

## CPM/CPU INTERACTION

- Host Commands from CPU
  - Change state of SCC Channel
  - Initialize SCC Channel
  - Consist of microcode routines which change state of microcode or SCC state machine.
- Buffer Descriptors
  - Give RISC data to transmit.
  - Tell RISC where to store received data.
  - Report transmit or receive errors.
- Event Registers
  - Hardware or microcode generated events
  - Can generate an interrupt to the CPU
- Configuration Registers
  - Determine the operation mode of the SCC
  - Generate Clocking
  - Determine the Physical Interface

# SCC PARAMETER RAM

Address	Name	Function
Base + \$00	RBASE	The DPRAM location of the first RBD
Base + \$02	TBASE	The DPRAM location of the first TBD
Base + \$04	RFCR	Function Code for Receive DMA
Base + \$05	TFCR	Function Code for Transmit DMA
Base + \$06	MRBLR	Max Receive Buffer Length
Base + \$08	RSTATE	RCV State information about channel
Base + \$0C	R_PTR	Pointer to next memory write location
Base + \$10	RBPTR	Pointer to current/next BD location
Base + \$12	R_CNT	Down count to end of frame or Buffer
Base + \$14	RTEMP	
Base + \$18	TSTATE	TX State information about channel
Base + \$1C	T_PTR	Pointer to next memory read location
Base + \$20	TBPTR	Porinter to current/next BD location
Base + \$22	T_CNT	Down count to end of Buffer
Base + \$24	TTEMP	
Base + \$28	R_CRC	Current Receive CRC
Base + \$2C	T_CRC	Current Transmit CRC

- Best two to check are RBPTR and TBPTR. In many “problem situations,” they are not pointing at the buffers at all.
- T\_PTR and R\_PTR changing indicate data TX/RX.
- R\_CNT and T\_CNT are set when the BD is opened.

# WHAT THE COMMANDS REALLY DO

- **INIT TX PARAMETERS**
  - Copies TBASE to TBPTR and sets TSTATE to zero.
- **INIT RX PARAMETERS**
  - Copies RBASE to RBPTR and sets RSTATE to zero
- **ENTER HUNT MODE**
  - Issues a command to the channel to look for an IDLE or FLAG and ignore all incoming data
- **STOP TX**
  - Tells the various transmit routines to take requests but not send any more data.
- **GRACEFUL STOP TX**
  - Tells the various transmit routines to transmit to the end of the current buffer/frame and then perform a STOP TX
- **RESTART TX**
  - Reverses the operation of STOP TX
- **CLOSE RX BD**
  - Closes the current BD if data has been received into it. Does nothing otherwise.

## HOW A FRAME GETS TRANSMITTED

- CPM looks at the BD pointed to by TBPTR
  - If the INIT TX PARAMETERS command was not executed, the TBPTR may be pointing to garbage.
- CPM Detects Ready bit has been set.
  - »After this SCC goes idle, the CPM polls the ready bit every 128 TX Clocks for Ethernet, (64 TX Clocks for HDLC/Transparent and every character time for UART).
  - This step cannot be detected by the user.
- CPM copies buffer length to T\_CNT (temporary count), and copies starting address to T\_PTR (temporary pointer)
  - This step is your clue that TX Clocks are working!! Check parallel port pins and clocking configuration.
- CPM does an SDMA cycle(s) to get the first 32-bits of transmit data.
  - You can set a special function code to see this happen on the bus. If it doesn't happen the SDMA arbitration priority may not be high enough. Check SDCR.

## HOW A FRAME GETS TRANSMITTED (2)

- CPM decrements T\_CNT and increments T\_PTR.
  - This means you are starting to fill the transmit FIFO. The TSTATE value should no longer be zeros.
- When TX FIFO contains at least 8 bytes, the idles or flags should stop being transmitted, and real data should be seen on the TXD pin. (In UART mode data starts as soon as one character is in the FIFO).
  - The CTS pin can prevent data from transmitting if the user programmed it as a sync, and never asserted CTS.
  - If the Time Slot Assigner does not see a sync, then no data will transmit.
  - If you still don't see data transmit, try internal loopback mode to eliminate the data and control pins as a source of the problem.
- As soon as there is one 32-bit FIFO entry available, the SCC generates a request to the RISC. The request remains asserted until the FIFO becomes full or the last byte of the frame is written to the FIFO.
  - This provides the maximum bus latency. Minimum bus latency allowed is therefore the time it takes to fill N-4 bytes, where N is the number of bytes in the FIFO (either 32 for SCC1, or 16 for the other SCCs).

## HOW A FRAME GETS TRANSMITTED (3)

- CPM will use multiple buffers to transmit a frame if needed.
  - Make sure all the buffers are “ready” before the first ready bit is set, or an “underrun” might occur between buffers. In this case TXE is set but there is no BD in which to report the underrun.
  - Use the TBPTR to watch the CPM sequence through the BDs.
- After the entire frame is transmitted, the CPM will check the next BD’s ready bit immediately. If 0, it will go back into 128 tx clock poll mode.

## HOW A FRAME GETS RECEIVED

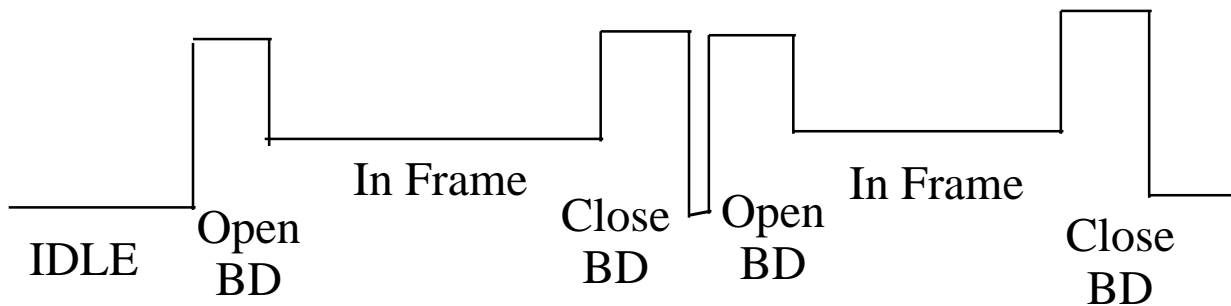
- CPM receives 32-bits of data from the RX FIFO
  - This step cannot be detected by the user.
- CPM looks at the BD pointed to by RBPTR
  - If the INIT RX PARAMETERS command was not executed, the RBPTR may be pointing to garbage.
- CPM Checks that the Empty bit has been set.
  - This step cannot be detected by the user.
- CPM copies buffer length to R\_CNT (temporary count), and copies starting address to R\_PTR (temporary pointer)
  - This step is your clue that RX Clocks are working!! Check parallel port pins and clocking configuration.
  - Echo mode can be used to check the SCC hardware without involving the RISC.
- CPM does an SDMA cycle(s) to write the first 32-bits of receive data.
  - You can set a special function code to see this happen on the bus. If it doesn't happen the SDMA arbitration priority may not be high enough. Check SDCR.

## HOW A FRAME GETS RECEIVED (2)

- CPM decrements R\_CNT and increments R\_PTR.
  - This means you are starting to fill the receivebuffer. The RSTATE value should no longer be zeros.
- As soon as there is one 32-bit entry of the receive FIFO that is available, a request is generated to the CPM.
  - This provides the maximum bus latency. Minimum bus latency allowed is therefore the time it takes to fill N-4 bytes, where N is the number of bytes in the FIFO (either 32 for SCC1, or 16 for the other SCCs).
- CPM will use multiple buffers to receive a frame if needed.
  - Use the RBPTR to watch the CPM sequence through the BDs.



## PERFORMANCE ISSUES



- Opening and closing BDs within a single frame is the worst issue for the CPM to deal with.
  - Worse than between frames because often frames have padding between them.
- The effect of a CPM performance problem can be
  - » SCC FIFO receive overrun
  - » SCC FIFO transmit underrun
  - FIFOs have patented emergency features that help lower priority SCCs keep up with the system.
    - » Extra pads between frames
    - » Extra idles between UART characters
- CPM pipelines its reads so up to 9 clocks of latency has no performance impact at all!
- Long frames are better because there is more time In Frame to recover from the high CPM loading of opening/closing BDs.
- Synchronize frame reception/transmission to stress the CPM for board tests.

## **SDMA Behavior**

- The SCC SDMA channels do not burst. The only bursts are from the IDMA's on the MPC860 and the ATM protocols on the 860sar. The 860 SCC microcode could have been re-written for bursts, but instead it was decided to be exactly compatible with the 68360.
- The SDMA channels do cycle stealing.
- The user will never see two back-to-back SDMA cycle steal cycles. (I.e. SCC1 TX immediately followed by SCC2 TX). There will always be a few clocks inbetween.
- If the SDMA is moving a 32-bit value to a smaller port, such as 16-bits the user will see back-to-back 16-bit cycles.
- On transmit, the SDMA will either read 16 or 32-bits at the start of a frame depending on the starting address. (If a 16-bit read is sufficient, the SDMA will only read 16-bits). It then throws away one of the bytes if it does not need all of them. Then for the rest of the transfers, the SDMA will always read 32-bits at a time. Note: The SDMA only reads 16-bits at a time in UART mode, and never reads 32-bits.