Track Recovery Module

Overview

Figure 1 shows a block diagram for the TRM.

The board is accesses by the SUCC via the VME bus connected through PL1.

Signals for processing arrive and leave via PL2.

Eight tracks signals are passed to the TRM after buffering and selection in the DIM. The signals are NRZM, single ended TTL. The board has eight identical processing circuits each handling one track completely independently.

Data are decoded and written to DRAM and read out in timecode synchronisation. This removes skew between tracks and synchronises the flow of data to the following boards, to the SU internal time keeping system.

The data receiver detects various quality indications, some of which are accumulated, and stores these with the data in DRAM as a four byte status word. Quality indicators are also passed to an interrupt generator where they can optionally be used to trigger micro intervention.

Sync status, Validity and Xilinx DONE bits from each FPGA are passed via an interface to front panel LEDs.

Microprocessor and VME interface

The VME interface is implemented in an EPLD. Messages to and from the Station Unit Control Computer (SUCC) are placed in a Dual Ported RAM which is shared with the Microcontroller. Via a common address and data bus the Micro accesses each of the data processing blocks where it can set configuration, read from the DRAM, read status information and control read-pointers to establish and maintain synchronisation.

Data processing blocks are implemented in Xilinx FPGA's which are programmed by the Micro at reset in asynchronous peripheral mode.

The local micro controller (MC68340) has two serial ports. One of these is brought out, at RS232 levels, to the front panel 9-pin D-type connector, PL3. During development this has been used as a diagnostic port. It is also the pROBE console port.

The micro controller can also be run in Background Debug Mode (BDM) via a special port which is brought out to PL4 on the front panel. This has been used in development as a means of loading and running software at source level.

Data Input Processing

One of the eight track processing circuits is described..

The input signal is buffered and passed to a AT&T clock recovery chip which extracts a clock and re-times the data. The clock recovery chip is programmable to a number of frequency ranges by seven static control lines. These are set by the Microcontroller and latched in U28. Frequency control is common to all eight recovery chips.

Extracted clock and re-timed data pass to the FPGA. Fig 2 shows the elements of the receiver.

Decoder

A delay and XOR circuit is used to detect transitions in the data and hence decode from NRZM to NRZL.

Sync Detection

The data are then clocked into a 36 bit shift register who's parallel output is continually compared with the sync pattern. When the sync pattern is detected a sync line is asserted which resets various mechanisms. Serial data are passed on for further processing from the output of the 36bit shift register. Thus, relative to the following circuits, the valid sync sequence is detected in advance of its arrival.

Demodulation

Data can be XORed with a pseudo random sequence generator, reset by the sync signal. This removes modulation if required as each bit is received.

Serial/Parallel conversion

Serial data are clocked into a 9 bit shift register. Using a feedback mechanism, reset by detected sync, this divides the serial data into 9bit bytes and provides a byte clock which is used to control the flow of data in following circuits.

Cyclic Redundancy Check (CRC)

Serial data also pass to a CRC check circuit which is enabled during the header (Timecode only for VLBA). If a CRC error is detected this is latched for inclusion in the status word and made available to the interrupt generator.

Parity Check

After Serial/parallel conversion, nine bit bytes are checked for parity using a wide XOR. The result of the check, 1 for odd, 0 for even, is latched alongside the eight data bits in place of the parity bit, and now represents the validity of the byte. Note that, at record time, the parity bit was added after the data were modulated. The replay modulation generator therefore includes a parity adjustment which inverts the validity bit if necessary. A signal derived from the Rx State Machine (see later) forces validity low for non-data bytes. Parity errors are accumulated in a counter which is reset each frame. The frame

error count is written into the frame status field in DRAM and added to an accumulator which can be read (and cleared) by the micro.

Rx State Machine (Rx-SMC)

Progress through each frame of data is monitored by a state machine shown in fig 3. The **RX-SMC** is unconditionally reset to State 0 when sync is detected. As each byte of sync data arrives the Rx-SMC advances through states 0, 1, 2 and 3. State 4 enables a counter which counts the eight bytes of timecode before enabling the next state. State 5 enables a second counter which counts the 2480 bytes of data (2500 for VLBA). .In State 6 a third counter is enabled to count the first seven bytes of Aux data before enabling State 7 which corresponds to the last byte of Aux. If no sync is detected the Rx-SMC advances to State 0 to repeat the cycle. The State machine flywheels indefinitely until a valid sync is detected.

Various decodes of the Rx-SMC and counters are extracted to control other functions.

Sync Condition

If Synchronisation is correct the detected sync should be coincident with the last bit of the last byte of Aux data. This “Sync Expected” line is logically combined with the “Sync Detected” line to produce three flags, Perfect-sync, Unexpected-sync and Missing-sync.

Receive Buffer

Bytes of received data are buffered by a seven stage FIFO before being passed to the RAM controller for writing to memory. The ram controller writes data to memory in fast page mode, four bytes at a time. The FIFO asserts Rx_RAM_REQ when four bytes are available. An extra bit alongside the data bytes is set high at the beginning of each frame to inform the RAM controller that it should start a new page.

Status Insertion

A multiplexer prior to the Receive Buffer replaces the four sync bytes with four status bytes. These are two bytes of Parity error count and two (duplicate) bytes of status. Status information is: CRC error, unexpected sync, missing sync, perfect sync, parity error count exceeded.

Status State Machine (Status_SMC)

The Status_SMC is responsible for controlling the paging of data in DRAM and the insertion of the four status bytes. The state diagram is shown in fig 4. Normally States 0,1,2,3 follow the corresponding states of the Rx-SMC, controlling the multiplexer that routes Status information to the FIFO whilst Sync data are being received. If the Rx_SMC is reset by the arrival of a late sync the Status_SMC waits for Timecode data before allowing data to pass to the FIFO. In Status_SMC State 0 the paging flag is set high.

Once the Status_SMC has moved to State 0 and beyond it cannot be reset until it arrives at State 5. State 5 is only permitted when the data frame

counter of the Rx_SMC has passed 2024 (i.e. MSB high). This ensures that during a period of lost sync the error in the SU frame count has to accumulate to more than 400 bytes before it can result in a discontinuity in the sequence of frames of data in the DRAM. In reality a bit slip is seldom more than a few bits.

Ram Write Address Control.

Status regarding the parity error count and the Timecode CRC check are not available until the frame is complete. This information has to be placed at the beginning of the page (i.e. frame) to which it refers. This is achieved as follows. When the page flag appears at the FIFO output the lower order RAM address is reset to zero. The four status bytes are thus written at the beginning of the page. Following this the page number is incremented without changing the rest of the address and Timecode begins to accumulate from byte 4 onward. Fig 5 illustrates.

RAM Read Page Address Control

Before the start of a scan the micro reads timecodes from DRAM and determines the location of the first frame to be transmitted. The required page number is loaded into the read address counter. Each time a frame of data is transmitted the page number is incremented. The precautions taken in the design of the receive circuit ensure that, normally, frames of data are sequentially and contiguously packed in the DRAM. Exceptionally this can be disturbed however so the micro routinely checks the TOT of the next frame to be transmitted. If this is incorrect the track is set invalid whilst the micro locates the page containing the correct TOT.

RAM Read Byte Address Control

The sequence of bytes to be read from the DRAM differs for MkIV and VLBA modes. In both cases the read byte address is generated by a counter which is pre loaded, once only before the scan, to ensure that data are extracted in the correct sequence.

In MkIV mode the counter is pre-loaded to begin with Auxiliary data. At the end of Auxiliary data the page is incremented and reading commences at location zero of the next page. This repeats throughout the scan. Pages are read sequentially unless the micro intervenes with a new page number.

In VLBA (non data replacement) mode Auxiliary, Status and Timecode data must be omitted. The counter is therefore pre-loaded to the beginning of the Data field. The page is incremented at the end of the Data field. This edits out the Auxiliary data. Status information and time code are then read from the next page and are removed on the read side of the FIFO.

FIFO Operation

When transmission is enabled the empty FIFO issues a request for data to the DRAM arbitration mechanism. Data are passed to the FIFO, four bytes at a time until it is full. Further requests are issued whenever there is room for another four byte access in the FIFO. Data are read from the FIFO under the

control of TRACK_FRAME from the CRM. In MkVI mode reading from the FIFO halts at the end of TRACK_FRAME. The number of bytes read from the FIFO during TRACK_FRAME exactly matches the number written from one page of DRAM. In VLBA (non data replacement) mode an extra counter is used to extract a further 12 bytes from the FIFO at the end of TRACK_FRAME. These bytes are not passed to the CRM so the header is edited out at this point. The Status byte is examined as it passes through the FIFO to check the **Set Frame Invalid** bit.

Validity control
The Transmitter is able to control validity of data in three ways:

1. Pass the validity stored with each byte unaltered
2. Set all validity bad in any frame which has the **Set Frame Invalid** bit in the status word set.
3. Set all validity bad until further notice under control of the Micro.

The Set Frame Invalid bit in the status is controlled by the receiver which can be programmed to set this bit if selectable quality criteria are not met, e.g. if the parity error count exceeds the programmed maximum for a frame.

Main Data Path
The following table gives the main data path through the TRM Xilinx.

Sheet Name (number)	Circuit ref. or module port name	Function
RX_INPUT(7)	U126	Input pin P9
RX_INPUT(7)	U145	Latched with RX_B_CLK
RX_INPUT(7)	U147 U146 U144	Delay and XOR to convert from NRZM to NRZL.
RX_INPUT(7)	BX57	36 bit shift register. Parallel O/P is compared with sync pattern
RX_INPUT(7)	U155	One extra delay for pipeline matching
RX-ALL(6)	Rx_In_Data	Up to sheet 6 then into sheet 9
RX_SHIFT(9)	U131	XOR with demodulation if active
RX_SHIFT(9)	BX52	Shift register. Converts from serial to 9bit parallel.
RX_SHIFT(9)	BX55	8 data bits directly, 1 parity bit via BX48,U521 and U134 which convert parity to validity.
RX-ALL(6)	Rx_HoldBuf	Up to sheet 6 then into sheet 11
PARSTAT(11)	BX59 BX64 BX65	Multiplexors insert status word in place of sync sequence.
RX-ALL(6)	B_Dat_In	Up to sheet 6 then into sheet 12
RX-BUFF(12)	BX98-104	7 stage FIFO Buffer.
RX-BUFF(12)	Rx_Data	Up to sheet 6, up to sheet 1 then into sheet 15
RAM-ALL(15)	BX112	Mux with uP data for DRAM write
RAM-ALL(15)	BX133	Latch
RAM-ALL(15)	BX119	DRAM I/O
RAM-ALL(15)	BX118	Latch. Now transmit data
RAM-ALL(15)	Tx-Dat	Up to sheet 1 then to sheet 2 and into sheet 3
T-BUFF(3)	Bx183 Bx177-12	7 stage FIFO Buffer.
T-BUFF(3)	BTx-Dat	Up to sheet 2 then into sheet 4
TX-SFT(4)	BX173	8th stage of FIFO buffer
TX-SFT(4)	BX170	Parallel-in Serial-out for 8 data bits
TX-SFT(4)	U460	Data selector re inserts Validity bit after gating in U463
TX-SFT(4)	Tx_Data_Out	Up to sheet 2
TX-ALL(2)	U417	Output latch and buffer
TX-ALL(2)	U438	Output pin P7

Figure 1 Block Diagram for the TRM card

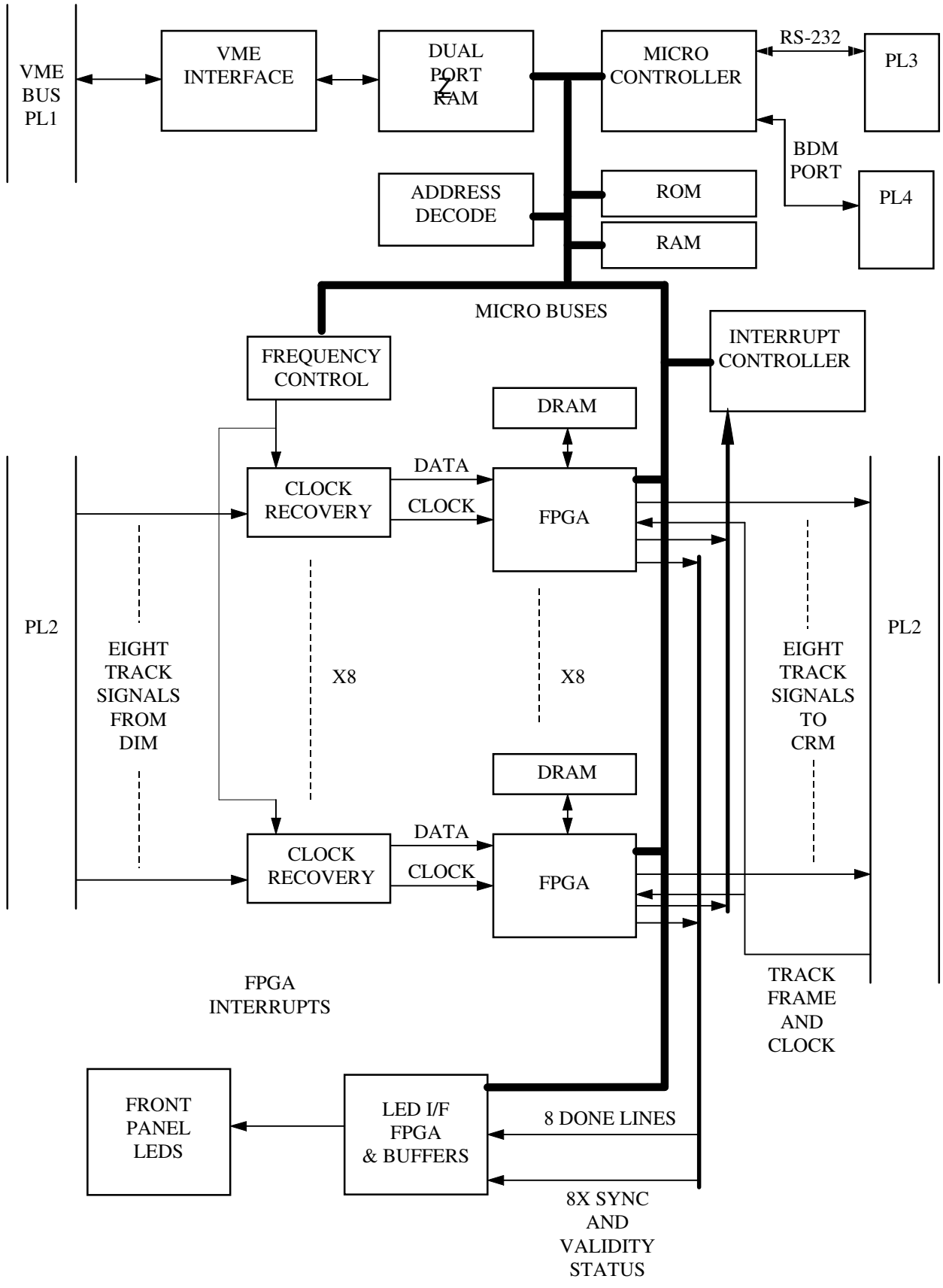


Fig 2. TRM RECEIVER

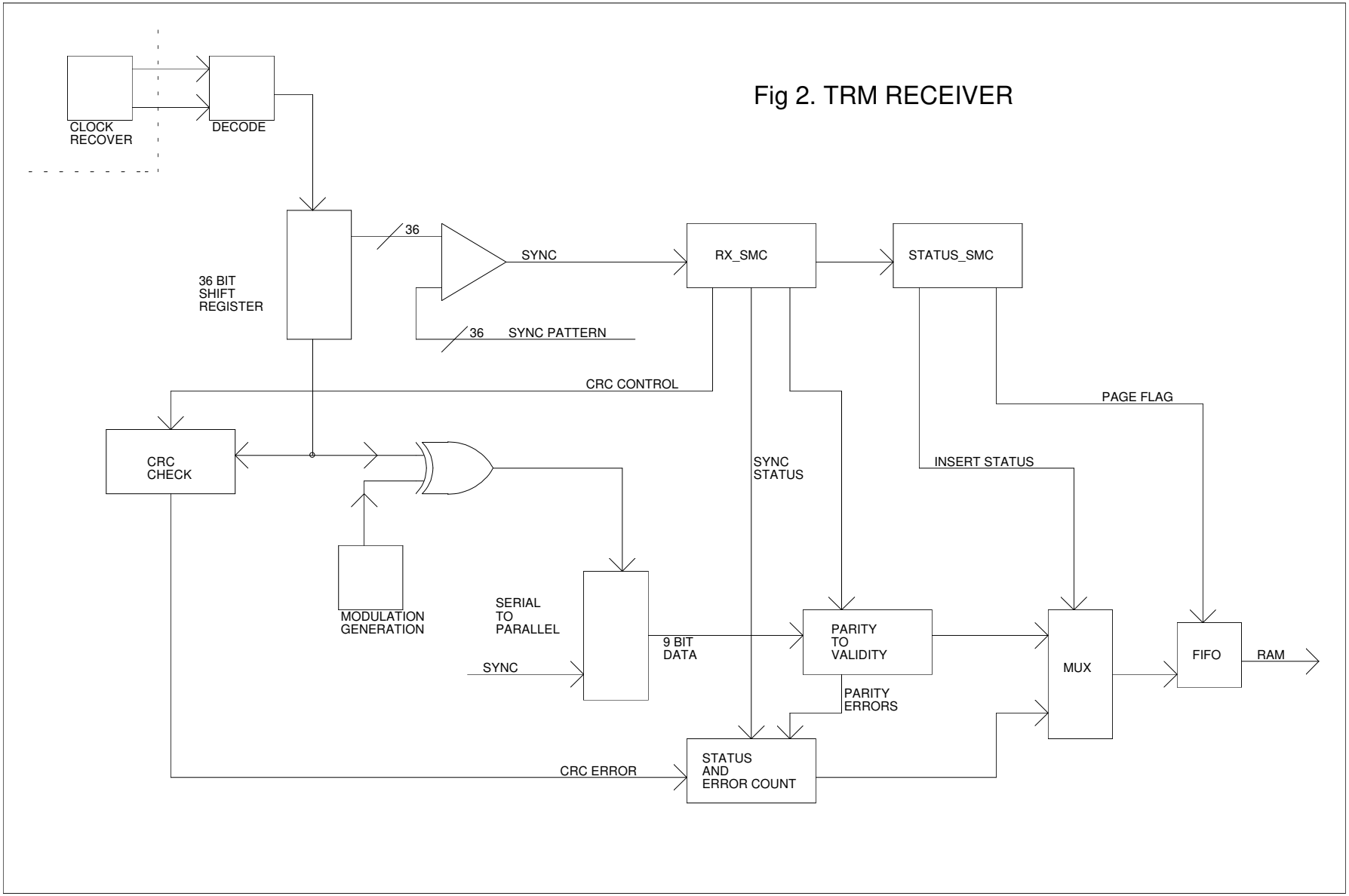


Fig 3. RX STATE MACHINE

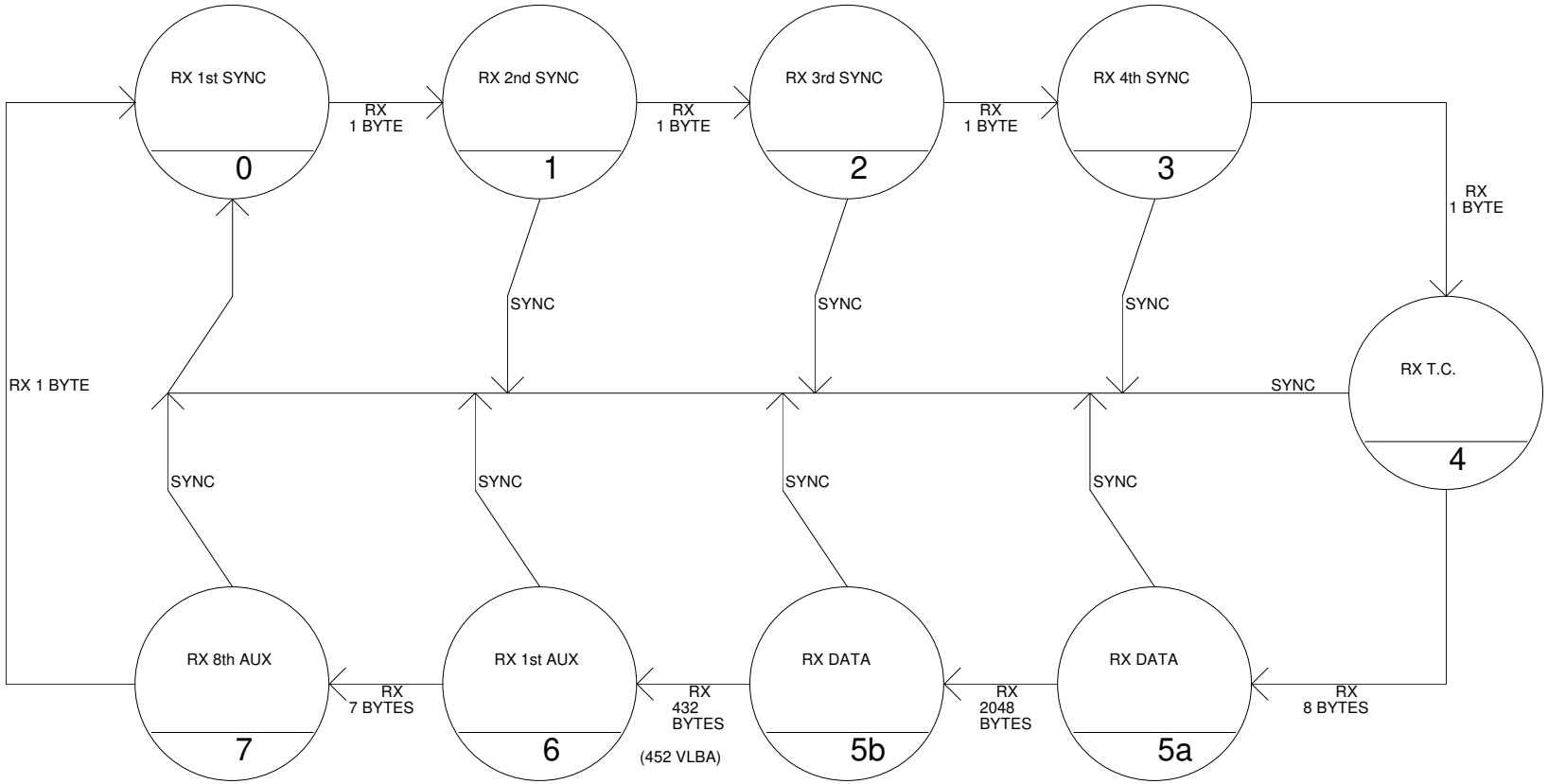
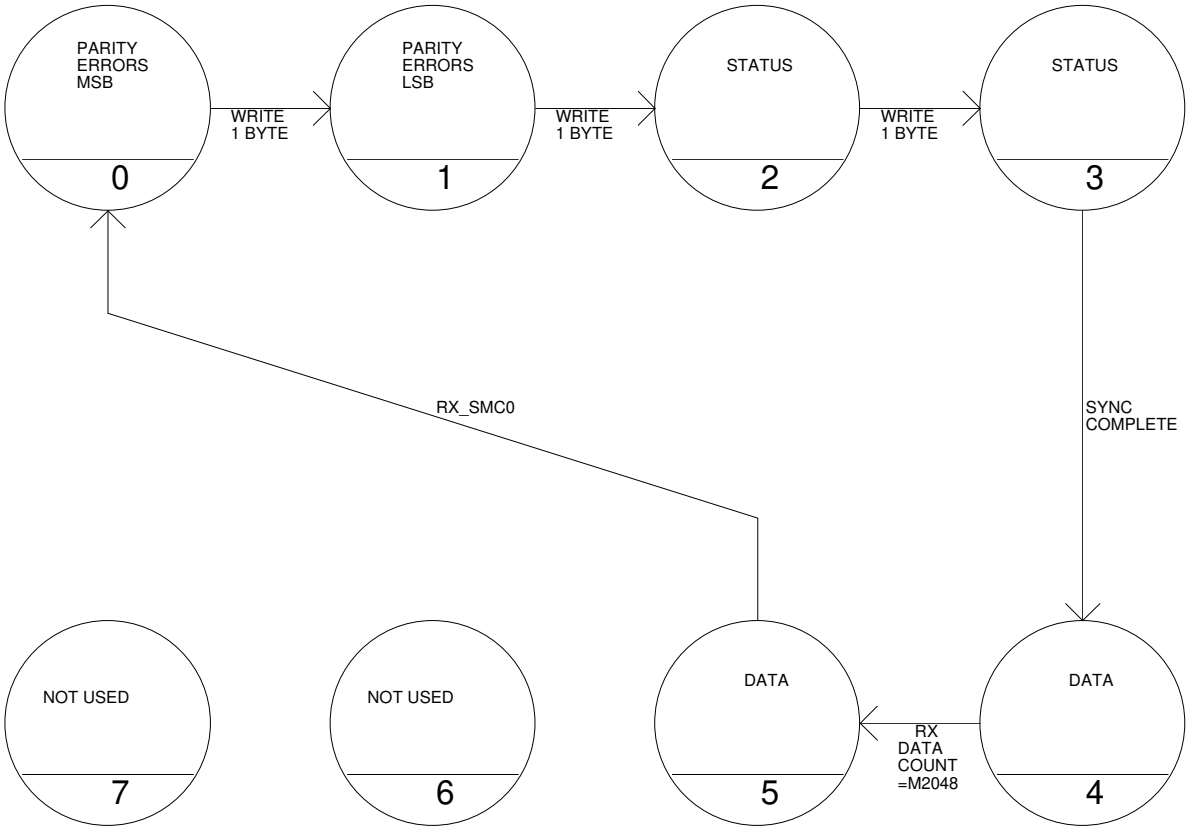


Fig 4. STATUS STATE MACHINE



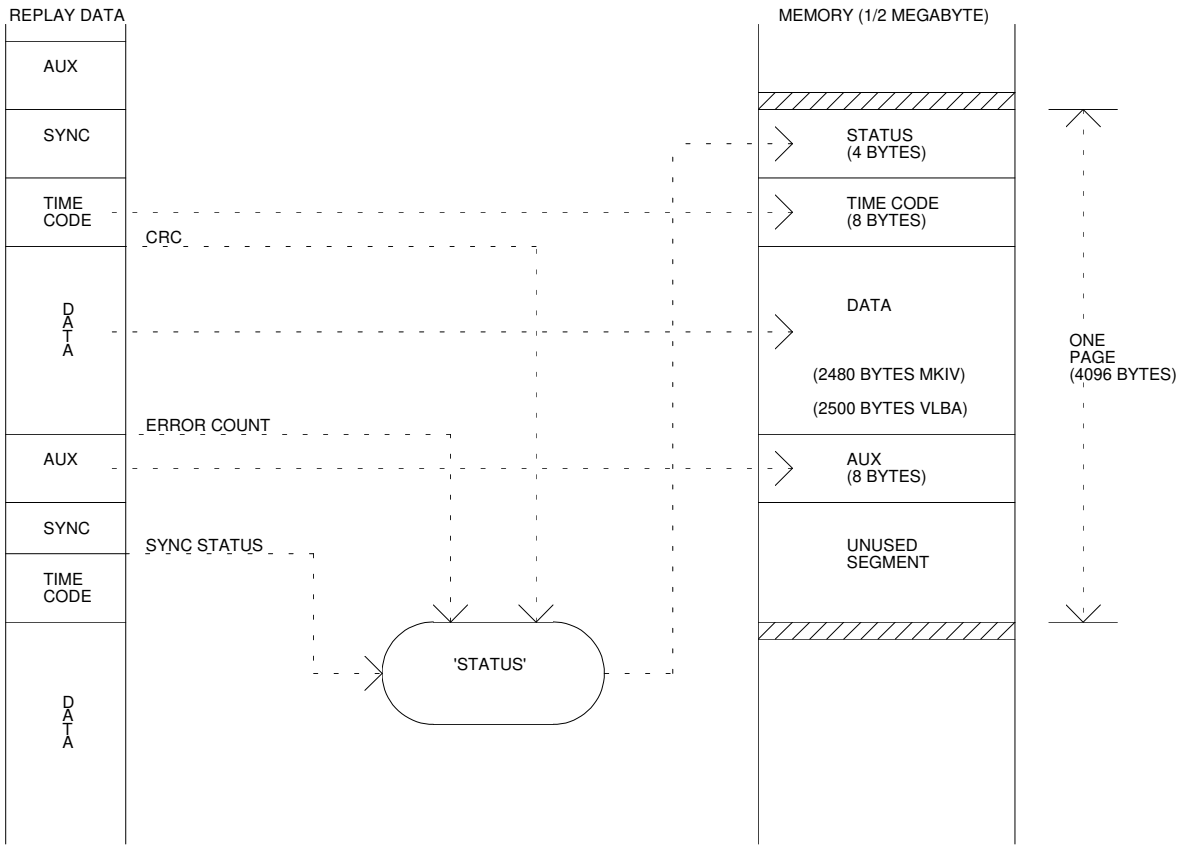


Fig 5. DRAM WRITE ADDRESSING

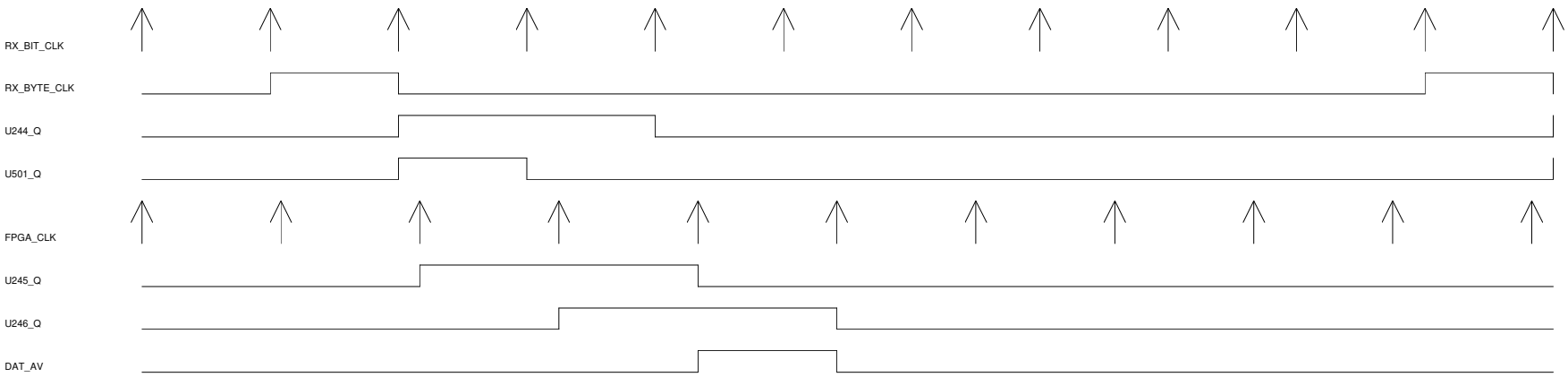


Figure 6. RX_BYTE_CLK to DAT_AV INTERFACE (RX_BIT_CLK > FPGA_CLK)

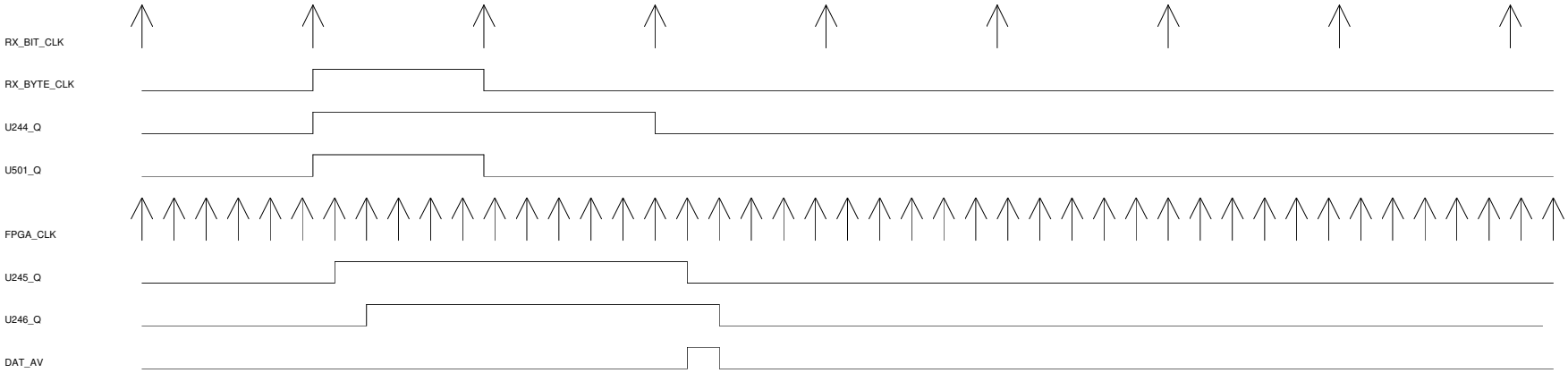


Figure 7. RX_BYTE_CLK to DAT_AV INTERFACE (RX_BIT_CLK << FPGA_CLK)

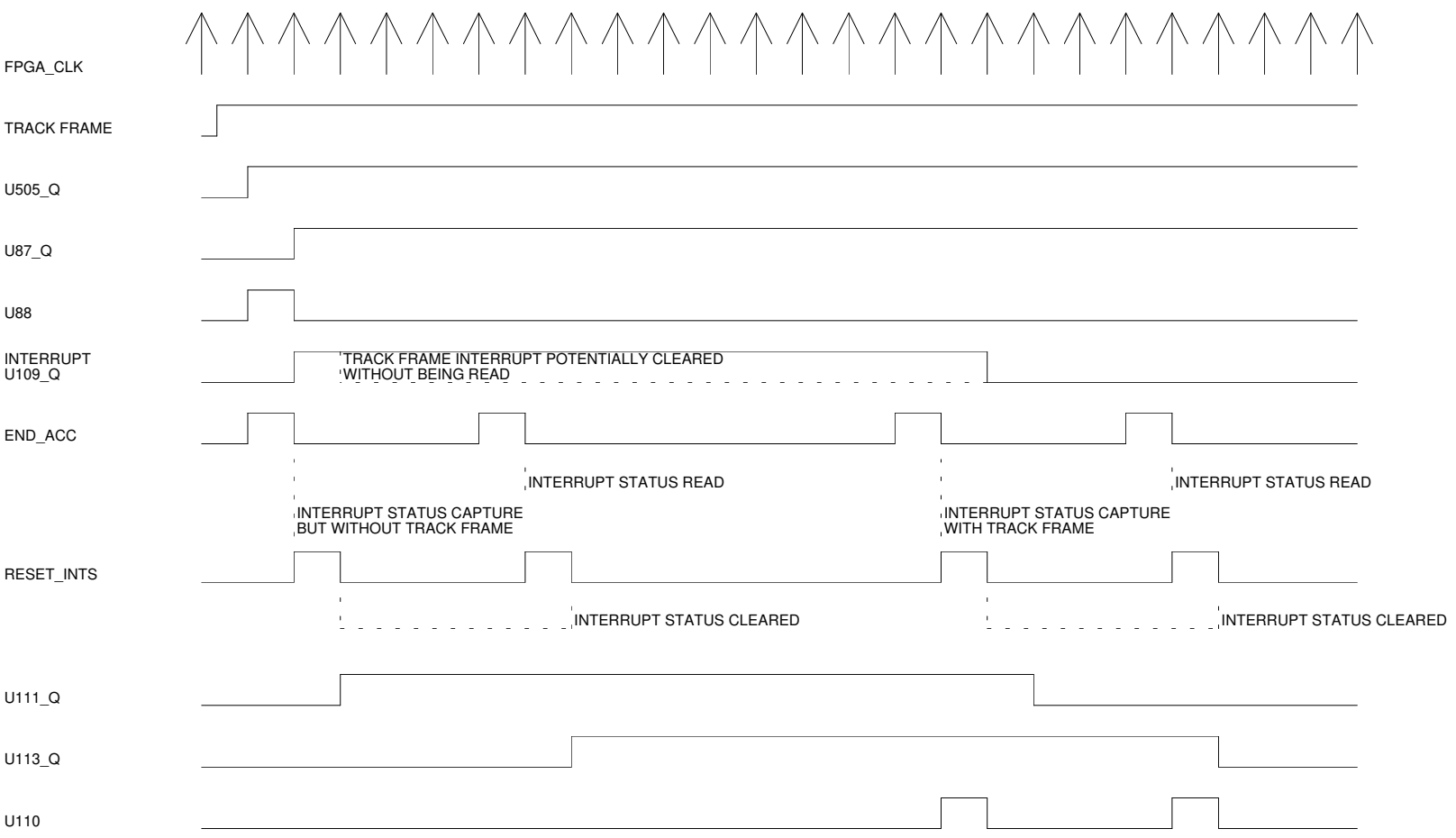


Figure 8. TRACK FRAME INTERRUPT

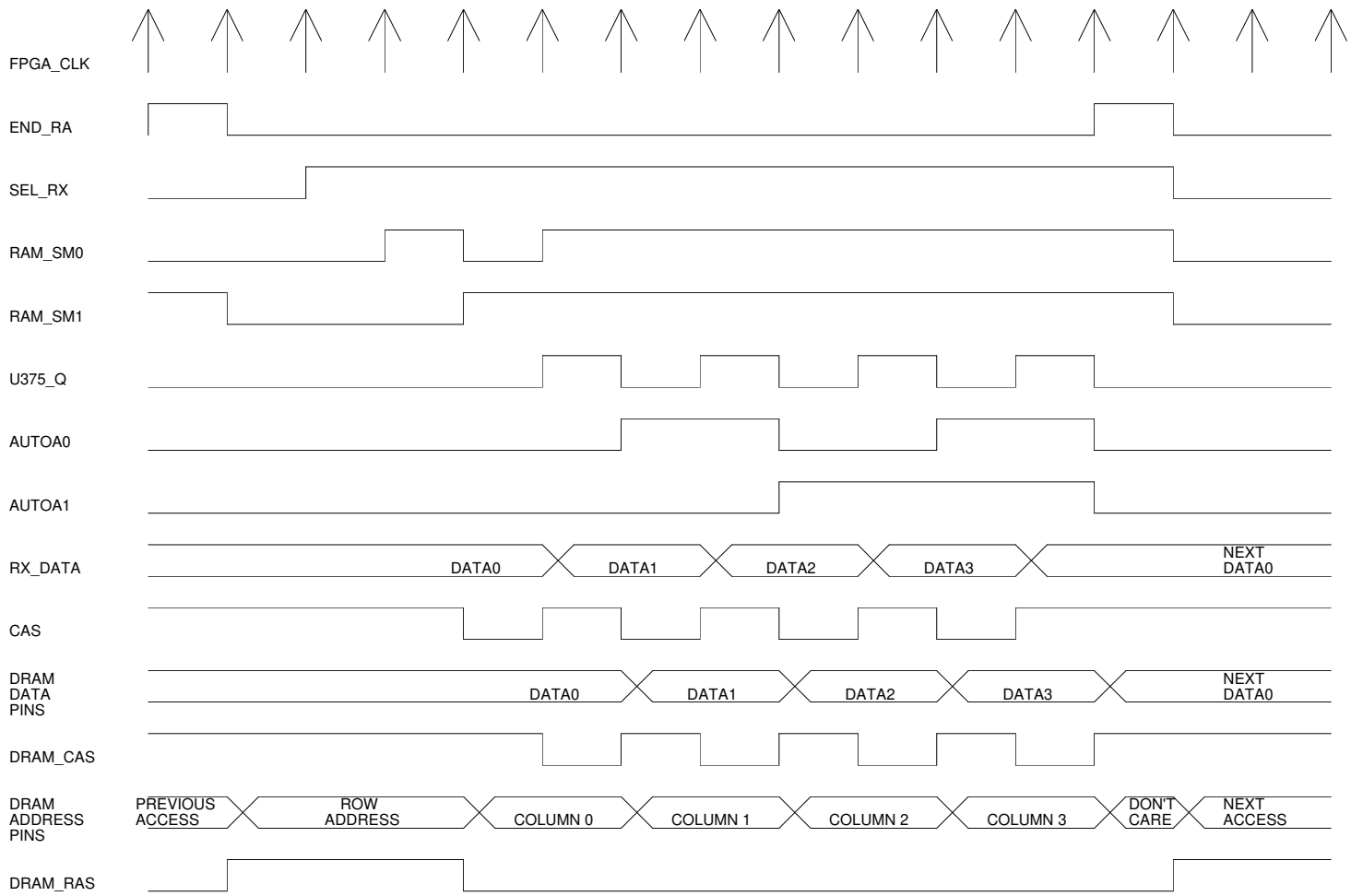


Figure 9. RX WRITE ACCESS TO DRAM's

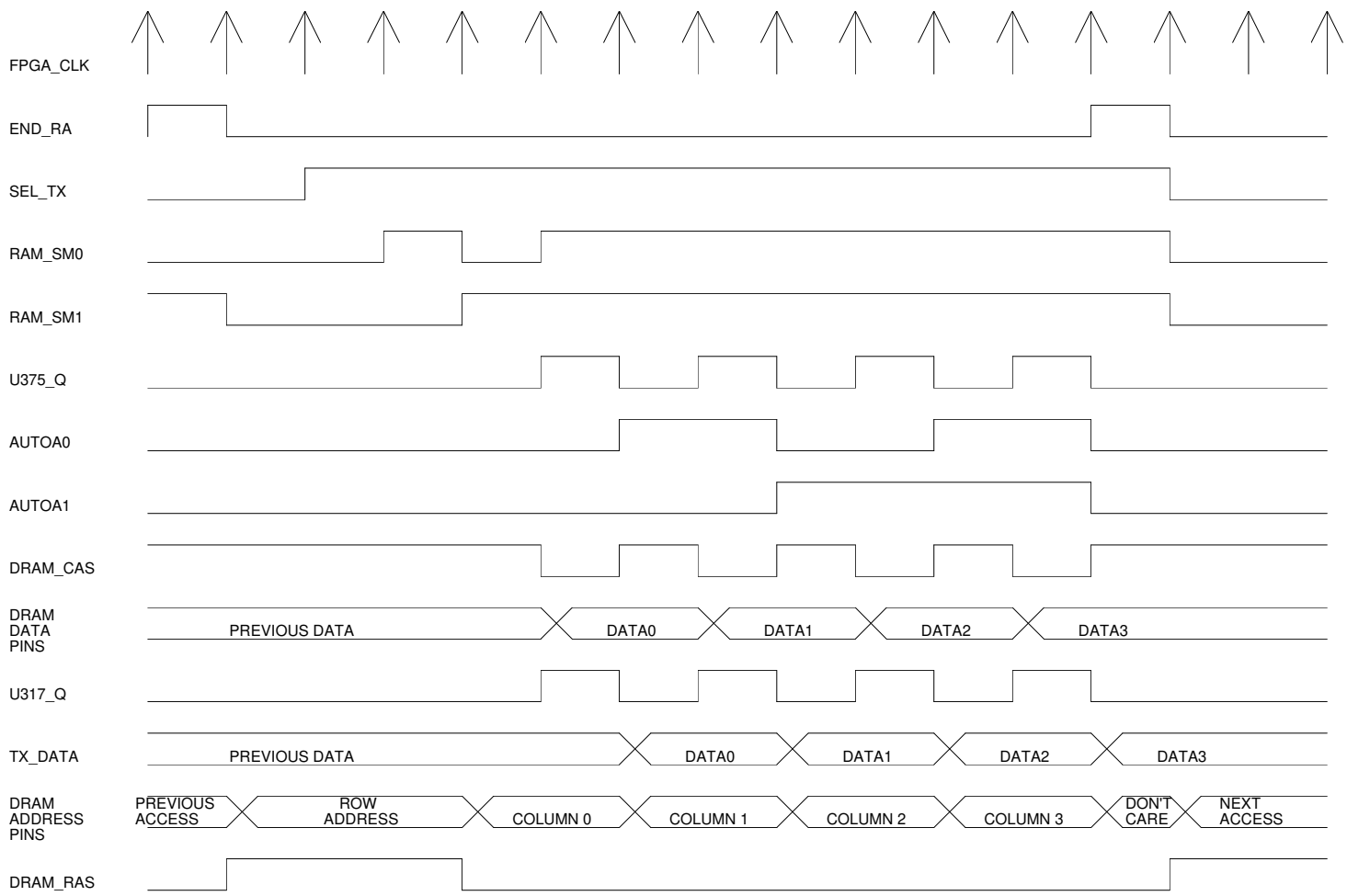


Figure 10. TX READ ACCESS TO DRAM's

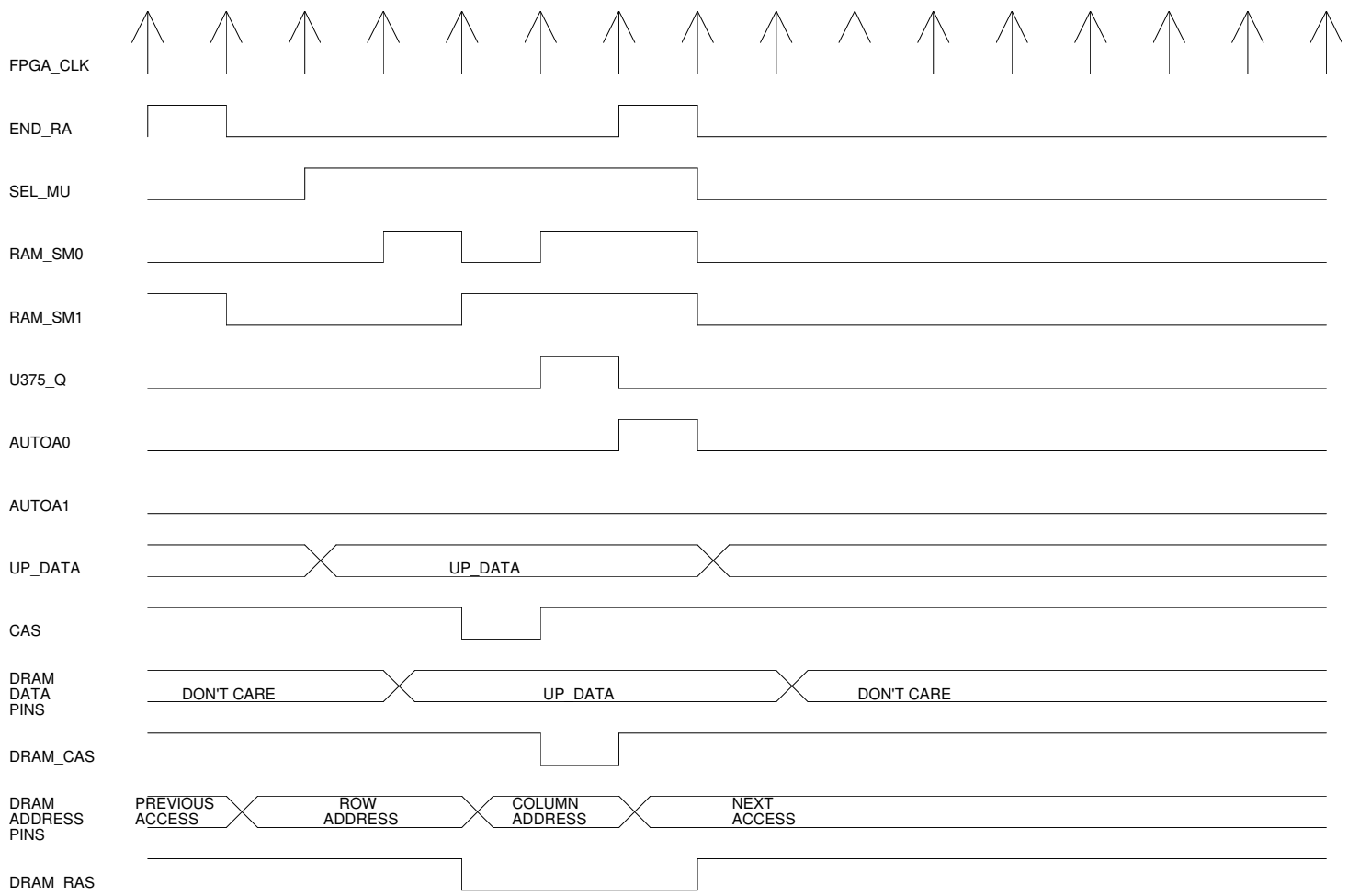


Figure 11. MICROCONTROLLER WRITE ACCESS TO DRAM's

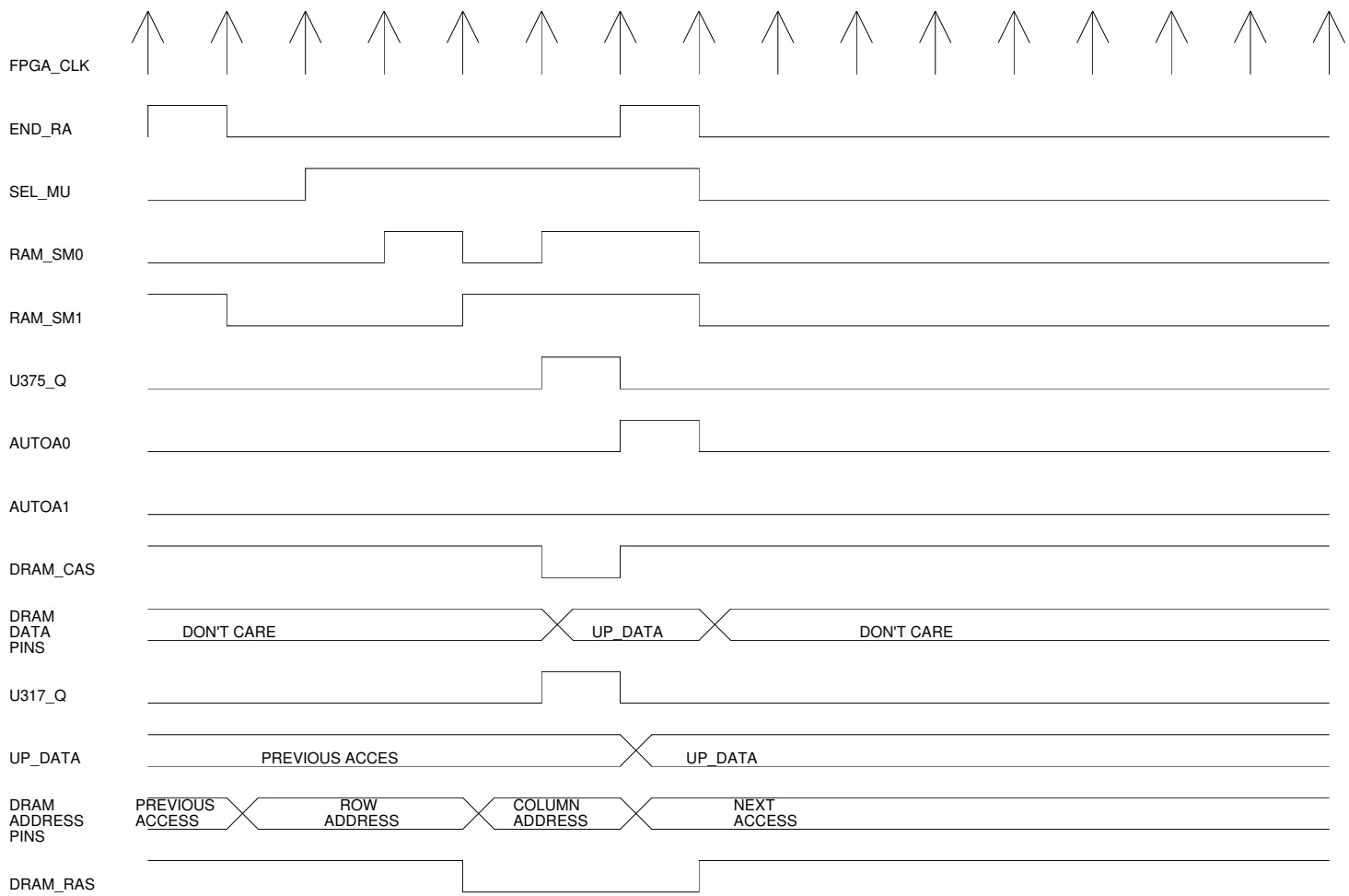


Figure 12. MICROCONTROLLER READ ACCESS TO DRAM's

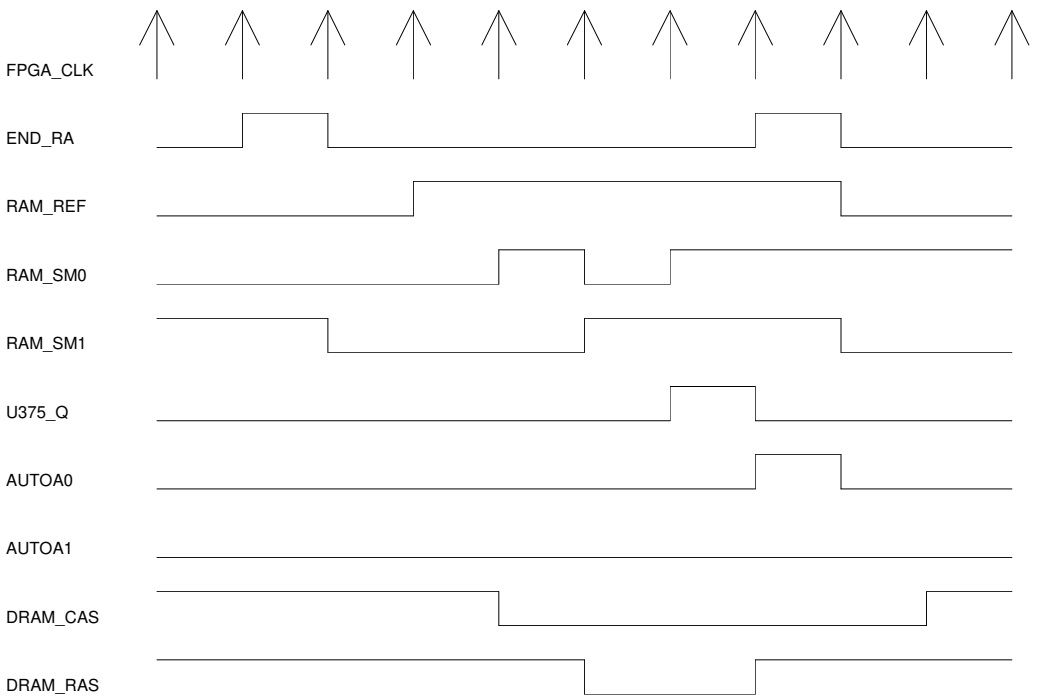


Figure 13. REFRESH ACCESS TO DRAM's

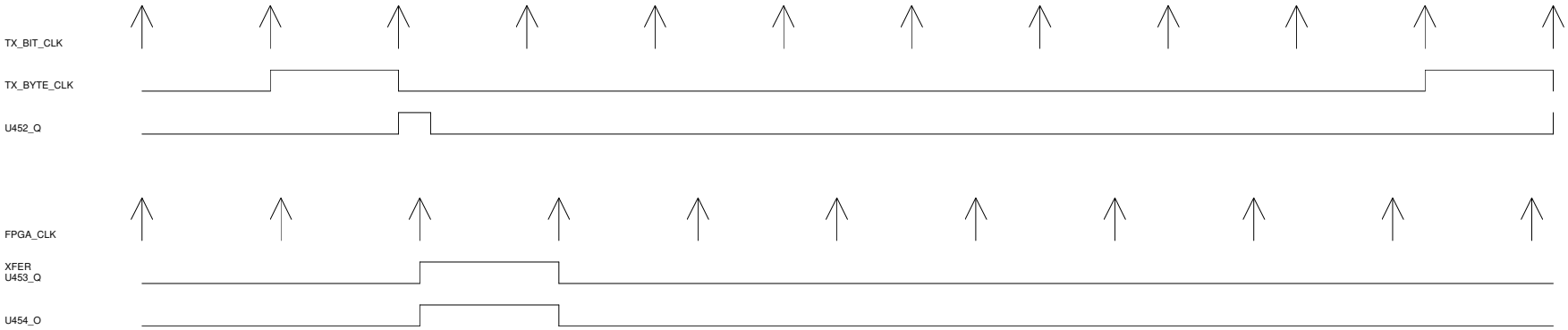


Figure 14. TX_BYTE_CLK to XFER INTERFACE (TX_BIT_CLK > FPGA_CLK)

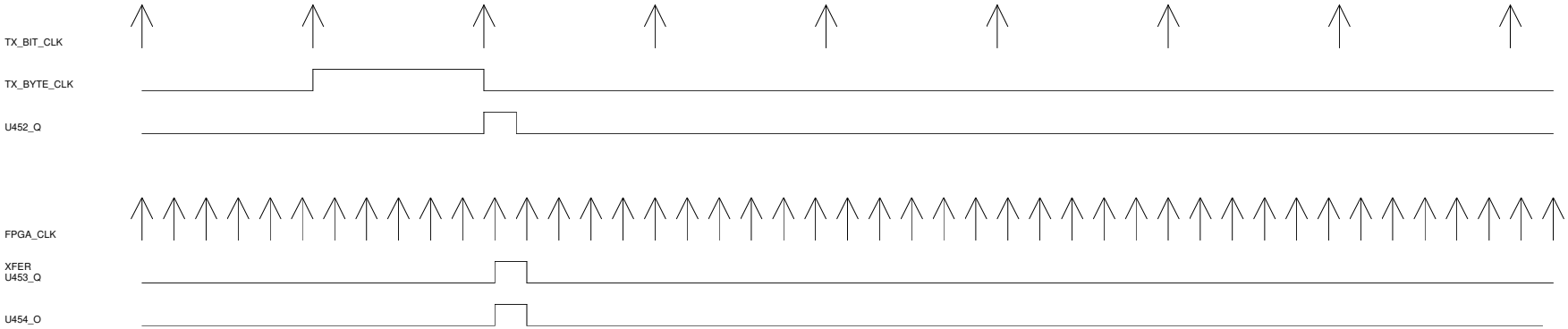


Figure 15. TX_BYTE_CLK to XFER INTERFACE (TX_BIT_CLK << FPGA_CLK)