COM-8003SOFT
SIGNAL DIVERSITY COMBINING
VHDL SOURCE CODE OVERVIEW

Overview

The COM-8003 ComBlock Module comprises two pieces of software:

- VHDL code to run within the FPGA for all signal processing functions.
- C/Assembly code running within the Atmel ATMega8515L microprocessor for non application-specific monitoring and control functions.

The VHDL code interfaces to the monitoring and control functions by exchanging byte-wide registers on the Atmel microcontroller 8-bit data bus. The control and monitoring registers are defined in the specifications [1].

The Atmel microprocessor code is generic (i.e. non application specific), not user-programmable and functionally transparent to the user. It is thus not described here.

The COM-8003 VHDL code runs on the generic COM-8000 hardware platform. The schematics [2] for this platform are available in this CD.

Reference documents

[1] specifications: com8003.pdf

[2] hardware schematics: com_8000schematics.pdf

[3] VHDL source code in directory
com-8003_001\src

[4] Xilinx ISE project files
com-8003_001\com-8003_ISE41.npl
com-8003_001\com-8003_ISE63.npl

[5] .ucf constraint files
com-8003_001\src\com8003.ucf

[6] .mcs FPGA bit files
8003_001\com8003_001.mcs

Configuration Management

The current software revision is 1.

Configuration Options

No option currently supported.

VHDL development environment

The VHDL software was developed using two development environments:

(a) Xilinx ISE 4.1 with Synopsys FPGA Express 3.6 as synthesis tool.

(b) Xilinx ISE 6.3 with XST as synthesis tool.

Target FPGA

The VHDL code was synthesized for the Xilinx Spartan-IIE XC2S300E-6PQ208 FPGA.

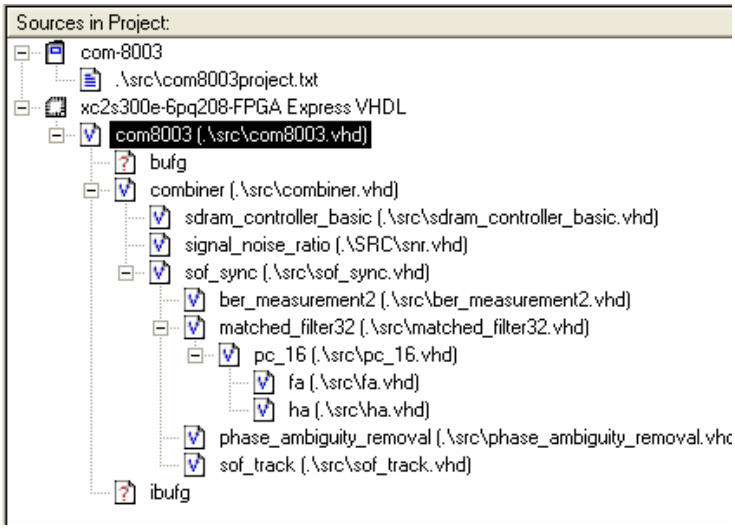
Xilinx-specific code

The VHDL source code was written in generic VHDL with few Xilinx primitives. No Xilinx CORE is used. The Xilinx primitives are:

- BUFG
- IBUFG

MSS • 18221 Flower Hill Way #A • Gaithersburg, Maryland 20879 • U.S.A.
Telephone: (240) 631-1111 Facsimile: (240) 631-1676 www.ComBlock.com

VHDL software hierarchy

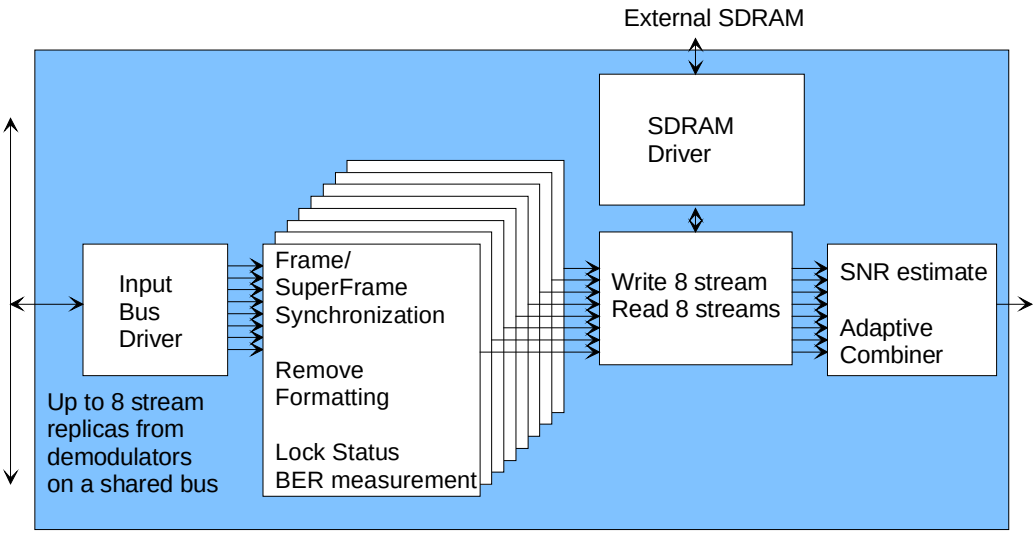


The code is stored with one, and only one, entity per file as shown above.

The root program (highlighted) is *com8003.vhd.*

The hierarchical nature of the VHDL code reflects the block diagram below:

- *com8003* is the root program which includes the input bus driver, the signal diversity combining in *combiner* and ancillary functions such as monitoring and control functions (interface with microprocessor).
- The *combiner* entity encapsulates most of the signal diversity combining functions, namely frame and superframe synchronization recovery in *sof_sync*, input elastic buffering, SDRAM access contention, SDRAM read and write transactions in *sdram_controller_basic*, signals quality estimation and final combining.
- The *sof_sync* entity detects frame and superframe synchronization using a 32-bit matched filter to detect the presence of unique words, confirms receiver lock by verifying the periodic nature of the 32-bit unique word, corrects for coherent BPSK/QPSK phase ambiguities and measures the BER from the unique word. Each frame consists of 4096 bits of data preceded by the 32-bit unique word 0x5A 0F BE 66. The superframe length is user-defined. It is an integer number of frames. The superframe unique word is the inverted version of the regular frame unique word.
- The *matched_filter32* entity detects the 32-bit unique word 0x5A0FBE66 in the input serial data stream through correlation. It also detect the unique word subject to 90deg,180deg,270deg phase ambiguity in a coherent BPSK/QPSK demodulator with gray encoding for the phase (see COM-1001 specifications). The detection threshold for the matched filter is set at approximately 10% bit errors (it tolerates 3 bit errors out of 32).
- The *sof_track* entity is a confirmation circuit for frame synchronization which eliminates false detections and missed detections at the matched filter. Periodic start of frame (SOF) pulses are generated reliably, even in the event of missed detection at the matched filter, in essence a kind of fly-wheel. *Sof_track* also generates a lock status based on the degree of confidence in detecting SOFs. The lock status is one of several input signal quality measurements to help with the subsequent combining algorithm.
- The *ber_measurement2* entity measures the bit error rate based on the periodic and known 32-bit unique words. The bit error rate is counted over 1024 bits. The BER is one of several input signal quality measurements to help with the subsequent combining algorithm.
- The *sdram_controller_basic* entity is the driver for the external 256MB SDRAM. It manages the SDRAM initialization, the periodic refresh cycles and random access read and write cycles.



COM-8003 Software Functional Block Diagram

Implementation

Overview

Signal diversity combining is a technique whereby a signal is transmitted over several media (satellite links for example) to improve the quality of service. The signal diversity combiner processes these input replicas by removing the differential delay at the receiver and summing the signals to yield the best signal to noise ratio.

Thus, the signal diversity combiner comprises three distinct circuits:

(a) a synchronization circuit, whereby the start of frame and start of superframe are detected for each input signal.

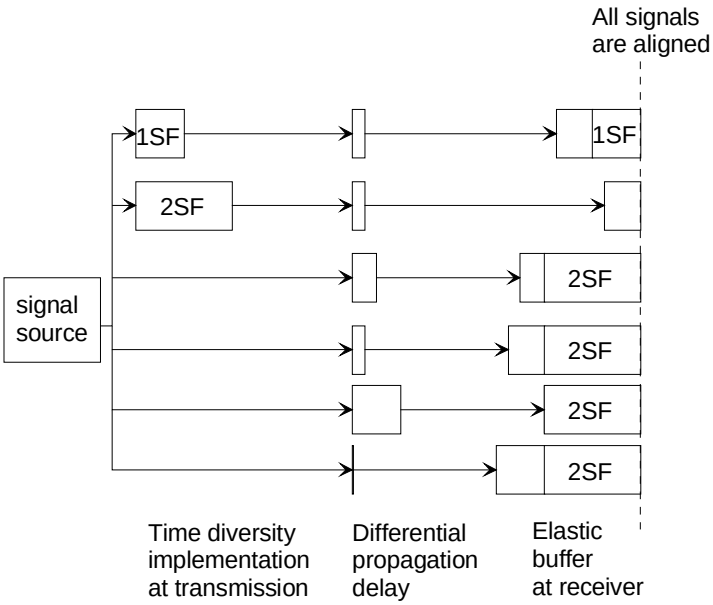
(b) An elastic buffer, whereby the differential delay among the various replica is removed.

(c) An intelligent combiner, whereby the synchronized replica are summed with adaptively controlled coefficients for maximum output SNR.

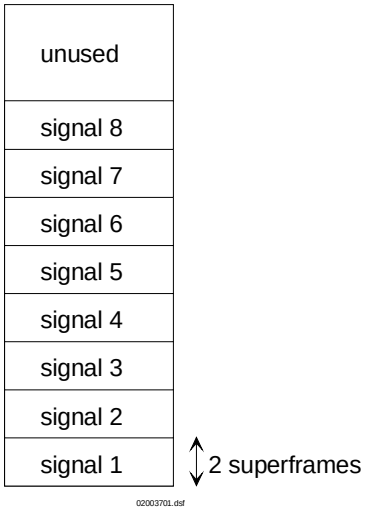
Synchronization Detection

Elastic Buffering

Elastic buffering is used to re-align in time all the signal replicas after they experience differential delays during propagation and at the transmitter.



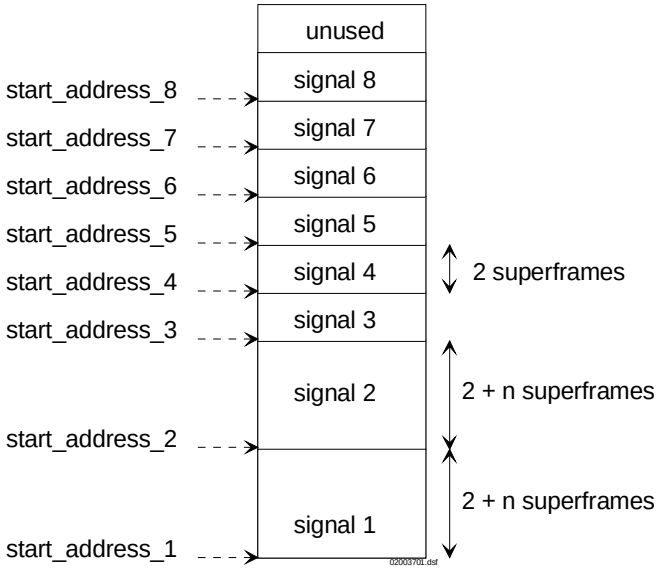
A single synchronous DRAM (SDRAM) module is shared among all signals. The 256 MB SDRAM module is partitioned in eight sections, to implement elastic buffering for each one of the eight input signals. Each section is sized so as to contain at least two superframes as shown below:



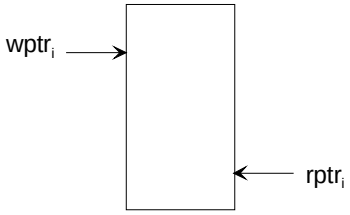
Each signal is formatted with a short (4096-bit) frame and a longer superframe. As the role of the superframe is to eliminate any time ambiguity occurring during propagation, the superframe size is selected as the smallest number of frames strictly greater than 2*(maximum differential propagation delay among replicas). By consequence, the superframe size depends on the data rate. Example:

- Data rate: 5 Kbps
- Maximum differential propagation delay among replicas: 4ms (12,000 km/c)
- Equivalent number of bits in differential delay: 20 bits
- Minimum superframe size = 1 frame = 4096 bits

There are, however, operational cases whereby a signal is purposely transmitted n superframes earlier than the other replicas. This method can be used to provided time diversity in a broadcast network for example. In such a case, the SDRAM section for a given signal can be increased from 2 to n+2.



Each section comprises a write pointer $wptr_i$ and a read pointer $rptr_i$.



During normal operation, the write pointer is slightly ahead of the read pointer.

The goal of the elastic buffer algorithm is to realign several streams in time. The algorithm is expected to be resilient to outside events such as intermittent receiver lock on one or several signals, blocking (all input signals disappear), drifting bit clocks and Doppler. The emphasis is on avoiding catastrophic failures while keeping the algorithm simple to implement and comprehend. No special effort is made here to assess the integrity of a given signal (this is for the follow-on combiner to decide). Instead, the emphasis in on maintaining the integrity of the read and write pointers only.

The data written to the SDRAM consists of the demodulated bit together with qualifiers such as bit quality (4-bit soft quantization), receiver lock status (SOF_LOCKx), start of superframe flag (SOSFx) and bit error rate summary BERx. The occupied space is thus 8 bits for each demodulated bit. The main objective is to keep demodulated bit and quality information together so that they are subject to the same delay through the SDRAM.

The frame synchronization sequence (unique word) is stripped off the data stream prior to writing to the SDRAM. Therefore, the occupied space for a frame is (4096 bits/frame) * (8bits/sample) / (64-bit SDRAM word) = 512 SDRAM words.

Write pointer management algorithm

Let us denote:
start_address$_i$ = lowest memory address in the sector reserved for signal i.
This address is always a multiple of superframe sizes and always in the form
n * ***sf_size*** *4096, where 4096 is the number of bits in a frame, ***sf_size*** is the number of frames in a superframe, and n is an integer. For ease of computing modulo operations, ***sf_size*** is restricted to the authorized values of 2,4,8,16,32,64,128 and n is selected to be a power of 2.

Let us denote
buffer_size$_i$ = the elastic buffer size for signal i.

The rules governing the write pointer for each signal i are as follows:

- definition: the write pointer indicates the next location to be written.

- In general. the write pointer is incremented after writing an incoming sample, even if the receiver does not clearly indicate lock. This means that all write pointers move independently of each others.
  In particular, the sampling clock frequency for each signal is distinct.
  The write pointer is incremented circularly in the range
  [***start_address***$_i$ **,** ***start_address***$_i$ **+** ***buffer_size***$_i$[

- Exception 1: “superframe is out of synchronization”:
  if a start of superframe is received while the write pointer points to a memory location inconsistent with a start of superframe location (i.e. not a multiple of ***sf_size****4096 samples), the write pointer is reset to ***start_address***$_i$, the first address in the sector.

- Exception 2: “frame is out of synchronization”:
  if a start of frame is received while the write pointer points to a memory location inconsistent with a start of frame location (i.e. not a multiple of 4096 samples), the write pointer is moved to the nearest SOF location. This exception is to take care of short loss of receiver lock (a few seconds) causing the bit clock to drift by a few bits. Particularly important if the receiver bit timing loop is a first order loop.

Read pointer management algorithm

Let us decompose the read pointer into an upper address ***rptr_superframe*** representing an integer number of superframes and a lower address ***rptr_bits*** representing the bits within a superframe.
rptr$_i$ = ***rptr_superframe***$_i$ * ***sf_size*** * 4096 + ***rptr_bits***$_i$ **+** ***start_address***$_i$

Let us denote ***sf_offset***$_i$ the fixed delay required at reception of signal i, expressed in number of samples This fixed and known delay is to compensate for the differential delay implemented at the transmitter in time-diversity systems. This delay is always a multiple number of superframes.

The rules governing the read pointer management for each signal i are as follows:

- definition: the read pointer indicates the next location to be read from.

- A single read clock, common to all signals, is used to move all read pointers.
  The read clock frequency is set slightly above (101%) the nominal data rate, to accommodate for local variations of crystal frequencies.

- In general, all active signals are read once per read clock period. The read pointers are incremented after reading. The read pointers are incremented circularly in the range [***start_address***$_i$ **,** ***start_address***$_{i+1}$[. As all read pointers are incremented simultaneously, their lower addresses ***rptr_bits***$_i$ are identical.

- Exception 1: “flow control check”.
  The read pointer is incremented at the read clock if and only if all qualified elastic buffers contain at least 1 sample to be read. By ‘qualified’, we mean that

(a) the receiver lock status for signal i (which can be checked by reading ahead the sample) indicates lock, and
(b) the alignment between read and write pointer is consistent with the desired superframe delay $\boldsymbol{sf\_offset_i}$ (see exception 2 below).
The number of samples ready to be read is computed as:
$(\boldsymbol{wptr_i} - \boldsymbol{rptr_i} - \boldsymbol{sf_offset_i})$modulo $\boldsymbol{buffer\_size_i}$

- Exception 2: superframe realignment.
  The elastic buffer algorithm will ensure that the delay between read and write pointer is consistent with the desired superframe delay $\boldsymbol{sf\_offset_i}$. The delay $(\boldsymbol{wptr_i} - \boldsymbol{rptr_i})$modulo $\boldsymbol{buffer\_size_i}$ must be within the window $[\boldsymbol{sf\_offset_i}, \boldsymbol{sf\_offset_i} + ½\ \boldsymbol{sf\_size} * 4096]$. In other words, the minimum delay is sf_offset, and the maximum delay sf_offset + ½ a superframe.

  If this condition is not met, two cases can occur:
  (a) It is correctable: the delay $(\boldsymbol{wptr_i} - \boldsymbol{rptr_i})$modulo $(\boldsymbol{sf\_size} * 4096)$ is within the window $[0, ½\ \boldsymbol{sf\_size} * 4096]$. the elastic buffer algorithm will adjust the upper address ***rptr_superframe*** so that the delay between read and write pointer is consistent with the desired superframe delay.

  (b) It is not correctable: the delay $(\boldsymbol{wptr_i} - \boldsymbol{rptr_i})$modulo $(\boldsymbol{sf\_size} * 4096)$ is outside the window $[0, ½\ \boldsymbol{sf\_size} * 4096]$. In this case, the elastic buffer is disqualified (see exception 1 above). No read pointer adjustment is implemented other than the regular increment.

SDRAM Controller

The code is written for interfacing with a generic 256MB SDRAM SODIMM memory module. A set of specifications can be obtained from Micron (MT48LC16M16A2 – 4 MEG X 16 X 4 BANKS). The clock provided to the SDRAM is 40 MHz, well below the specified maximum for PC100/PC133 SDRAMs. The read and write pointers representing SDRAM addresses are written with 27-bit precision so as to allow easy extension to 1GB SDRAM memory modules if needed.

In order to minimize the number of FPGA flip-flops used for serial to parallel conversions, a basic SDRAM write transaction deals with a single 64-bit word. Burst mode, although conducive to a higher SDRAM throughput, is not used because of the extra FPGA resources required.

In addition to the read, and write operations, the *sdram_controller_basic* entity takes care of the initial SDRAM initialization, of periodic auto refresh and of the contention avoidance mechanism between the asynchronous read/write transactions and the auto refresh transaction.

Signal Quality Criteria

In order to assist the combiner in selecting the best combinations of input signals, multiple signal quality measurements are performed throughout the FPGA. Some measurements are performed prior to the selective SDRAM delay:

- receiver lock status, based on the detection of periodic start of frame synchronization markers.
- BER measurement performed on the 32-bit unique words over 32 successive frames.
- 4-bit soft-quantized samples gathered at the demodulators.

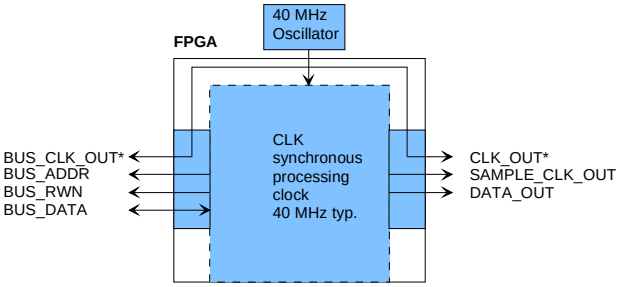
These early quality measurements are performed for each input signal replica. Their timing alignement with the demodulated bit is preserved through the SDRAM. In other words, the quality measurements are subject to the same delay through the SDRAM as the demodulated information bit.

Some other quality measurements are performed after the selective SDRAM delay:

- SNR measurement using the standard deviation of the 4-bit soft-quantized samples.
- detection of misaligned streams.

Clock / Timing

The clock distribution scheme embodied in the COM-8003 is illustrated below.



Baseline clock architecture

Darker blue = internal 40 MHz clock
**** indicates edge-trigger signal***

The core signal processing performed within the FPGA and all the inputs/outputs are synchronous with the 40 MHz internal oscillator.

VHDL Simulation

Representative simulation screens for salient internal signals are captured and discussed below.

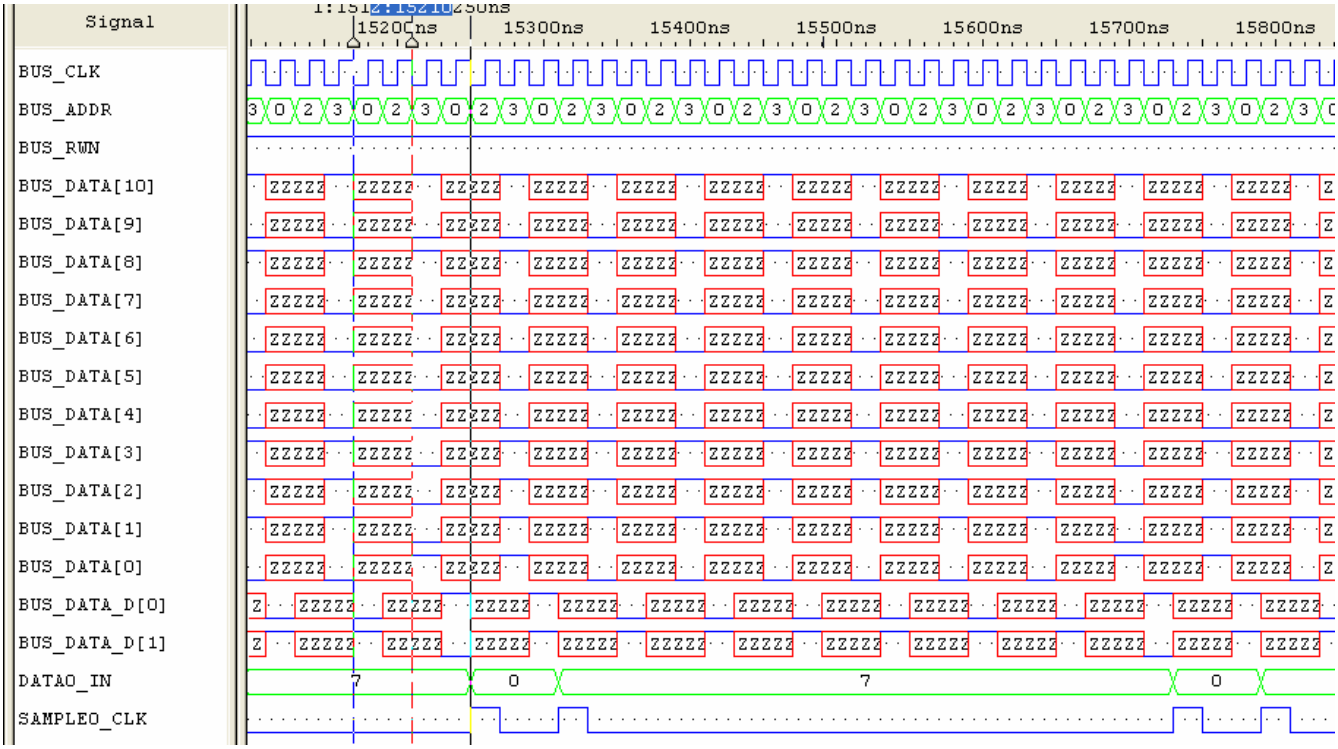
Input Data Bus

Definition:

Input Module Interface Shared bus	**Definition**
BUS_CLK_OUT	40 MHz output reference clock for use on the synchronous bus.
BUS_ADDR[3:0]	Bus address. Output (since this module is the bus Master). Designates which slave module is targeted for this read or write transaction. All 1's indicates that the write data is to be broadcasted to all receiving slave modules. Read at the rising edge of BUS_CLK_OUT.
BUS_RWN	Read/Write#. Output (since this module is the bus Master). Indicates whether a read (1) or write (0) transaction is conducted. Read at the rising edge of BUS_CLK_OUT. Read and Write refer to the bus master's perspective.
BUS_DATA[15:0]	Bi-directional data bus. Output when BUS_RWN = '0'. Input when BUS_RWN = '1'. Read data is delayed by two clock periods with respect to the read request from the master. Functional definition during read: • bit 0 SAMPLE_CLK_IN. '1' when 4-bit soft-quantized samples are available. • bits(4:1) DATA_IN(3:0) soft quantized samples from demodulator. MSb is the information bit. 3 LSBs are quality bits. • bits(15:5) undefined

The COM-8003 is capable of reading multiple data streams from multiple demodulators over a shared data bus. The COM-8003 is bus 'master', i.e. it drives the BUS_CLK_OUT, BUS_ADDR, BUS_RWN signals. The 16-bit BUS_DATA is bi-directional. The simulation below shows that the COM-8003 is reading data from four demodulators at bus addresses 0,1,2 and 3 respectively. For simplicity, only one demodulator, responding to bus address 0, is simulated.

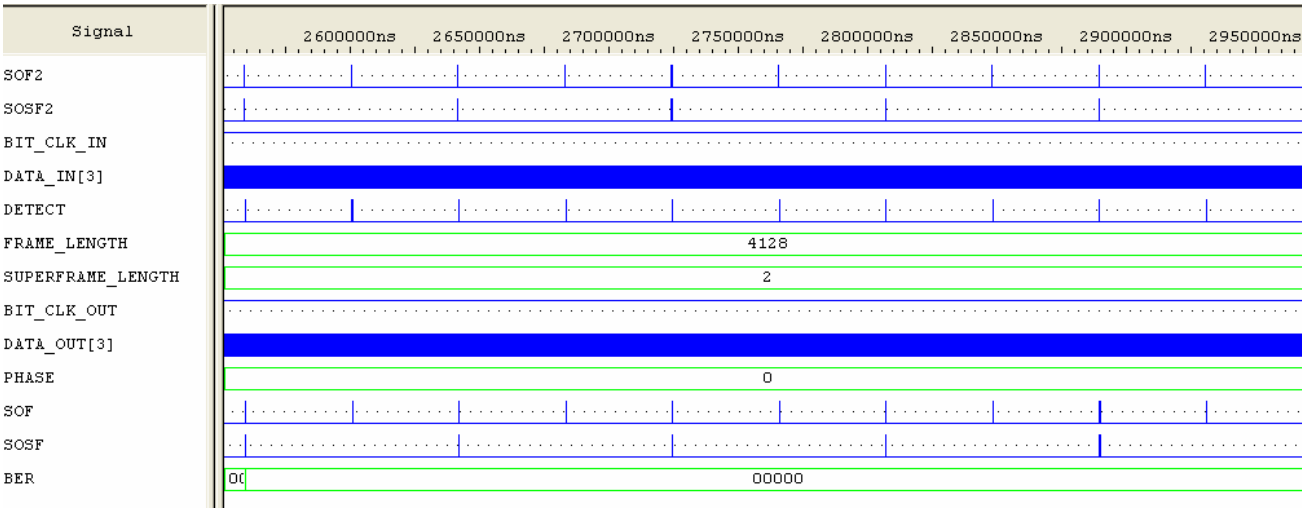
The simulation shows that the read data (read marker) is delayed by two clock periods with respect to the read request (blue marker). The bus data is converted to internal serial data streams (DATA0_IN/SAMPLE0_CLK) two clocks later (black marker).



Data bus simulation. COM-1018 DSSS demodulator -> COM-8003 signal diversity combiner

Frame / SuperFrame Synchronization

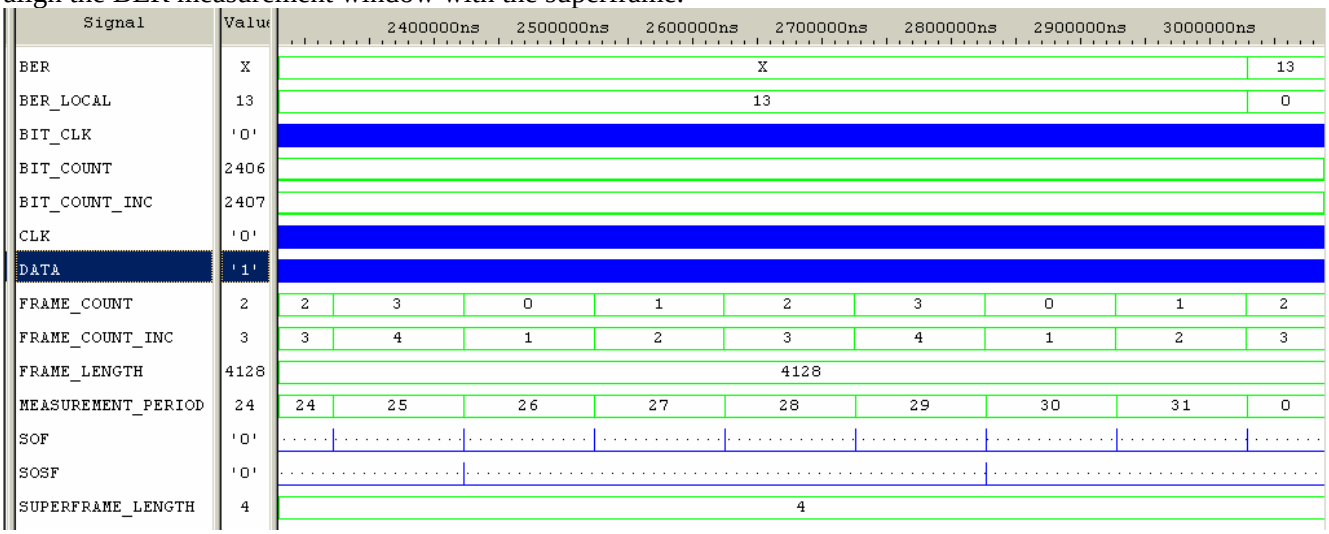
Key signals for the *sof_sync* component are shown below for a configuration consisting of a 4128-bit long frame and a 2-frame long superframe. DATA_IN, BIT_CLK_IN are the serial input signal. SOF2/SOSF2 are the start of frame and start of superframe at the generator, shown for the sole purpose of comparison with the recovered start of frame (SOF) and start of superframe (SOSF). The detected phase ambiguity is 0, meaning no inversion or phase ambiguity at the demodulator. The bit error rate, shown here as zero, is computed over 1024 bit (32 consecutive 32-bit unique words). DETECT is the unprocessed output of the *matched_filter32* 32-bit matched filter entity to detect one of four possible versions of the unique word.



Frame / SuperFrame Synchronization

Bit Error Rate Measurement

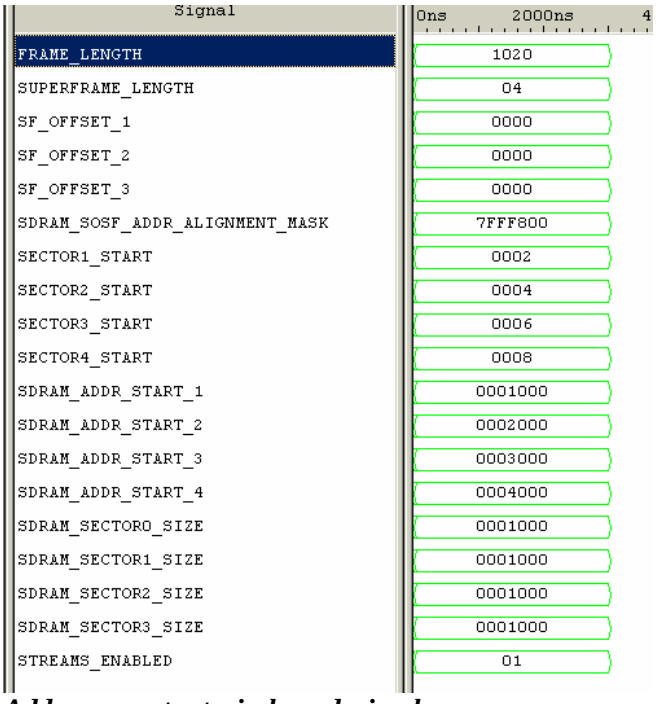
The *ber_measurement2* component measures the bit error rate (uncoded) over 1024 bits, using the known bits within the unique word synchronization sequence. This method provides a good estimate of the signal quality on-the-fly, without having to substitute the payload data with a PRBS sequence. The measurement period is 32 frames (to reach a total of 32*32-bit unique words = 1024 bits). The signal BER_LOCAL is the running bit error count while the signal BER is the final count for the last detection window. There is no requirement to align the BER measurement window with the superframe.



Combiner configuration

The main configuration parameters of the *combiner* component are:

- the superframe length SUPERFRAME_LENGTH (expressed as number of fixed-length frames)
- the known delay SF_OFFSET_i of received stream i with respect to the reference stream 0 (expressed as number of superframes)
- the list of active streams STREAMS_ENABLED



Address constants, in hexadecimal

The SDRAM is automatically apportioned into contiguous sectors, one for each stream. The simulation above shows the start addresses. When computing SDRAM addresses, the following should be taking into account:

(a) the SDRAM data width is 64-bit, thus each address represents 8 bytes of data.

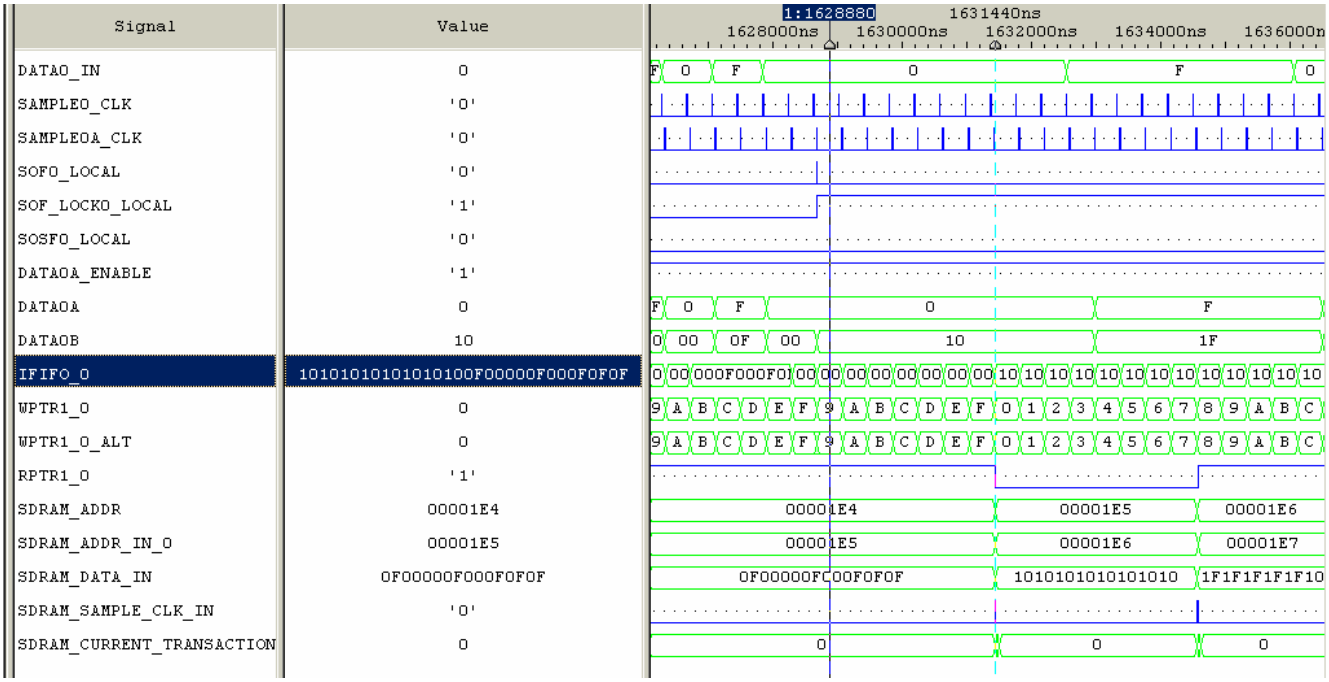
(b) Each demodulator sample occupies one byte of SDRAM storage (4 bit soft quantization, lock status, start of superframe marker and BER summary).

Signal path to SDRAM

The signal path within the *combiner* component, from the input to the SDRAM write is illustrated by the one-stream simulation below. The 4-bit soft-quantized demodulator output is DATA0_IN with its sample clock SAMPLE0_CLK. After going through the *sof_sync* component, the data stream DATA0A is corrected for demodulator phase ambiguities. Ancillary information such as the synchronization information (SOF_LOCK0) is appended to form the one-byte sample DATA0B. The capture below clearly shows the start-of-frame lock status information changing in bit 4 DATA0B as lock is detected.

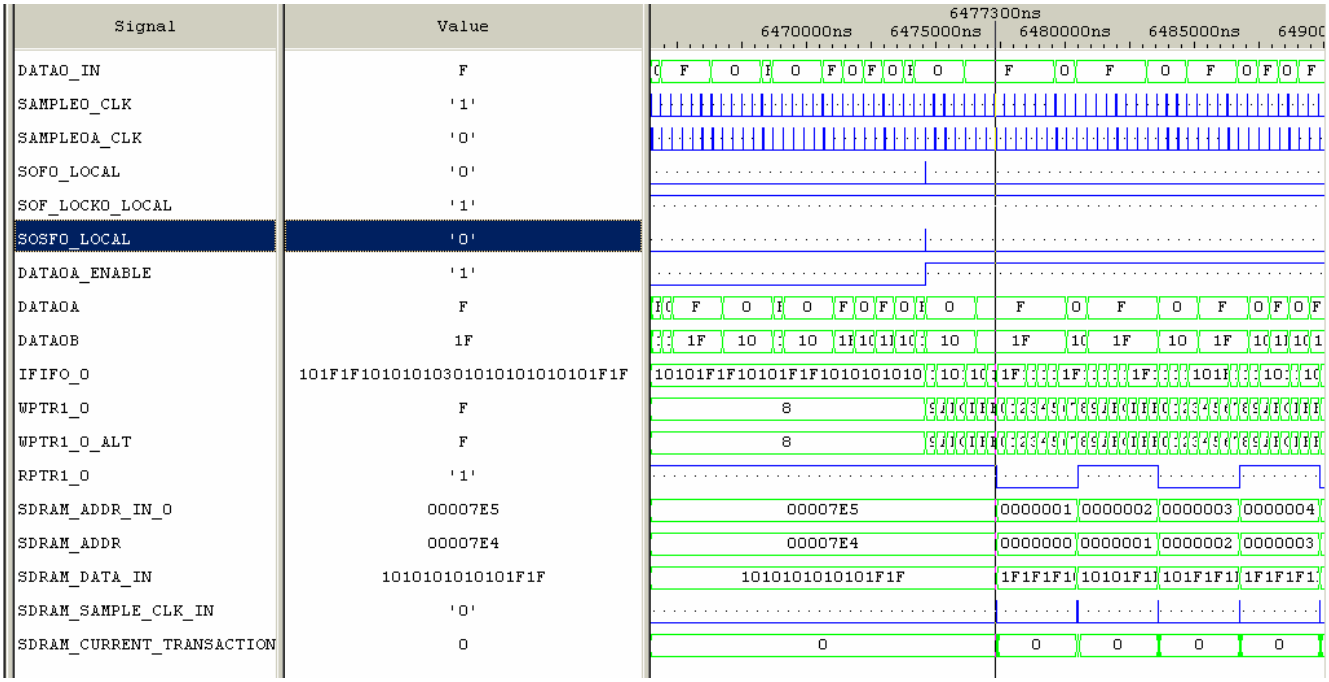
The input FIFO IFIFO_0 aggregates the 8-bit wide DATA0B samples into 64-bit wide SDRAM_DATA_IN samples. The FIFO itself is 128-bit long so as to allow simultaneous read and write in opposite halfs. WPTR1_0 and RPTR1_0 are the input FIFO write and read pointers, respectively.

The capture shows an instance of write pointer realignement at the blue marker because the detected start of frame is not aligned with the 64-bit word boundary in accordance with the write pointer management algorithm described in a previous section The write pointer WPTR1_0 is reset from F to 8.



Data path (single stream 0) from demodulated sample to SDRAM

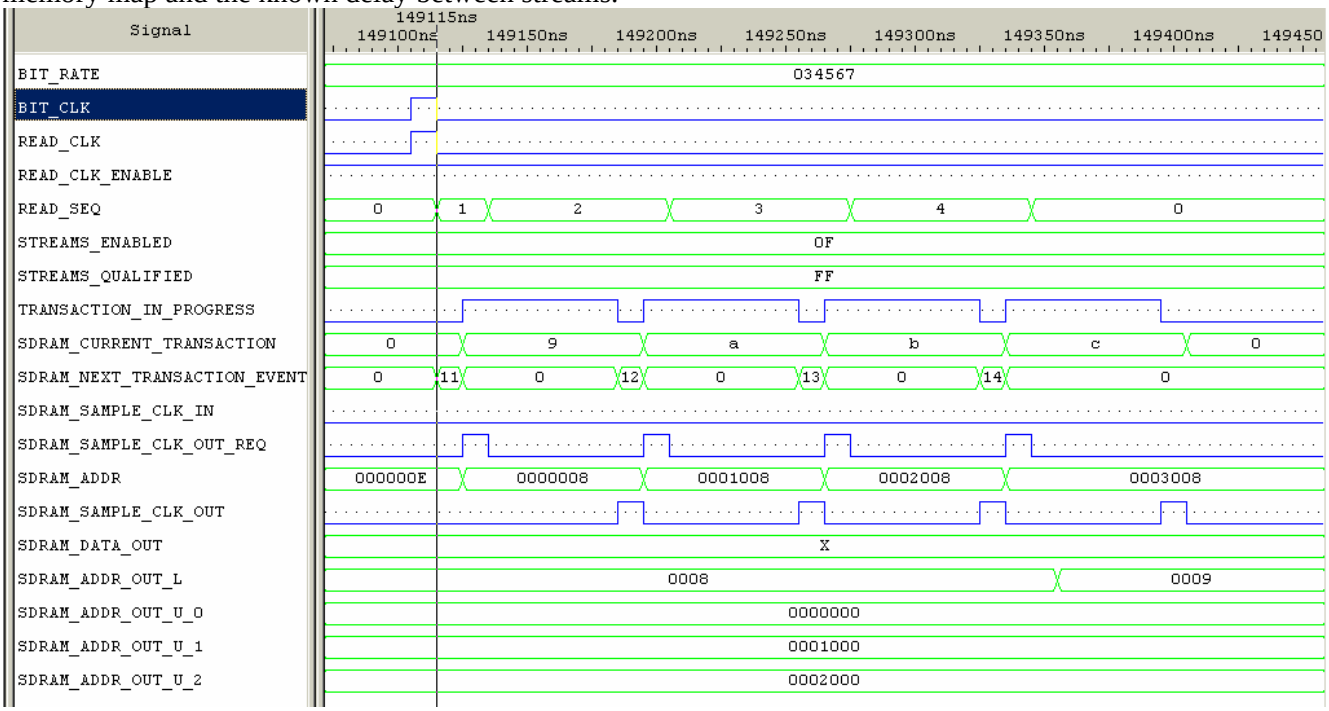
Another instance of write pointer realignement is shown in the capture below. In this case, the start of superframe SOSF0_LOCAL is not aligned with an SDRAM superframe boundary. The SDRAM address is reset to the start address for this stream.



Superframe is out of synchronization with SDRAM write pointer. SDRAM write pointer is reset.

Signal path from SDRAM

All active streams are read from the SDRAM as a group. The bit rate is controlled by a numerically controlled oscillator which generates BIT_CLK. Since the SDRAM samples are 8 byte wide, the SDRAM read rate is READ_CLK (BIT_CLK / 8). Each READ_CLK pulse triggers a sequence of SDRAM read transactions, one for each active stream. The simulation capture below illustrates this in the case of 4 active streams (STREAMS_ENABLED = “00001111”). The SDRAM lower address bits SDRAM_ADDR_OUT_L are the same for all active streams. The SDRAM upper address bits SDRAM_ADDR_OUT_U_x reflect the SDRAM memory map and the known delay between streams.

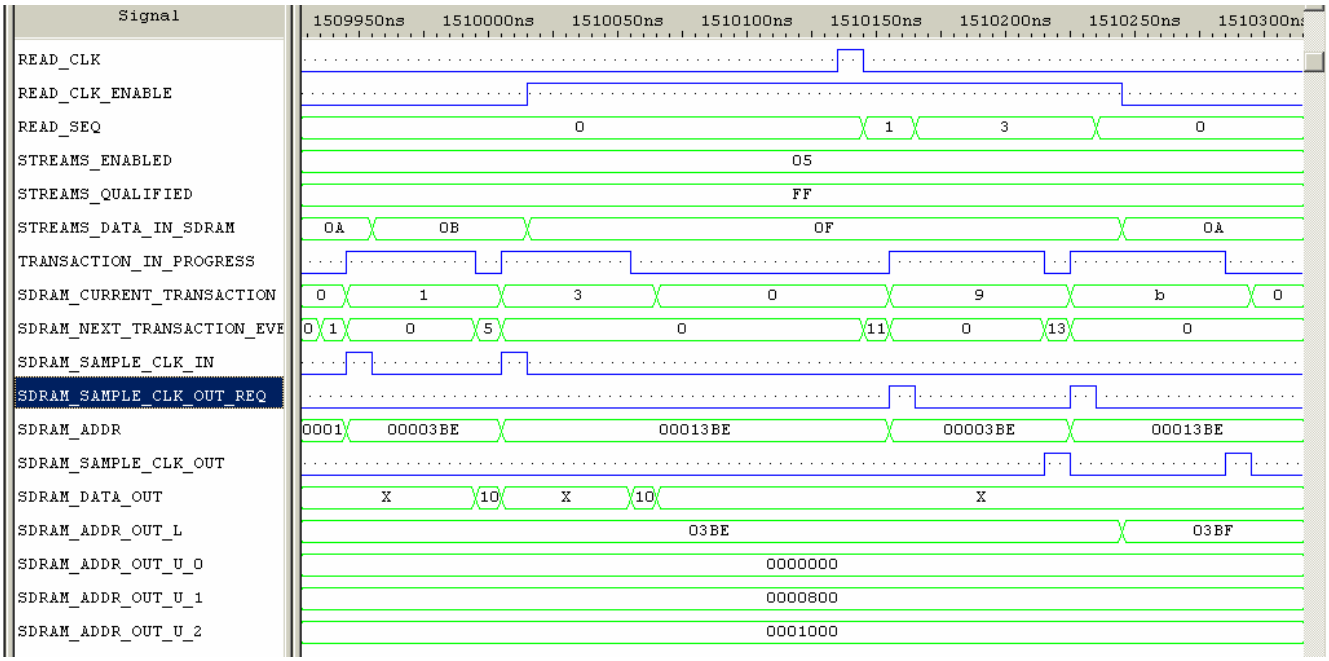


Reading 4 streams from the SDRAM

Flow Control

Demodulated data streams are written to the SDRAM at data rates determined by the demodulators. All data streams are nominally at the same data rate. However, each data stream may experience small data rate variations due to such effect as Doppler.

On the other side (the read side) of the SDRAM, all data streams are read at the same rate which is set by a numerically controlled oscillator. The NCO is set slightly higher (1 + 1/128) than the nominal demodulator rate. The READ_CLK is generated by the NCO. A flow control mechanism ensures that the read pointers do not move past the write pointers. When the SDRAM virtual buffers are empty, READ_CLK pulses are temporarily disabled by the READ_CLK_ENABLE signal. The simulation capture below illustrates this flow control mechanism. It can be observed that the samples are addresses 3BE and 13BE are read from SDRAM shortly after being written to SDRAM.

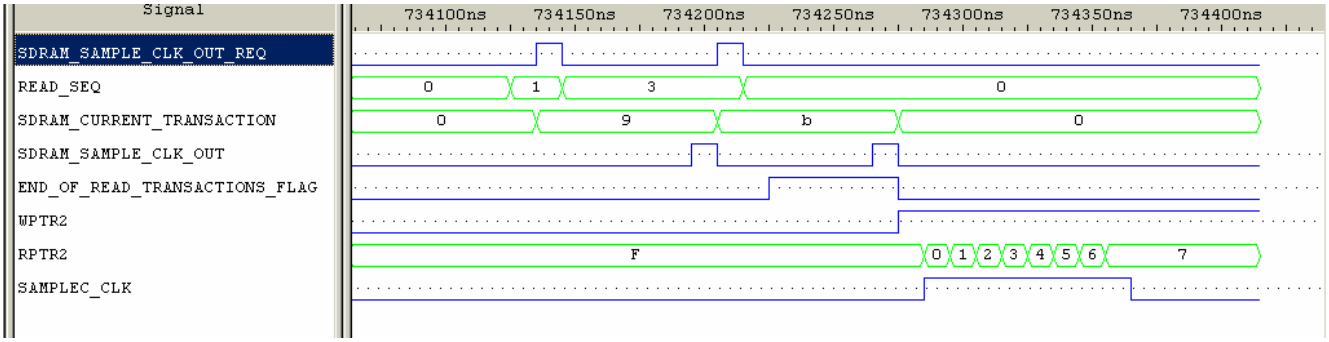


Flow control

Output FIFO

Data is read from the SDRAM 64-bit at a time. Each SDRAM output word consists of 8 x 8-bit samples. The 64-bit words are written to the output FIFO OFIFO_x , where x identifies which stream. The 128-bit wide FIFOs are used as “A/B” buffers. Data is written to the “A” half while being read from the “B” half and vice versa. The write pointer WPTR2 points to the FIFO half to be written to. The read pointer RPTR2 points to the FIFO byte to be read. Since all active streams are synchronized at this point, the write and read pointers are common to all streams.

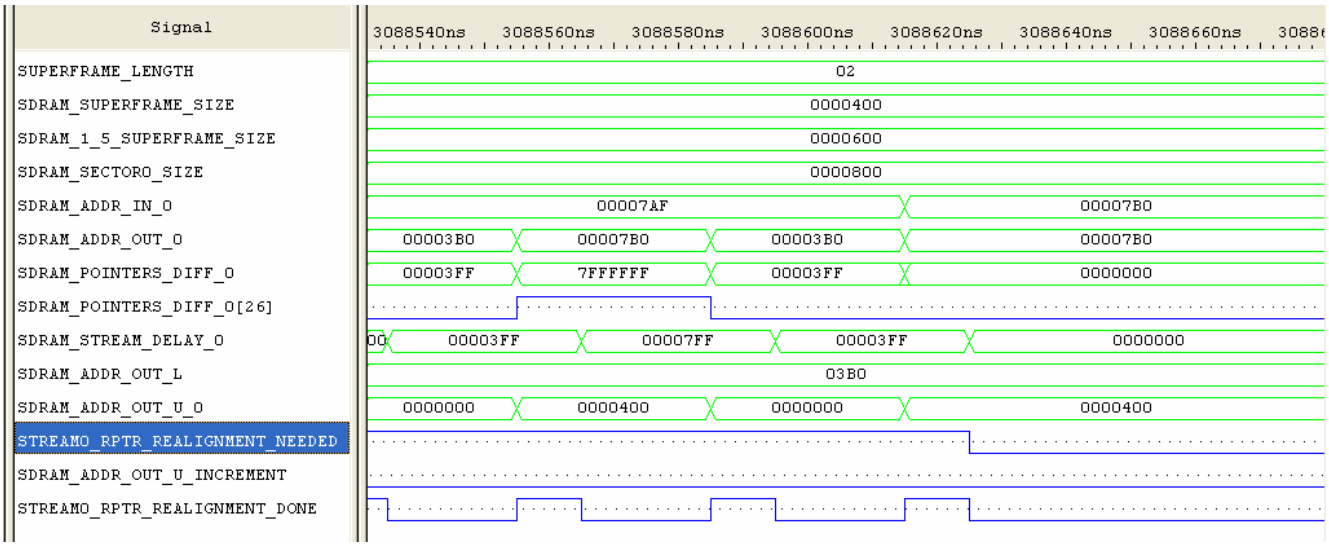
Below is a simulation capture illustrating the output FIFO operation in the parallel to serial conversion. Two streams are active in this simulation.



Output FIFO. 64-bit parallel to 8-bit serial conversion.

Read pointer / Write pointer synchronization

The VHDL code automatically detects any misalignment between the write pointer and the read pointer for each stream. Any misalignment, as per the rules described earlier in this document, is immediately detected and the flag STREAMx_RPTR_REALIGNMENT_NEEDED is raised. Consequently, the upper address field is incremented one half superframe at a time until the misalignment is corrected. If the misalignment is not correctable, the flag STREAMx_RPTR_REALIGNMENT_NEEDED stays high and the corresponding stream is ignored.



Misalignment detected and corrected.

FPGA Occupancy

```
Design Summary:
   Number of errors:      0
   Number of warnings:    4
   Number of Slices:                3,070 out of  3,072   99%
   Number of Slices containing
      unrelated logic:                177 out of  3,070    5%
   Number of Slice Flip Flops:      2,561 out of  6,144   41%
   Total Number 4 input LUTs:       4,266 out of  6,144   69%
      Number used as LUTs:                           3,946
      Number used as a route-thru:                     291
      Number used as Shift registers:                   29
   Number of bonded IOBs:             123 out of    142   86%
   Number of GCLKs:                     3 out of      4   75%
   Number of GCLKIOBs:                  3 out of      4   75%
Total equivalent gate count for design:  55,153
```

```
Additional JTAG gate count for IOBs:  6,048
```

Acknowledgments

This product development was funded by SBIR Phase II research contract “Signal Diversity Combining for Improved Satellite Communications” awarded by the US Air Force Research Laboratory, Rome, NY.

Contact Information

MSS • 18221 Flower Hill Way #A •
Gaithersburg, Maryland 20879 • U.S.A.
Telephone: (240) 631-1111
Facsimile: (240) 631-1676
E-mail: info@comblock.com