DATM: DBI ATM Interface

Ian Leslie

DRAFT 3 May 1994

Abstract

The DATM card is an ATM interface based on the EISA ATM card developed in the Computer Laboratory. A structural overview is shown in figure 1.

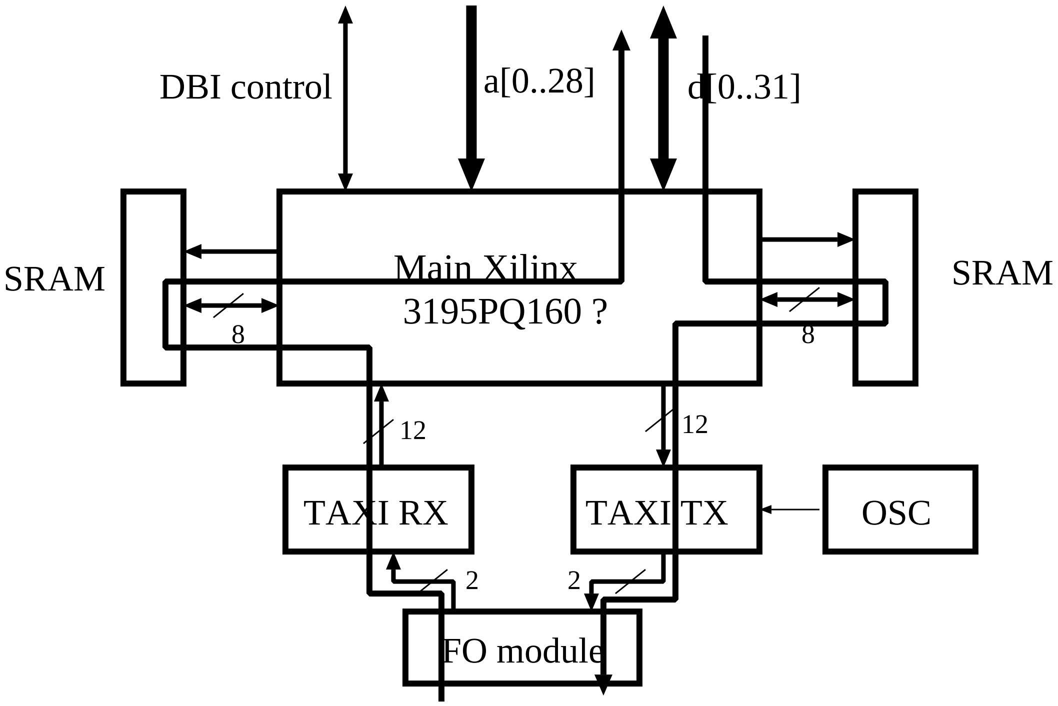

Figure 1: DATM Components and Main Data Paths

1 Differences from EATM Card

It is recommended that the description of the EATM card be read before reading this document. While the interference of reads and writes with the internal DATM state machine is not quite as tortured as in EATM, the fact remains that the card expects accesses in a certain order.

There is no CRC continue nonsense in DATM; the pipeline is slightly better designed. Thus data is dragged through the CRC logic *after* it has been read; as long as the CRC residue before reading is correct the result after will be correct.

2 Registers

All accesses to the card should be 32 bits with the possible exception of access to the ID space which may be word or byte. Addresses are relative to the base address of the EASI podule address. With the exception of the ID space, the card only decodes LA[8..11]. (The ID space also decodes LA[2..4]. The register address map in shown in table 1.

Name	Offset	R/W	Function
(id)	0x0XX	R	identification etc
STATCON	0x1XX	R/W	status and control
RXHDR	0x2XX	R	rx cell header
RXDATA	0x3XX	R	rx cell data
RXCRC	0x4XX	R/W	rx CRC-32
STARTDMA	0x5XX	W	start DMA
TXHDR	0x8XX	W	tx cell header
TXDATA	0x9XX	W	tx cell data
TXCRC	0xaXX	R/W	tx CRC-32

Table 1: DATM Register addresses.

All are reasonably straightforward with the exception of the STATCON and STARTDMA registers.

The status register format is shown in table 2. **Bits not specified should be left zero.**

At initialisation, a one should be written into the reset bit and then it should be cleared. The interrupt will drive the both podule IRQ and FIQ; these can be enabled or disabled by IOMD.

A write to the STARTDMA (not yet implemented) register puts the DATM card into a state where it will begin requesting DMA cycles until it sees a transfer complete from IOMD. If the DMA cycles are writes then these are treated just like writes into TXDATA, if reads then just like reads from RXDATA.

The RXBIT is set by hardware or software but can only be cleared by software. After clearing the bit, the RXNOTEMPTY should be polled.

3 Receive CRC

The board provides CRC-32 support for receiving AAL5 units. In order to check an AAL5 adaption unit for a correct CRC, the following steps should be taken.

- at the beginning of an AAL5 unit the RXCRC register should be set to `0xffffffff`.

Name	Bit	R/W	Function
RXNOTEMPTY	15	R	1 indicates at least one cell in card
RXBIT	13	R/W	indicates a cell with AUU bit set has been received
RXFULL	10	R	1 indicates rx buffer overflow
FIQEN	8	R/W	1 will cause interrupts to raise FIQ rather than IRQs
INTNOTEMPTY	7	R/W	1 will enable interrupts on RXNOTEMPTY
INTRXBIT	5	R/W	1 will enable interrupts of RXBIT
INTOVERFLOW	2	R/W	1 will enable interrupts on RXFULL
RESET	0	R/W	1 resets most of card (not taxis)

Table 2: STATCON register contents.

- assuming all cells of the AAL5 unit are received contiguously, when the last cell is received, the residue in the RXCRC (after the CRC present in the PDU has been read) should be 0xc704dd7b.
- if cells from different AAL5 PDUs are interleaved (ie. they have different VCIs) then the RXCRC should be saved/restored after the RXHDR is read.

4 Transmit CRC

When the board provides CRC-32 support for transmitting AAL5 PDUs, the following will apply for sending an AAL5 PDU:

- at the beginning of an AAL5 unit the TXCRC register should be set to 0xffffffff.
- assuming all cells of the AAL5 unit are transmitted contiguously, in the last cell the first 11 data words should be transmitted, then the TXCRC should be read, xor'd with 0xffffffff and then sent as the 12^{th} and final word.
- if cells from different AAL5 PDUs are interleaved then the TXCRC should be saved/restored between cells.

5 DMA Operation

When a DMA start operation is started (ie the board is prodded by a write to DMA start) then the board will transfer 12 words of payload and then desist from transferring. If transfer complete is asserted it will return to a quiescent state otherwise it will go into a standby state, noting whether it was transferring to or from the device. When a header is read or written in a direction corresponding to the direction of the transfer before the card entered standby, the card will resume transferring, again for 12 words.

There is no checking for the same VCI/VPI in the header; interleaved cells from different VCs have to be sorted out after they appear in memory.

The location at offset 0x6xx (reset DMA) can be used to move the card from a standby (or indeed any) state to the quiescent state.

Note that a version of the DATM card used for shaping purposes has different DMA properties. The shaping version also has no support for CRC32. DMA is only supported on receive. The system will DMA headers (4 bytes) and payloads into memory so that 52 byte cells appear contiguously in memory. When DMA start is asserted, the card will begin DMAing cells into memory. This operation will pause if there are no receive cells and stop when the DMA operation is complete (as indicated by the transfer complete signal).