cells.

The user must write the EHEAD and EPAYLOAD parameters as required and these two locations must be written in little endian byte order. Unassigned cells are used as empty cells when EHEAD = \$0000 0000 and idle cells when EHEAD = \$0100 0000. In both cases, EPAYLOAD should be initialised to \$6A6A 6A6A.

#### 4.0.12. Status Information (ASTATUS)

The ATOM1 Status parameter provides the user with additional status information concerning FIFO errors and receiver synchronisation status. ASTATUS is shown in Figure 5.



**Figure 5. ASTATUS**

Bits in the upper byte is set by ATOM1 and cleared by the user. Bits in the lower byte are set and cleared by ATOM1. During initialisation, the user must clear all bits in ASTATUS. During operation, the user can read and clear (write zero) to the upper byte but must only read the lower byte.

FIFO overruns and underruns are reported in the SCCE FIFO bit and further information is given in the ORUN and URUN bits:

ORUN	0	No Receiver FIFO Overrun.
	1	Receiver FIFO Overrun.
 URUN	 0	 No transmitter FIFO underrun.
	1	Transmitter FIFO underrun.

Each change in state of the receiver cell delineation is reported in the SCCE SYNC bit. The LOCK bit shows the current delineation status:

LOCK	0	The ATOM1 receiver is out of synchronisation and is not receiving cells.
	1	The ATOM1 receiver has gained cell delineation and is receiving cells.

#### 4.0.13. Non-Matching Header Storage and Counter (NMA\_HEAD and NMA\_CNT)

When a cell is received with a non-matching header, the header (minus the HEC) is written to NMA\_HEAD, NMA\_CNT is incremented and the NMA bit set in the SCCE. NMA\_HEAD is a location where headers from non-matching cells are logged and NMA\_CNT is a counter of non-matching cells. The user is free to read and clear NMA\_CNT at any time.

#### 4.0.14. HEC Error Counter (HEC\_ERR)

HEC\_ERR is a 16-bit counter of incoming cells with HEC errors. The user is free to read and clear HEC\_ERR at any time.

#### 4.0.15. Incoming Header Mask and Look-up Table (HEADMASK and HEADn)

When the CAM option is disabled, ATOM1 masks the header of each incoming cell with HEADMASK and then searches for the result in the HEADn table. When a match is found, the cell is received otherwise the cell header is logged in NMA\_HEAD and the remainder of the cell is discarded.

The HEADMASK masking process uses a bitwise AND function so bits can be masked out by clearing the relevant bit in HEADMASK. The bit locations for HEADMASK are shown in Figure 6. Bit 0 must always be written with zero.

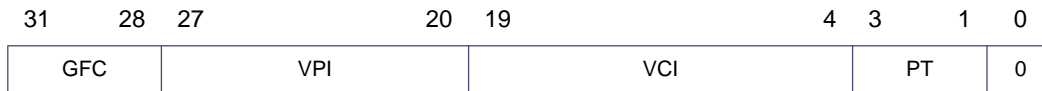


Figure 6. HEADMASK

The result of the masking is then compared with each of the HEADn values, starting with HEAD1. Although sixteen locations are shown in the HEADn table, the user may define fewer. The final location in the table is denoted with the Last bit being set. When multiple HEADn locations are being used, the user should take care because the look-up table can overlap other peripheral areas in memory (see the MC68360 User's Manual section 3.2 for further details). If the look-up table is extended to overlap other peripheral's parameter RAM areas, the other peripherals cannot be used.

The bit locations for the HEADn locations are shown in Figure 7. Bit 0 should be written with zero except for the last entry in the table.

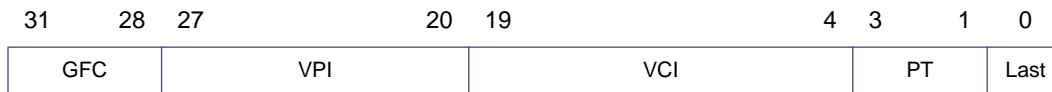


Figure 7. HEADn

ATOM1 searches the HEADn table sequentially so the headers for the most frequently connections should be at the beginning of the table.

#### 4.0.16. Temporary Data Storage

During ATM cell transmission and reception, ATOM1 uses RTEMP, TTEMP, RSTUFF, RHECTEMP, THECTEMP, RSCRAM, RSCRAM1, TSCRAM and TSCRAM1 to store temporary data. This data is for internal use and the user must not write to these locations.

## 5. ATOM1 COMMANDS

The following commands are understood by ATOM1:

- Initialise Receiver and Transmitter Parameters
- Initialise Receiver Parameters

- Initialise Transmitter Parameters
- Stop Transmit
- Restart Transmit
- Restart Reception
- CRC-32 Calculator

Execution of other SCC commands may cause errant behaviour. All commands are given through the Command Register and operate as with other protocols except Restart Reception.

The ATOM1 commands to initialise the receiver and transmitter parameters operate the same as for other SCC protocols. If the initialisation commands are given during cell transmission and reception, cell synchronisation will be lost.

The Stop Transmit Command stops cell transmission immediately and the transmit line goes high. To stop transmission of data cells but continue with empty cells, the user must wait for the transmit BD queue to empty.

The commands are run by writing the command and channel number to the QUICC's Command Register (CR). The op-codes for the initialisation, Stop Transmit and Restart Transmit command are the same as for the QUICC's other protocols. The op-code for the Restart Reception command is and the CRC-32 calculator 1111. All other bits in the CR remain unchanged.

Command execution latency for the initialisation, Stop Transmit and Restart Transmit command are the same as for the QUICC's other protocols. The Restart Reception command is executed when the current cell being received is complete and execution time therefore depends on the data rate and when the command is given. For all commands, ATOM1 clears the FLG bit in the CR to indicate completion of the command.

### **5.1. Restart Reception Command**

The Restart Reception Command is similar to Enter Hunt Mode. If ATOM1 is currently receiving an empty cell, cell delineation is immediately lost and the resynchronisation process begins. If the current cell being received is a data or control cell (being received or discarded by ATOM1), the current buffer is closed and cell delineation is lost at the end of the cell. The Enter Hunt Mode command should not be used with ATOM1.

To restart the receiver immediately regardless of the current state, the Init Rx Parameter command can be used.

The Restart Reception Command and the CRC-32 Calculator share the same op-code in the command register. To issue a Restart Reception Command, the user must ensure that the current CRC-32 calculation is complete and the CRC\_CNT parameter is zero.

### **5.2. CRC-32 Calculator**

The CRC-32 Calculator command calculates the 32-bit CRC sequence used in AAL5 frames on a data buffer defined by a pointer and length. Multiple buffers may be strung together by the

user to calculate the CRC-32 value over a large AAL5 frame made up of multiple CRC-32 buffers. ITU I.363 gives more details and examples of AAL5 CRC calculations.

Note: The QUICC's CP runs the ATOM1 CRC-32 calculator as its lowest priority task. High traffic throughput on the SCCs and other CP activity will adversely affect the CRC-32 calculator's performance.

The ATOM1 CRC-32 calculator replaces four parameters in QUICC's RISC Timer Parameter RAM area (starting at DPRBASE + \$DB0) as shown in Table 7. The user must initialize the CRC-32 parameter RAM with the values given after reset or when re-initializing ATOM1.

Address	Name	Width	Description	User Writes (hex)
DPRBASE + \$DB0	CRC_PTR	Long	CRC-32 Calculator Pointer	UD
DPRBASE + \$DB4	CRC_CNT	Word	CRC-32 Calculator Counter	UD
DPRBASE + \$DB6	CRC_FC	Byte	CRC-32 Calculator Function Code	UD
DPRBASE + \$DB8	CRC_RESULT	Long	CRC-32 Calculator Seed Value and Result	0000 0000 or UD

UD = User Defined

**Table 7: ATOM1 CRC-32 Calculator Parameter RAM Memory Map**

To start a CRC-32 calculation, the CRC\_PTR, CRC\_CNT, CRC\_RESULT and CRC\_FC parameters must be initialised to give the start address of the data block for the calculation, block length, starting CRC-32 seed and function code. ATOM1 reads and writes these four parameters during a CRC-32 calculation so they must not be altered by the user while the calculation is in progress.

The CRC\_PTR may be programmed with any address that corresponds to the start of the CRC-32 buffer in memory. The length of the buffer must be programmed into the CRC\_CNT parameter. If CRC\_CNT is zero at the start of the command, ATOM1 will run a Restart Reception command rather than the CRC-32 calculator.

The CRC\_RESULT parameter must be preset to zero at the start of a new CRC-32 calculation. On completion of the calculation, CRC\_RESULT holds the resulting 32-bit CRC value. At the beginning of a CRC-32 calculation, CRC\_RESULT may be programmed with the result of a previous CRC-32 calculation if buffers are being strung together.

The user configures CRC\_FC to define the SDMA function code pin settings when the SDMA reads the CRC-32 buffer. Further information is given in the MC68360 User's Manual, section 7.10.7.2.

A CRC-32 calculation is started by writing \$0F01 to the command register. When the command has been accepted and CRC-32 processing begins, ATOM1 clears the FLG bit in the command register. Processing continues through the whole CRC-32 buffer and when the calculation is complete, ATOM1 sets bit 15 in the RISC Timer Even Register (RTER). This can be used to generate a processor interrupt by programming the QUICC's interrupt controller. Bit 15 of the RTER should be cleared before a CRC-32 calculation is started.

## 6. BUFFER DESCRIPTORS & BUFFERS

ATOM1 uses a modified version of the standard MC68360 BD and buffer data structure better suited to the high number of and small size of ATM cells. There is a single transmit BD queue and a single receive queue. Associated with each BD queue is a buffer queue. All ATOM1 BDs are two bytes long and buffers are all 52 bytes long. These BD and buffer sizes maximise the number of BDs that can be located in the QUICC's dual port RAM when the ATOM1 microcode is also present and running.

The BDs are a subset of the standard QUICC BDs containing only a 16-bit status and control word. Buffer length and pointer fields are not implemented as for other QUICC serial protocols. BDs are located in the QUICC's dual port RAM.

All data buffers have a fixed length of 52 bytes and are located within 16K byte buffer areas in memory. Receive BD1 is associated with buffer 1, receive BD2 with buffer 2, etc. Transmit and receive buffers are scattered within a 16K byte memory area and use an eleven bit pointer within the transmit BD to locate the buffer exactly. The transmitter and receiver each use different 16K byte buffer areas which may be located anywhere in memory.

### 6.1. ATOM1 Buffer Descriptor and Buffer Queues

Figure 8 shows an example of the transmitter BD and data buffer queues and the pointers associated with them. In this example, there are 20 BDs, each linked to a buffer. TBASE points to the start of the BD queue and TX\_BUFF1 points to the start of the buffer area. ATOM1 calculates the exact start address of a transmit buffer by adding TX\_BUFF1 to the address field in the transmit BD. Buffers may thus be scattered randomly throughout the 16K byte area.

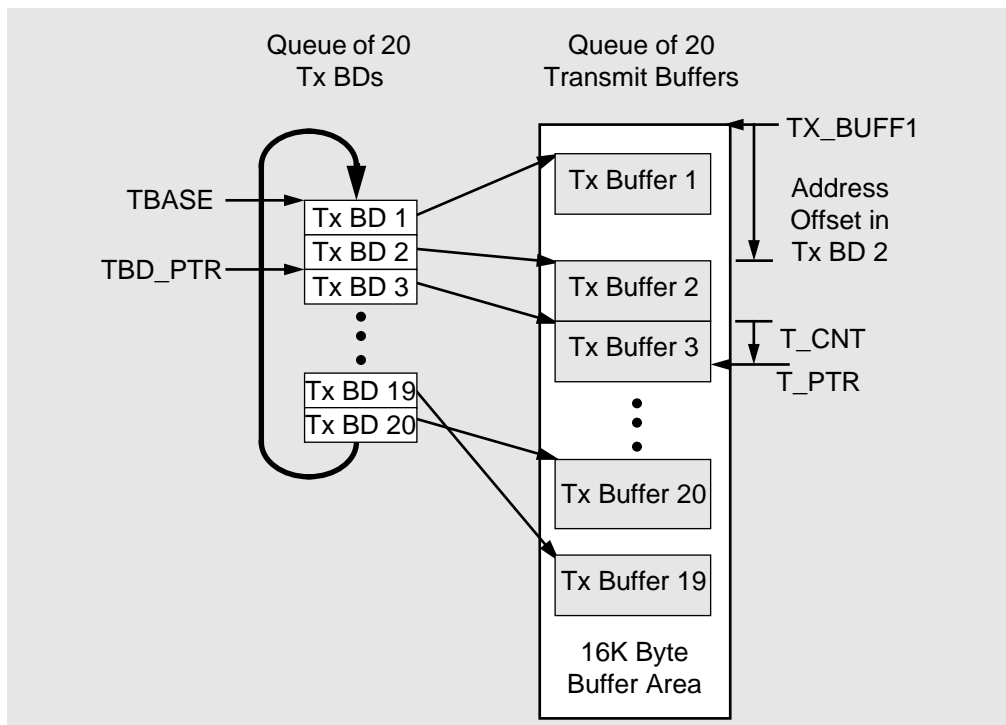


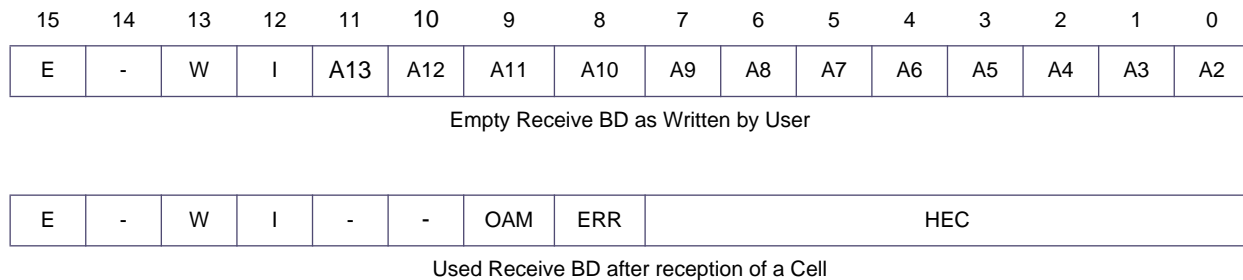
Figure 8. ATOM1 Transmit Buffer and BD Queue Example

In this example, the first two buffers have been transmitted and ATOM1 has advanced the TBD\_PTR pointer to point at the current BD. Similarly, the T\_PTR pointer has been advanced to point to the current data being transmitted by ATOM1. T\_CNT is the offset of T\_PTR from the start of the data buffer. When the end of the BD queue is reached, TBD\_PTR is wrapped back by loading it from TBASE. Receive BD and buffer queues follow the same structure as shown here for transmission.

## 6.2. Receive Buffer Descriptor and Data Buffer

Receive BDs are two bytes long as shown in Figure 9. Each BD is linked to one buffer that can hold a received ATM cell. Receive buffers may be scattered around a 16K byte area of memory but each buffer must start on a long word aligned address. If the start address of a buffer is at the very end of the 16K byte area (for example, the BD is programmed with \$8FFF) then the end of the received cell will be 48 bytes outside of the 16K byte area.

When the user prepares an empty receive BD, the address pointer to the start of the data buffer must be written into the BD. When the Empty bit is set, ownership of the BD is handed over from the user to the QUICC's CP and the user must not modify any part of the BD. Once a cell is received and ATOM1 returns the BD and buffer to the user, the address pointer field is replaced with OAM and error status flags and the HEC of the incoming cell. User software must maintain a copy of the address pointer for each receive BD in the receive queue.



**Figure 9. ATOM1 Receive Buffer Descriptor**

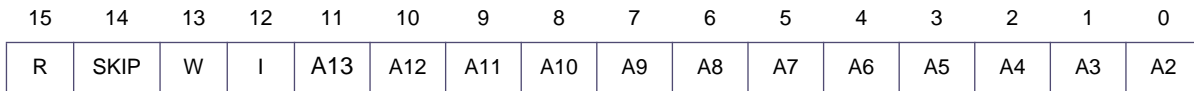
E	Empty	Control bit set by the user to indicate that the associated buffer is empty and ATOM1 can receive a cell into it.
W	Wrap	Control bit set by the user to wrap the receive BD queue after this BD.
I	Interrupt	Control bit set by the user. When a cell is received into the buffer associated with this BD, a receive interrupt is generated.
A2-A13	Address	Address offset to the start of the data buffer from the start of the 16K byte receive data area.
OAM	OAM Cell	Status bit set by ATOM1. The associated cell is an OAM or reserved cell (cell PT field = 1xx).
ERR	HEC Error	Status bit set by ATOM1. The associated cell was received with an error in the header as detected by the HEC check.

HEC      HEC Byte    Byte containing the received cell's HEC field.

ATOM1 stores received ATM cells in receive buffers after removing the HEC. Receive data buffers have the same structure as transmit buffers. The first byte of the buffer is the first byte received (the GFC field and part of the VPI). The HEC field is removed and stored in the receive BD.

### 6.3. Transmit Buffer Descriptor and Data Buffer

Transmit BDs are two bytes long as shown in Figure 10. Each BD is linked to one buffer that must hold the complete ATM cell to be transmitted. Transmit buffers may be scattered around a 16K byte area of memory but must start on a long word aligned address. If the start address of a buffer is at the very end of the 16K byte area (for example, the transmit BD is programmed with \$8FFF) then the end of the transmit cell will be 48 bytes outside of the 16K byte area.



**Figure 10. ATOM1 Transmit Buffer Descriptor**

R	Ready	Control bit set by the user to initiate transmission of the associated cell. ATOM1 clears the Ready bit when the associated buffer has been transmitted.
SKIP	Skip BD	Control bit set by the user. When ATOM1 encounters a transmit BD with the Skip bit set, an empty cell is transmitted and control passes to the next BD. When the Skip bit is set, the Ready bit is ignored.
W	Wrap	Control bit set by the user to wrap the transmit BD queue after this BD.
I	Interrupt	Control bit set by the user. When the cell associated with this BD is transmitted, a transmit interrupt is generated. This bit is ignored when the Skip bit is also set.
A2-A13	Address	Address offset to the start of the data buffer from the start of the 16K byte transmit data area.

ATOM1 transmits the ATM cell stored in the transmit buffer and automatically inserts the HEC. Data buffers are all 52 bytes long with the first four bytes being the cell header as shown in Figure 11. The first byte of the buffer is the first byte to be transmitted.

GFC	VPI	VCI	PT	CLP
Payload				
Payload				
• • •				
Payload				

**Figure 11. Transmit Buffer**

### 6.3.1. Transmit Bandwidth Reservation

ATOM1 includes a mechanism to reserve transmit bandwidth to aid tasks that require low latency transmission. An example of such a system might be one that simultaneously handles voice and data transmissions: the data transmission task can queue a large number of ATM cells that are not delay sensitive while the voice transmission circuit has less traffic but must be serviced at regular intervals.

The mixed voice and data example is used here to illustrate the bandwidth reservation system.

In most systems, the number of transmit and receive buffer descriptors and buffers per SCC is fixed when the system software is designed. For example, the ATOM1 transmit queue may be fixed at ten buffers. The data transmission software can maintain its own software queue that may hold several hundred cells. The voice task must transmit each cell as soon as it is generated and not store them in a queue. If the voice cells were to be placed at the end of the software data transmission queue, the latency would be too high.

The transmit bandwidth reservation system avoids this problem by allowing selected BDs to be reserved for the voice transmission. When ten BDs are allocated to the data and voice transmission software tasks, nine can be assigned for data and one for voice. This reserves 10% of the transmission bandwidth for voice channels.

The reservation mechanism uses the Skip bit in the transmit BD. As ATOM1 processes the transmit BD queue, it transmits cells where the Skip bit is zero. When a BD with the Skip bit is encountered, it is skipped and an empty cell is transmitted in its place.

In the mixed data and voice example, the data transmission software can control nine of the ten BDs and set their Ready bits to initiate transmission. The voice transmission software reserves one BD by setting its Skip bit. When a voice cell is ready for transmission, the voice transmission software inserts the header and payload in the relevant transmit data buffer, clears the Skip bit and sets the Ready bit. Next time that ATOM1 checks this BD, it will see only the Ready bit set and therefore transmit the voice cell.

Figure 12 illustrates this example. The tenth transmit BD is reserved for voice traffic. While the voice transmission task is constructing an ATM cell and has nothing in the transmit BD queue,



it reserves bandwidth by setting the Skip bit. When the voice cell has been constructed and loaded into the tenth transmit buffer, the Skip bit is cleared and the Ready bit set. The voice cell is then transmitted with a maximum delay of nine cells.

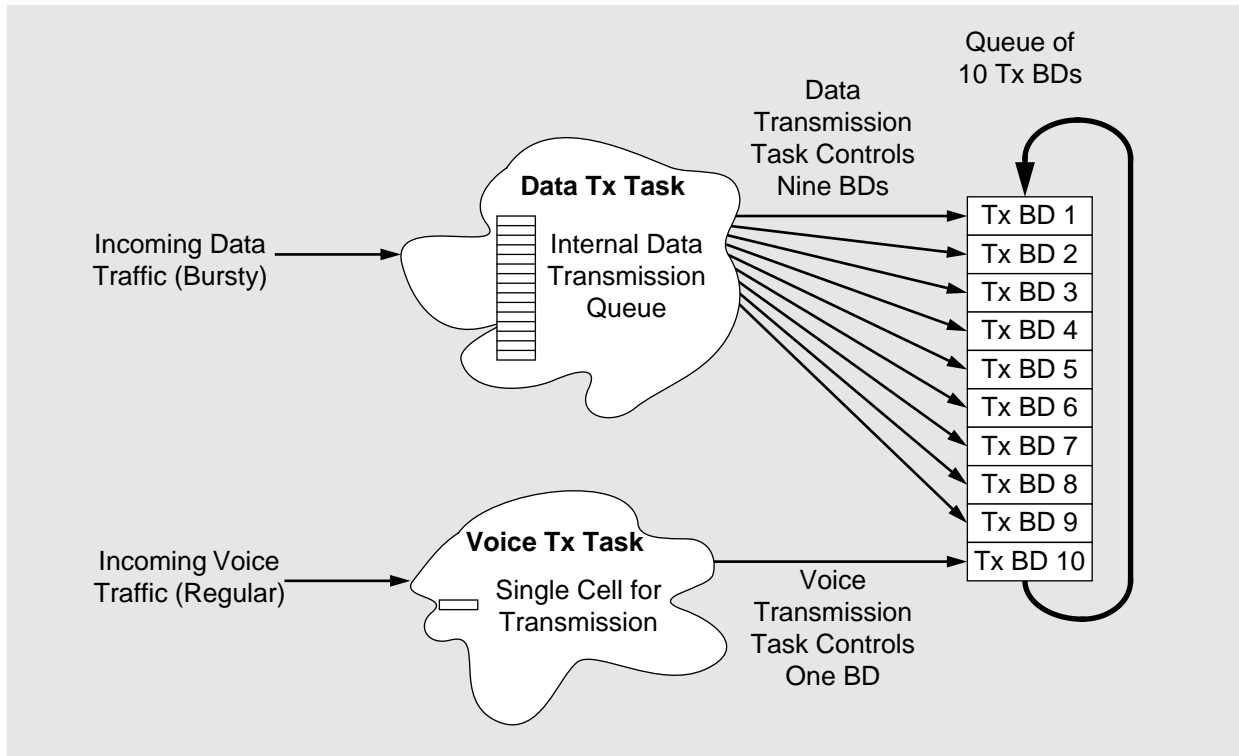
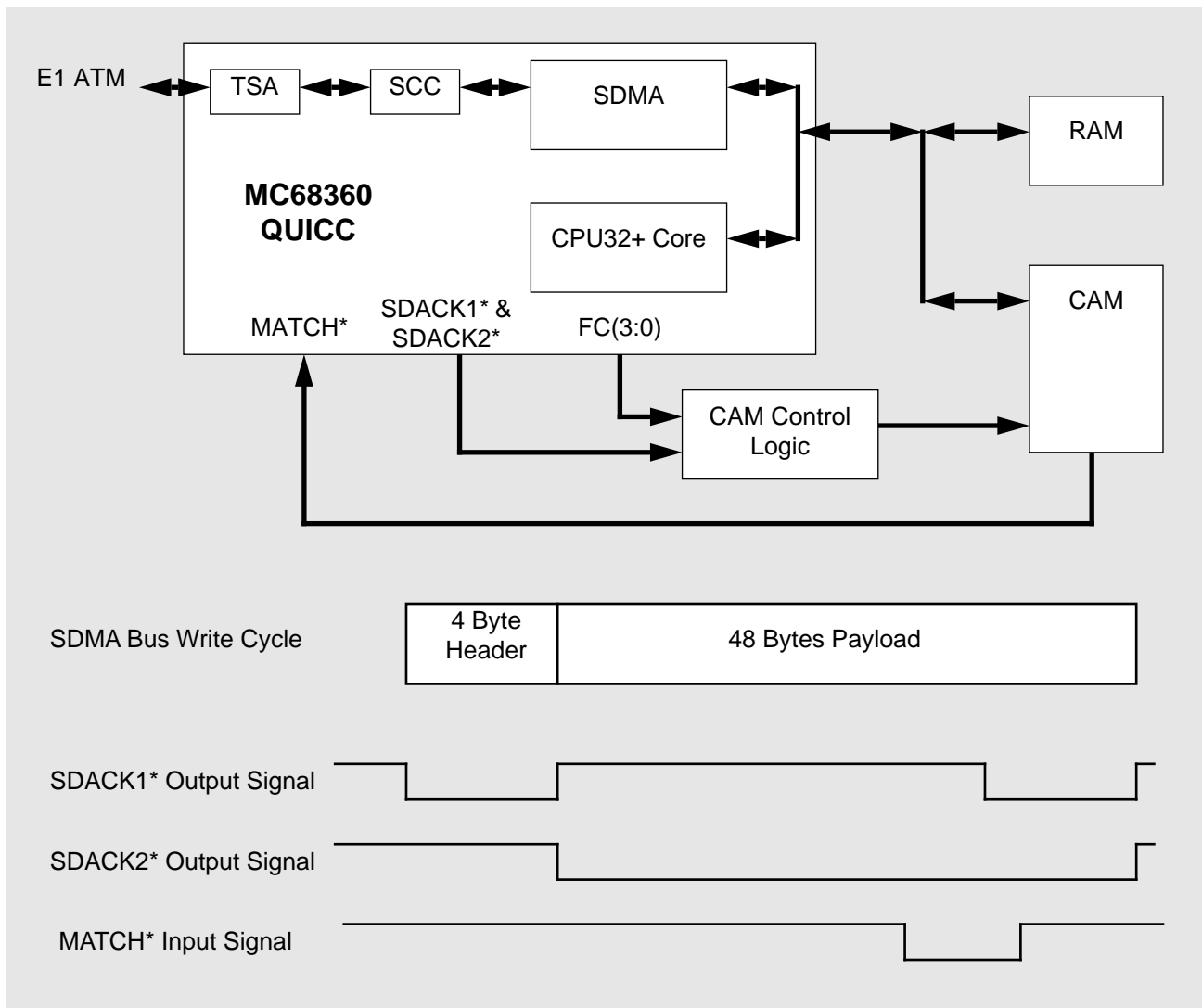


Figure 12. Mixed Voice and Data Transmission Example

## 7. ATOM1 CAM INTERFACE

Figure 13 shows the ATOM1 CAM interface and a timing diagram illustrating operation. Operation is similar to the Ethernet CAM interface (see the MC68360 User's Manual section 7.10.23.7 for further details). The SDACK1\* signal delimits a header write cycle on the data bus and SDACK2\* delimits payload write cycles. Both SDACK1\* and SDACK2\* are asserted during the final payload write cycle. ATOM1 samples the MATCH\* signal during the last data transfer of the cell.



**Figure 13. ATOM1 CAM Interface**

The user selects the pin to be used as the MATCH\* input from the port B PB0-PB15 pins using the CAM\_PORT parameter. Multiple CAMs can be connected to a single SCC running ATOM1 by using multiple MATCH\* inputs. ATOM1 looks for one or more of the MATCH\* inputs to be active. The port B pins being used as the MATCH\* input must be configured as input ports using the port B configuration registers PBPARG and PBDIR.

When ATOM1 is operating on several SCCs concurrently, each may be connected to a CAM. To distinguish between the SDMA cycles for each SCC, each receiver function codes should be programmed to a different value.

When the CAM interface is operating with ATOM1, the Ethernet CAM interface can still be used. The CAM control logic will need to check the function codes and SDACK signals to determine which SCC has initiated a particular bus cycle and enable the appropriate CAM chip.

## 8. LOADING & CONFIGURING ATOM1

Before operation, the ATOM1 microcode must be loaded into the MC68360's internal RAM. The microcode is delivered in a file of S-records ready to download to a Motorola QUADS board and a general M68000 assembly code file. Further details of QUADS download procedures are given in the QUADS User's Manual.

The microcode occupies the internal RAM blocks DPRBASE+0 to DPRBASE+3FF and DPRBASE+600 to DPRBASE+6FF. These areas are unavailable for use by the user.

Once the ATOM1 microcode is loaded into the MC68360's internal RAM, it is initialized as follows:

1. Configure SDMA.
2. Configure parallel ports.
3. Configure serial interface.
4. Enable TSA.
5. Configure CP registers.
6. Enable RAM microcode.
7. Configure SCC registers. Configure GSMR, PSMR, SCCE and SCCM to select required operation.
8. Configure ATOM1 parameter RAM.
9. Initialize queue of empty receive BDs and buffers.
10. Clear transmit BDs.
11. Enable SCC by setting ENT and ENR bits in the GSMR lower long word.

Once initialisation is complete, ATOM1 can be controlled through the transmit and receive BD queues and commands.

Example source code for initialisation and operation of ATOM1 is given in Appendix B.

## 9. PERFORMANCE

Tables 8 and 9 show the results of some performance characterisation experiments where ATOM1 is run concurrently with other SCCs. All results are for zero wait state access to memory. In the experiments, all SCCs were continually transmitting and receiving data cells (no commands or CRC-32 calculations) and frames of the following size:

HDLC frames	24 bytes data + 2 bytes CRC
Ethernet frames	64 bytes data + 4 bytes CRC
ATM cells	5 byte header + 48 bytes payload

Configuration	ATOM1 SCCs	Mode	Frequency Ratio	System Frequency
1 ATM	SCC1	ATOM1 - No Scrambling	1:3	25 MHz / 33 MHz
1 ATM	SCC1	ATOM1 - Scrambling	1:4.5	
2 ATM	SCC1, 2	ATOM1 - No Scrambling	1:8	
2 ATM	SCC1, 2	ATOM1 - Scrambling	1:10	
3 ATM	SCC1, 2, 3	ATOM1 - No Scrambling	1:10	
3 ATM	SCC1, 2, 3	ATOM1 - Scrambling	1:14	
4 ATM	SCC1, 2, 3, 4	ATOM1 - No Scrambling	1:13	
4 ATM	SCC1, 2, 3, 4	ATOM1 - Scrambling	1:19	

**Table 8: ATOM1 Performance**

Configuration	ATOM1 SCC			Other SCCs			System Frequency
	SCC	Mode	Frequency Ratio	SCC	Mode	Frequency Ratio	
1 ATM channel + 3 HDLC channels	SCC1	ATOM1 No Scrambling	1:12 (note 1)	SCC2/3/4	HDLC	1:8	25 MHz / 33 MHz
	SCC1	ATOM1 Scrambling	1:12 (note 1)	SCC2/3/4	HDLC	1:9	
	SCC1	ATOM1 No Scrambling	1:3.5	SCC2/3/4	HDLC	1:390 (note 2)	
	SCC1	ATOM1 Scrambling	1:5	SCC2/3/4	HDLC	1:390 (note 2)	
1 ATM + 1 Ethernet channel	SCC1	ATOM1 No Scrambling	1:6	SCC2	Ethernet	1:2.5	25 MHz
	SCC1	ATOM1 Scrambling	1:7	SCC2	Ethernet	1:2.5	
1 ATM, 1 Ethernet + 1 HDLC channel	SCC1	ATOM1 No Scrambling	1:12 (note 1)	SCC2	Ethernet	1:2.5	25 MHz
				SCC3	HDLC	1:4	
	SCC1	ATOM1 Scrambling	1:12 (note 1)	SCC2	Ethernet	1:2.5	
				SCC3	HDLC	1:5	
1 ATM + 1 Ethernet channel	SCC1	ATOM1 No Scrambling	1:3.5	SCC2	Ethernet	1:3.3	33 MHz
	SCC1	ATOM1 Scrambling	1:4	SCC2	Ethernet	1:3.3	
1 ATM, 1 Ethernet + 1 HDLC channel	SCC1	ATOM1 No Scrambling	1:16 (note 3)	SCC2	Ethernet	1:3.3	33 MHz
				SCC3	HDLC	1:4	
	SCC1	ATOM1 Scrambling	1:16 (note 3)	SCC2	Ethernet	1:3.3	
				SCC3	HDLC	1:5	

Notes:

1. A frequency ratio of 1:12 equates to 2.048 MHz at 25 MHz.
2. A frequency ratio of 1:390 equates to 64 KHz at 25 MHz.
3. A frequency ratio of 1:16 equates to 2.048 MHz at 33 MHz.

**Table 9: ATOM1 and other Protocol Performance**

This data traffic in the experiments represents typical worst case scenarios for real life applications where QUICC data structures were organised to maximise performance. All frames were restricted to single buffers that were aligned to long-word addresses and receive frame address comparisons were disabled. The Ethernet figures quoted in Table 8 are for half duplex operation<sup>1</sup>.

The frequency ratios quoted are between the main QUICC clock and SCC bit clock.

### 9.1. Command Execution Latency

When ATOM1 is operating, typical and worst-case command execution latencies are slightly longer than when only the standard ROM microcodes are running.

Note: The ATOM1 Restart Reception command is not executed until ATOM1 processes the end of the current cell being received. Therefore command latency for the Restart Reception command depends on when the command is given and data rate.

### 9.2. CRC-32 Calculation Execution

Table 10 shows the results of some performance characterisation experiments where ATOM1 is running on one or more SCCs while CRC-32 calculations are also running. The QUICC's CP runs the CRC-32 calculator as its lowest priority task so the duration of a calculation varies depending on the exact loading and state of the CP. The CRC-32 calculations were run over a buffer of 44 bytes representing the AAL5 payload of a single cell AAL5 frame with start of the data buffer aligned to a longword address. All results shown in Table 10 are for system operation at 25MHz and can be scaled for operation at other frequencies.

Configuration	ATOM1 SCCs	Mode	Frequency Ratio	CRC-32 Calculation Duration
1 ATM	SCC1	ATOM1 - No Scrambling	1:12 (note 1)	20 - 24 μs
1 ATM	SCC1	ATOM1 - Scrambling	1:12 (note 1)	20 - 28 μs
2 ATM	SCC1, 2	ATOM1 - No Scrambling	1:12 (note 1)	28 - 37 μs
2 ATM	SCC1, 2	ATOM1 - Scrambling	1:12 (note 1)	52 - 72 μs
3 ATM	SCC1, 2, 3	ATOM1 - No Scrambling	1:12 (note 1)	52 - 76 μs
3 ATM	SCC1, 2, 3	ATOM1 - No Scrambling	1:25 (note 2)	23 - 30 μs
3 ATM	SCC1, 2, 3	ATOM1 - Scrambling	1:25 (note 2)	27 - 52 μs
4 ATM	SCC1, 2, 3, 4	ATOM1 - No Scrambling	1:25 (note 2)	26 - 34 μs
4 ATM	SCC1, 2, 3, 4	ATOM1 - Scrambling	1:25 (note 2)	40 - 65 μs

Notes:

1. A frequency ratio of 1:12 equates to 2.083 MHz at 25 MHz.
2. A frequency ratio of 1:25 equates to 1 MHz at 25 MHz.

**Table 10: CRC-32 Calculator Performance**

---

1. The experimental Ethernet data rate was 5 Mbps full duplex which equates to 10 Mbps on a conventional half duplex Ethernet system.

## 10. ATOM1 DISK STRUCTURE

ATOM1 is shipped on a PC format disk. Figure 14 shows the disk structure. The microcode is held in the UCODE directory and demonstration software is stored in the DEMO directory. ATOM1.SRX is the S-record microcode file for the QUADS board memory map. ATOM1.SRC is an M68000 assembly code file for the ATOM1 microcode that can be assembled and linked to any system memory map.

The file ATOM1.C contains the C source code for the initialisation and ping-pong demonstration software given in Appendix B. ATOM1 and ATOM1.CMD are the makefiles for the C demonstration software and ATOM1.X is the S-record file of the demonstration software, ready to download to a QUADS board.

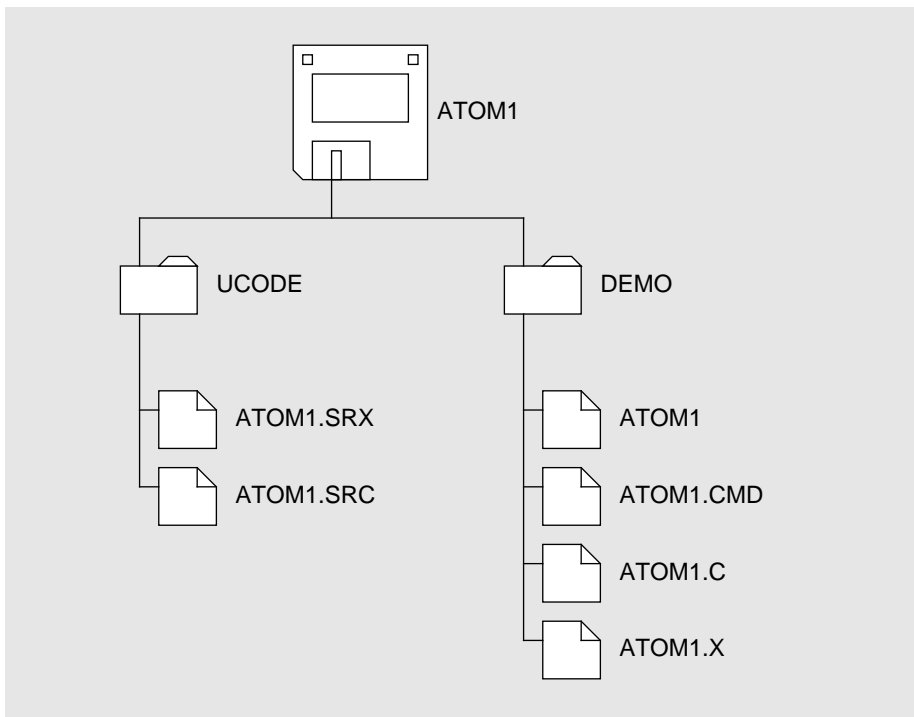


Figure 14. ATOM1 Disk Structure

## 11. MC68360 MASK SET APPLICABILITY

ATOM1 microcode operates on revision B or later MC68360s and MC68EN360s. Note: there are slight differences to the initialisation software for revision B1 and later devices (see section 3.1.2). The ATOM1 microcode cannot be used in conjunction with the QMC protocol for multi-channel operation on an MC68MH360 due to on-chip RAM space constraints.

## **12. ORDERING INFORMATION**

The ATOM1 package contains relocatable ATOM1 object code, object code for the QUADS board, a User's Manual and demonstration software for the QUADS board. ATOM1 is available on DOS media by using the following part number:

M68360SWATOM1-DOS    PC disk.

## **13. REFERENCES**

1. MC68360 User's Manual, MC68360UM/AD Rev 1, Motorola, 1995.
2. ITU I.432, B-ISDN User Network Interface - Physical Layer Specification, 1991.
3. ATM Forum User Network Interface Specification, version 3.0, 1993.
4. Draft Recommendation ITU G.804, ATM Cell Mapping into Plesiochronous Digital Hierarchy (PDH), 1993.
5. ITU I.363, B-ISDN ATM Adaptation Layer Specification.

# Appendix 1. STATE MACHINE DETAILS

## A1.1 ATOM1 Transmitter

Figure 15 shows the ATOM1 transmitter procedure. After start-up when MRBLR is zero, an empty cell is transmitted before the first transmit BD is polled. If the BD has the Skip bit set, an empty cell is transmitted and the transmitter moves to the next BD. If the BD is ready, a user defined ATM cell is transmitted, otherwise another empty cell is transmitted. Operation continues in this manner until the Stop Transmit command or a FIFO underrun error occurs or the user disables the SCC transmitter. The transmitted cells may have their payloads scrambled as described below.

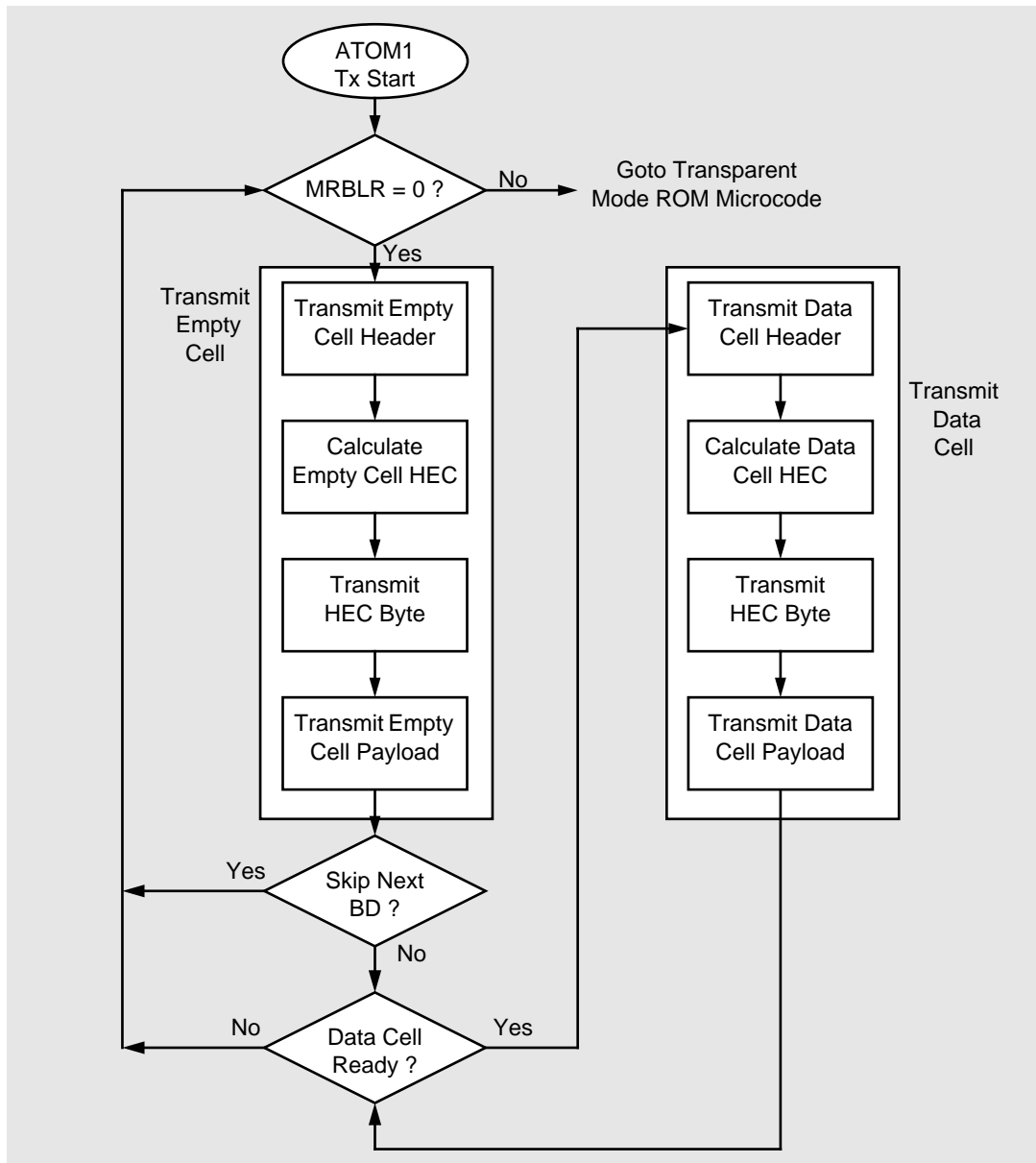
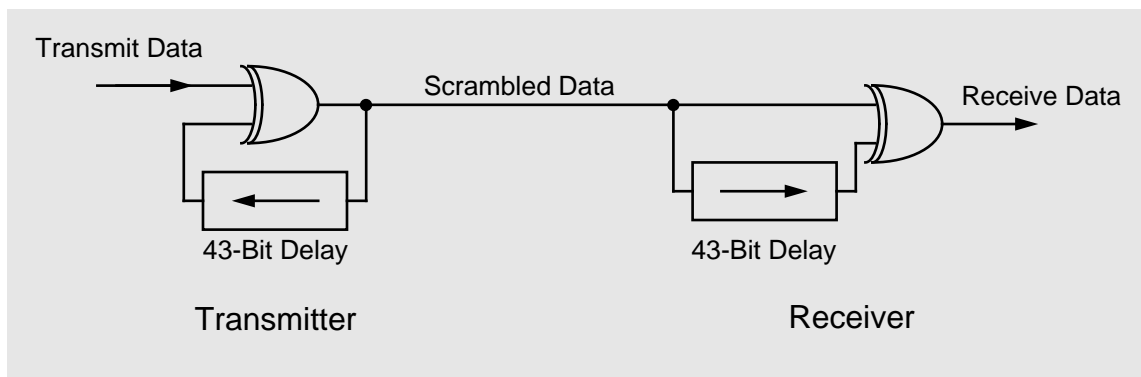


Figure 15. ATOM1 Transmitter Procedure



## A1.2 ATM Cell Payload Scrambling

ATOM1 provides a scrambling option on a per line basis for cell payload bytes using the polynomial  $X^{43}+1$ . Figure 16 illustrates the payload transmitter and receiver scrambling mechanism used by ATOM1.



**Figure 16. ATM Cell Payload Scrambling Mechanism**

The context of the transmit and receive scrambling mechanism is independent of the cell's virtual connection. The seed value for the scrambling algorithm for a particular cell is obtained from the 43 cell payload bits transmitted or received immediately prior to the current cell. Two consecutive cells may be from different connections and scrambling mechanism ignores this.

## A1.3 ATOM1 Receiver

Figure 17 shows the ATOM1 receiver procedure. After start-up when MRBLR is zero, the HEC delineation procedure begins (see below). Once complete, cell reception commences.

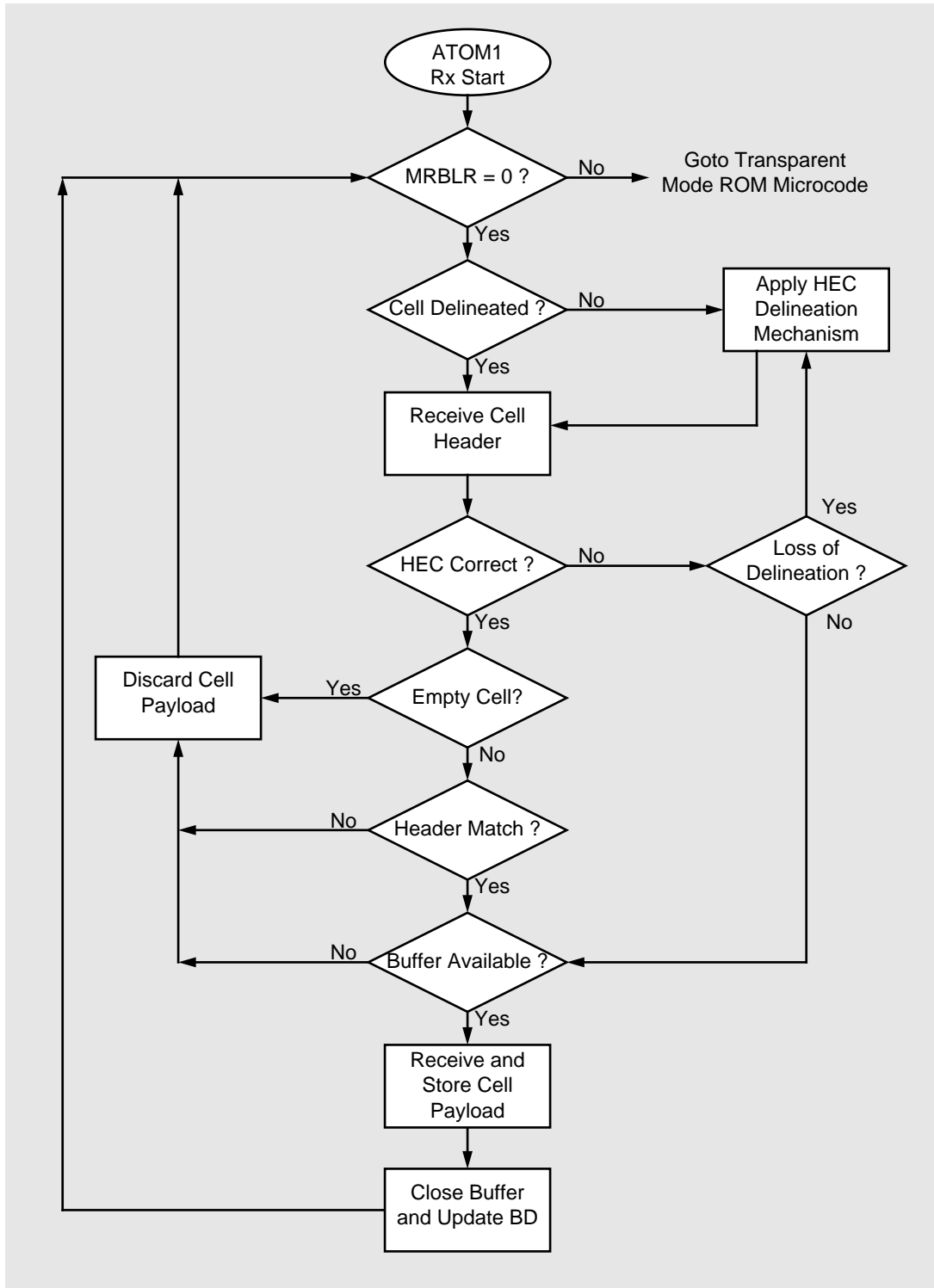
For each cell received, the HEC is checked and if incorrect the cell is usually received. It is not received only when the receiver loses cell delineation. In the cell is an empty cell, the payload is discarded. If there is a header match using internal the header lookup table then the cell payload is received. When the external CAM option is enabled, the cell is received before the CAM match is checked.

Operation continues in this manner until the Restart Reception command or a FIFO overrun error occurs or the user disables the SCC receiver. Received cells may have their payloads descrambled as described earlier.

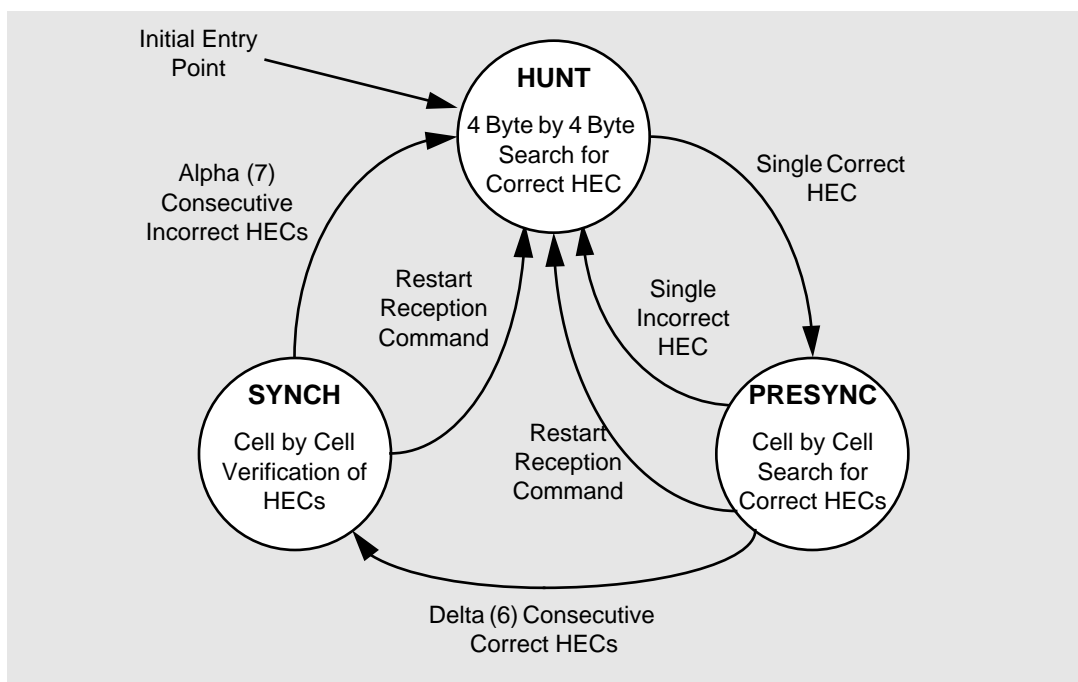
## A1.4 HEC Delineation Mechanism

ATOM1 applies the cell delineation mechanism specified in I.432 and shown in Figure 18 to synchronise to the incoming cell stream. The SDH-based physical layer values for alpha and delta, 7 and 6 respectively, are used by ATOM1.

At start-up, the hunt mode is entered where the CRC-8 value is calculated on each incoming longword. When this matches the next received byte, it is assumed that the HEC is found and the state machine advances to the presync state. Because the hunt state only finds a HEC field that is aligned to the start of reception, it can take up to four correct cells to leave the hunt state.



**Figure 17. ATOM1 Receiver Procedure**



**Figure 18. Cell Delineation State Diagram**

I.432 specifies that the hunt state searches for a correct HEC on a bit-by-bit basis but ATOM1 uses a longword by longword search. For G.804 framing of ATM cells, the HEC field is always aligned with the E1 or T1 frame octet timeslots so bit-by-bit searching is not required. To ease implementation, this was taken a stage further and ATOM1 uses a longword by longword search. Over the course of four cells received (212 bytes) with correct HECs, there will be one matching and aligned HEC. This is the point where the state machine jumps to the presync state.

In the presync state, ATOM1 has detected cell boundaries and is verifying that they are correct. ATOM1 leaves the presync state after a single cell with an incorrect HEC is received or the Restart Reception command is given. When six consecutive cells with valid HECs are received, ATOM1 jumps to the sync state and begins reception. The state machine advances immediately after reception of the sixth correct HEC and is not delayed until the end of the cell. This means that five cells with valid HECs are discarded and the sixth is received into a data buffer.

In the synch state, ATOM1 receives data and control cells and discards incoming empty cells. When a stream of cells with incorrect HECs are received, the first six consecutive cells are received and on reception of the seventh incorrect HEC, ATOM1 immediately jumps to the hunt state and does not receive the cell with the seventh incorrect HEC. ATOM1 also jumps to the hunt state after the Restart Reception command.

When cell payload scrambling is enabled, the descrambler will self-synchronise with the incoming data after 43 bits. Hence, after starting the receiver, the descrambler will attain synchronisation during reception of the first cell in the presync state.

## Appendix 2. ATOM1 DEMONSTRATION SOFTWARE

This appendix contains demonstration software for ATOM1. The software is assembled for use on the QUADS board and assumes that the ATOM1 microcode is downloaded separately.

Before running this software, the user should either link two QUADS boards, both running ATOM1, or externally loopback the master MC68360 transmitter and receiver on one QUADS board.

```
/******  
* ATOM1 Microcode - *  
* Initialisation and Ping-Pong Software. *  
*****  
* Last Modified : 21 March 1996 Rev : 1 *  
*****  
* Filename: atom1.c *  
*****  
* <C> Copyright Motorola Inc 1996 *  
* Written by Kevin Godfrey, Motorola EKB. *  
*****  
* Revision History: *  
* 0 Initial software. *  
* 1 CRC-32 Calculator and buffer scattering added. *  
*****  
* Associated Documentation: *  
* ATOM1 User's Manual DATE: 21 March 1996 REV: 1.0 *  
*****  
* Target System: Motorola QUADS board running QUICCCbug. *  
* The target system must use an MC68360 or MC68EN360 Rev B or *  
* later. *  
*****/
```

```
/******  
* This software initialises SCC1 to transmit and receive ATM *  
* cells. SCC1 runs ATOM1 microcode. *  
* *  
* Ping-Pong demonstration code is included to demonstrate simple *  
* receive and transmit buffer handling and CRC-32 calculations. *  
* *  
* The ATOM1 microcode is loaded by other means before running *  
* this program. *  
* *  
* To demonstrate ATOM1 receiver, an external loopback is required *  
* between L1TXDA and L1RXDA or the QUADS board can be connected *  
* to another QUICC running ATOM1 microcode. *  
* *  
* External clock and synchronisation signals are required. *  
*****/
```

```
/****** Definitions *****/
```

```
typedef unsigned char ubyte;  
typedef unsigned short uword;  
typedef unsigned long ulong;
```

```

/* Microtec compiler treats char variables as bytes, short integers */
/* as 32-bit quantities and long integers as 32-bit quantities.      */

/***** QUADS Board memory Map *****/

#define QUICC 0x20000 /* QUICC address = 0x20000 */
#define BDArea 0x20400 /* 256 bytes of BD space starts at 0x20400 */
#define ARXBD0 0x20400 /* ATOM1 rx BDs start at 0x20400 */
#define ATXBD0 0x20440 /* ATOM1 tx BDs start at 0x20440 */
#define TXBUFF 0x4f0000 /* ATOM1 tx Buffer page starts at 0x4f0000 */
#define RXBUFF 0x4e0000 /* ATOM1 rx Buffers start at 0x4e0000 */

/***** ATOM1 Demo Buffer Sizes, Limits, etc. *****/

#define BUFFSIZE 52 /* ATM buffer size 52 bytes */
#define CELLSIZE 53 /* ATM cell length 53 bytes */
#define MAXBUFF 9 /* Maximum number of buffers = 10 */
#define SI_RAM_SIZE 128 /* 128 words of SI RAM */
#define MAXBDAREA 128 /* 256 bytes of BD space */
#define NO 0
#define YES 1

/***** Define QUICC Registers and Parameter RAM Data Structures *****/

struct CP_regs {
    volatile uword CR ; /* Command Register */
    uword reserved1 ;
    volatile uword RCCR ; /* RISC Configuration Register */
    uword reserved2 ;
    uword reserved3 ;
    volatile uword CPC1 ; /* CP Control Register 1 */
    volatile uword CPC2 ; /* CP Control Register 2 */
    volatile uword CPC3 ; /* CP Control Register 3 */
    volatile uword CPC4 ; /* CP Control Register 4 */
    uword reserved4 ;
    volatile uword RTE ; /* RISC Timers Event Register - used by CRC-32 calculator */
    volatile uword RTMR ; /* RISC Timers Mask Register */
} ;

struct Int_regs {
    volatile uword CICR ; /* CP Interrupt Control Register */
    volatile uword CIPR ; /* CP Interrupt Pending Register */
    volatile uword CIMR ; /* CP Interrupt Mask Register */
    volatile uword CISR ; /* CP Interrupt In-Service Register */
} ;

struct PortA_regs {
    volatile uword PADIR ; /* Port A Data Direction Register */
    volatile uword PAPAR ; /* Port A Pin Assignment Register */
    volatile uword PAODR ; /* Port A Open Drain Register */
    volatile uword PADAT ; /* Port A Data Register */
} ;

struct PortC_regs {
    volatile uword PCDIR ; /* Port C Data Direction Register */
    volatile uword PCPAR ; /* Port C Pin Assignment Register */
    volatile uword PCSO ; /* Port C Special Option Register */
    volatile uword PCDAT ; /* Port C Data Register */
    volatile uword PCINT ; /* Port C Interrupt Control Register */
} ;

```

```

    } ;

struct PIP_regs {
    volatile uword PIPC      ; /* PIP Configuration Register */
        uword reserved1 ;
    volatile uword PTPR     ; /* PIP Timing Parameter Register */
    volatile ulong PBDIR   ; /* Port B Data Direction Register */
    volatile ulong PBPAR   ; /* Port B Pin Assignment Register */
    volatile uword PBODR   ; /* Port B Open Drain Register */
        uword reserved2 ;
    volatile ulong PBDAT   ; /* Port B Data Register */
    } ;

struct SI_regs {
    volatile ulong SIMODE   ; /* SI mode register */
    volatile ubyte SIGMR   ; /* SI global mode register */
        ubyte reserved1 ;
    volatile ubyte SISTR   ; /* SI status register */
    volatile ubyte SICMR   ; /* SI command register */
        ulong reserved2 ;
    volatile ulong SICR    ; /* SI clock route */
        uword reserved3 ;
    volatile ulong SIRP    ; /* SI RAM pointer */
    } ;

struct SDMA_regs {
    volatile uword SDCR     ; /* SDMA Configuration Register */
    volatile ulong SDAR    ; /* SDMA Address Register */
    } ;

struct SCC_regs {
    volatile ulong GSMRL   ; /* SCC General Mode Register - lower long-word */
    volatile ulong GSMRH   ; /* SCC General Mode Register - upper long-word */
    volatile uword PSMR    ; /* SCC protocol specific mode register */
        uword reserved1 ;
    volatile uword TODR    ; /* SCC tx on demand register */
    volatile uword DSR     ; /* SCC data sync register */
    volatile uword SCCE    ; /* SCC event register */
        uword reserved2 ;
    volatile uword SCCM    ; /* SCC mask register */
        ubyte reserved3 ;
    volatile ubyte SCCS    ; /* SCC status register */
    } ;

/* ATOM1 SCC Parameter RAM */

struct ATOM1_Params {
    volatile uword RBASE   ; /* ATOM1 rx BD Base Address */
    volatile uword TBASE   ; /* ATOM1 tx BD Base Address */
    volatile ubyte RFCR    ; /* ATOM1 SDMA rx Function Code */
    volatile ubyte TFCR    ; /* ATOM1 SDMA tx Function Code */
    volatile uword MRBLR   ; /* Transparent receiver max buffer length */
    volatile ulong RSTATE  ; /* ATOM1 rx internal state storage */
    volatile ulong R_PTR   ; /* ATOM1 rx data pointer */
    volatile uword RBD_PTR ; /* ATOM1 rx BD Pointer */
    volatile uword R_CNT   ; /* ATOM1 rx counter */
    volatile ulong RTEMP   ; /* Receiver Temporary Storage */
    volatile ulong TSTATE  ; /* ATOM1 tx internal state storage */
    volatile ulong T_PTR   ; /* ATOM1 tx data pointer */
    volatile uword TBD_PTR ; /* ATOM1 tx BD pointer */
    volatile uword T_CNT   ; /* ATOM1 tx counter */

```

```

volatile ulong TTEMP      ; /* Transmitter Temporary Storage */
volatile uword ALPHA     ; /* Receiver Synchronisation Alpha Counter */
volatile uword DELTA     ; /* Receiver Synchronisation Delta Counter */
volatile uword AVERSION  ; /* ATOM1 version number */
volatile uword CAM_PORT  ; /* ATOM1 CAM Port Mask */
volatile ulong RX_BUFF1  ; /* Receive Buffer Area Start Pointer */
volatile ulong TX_BUFF1  ; /* Transmit Buffer Page Start Pointer */
volatile ulong EHEAD     ; /* Empty Cell Header */
volatile ulong EPAYLOAD  ; /* Empty Cell Payload */
volatile ulong RSTUFF    ; /* Receive Data Stuffing Location */
volatile uword SHUFFLESTATE ; /* Receive Data Stuffing State */
volatile uword RHECTEMP  ; /* Temporary HEC storage area */
volatile uword THECTEMP  ; /* Temporary HEC storage area */
volatile uword ASTATUS   ; /* ATOM1 status word */
volatile ulong NMA_HEAD  ; /* Non-Matching Address Cell Header */
volatile uword NMA_CNT   ; /* Non-Matching Address Cell Counter */
volatile uword HEC_ERR   ; /* HEC Error Counter */
volatile ulong RSCRAM    ; /* Receiver Scrambling Storage */
volatile ulong RSCRAM1   ; /* Receiver Scrambling Storage */
volatile ulong TSCRAM    ; /* Transmitter Scrambling Storage */
volatile ulong TSCRAM1   ; /* Transmitter Scrambling Storage */
volatile ulong HEADMASK  ; /* Cell Header Comparison Mask */
volatile ulong HEAD1     ; /* Cell Header Look-Up Table Entry 1 */
volatile ulong HEAD2     ; /* Cell Header Look-Up Table Entry 2 */
volatile ulong HEAD3     ; /* Cell Header Look-Up Table Entry 3 */
volatile ulong HEAD4     ; /* Cell Header Look-Up Table Entry 4 */
volatile ulong HEAD5     ; /* Cell Header Look-Up Table Entry 5 */
volatile ulong HEAD6     ; /* Cell Header Look-Up Table Entry 6 */
volatile ulong HEAD7     ; /* Cell Header Look-Up Table Entry 7 */
volatile ulong HEAD8     ; /* Cell Header Look-Up Table Entry 8 */
volatile ulong HEAD9     ; /* Cell Header Look-Up Table Entry 9 */
volatile ulong HEAD10    ; /* Cell Header Look-Up Table Entry 10 */
volatile ulong HEAD11    ; /* Cell Header Look-Up Table Entry 11 */
volatile ulong HEAD12    ; /* Cell Header Look-Up Table Entry 12 */
volatile ulong HEAD13    ; /* Cell Header Look-Up Table Entry 13 */
volatile ulong HEAD14    ; /* Cell Header Look-Up Table Entry 14 */
volatile ulong HEAD15    ; /* Cell Header Look-Up Table Entry 15 */
volatile ulong HEAD16    ; /* Cell Header Look-Up Table Entry 16 */
} ;

struct CRC32_calculator {
    volatile ulong CRC_PTR      ; /* CRC-32 calculator pointer */
    volatile uword CRC_CNT      ; /* CRC-32 calculator counter register */
    volatile ubyte CRC_FC       ; /* CRC-32 calculator Function Code */
    ubyte reserved1 ;
    volatile ulong CRC_RESULT   ; /* CRC-32 calculator preset and result register */
} ;

/* Define Pointers to QUICC Registers and SCC Parameter RAM */

struct CP_regs      *CP      ;
struct Int_regs     *Intr    ;
struct PortA_regs   *PortA   ;
struct PortC_regs   *PortC   ;
struct PIP_regs     *PIP     ;
struct SI_regs      *SI      ;
struct SDMA_regs    *SDMA    ;
struct SCC_regs     *SCC1    ;
struct ATOM1_Params *ATOM1_1 ;

```

```

struct CRC32_calculator *CRC32 ;

uword *SIRAM ; /* Pointer to 256 bytes of SI routing RAM */
ulong *BRGC3 ; /* Baud Rate Generator 3 control register */
uword *BDSpace ; /* Pointer to start of 256 byte BD space */
uword *ATXBD ; /* Pointer to ATOM1 Tx BD Queue */
uword *ARXBD ; /* Pointer to ATOM1 Rx BD Queue */
ubyte *ATXBuff ; /* Pointer to ATOM1 Tx data queue */
ubyte *ARXBuff ; /* Pointer to ATOM1 Rx data queue */

main()
{
/* Local Variables */

int buffnum ; /* General buffer counting variable (32-bit) */
int bytecnt ; /* General byte counting variable (32-bit) */
int count ; /* General counting variable (32-bit) */
char dataval ; /* General 8-bit data value */
int txcount = 0, rxcount = 0 ; /* Ping-Pong receive and transmit cell counters */
int ppperror = 0 ; /* Ping-pong error flag */

/* Initialise Memory Pointers */

CP = (struct CP_regs *) (QUICC+0x15c0) ;
Intr = (struct Intr_regs *) (QUICC+0x1540) ;
PortA = (struct PortA_regs *) (QUICC+0x1550) ;
PortC = (struct PortC_regs *) (QUICC+0x1560) ;
PIP = (struct PIP_regs *) (QUICC+0x16b2) ;
SI = (struct SI_regs *) (QUICC+0x16e0) ;
SDMA = (struct SDMA_regs *) (QUICC+0x151e) ;
SCC1 = (struct SCC_regs *) (QUICC+0x1600) ;
ATOM1_1 = (struct ATOM1_Params *) (QUICC+0x0c00) ;
CRC32 = (struct CRC32_calculator *) (QUICC+0x0db0) ;

SIRAM = (uword *) (QUICC+0x1700) ;
BRGC3 = (ulong *) (QUICC+0x15f8) ;
ATXBD = (uword *) ATXBD0 ;
ARXBD = (uword *) ARXBD0 ;
ATXBuff = (ubyte *) TXBUFF ;
ARXBuff = (ubyte *) RXBUFF ;
BDSpace = (uword *) BDArea ;

/* Configure QUICC Parallel Ports */

config_parallel_ports() ;

/* Reset QUICC CPM */

CP->RCCR = 0 ; /* Clear RCCR */
CP->CR = 0x8001 ; /* Reset CPM */

/* Configure BRG3 for external use */

*BRGC3 = 0x0001000a ; /* BRG3 = 4.17MHz clock at 25MHz */

```



```

/* Configure SDMA */

    SDMA->SDCR = 0x0740 ; /* SDMA configuration register */

/* Configure Serial Interface and TSA */

    config_tsa() ;

/* Disable all interrupts */

    Intr->CIMR = 0 ;

/* Initialise and enable RAM microcode */

    config_ucose() ;

/* Configure SCC1 for ATOM1 */

    atom_init(SCC1, ATOM1_1) ;
    CP->CR = 0x0001          ; /* Initialise SCC1 rx and tx parameters */

/* Configure ATOM1 Data Buffers & Buffer Descriptors */

/* Clear out the whole BD area */

    for (count=0 ; count<MAXBDAREA ; count++)
        *(BDSpace + count) = 0 ;

/* Clear and set up ten empty receive BDs. */
/* All are made ready and the wrap bit in tenth BD is set. */
/* The first nine buffers are actually in contiguous memory starting at the */
/* begining of the receive buffer area. The tenth buffer at the end of the */
/* receive buffer area. */

    for (count=0 ; count<MAXBUFF ; count++)
        BD_init((ARXBD + count), NO, YES, NO, RXBUFF, RXBUFF + count*52) ;
    BD_init((ARXBD + MAXBUFF), NO, YES, YES, RXBUFF, RXBUFF + 0x3ffc) ;

/* Clear receive buffer area */

    for (bytecnt=0 ; bytecnt<(MAXBUFF*BUFFSIZE) ; bytecnt++)
        *(ARXBuff + bytecnt) = 0 ;

/* Clear and set up ten transmit BDs. */
/* All are made ready and the wrap bit in tenth BD is set. */
/* The first nine buffers are actually in contiguous memory starting at the */
/* begining of the transmit buffer area. The tenth buffer at the end of the */
/* transmit buffer area. */

    for (count=0 ; count<MAXBUFF ; count++)
        BD_init((ATXBD + count), NO, YES, NO, TXBUFF, TXBUFF + count*52) ;
    BD_init((ATXBD + MAXBUFF), NO, YES, YES, TXBUFF, TXBUFF + 0x3ffc) ;

```

```

/* Fill transmit buffer area with ascending numbers */

    dataval=1 ;
    for (bytecnt=0 ; bytecnt<(MAXBUFF*BUFSIZE) ; bytecnt++)
        *(ATXBuff + bytecnt) = dataval++ ;

/* Initialisation is complete so enable SCC1 for ATM Reception. */

    SCC1->GSMRL = 0x00000030 ; /* SCC1 Enabled */

/* Initialise the CRC-32 calculator and perform one sample calculation over the transmit */
/* buffer that was just created. */

    CRC32->CRC_PTR = TXBUFF ;
    CRC32->CRC_CNT = 48 ;
    CRC32->CRC_FC = 0x18 ;
    CRC32->CRC_RESULT = 0 ; /* Result register must be preset to zero at the start */

    CP->RTER = 0xffff ; /* Clear RTER so that no interrupts are pending. */
    /* RTER is used to indicate that a CRC-32 calculation has completed. */

    CP->CR = 0x0f01 ; /* Start the CRC-32 calculation. */

/* Return to QUADS debugger. TEMPORARY ENDING AFTER CONFIGURATION */

    asm(" trap #15");
    asm(" dc.w $0063");

/* Now Enter the ATOM1 Ping-Pong Demonstration. */

/***** ATOM1 PING-PONG SOFTWARE *****/

/*****
/* This routine uses SCC1 to transmit and receive ATM cells in a game */
/* of ping-pong. It is assumed that SCC1 is either connected with an */
/* external loopback or to another QUADS board also running this */
/* ping-pong software. */
/* */
/* SCC1 must be initialised before running this software. */
/* */
/* The PA15 pin is driven high by software for the duration of the */
/* single cell CRC-32 calculations to allow an external counter or */
/* oscilloscope to measure the CRC-32 calculation duration. */
*****/

/* Tx enough frames to empty the receive queue. */
/* The tx queue was emptied by the initialisation code. */

/* First transmit enough frames to empty the receive queue */

    while (ATOM1_1->RBD_PTR != ATOM1_1->RBASE)
        BD_init(ATXBD, NO, NO, YES, TXBUFF, TXBUFF + 0) ;

```

```

/* Configure tx buffer area with AAL5 sample frame given in I.363 appendix. */

    (ulong *) *(ATXBuff + 0) = 0x12345678 ; /* cell header = 0x12345678 */

/* AAL5 single cell frame payload consists 40 bytes of zero, CPCS-UU = 0, */
/* length = 40 and CRC32 = 0x864d7f99 */

    for (bytecnt=4 ; bytecnt<44 ; bytecnt++)
        *(ATXBuff + bytecnt) = 0x00 ; /* AAL5 frame payload */
    (ulong *) *(ATXBuff + bytecnt) = 0x00000028 ; /* AAL5 frame control long word */
    bytecnt = bytecnt + 4 ;
    (ulong *) *(ATXBuff + bytecnt) = 0x864d7f99 ; /* AAL5 frame CRC */

    pperror = 0 ;
    while (pperror == 0)
    {
        SCC1->SCCE = 0xffff ; /* Clear SCCE1 */

/* Set up one rx BD with Int and Wrap bits set */

        BD_init(ARXBD, NO, YES, YES, RXBUFF, RXBUFF + 0) ;

/* Tx one ATM cell - BD Int and Wrap bits set */

        BD_init(ATXBD, NO, YES, YES, TXBUFF, TXBUFF + 0) ;

        txcount++ ; /* Increment transmit cell counter */

        while ((SCC1->SCCE & 0x007d) == 0x0000) ; /* Wait for reception or error */

        if ((SCC1->SCCE & 0x00f4) == 0x0000) /* On error, set error flag. */
            rxcount++ ; /* Otherwise increment rx counter. */
        else pperror = 1 ;

/* Check that received cells have the correct AAL5 CRC-32 value */

        CRC32->CRC_PTR = RXBUFF + 4 ;
        CRC32->CRC_CNT = 44 ;
        CRC32->CRC_FC = 0x18 ;
        CRC32->CRC_RESULT = 0 ; /* Result register must be preset to zero at the start */
        CP->RTER = 0x8000 ;
        /* RTER is used to indicate that a CRC-32 calculation has completed. */
        CP->CR = 0x0f01 ; /* Start the CRC-32 calculation. */
        PortA->PADAT = 0x8000 ; /* Set PA15 pin */

        while (CP->CR != 0x0f00) ; /* Wait for the CP command to complete */

        while (CP->RTER != 0x8000) ; /* Wait for the CRC32 calculator to complete */
        PortA->PADAT = 0x0000 ; /* Reset PA15 pin */

/* If incorrect CRC32 receiver, set error flag */
        if (CRC32->CRC_RESULT != 0x864d7f99 ) pperror = 1 ;

    }

/* Return to QUADS debugger. */

```

```

asm(" trap #15");
asm(" dc.w $0063");

} /* End of Main */

/*****
/*  QUICC PARALLEL PORT CONFIGURATION          */
/*****
/*  Function name:  config_parallel_ports      */
/*****
/*  Configure QUICC Parallel Ports as follows: */
/*  L1TXDA   (PA6)   Transmit data output     */
/*  L1RXDA   (PA7)   Receive data input       */
/*  L1RSYNCA (PC11)  Synchronisation input    */
/*  L1RCLKA  (PA8)   Clock input              */
/*  L1ST1    (PB12)  TSA strobe output        */
/*  L1ST2    (PB13)  TSA strobe output        */
/*  L1ST3    (PB14)  TSA strobe output        */
/*  L1ST4    (PB15)  TSA strobe output        */
/*  SDACK1*  (PC10)  SDMA cycle output        */
/*  SDACK2*  (PC8)   SDMA cycle output        */
/*  PA15     (PA15)  CRC-32 PIO output pin for software use */
/*  BRGO3    (PB16)  BRG3 Output Signal      */
/*****
/*  Input Parameters:  global data structures PIP, PortA and PortC */
/*  */
/*  Output Parameters:  PIP, PortA and PortC modified              */
/*  */
/*****

config_parallel_ports()
{
    PIP->PIPC = 0x0000    ;    /* PIP configuration register */

    PortA->PADAT = 0x0000 ;    /* Port A data register */
    PortA->PADIR = 0x80c0 ;    /* Port A data direction register */
    PortA->PAPAR = 0x01c0 ;    /* Port A pin assignment register */
    PortA->PAODR = 0x0000 ;    /* Port A open drain register */

    PortC->PCDIR = 0x0500 ;    /* Port C data direction register */
    PortC->PCPAR = 0x0d00 ;    /* Port C pin assignment register */
    PortC->PCSO = 0x0000 ;    /* Port C social option register */
    PortC->PCINT = 0x0000 ;    /* Port C interrupt control register */

    PIP->PBDIR = 0x00010000 ; /* Port B data direction register */
    PIP->PBPAR = 0x0001f000 ; /* Port B pin assignment register */
    PIP->PBODR = 0x0000    ;    /* Port B open drain register */

} /* End of config_parallel_ports */

/*****
/*  QUICC TSA CONFIGURATION & INITIALISATION          */
/*****

```

```

/* Function name:  config_tsa                                     */
/*****/
/* Configure Serial Interface and TSA as follows:                */
/* TDMA used to tx and rx 53 byte ATM cells via SCC1 and SCC2.  */
/* Common rx and tx clock and synchronisation signals.          */
/* Tx data on falling clock edges and rx on rising edges.       */
/* Frame sync signal recognised on a rising clock edge.         */
/*****/
/* Input Parameters:  global data structure SI and pointer SIRAM */
/*                                                            */
/* Output Parameters:  SI modified                             */
/*                                                            */
/*****/

config_tsa()
{
int count ;
/* SI registers: */

    SI->SIGMR = 0x00 ;      /* Disable SI */
    SI->SIMODE = 0x00000058 ; /* SI mode register without loopback */
/* SI->SIMODE = 0x00000858 ;  SI mode register with TSA loopback */
    SI->SICR = 0x00004040 ; /* SI clock route */

/* Clear SI RAM: */

    for (count=0 ; count<SI_RAM_SIZE ; count++)
        *(SIRAM + count) = 0x0000 ;

/* Configure SI RAM:
    LlST1 asserted during the first 4 tx bytes (cell header except HEC).
    LlST2 asserted during the fifth tx byte (HEC).
    LlST3 asserted during the first 4 rx bytes (cell header except HEC).
    LlST4 asserted during the fifth rx byte (HEC). */

    count=0 ;
    *(SIRAM + count++) = 0x104e ; /* Rx cell header - SCC1          */
    *(SIRAM + count++) = 0x2042 ; /* Rx cell HEC - SCC1          */
    *(SIRAM + count++) = 0x007e ; /* Rx cell payload - 16 bytes - SCC1 */
    *(SIRAM + count++) = 0x007e ; /* Rx cell payload - 16 bytes - SCC1 */
    *(SIRAM + count++) = 0x007f ; /* Rx cell payload - 16 bytes - SCC1 */

    count=0x40 ;
    *(SIRAM + count++) = 0x044e ; /* Tx cell header - SCC1          */
    *(SIRAM + count++) = 0x0842 ; /* Tx cell HEC - SCC1          */
    *(SIRAM + count++) = 0x007e ; /* Tx cell payload - 16 bytes - SCC1 */
    *(SIRAM + count++) = 0x007e ; /* Tx cell payload - 16 bytes - SCC1 */
    *(SIRAM + count++) = 0x007f ; /* Tx cell payload - 16 bytes - SCC1 */

/* Enable SI */

    SI->SIGMR = 0x04 ;      /* Enable TDMA */

} /* End of config_tsa */

/*****/
/* QUICC RAM MICROCODE CONFIGURATION & INITIALISATION          */
/*****/

```

```

/* Function name:  config_ucose                                     */
/*****
/* Configure CP control registers and enable RAM microcode.      */
/*****
/* Input Parameters:  global data structure CP                   */
/*                                                             */
/* Output Parameters:  CP modified                               */
/*                                                             */
/*****

config_ucose()
{
    CP->CPCR1 = 0x8056 ;    /* CP Control Register 1 */
    CP->CPCR2 = 0x8076 ;    /* CP Control Register 2 */
    CP->CPCR3 = 0x8036 ;    /* CP Control Register 3 */
/* CP->CPCR4 = 0x8e18 ;    CP Control Register 4 - QUICC rev B1 */
    CP->CPCR4 = 0x8818 ;    /* CP Control Register 4 - QUICC rev B2 */

    CP->RCCR = 0x0003 ;    /* Large RAM microcode now running */
} /* End of config_ucose */

/*****
/*  ATOM SCC CONFIGURATION & INITIALISATION                       */
/*****
/* Function name:  atom_init                                       */
/*****
/* This routine initialises an SCC to transmit and receive ATM cells.*/
/* The SCC is configured but not enabled.                          */
/*****
/* Input Parameters:  Pointers to the SCC register, SCC parameter */
/*                   RAM and CP register data structures.         */
/*                   Accesses global data structure CP.           */
/*                                                             */
/* Output Parameters:  CP modified                               */
/*                                                             */
/*****

atom_init(SCC, ATOM)

    struct SCC_regs    *SCC ;
    struct ATOM1_Params *ATOM ;
{
/* SCC registers: */

    SCC->GSMRL = 0                ; /* SCC Disabled */
    SCC->GSMRH = 0x00001F80       ; /* Set sync and transparent bits */
    SCC->PSMR  = 0x2100           ; /* Enable scrambling and HEC coset */
/* SCC->PSMR  = 0                ; /* Disable scrambling and HEC coset */
    SCC->SCCM  = 0                ; /* Mask out interrupts */
    SCC->SCCE  = 0xffff           ; /* Clear SCC event register */

/* Configure SCC parameter RAM for ATOM1: */

    ATOM->RBASE = (ARXBD0-QUICC) ; /* Rx BD Base Address */
    ATOM->TBASE = (ATXBD0-QUICC) ; /* Tx BD Base Address */
    ATOM->RFCR  = 0x18            ; /* SDMA Rx Function Code */
    ATOM->TFCR  = 0x19            ; /* SDMA Tx Function Code */

```

```

ATOM->MRBLR    = 0           ; /* Transparent Receiver Max Buffer Length */
ATOM->ALPHA     = 0           ; /* Receiver Synchronisation Alpha Counter */
ATOM->DELTA     = 0           ; /* Receiver Synchronisation Delta Counter */
ATOM->CAM_PORT  = 0x0001     ; /* CAM Match* signal on PB0 */
ATOM->RX_BUFF1  = RXBUFF     ; /* Receive Buffer Area Start Pointer */
ATOM->TX_BUFF1  = TXBUFF     ; /* Transmit Buffer Page Start Pointer */
ATOM->EHEAD     = 0x01000000 ; /* Empty Cell Header (little-endian) */
ATOM->EPAYLOAD  = 0x6a6a6a6a ; /* Empty Cell Payload */
ATOM->RSTUFF    = 0           ; /* Receive Data Stuffing Location */
ATOM->SHUFFLESTATE = 0       ; /* Receive Data Stuffing State */
ATOM->RHECTEMP  = 0x00ff     ; /* Preset rx temp HEC storage area */
ATOM->THECTEMP  = 0           ; /* Temporary HEC storage area */
ATOM->ASTATUS   = 0           ; /* ATOM1 status word */
ATOM->NMA_HEAD  = 0           ; /* Non-Matching Address Cell Header */
ATOM->NMA_CNT   = 0           ; /* Non-Matching Address Cell Counter */
ATOM->HEC_ERR   = 0           ; /* HEC Error Counter */
ATOM->RSCRAM    = 0           ; /* Receiver Scrambling Storage */
ATOM->RSCRAM1   = 0           ; /* Receiver Scrambling Storage */
ATOM->TSCRAM    = 0           ; /* Transmitter Scrambling Storage */
ATOM->TSCRAM1   = 0           ; /* Transmitter Scrambling Storage */
ATOM->HEADMASK  = 0           ; /* Cell Header Comparison Mask */
ATOM->HEAD1     = 0           ; /* Cell Header Look-Up Table Entry 1 */
ATOM->HEAD2     = 0           ; /* Cell Header Look-Up Table Entry 2 */
ATOM->HEAD3     = 0           ; /* Cell Header Look-Up Table Entry 3 */
ATOM->HEAD4     = 0           ; /* Cell Header Look-Up Table Entry 4 */
ATOM->HEAD5     = 0           ; /* Cell Header Look-Up Table Entry 5 */
ATOM->HEAD6     = 0           ; /* Cell Header Look-Up Table Entry 6 */
ATOM->HEAD7     = 0           ; /* Cell Header Look-Up Table Entry 7 */
ATOM->HEAD8     = 0           ; /* Cell Header Look-Up Table Entry 8 */
ATOM->HEAD9     = 0           ; /* Cell Header Look-Up Table Entry 9 */
ATOM->HEAD10    = 0          ; /* Cell Header Look-Up Table Entry 10 */
ATOM->HEAD11    = 0          ; /* Cell Header Look-Up Table Entry 11 */
ATOM->HEAD12    = 0          ; /* Cell Header Look-Up Table Entry 12 */
ATOM->HEAD13    = 0          ; /* Cell Header Look-Up Table Entry 13 */
ATOM->HEAD14    = 0          ; /* Cell Header Look-Up Table Entry 14 */
ATOM->HEAD15    = 0          ; /* Cell Header Look-Up Table Entry 15 */
ATOM->HEAD16    = 0          ; /* Cell Header Look-Up Table Entry 16 */

```

```

} /* End of atom_init */

```

```

/*****
/*  ATOM1 TRANSMIT & RECEIVE BD INITIALISATION ROUTINE          */
/*****
/*  Function name:      BD_init                                  */
/*****
/*  This routine initialises an ATOM1 transmit or receive BD.   */
/*****
/*  Input Parameters:   Pointer to the BD.                       */
/*                      Flags for Skip, Interrupt and Wrap bits. */
/*                      The addresses of the start of data buffer */
/*                      area and the actual data buffer within that. */
/*                      area to tx                               */
/*                      */
/*  Output Parameters:  BD queue modified                        */
/*                      */
/*****

```

```

BD_init(new_BD, skip, i_rupt, wrap, page, address)

```

```

uword  *new_BD ;
int     skip, i_rupt, wrap ;
ulong   page, address ;
{
    uword  BD_contents ;

    BD_contents = 0x8000 ;

    if (skip != 0)
        (BD_contents = BD_contents | 0x4000) ; /* Set tx BD skip bit */
    if (i_rupt != 0)
        (BD_contents = BD_contents | 0x1000) ; /* Set BD interrupt bit */
    if (wrap != 0)
        (BD_contents = BD_contents | 0x2000) ; /* Set BD wrap bit */

    /* Append address */
    BD_contents = BD_contents + (uword) ((address - page) >> 2) ;

    *new_BD = BD_contents ;
} /* End of BD_init */

/* End File */

```