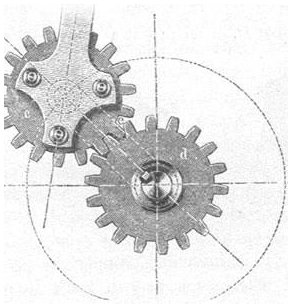


SV3D_09 Command Interface Design

1 Revision History

1.0	2016/07/26	DE: created SV3D_01 command interface document. Captures common use cases.
1.1	2016/07/27	DE: modified 'Port' column in Table 9. Now indicates which commands must be set on both PortA and PortB.
1.2	2016/11/08	DE: added missing pattern start command
1.3	2017/01/05	DE: update pattern sequencer slots.
1.4	2017/01/13	DE: added instruction sequencer commands. Added 'Pattern Mode' and 'Pattern Disabled State'
1.5	2017/04/21	DE: added User Clock Modes and Frequency commands.
1.6	2017/05/03	DE: removed reference to cmd proc ready flag.
1.7	2017/08/11	DE: Added TX fine phase delay [0x0550] and added trigger pin description to Table 16.
1.8	2017/08/18	DE: Added block write for Rx Phase [0x0440] and added Tx Pattern Status [0x0582].
1.9	2017/11/10	DE: Added commands to support Rx User Pattern Memory. Added 'Memory Underflow' command.
2.0	2018/01/15	DE: Added command for fan out mode. Updated Pattern sequencer description.
2.1	2018/03/16	DE: Added commands for force zero, coarse skew write, fine skew write, and Rx Rate Ratio.
2.2	2018/05/01	DE: Added command to read module speed grade.
2.3	2018/07/30	DE: Added command to disable tx alignment data.
2.4	2018/11/29	DE: Added calibration control registers.
2.5	2019/01/14	DE: Added PTI registers.
2.6	2019/02/20	DE: Modified PTI Write command and added PTI Read Memory (0x0371)
2.7	2023/11/21	DE: Updated number of user slots and programming examples. Updated Global and Hardware status codes. Added Fan Out Mode descriptions. Added information on Tx Channel Groupings (see Global Target Mask)

2 Motivation

Provides design detail for the command interface exposed by the SV3. The SPI ports and its timing parameters are documented elsewhere.

3 Conventions

Unless otherwise noted, multi-byte words are always transmitted MSB-first. Signed numbers are always expressed in 2's-compliment notation.

4 Introduction

All control of SV3 is carried out through its command interface, depicted in Figure 1. Control is carried out through the slave SPI ports. A soft-core processor on the module parses incoming SPI data, extracts and executes the commands embedded in it, and responds appropriately. The soft-core processor exerts control over all aspects of the module's functionality, shown simply as "controlled components" in the figure.

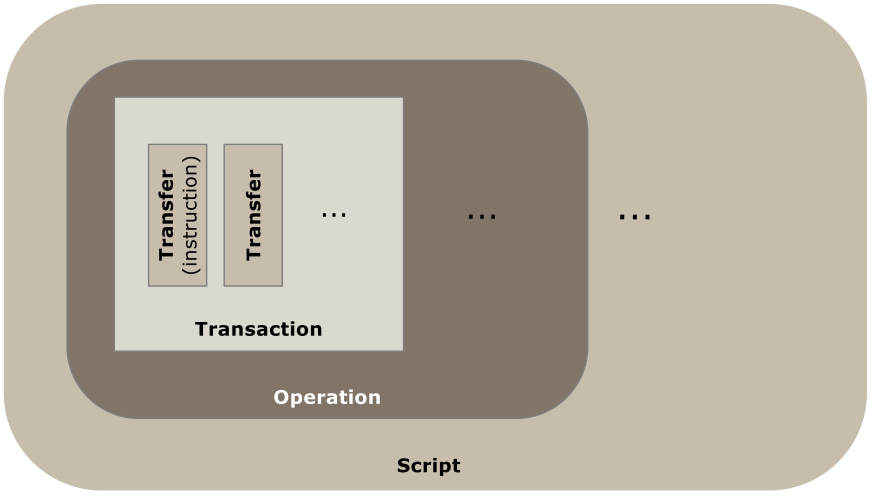


Figure 1: Hierarchy of the command interface

- **Program:** A *program* runs on the host and controls the module to carry out a high-level task. Examples are an eye diagram measurement, and a mask test. Programs consist of many *operations* chained together.
- **Operation:** *Operations* coincide with atomic module functions. The host carries out an operation by performing a short series of *transactions*. Examples are performing a bit error rate test (BERT) and setting a threshold voltage.
- **Transaction:** *Transactions* are the individual reads and writes that make up an operation. For example, a BERT operation consists of two transactions: one register write to start the BERT, and one register read to retrieve the result. Each transaction consists of one or more *transfers*, and the first transfer of every transaction is an *instruction*.
- **Transfer:** A *transfer* communicates a specific kind of information between the host and module. Examples of transfers are an instruction, a data block, and an acknowledge. A set of transaction models prescribes the sequences of transfers to follow in all communication with the module.

5 Command Interface

5.1 The Command Hierarchy

The command interface can be described hierarchically, as depicted in Figure 1. Terminology is summarized in the figure, and each level of the hierarchy is described in the sections that follow.

5.1.1 Command Hierarchy Example

A concrete example demonstrating this hierarchy is shown in Figure 2. At the left lies part of a program, shown as a series of operations. Each of these operations will have an implementation on the host, though in this example only the implementation of the Single-Shot BERT operation is shown. Moving to the right, we see the Single-Shot BERT operation expanded into its two constituent transactions: a Write Data Block, to start up to 16 BERTs, and Read Data Block, to obtain the results. Each transaction consists of several transfers: the first is always an instruction, and for data block transfers the second transfer contains the data block, and the third an Acknowledgement. The following sections will elaborate on the transaction models.

The wait state between the two transactions is necessary to allow the module to complete the BERT before reading the result – this is discussed further in Section 5.2.1. The shorter, fixed wait state between transfers reflects the time needed for the Command Processor to retrieve and act on the incoming SPI data.

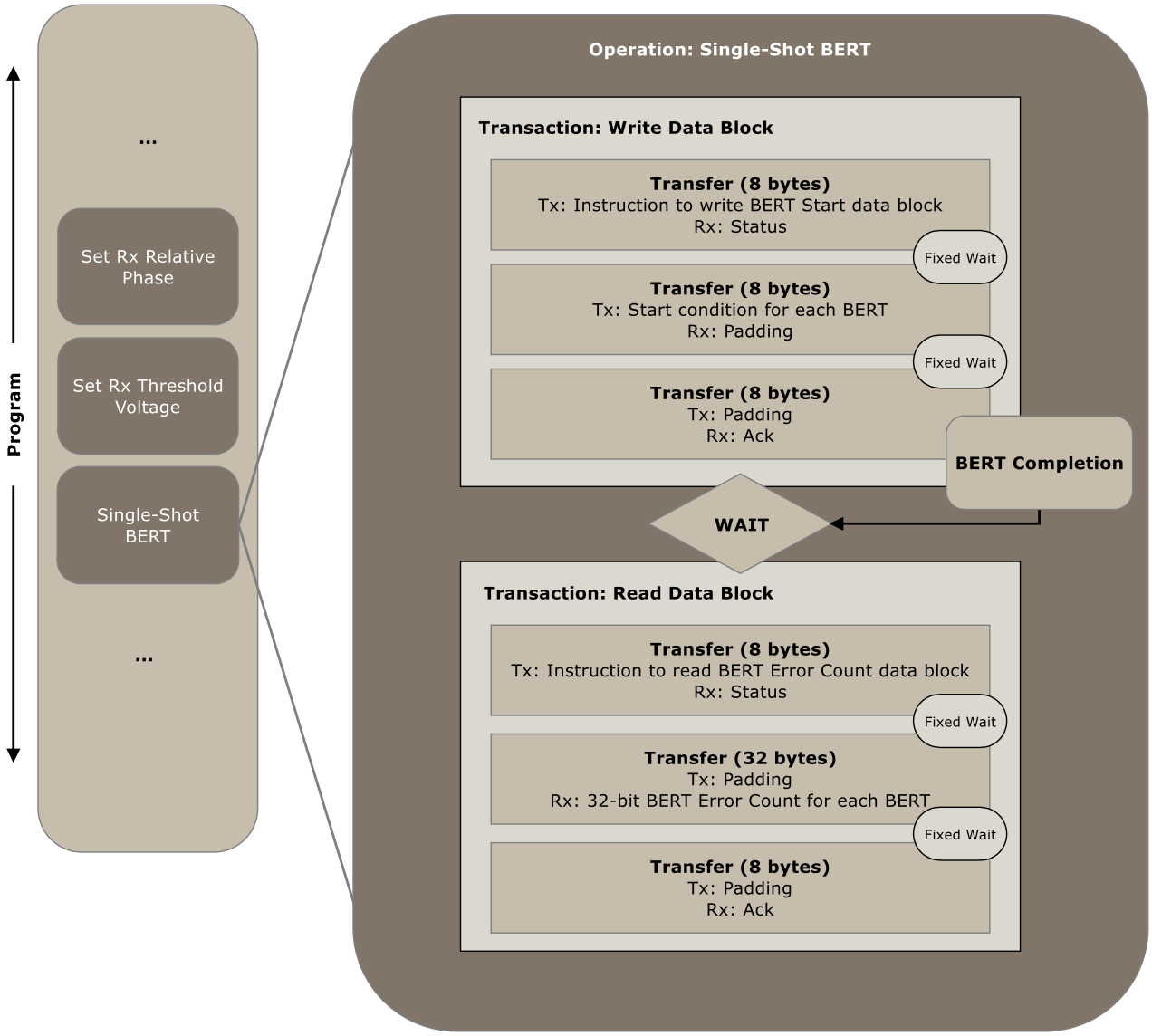


Figure 2: An example demonstrating the command hierarchy

5.2 Operations

Operations are sets of transactions, implemented on the host, designed to accomplish some goal, such as performing a Single-Shot BERT. In general, an operation consists of many transactions, though some simple operations, such as retrieving an immediately-available value, are single-transaction operations. All operations follow the operation models summarized in Table 1.

Operation model	Transaction	Transaction Description	Example
Get/set immediate	1	Read/write register or data	Read module status
Extended operation without feedback	1 < wait >	Write register / data to start operation < wait for completion >	Set threshold voltage on multiple Rx channels
Extended operation with feedback	1 < wait > 2	Write reg / data to start operation < wait for completion > Read result reg / data	Perform BERT

Table 1: Operation models and their constituent transactions

5.2.1 Wait States

Most of the operation models include a *wait* state, during which the module is allowed to complete a requested action before subsequent transfers are accepted. To determine completion of an action on the module, two mechanisms are available:

1. Status polling over SPI
2. Fixed worst-case timing

Fixed timing needs to reflect the worst-case duration of variable-duration commands, making it unnecessarily pessimistic. SPI polling, on the other hand, requires the host to be actively interrogating the device, meaning it can't be doing something else in the meantime.

Status Polling

During a read register transaction, described below, status bits are returned in the first transaction, and reflect the busy state of the module. By repeatedly issuing a read register transaction (targeting any register) and checking the status that's returned, the busy state of the module can be assessed. The contents of the status bits are discussed in Section 5.4.2.

Fixed worst-case timing

This represents the longest the operation could possibly take, meaning that a fixed delay can be used by the host, after which the module is guaranteed to be ready.

One limitation of this approach is that externally-triggered operations begin at a time determined by the trigger, and so the overall completion time needs to take into account the trigger time. It's up to the host to compute the overall timing, based on documented worst-case trigger-to-completion times, and knowledge of when the trigger is asserted.

Fixed timing information is implementation-dependent, and is TBD for the moment.

5.3 Transactions

A transaction is the collection of transfers required to perform a complete transmission of information between host and module. Every transaction begins with the host transmitting an instruction to the module. From there, the module interprets the op code to determine which transaction model to follow. Table 2 summarizes the transaction models, showing their constituent transfers.

Note that, since SPI is a full-duplex standard, each byte written to the module is accompanied by a byte read by the host. The data shown in the Tx column is the data transmitted to the module. The data in the Rx column is transferred simultaneously, from the module to the host.

Transaction model	Transfer	Tx	Rx
Read register	1	Instruction: register address	Status
	2	Tx Padding (8 bytes)	Ack, Register value
Write register	1	Instruction: register address, new value	Status
	2	Tx Padding (8 bytes)	Ack
Read data block	1	Instruction: data address, data length	Status
	2	Tx Padding (arbitrary length)	Data block
	3	Tx Padding (8 bytes)	Ack
Write data block	1	Instruction: data address, data length	Status
	2	Data block to write (arbitrary length)	Rx padding
	3	Tx Padding (8 bytes)	Ack

Table 2: Transaction models and their constituent transfers

As shown, different subsets of the instruction are used depending on the transaction model. Some instruction fields are omitted from this table, such as the target field, which is used in a subset of the write register transactions. Individual transfer types are described below.

5.4 Transfers

At the lowest level of the command interface hierarchy is the transfer. A SV3 transfer corresponds to a single SPI transfer, meaning the slave select line is asserted, bytes are transferred (in both directions), and the slave select line deasserted. The following sections describe the data types that make up the transfers in Table 2.

As indicated in the text accompanying Figure 2, a fixed, minimum delay must exist between transfers, to allow the command processor to retrieve and act on the incoming SPI data. Alternatively, a flag can be used to indicate when the Command Processor is ready to accept a new transfer.

5.4.1 Instruction

Every instruction is 8 bytes long, and follows the format depicted in Table 3. Table 4 summarizes the meaning of each field of the instruction, and Table 5 enumerates the transfer codes used in the opcode field. Example instructions are provided in Table 6.

Field	Opcode		Address		Target	Data	
Subfield	Rsvd	Transfer	Comp	Sub			Total
Byte cnt	1		2		1	4	8 bytes
Byte addr	7		6:5		4	3:0	
Bit cnt	6	2	8	8	8	32	64 bits
Bit addr	63:58	57:56	55:48	47:40	39:32	31:0	

Table 3: The 8-byte instruction format

Field	Description
Opcode	8 bits containing a 2-bit transfer code (see below), and 6 reserved bits which should be written as zero
Address	16 bits total consisting of component address and sub-address
Target	8 bits used to specify destinations / sources for some registers – e.g. the User Pattern Slot ID is specified as a target when addressing the User Data Pattern Memory
Data	32 bits used to set register values; also used as data lengths in data block transfers – interpretation differs depending on address and opcode

Table 4: Instruction fields

Code	Meaning
00	Read register
01	Write register
10	Read data block
11	Write data block

Table 5: Transfer codes

Instruction	Opcode	Address	Target	Data
Read status	0b00000000 (Read register)	0x01 02 (Status)	0x00 (N/A)	0x00000000 (N/A)
Set PRBS Polynomial Order	0b00000001 (Write register)	0x03 30 (PRBS Poly)	0x00 (N/A)	0x00000005 (PRBS-5)
Download data pattern	0b00000010 (Read data block)	0x03 10 (User data pat)	0x00 (N/A)	0x00001000 (length)
Upload data pattern	0b00000011 (Write data block)	0x03 10 (User data pat)	0x00 (N/A)	0x00001000 (length)

Table 6: Instruction examples

5.4.2 Status transfer

While an 8-byte instruction is being transmitted, the module will respond with its *status*. The status is simply a bitfield, encoded as follows:

Bit	Meaning	Details
0	Presence (1)	Always returns 1 when the module is present – useful as a debugging tool
1	Busy (0) / Okay (1)	0: Busy: indicates the module is busy with a previously-issued command; the transaction will not proceed 1: Okay: the transaction will proceed normally
2	Global Status Error (0) / Okay (1)	When set to 0, indicates that one of the Errors reported in the Global status register has been asserted. 0: Global Status Error 1: Okay
63:3	Reserved	

Table 7: Status bitfield

5.4.3 Ack transfer

8-byte Ack transfers are used to indicate the validity of instructions and successful transmission of data blocks. They follow a similar encoding to the status bitfield, as shown below.

Bit	Meaning	Details
0	Presence	Always returns 1 when the module is present – useful as a debugging tool
1	Transaction valid	A 0 indicates the transaction was invalid and will be ignored – this would be the case for an invalid instruction, for example; A 1 indicates the transaction was successful
2	Parameters in range and valid	A 0 indicates the parameters of the transaction were invalid and so the transaction will be ignored – this would be the case for an out-of range voltage setting, for example; A 1 indicates the transaction was successful;
31:3	Reserved	
63:32	Sometimes used for Register Value, otherwise Reserved	Some fields piggyback on the upper 4 bytes of the Ack transfer – see Register Value below

Table 8: Ack bitfield

Register value

In a read register transaction, the register's contents are included in the upper 4 bytes of the Ack transfer.

5.4.4 Padding Transfers

Padding is used when only half of the full-duplex SPI is in use. The exact content of Tx Padding isn't strictly important, but choosing a bit-balanced padding such as all 0xAA's is a sensible approach.

Rx Padding is all 0xAA's for successfully-received data. If, for some reason, data is not being received correctly, Rx padding switches to 0xFF's, and the transaction should be aborted.

6 Address Map

Table 9 depicts the address map of the SV3, including both register and data block addresses. Access modes, data types and additional details are provided in subsequent sections.

Commands must be directed to the required SPI port. Unless noted otherwise, commands which target channels 1-16 must be sent to SPI port A. Commands targeting channels 17-32 must be sent via SPI port B.

Generally the 'Global Target Mask [0x0232]' is used to indicate which channels will be addressed. For example, a mask of 0x01 sent via SPI port A would be addressing Channel 1; while a mask of 0x01 sent via SPI port B would be addressing Channel 17.
In general all commands can be sent to either port A or B (A/B). Some hardware specific commands need to be sent to port A first, and then to port B (A->B). Note that there are a handful for commands which will only be accepted on port A (A).

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Module Control	**0x01**							
Global Status	0x02	R	Reg	A/B	0	Global Error Bitfield 0 means no errors (Section 8.6, Table 17)	-	Reflects the global module state; in case of an invalid command, out-of-range parameter, trigger timeout this register is set to the specific error code
Global Status Error Mask	0x04	RW	Reg	A/B	0xFFFFFFFF	Global Status Error Mask	-	Sets which Global Status error bits will be used in triggering the Global Status Error, and Status Transfer bits. By Default all errors are monitored. Logic 0 would mean a status bit is masked and therefore will not generate a Global status assertion. Errors will always be reported in the Global Status register.
Clear Global Status	0x06	W	Reg	A/B	-	32-bit Error Bitfield	-	Clears selected bits in the Global Status register [0x0102]. Logic 1 indicates the Error bit will be cleared.
Temperature	0x10	R	Data	A		Block of 2 32 bit temperatures.	-	Temperature is specified in 1 degree (C) increments. First temperature is measured on dye. Second is pcb temperature.
Delay	0x20	W	Reg	A/B	-	Number of microseconds	-	Generate delay, holding command processor busy.
Underflow	0x44	R	Reg	A	-	0 - No underflow, 1 - Underflow Detected	-	Indicates if bandwidth limitations are being exceeded. Bandwidth is affected by data rate and tx pattern composition.

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Fan Out Mode	0x46	RW	Reg	A	0	Value indicating fan out mode	-	Sets the fan out mode for the module. 0 = x1, 1 = x2, 2 = x4, 3 = x8. If the same data is being played on multiple channels the fan out mode can be used to mitigate bandwidth issues. X1 Mode Fan Out Disabled X2 Groups 1,3,5 and 7 of TX are repeated on next group X4 Groups 1 and 5 of TX are repeated on all groups of respective core X8 Group 1 is repeated on all groups of both cores
PTI Transfer Status	0xC0		R	Reg	-	Status bit indicating ready/busy	-	PTI Transfer status indicates whether the PTI is currently ready or busy. 0 - Busy, 1- Ready.
PTI Transfer Length	0xC4		RW	Reg	-	32-bit transfer length in bytes	-	Set the transfer length in bytes for the next PTI transfer.
PTI Transfer Start	0xC6		W	Reg	-	32-bit transfer direction	-	Writing to this register starts either a transfer read or transfer write. 0x01 - Read, 0x02 - Write
PTI Transfer Reset	0xC8		W	Reg	-	-	-	Resets and clears existing PTI transfer.
Configuration Manager	**0x02**							
HW, FW and SW IDs	0x02	R	Data	A/B	-	Three 8-byte identifiers, in the order [HW, FW, SW];	-	Identifies the type and version of the hardware, firmware and software
Module Revision	0x06	R	Reg	A/B	-	Module Revision Letter	-	Identifies the revision of the module.
Personality ID	0x08	R	Data	A/B	-	A 24-byte Personality ID String (Section 8.5)	-	Identifies the currently-running personality
SV3 Serial number	0x0E	R	Data	A/B	-	A 16 byte serial number.		Identifies the SV3 serial number in ASCII.

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Global Target Mask	0x32	RW	Reg	A/B	-	16-bit channel bitfield or 4-bit group bitfield	-	Mask indicates which channels or groups are affected by various SPI commands. When the mask is used as a channel bitfield, the 16 LSB of the register contents represent channels [15:0]. When the mask is used as a group bitfield, the 4 LSB of the register contents represent groups [3:0]. Group1 = Tx Channels 1,2,3,4 Group2 = Tx Channels 5,6,7,8 Group3 = Tx Channels 9,10,13,14 Group4 = Tx Channels 11,12,15,16. The groups and channels are similarly mapped on Core B.
Soft Reset	0xF0	W	Reg	A->B	-	-	-	Writing anything to this register causes all settings to revert to their power-on default values for the currently-running personality, except the following are unchanged: -Personality type -User pattern memory -Pattern sequence instructions -Data Rate -System reference clock source and frequency -Scripts -Calibration values (Note that jitter injection is disabled)
Reset to Default Settings	0xFE	W	Reg	A->B	-	-	-	Writing anything to this register causes all settings to revert to their power-on default values for the currently-running personality; does not change the personality – see Firmware Reconfiguration for that functionality

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Pattern Source	**0x03**							
Pattern Start Condition	0x04	RW	Dual	A/B	0	Dual Access (Section 7.5), 8-bit word per channel: Trigger ID (Section 8.3.1)	[Channel Bitfield] Set via 'Global Target Mask'	Assign start condition based on Trigger ID; this allows an 'immediate' start or no start on each channel. See 0x0307 for Pattern/Trig Start.
Pattern Stop	0x06	W	Reg	A/B	-	Channel Bitfield: Stop (1), or No Change (0)	-	Stops a previously-started pattern and reverts to 'Tx Pattern Disabled State [0x0334]'. If 'Tx Pattern Mode = Instruction Sequence [0x0336]' then all patterns will be disabled regardless of the channel bitfield.
Pattern Force/Trig Start	0x07	W	Reg	A	-	-	-	Starts all channels/patterns (1-32) that have been set to 'start immediate' via command 0x0304.
Instruction Sequencer Start Condition	0x08	RW	Reg	A/B	0	Trigger ID (Section 8.3.1)	-	Assign start condition based on Trigger ID; this allows an 'immediate' start or no start on the instruction sequencer. See 0x0307 for Pattern/Trig Start.
User Data Pattern Memory	0x10	RW	Data	A/B	Empty	A preformatted data pattern (Section 8.2.2)	User Pattern Slot ID (Section 8.2.3) Set via 'Global Target Mask'	Write loads a user data pattern into the User Pattern Slot identified by Target; reading retrieves a previously-written pattern; reading from an empty slot returns a block of zeros; Uploading to memory while sourcing user patterns or pattern sequencers is not recommended (Section 8.2.4); A user pattern can be cleared by writing a pattern of length zero (Section 8.2.5); Patterns can be read in chunks using the scrolling read feature (Section 8.2.6). Patterns can be written in chunks; and must be when their length exceeds 32KB (Section 8.2.7). Note that user pattern lengths will often be expanded to meet with internal requirements.

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
User Data Pattern Length	0x11	R	Reg	A/B	0	Block length, in bytes	User Pattern Slot ID Set via 'Global Target Mask'	Returns the length of the pattern loaded into the User Pattern Slot identified by Target; useful as a check, and when reading back user data patterns, as this is the length of data block to read
User Data Pattern Memory Allocate	0x14	W	Reg	A/B	-	Size of memory to be allocated in bytes. This size should correspond to a pattern length that will soon be written.	User Pattern Slot ID (Section 8.2.3) Set via 'Global Target Mask'	For patterns over 32 KB pattern memory must first be allocated. After a pattern slot has been allocate you can write the user pattern in chunks using the 'User Data Pattern Memory' [0x0310] command.
User Data Pattern Clear Memory	0x16	W	Reg	A/B	-	-	-	Write clears entire pattern memory and all user pattern slots.
Pattern Sequencer Memory	0x20	RW	Data	A/B	Empty	A Pattern Sequencer Program (Section 8.2.8)	Pattern Sequence Slot ID (Section 8.2.9) Set via 'Global Target Mask'	Write loads a pattern sequencer program into the Pattern Sequence Slot identified by Target; read retrieves a previously-written sequence; reading from an empty slot returns a block of zeros
Pattern Sequencer Program Length	0x21	R	Reg	A/B	0	Data block length, in bytes	Pattern Sequence Slot ID Set via 'Global Target Mask'	Returns the length of the Pattern Sequencer Program identified in Target; useful as a check, and when reading back the sequencer programs, as this is the length of data block to read
Tx PRBS Polynomial Order	0x30	RW	Reg	A/B	7	Polynomial order of the PRBS pattern generator (Section 8.2.10)	[Channel Bitfield] Set via 'Global Target Mask'	Sets the polynomial order of the tx PRBS generator on a per channel basis; available orders are 5, 7, 9, 11, 13, 15, 21, 23 and 31
Rx PRBS Polynomial Order	0x32	RW	Reg	A/B	7	Polynomial order of the rx PRBS pattern generator (Section 8.2.10)	[Channel Bitfield] Set via 'Global Target Mask'	Sets the polynomial order of the rx PRBS generator on a per channel basis; available orders are 5, 7, 9, 11, 13, 15, 21, 23 and 31

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Tx Pattern Disabled State	0x34	RW	Reg	A/B	0	Electrical Idle (0), All Ones (1), All Zeroes (2)	-	When a pattern is stopped this is the state which will be output. This state will also be output during a clock commit.
Tx Pattern Mode	0x36	RW	Reg	A/B	0	Basic (0), Instruction Sequencer (1)	-	Control how all patterns are controlled. In 'Basic (0)' mode the pattern generators are not controlled via the instruction sequencer. Once enabled they are active until explicitly stopped.
Instruction Sequencer Force Trigger	0x38	W	Reg	A/B	-	-	-	If the current instruction is waiting on a trigger this command will force the instruction sequencer to proceed to the next instruction.
Rx User Data Pattern Memory	0x60	RW	Data	A/B	Empty	A preformatted data pattern (Section 8.2.2)	User Pattern Slot ID (Section 8.2.3) Set via 'Global Target Mask'	Write loads a user data pattern into the Rx User Pattern Slot identified by Target; reading retrieves a previously-written pattern; reading from an empty slot returns a block of zeros; Uploading to memory while sourcing user patterns or pattern sequencers is not recommended (Section 8.2.4); A user pattern can be cleared by writing a pattern of length zero (Section 8.2.5); Patterns can be read in chunks using the scrolling read feature (Section 8.2.6). Patterns can be written in chunks; and must be when their length exceeds 32KB (Section 8.2.7). Note that user pattern lengths will often be expanded to meet with internal requirements.
Rx User Data Pattern Length	0x62	R	Reg	A/B	0	Block length, in bytes	User Pattern Slot ID Set via 'Global Target Mask'	Returns the length of the pattern loaded into the Rx User Pattern Slot identified by Target; useful as a check, and when reading back user data patterns, as this is the length of data block to read

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Rx User Data Pattern Memory Allocate	0x64	W	Reg	A/B	-	Size of memory to be allocated in bytes. This size should correspond to a pattern length that will soon be written.	User Pattern Slot ID (Section 8.2.3) Set via 'Global Target Mask'	For patterns over 32 KB pattern memory must first be allocated. After a pattern slot has been allocate you can write the user pattern in chunks using the 'Rx User Data Pattern Memory' [0x0360] command.
Rx User Data Pattern Clear Memory	0x66	W	Reg	A/B	-	-	-	Write clears entire rx pattern memory and all user pattern slots.
PTI Memory Write	0x70	W		Reg	-	Block length, in bytes	User Pattern Slot ID (Section 8.2.3) Set via 'Global Target Mask'	Write loads the most recent PTI transfer into the User Pattern Slot identified by Target. Uploading to memory while sourcing user patterns or pattern sequencers is not recommended (Section 8.2.4); Patterns can be read in chunks using the scrolling read feature (Section 8.2.6). Patterns can be written in chunks; and must be when their length exceeds 8MB (Section 8.2.7). Note that user pattern lengths will often be expanded to meet with internal requirements.
PTI Memory Read	0x71	W		Reg	-	Block length in bytes	User Pattern Slot ID (Section 8.2.3) Set via 'Global Target Mask'	Retrieves the User Pattern identified by the target in preparation for a 'PTI Start' read operation.
Rx Configuration	**0x04**							
Polarity Invert	0x02	RW	Dual	A/B	0	Dual Access (Section 7.5), single byte per channel: Inverted (1) / Normal (0)	[Channel Bitfield] Set via 'Global Target Mask'	Selective inversion of received bitstreams on a per-channel basis
Comparator Threshold Voltage	0x10	RW	Dual	A/B	0	Dual Access, 32-bit word per channel: Signed Voltage (μV)	[Channel Bitfield] Set via 'Global Target Mask'	Rx threshold voltage for each channel, expressed in microvolts.

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Comparator Threshold Offset	0x14	RW	Dual	A/B	0	Dual Access, 32-bit word per channel: Signed Voltage (uV)	[Channel Bitfield] Set via 'Global Target Mask'	Rx threshold offset data for each channel, expressed in microvolts. Applies to all data rates.
Comparator Threshold Offset Cal	0x15	W	Data	A	-	32-bit word per channel: Voltage (uV)	-	Allows programming of the threshold offset calibration data. Data block must be 128bytes. 32 Channels x 4 bytes per offset. Note that system must be power cycled after writing.
Comparator Threshold Linear Factor	0x16	RW	Dual	A/B	0	Dual Access; 32-bit word per channel: Fixed point notation <<28. ie int(float(2^28))	[Channel Bitfield] Set via 'Global Target Mask'	Rx threshold linear factor data for each channel. Applies to all data rates.
Comparator Threshold Linear Factor Cal	0x17	W	Data	A	-	32-bit word per channel: Fixed point notation <<28. ie int(float(2^28))	-	Allows programming of the threshold linear factor calibration data. Data block must be 128bytes. 32 Channels x 4 bytes per factor. Note that system must be power cycled after writing.
Comparator Threshold Square Factor	0x18	RW	Dual	A/B	0	Dual Access; 32-bit word per channel: Fixed point notation <<28. ie int(float(2^28))	[Channel Bitfield] Set via 'Global Target Mask'	Rx threshold square factor data for each channel. Applies to all data rates.
Comparator Threshold Square Factor Cal	0x19	W	Data	A	-	32-bit word per channel: Fixed point notation <<28. ie int(float(2^28))	-	Allows programming of the threshold square factor calibration data. Data block must be 128bytes. 32 Channels x 4 bytes per factor. Note that system must be power cycled after writing.
Comparator Threshold Cubic Factor	0x1A	RW	Dual	A/B	0	Dual Access; 32-bit word per channel: Fixed point notation <<28. ie int(float(2^28))	[Channel Bitfield] Set via 'Global Target Mask'	Rx threshold cubic factor data for each channel. Applies to all data rates.

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Comparator Threshold Cubic Factor Cal	0x1B	W	Data	A	-	32-bit word per channel: Fixed point notation <<28. ie int(float(2^28))	-	Allows programming of the threshold cubic factor calibration data. Data block must be 128bytes. 32 Channels x 4 bytes per factor. Note that system must be power cycled after writing.
Comparator Threshold Quad Factor	0x1C	RW	Dual	A/B	0	Dual Access; 32-bit word per channel: Fixed point notation <<44. ie int(float(2^44))	[Channel Bitfield] Set via 'Global Target Mask'	Rx threshold quad factor data for each channel. Applies to all data rates.
Comparator Threshold Quad Factor Cal	0x1D	W	Data	A	-	32-bit word per channel: Fixed point notation <<44. ie int(float(2^44))	-	Allows programming of the threshold quad factor calibration data. Data block must be 128bytes. 32 Channels x 4 bytes per factor. Note that system must be power cycled after writing.
Comparator Threshold Quint Factor	0x1E	RW	Dual	A/B	0	Dual Access; 32-bit word per channel: Fixed point notation <<44. ie int(float(2^44))	[Channel Bitfield] Set via 'Global Target Mask'	Rx threshold quint factor data for each channel. Applies to all data rates.
Comparator Threshold Quint Factor Cal	0x1F	W	Data	A	-	32-bit word per channel: Fixed point notation <<44. ie int(float(2^44))	-	Allows programming of the threshold quint factor calibration data. Data block must be 128bytes. 32 Channels x 4 bytes per factor. Note that system must be power cycled after writing.
CDR Data Path Mode	0x32	RW	Dual	A/B	0	Dual Access, single byte per channel: Data Path CDR (0) / Measurement Path Rx Reference (1)	[Channel Bitfield] Set via 'Global Target Mask'	Controls the CDR path mode on a per-channel basis, causing the Rx channel to clock either based on the CDR (clock is recovered from data), or based on the Rx Reference clock; the Rx reference clock is configured by the Rx Reference Clock Source register, below.
CDR LTD/LTR Lock Status	0x34	R	Reg	A/B	-	Channel Bitfield of Locked (1) or Not (0)	-	Reflects the status of the RX CDR PLL on each Rx channel. Does not apply to Digital CDR

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Rx Reference Clock Source	0x3C	RW	Reg	A/B	0	System Ref Clock (LTR) (0) / Extracted (LTD) (1)	[Channel Bitfield] Set via 'Global Target Mask'	The Rx Reference Clock is used when CDR Mode is set to "Reference"; "System Ref Clock" (0) will use the system reference clock, configurable by the System Reference Clock Source register; "Extracted" (1) will use either a forwarded or a recovered clock, as appropriate for the currently-loaded personality.
Rx Reference Core Clock Source	0x3F	RW	Reg	A/B	0	System Ref Clock (0) / Extracted (1)	[Channel Bitfield] Set via 'Global Target Mask'	Sets the system core clock source. Change takes effect immediately.
Rx Reference Phase Delay	0x40	RW	Dual	A/B	0	Dual Access, 32-bit word per channel: Signed Phase Delay (fs)	[Channel Bitfield] Set via 'Global Target Mask'	Controls the Rx phase by specifying a signed delay in fs. The delay range is data rate dependant.
Rx Reference Phase Delay Increment	0x44	W	Reg	A/B	-	32-bit word Signed Phase Delay (fs)	[Channel Bitfield] Set via 'Global Target Mask'	Modifies the current Rx phase by adding the (signed fs) value to the current phase setting. Affects the values read by commands 0x0440 above.
Rx Digital Reset	0x52	W	Reg	A/B	-	-	-	Performs a reset of the Rx FIFOs.
Rx Reference Phase Step	0x54	R	Reg	A/B	-	-	-	Reads the minimum step size / increment of the rx reference phase delay in fs.
Rx EQ DC Gain	0x56	RW	Dual	A/B	0	Dual Access (Section 7.5), single byte per channel.	[Channel Bitfield] Set via 'Global Target Mask'	Sets and reads DC gain level. Valid range is 0 – 4.
Rx EQ Control	0x58	RW	Dual	A/B	0	Dual Access (Section 7.5), single byte per channel.	[Channel Bitfield] Set via 'Global Target Mask'	Sets and reads Rx Equalization Control level. Valid range is 0 – 15.

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Bandwidth Control	0x60	RW	Reg	A/B	0	CDR bandwidth setting. High (0) / Medium (2) / Low (3)	[Channel Bitfield] Set via 'Global Target Mask'	Set the cdr bandwidth on a per-channel basis. Requires a clock commit to take effect.
Clock Monitor Control	0x62	RW	Reg	A/B	0	Disable (0) / Enable (1)	[Channel Bitfield] Set via 'Global Target Mask'	Enables the Clock Monitor. Requires a clock commit to take effect.
Tx Configuration	**0x05**							
Polarity Invert	0x02	RW	Dual	A/B	0	Dual Access (Section 7.5), single byte per channel: Inverted (1) / Normal (0)	[Channel Bitfield] Set via 'Global Target Mask'	Selective inversion of transmitted bitstreams on a per-channel basis
Data Pattern Provider	0x04	RW	Dual	A/B	K28.5	Dual Access, 32-bit word per channel: Pattern Selector (Section 8.2.1)	[Channel Bitfield] Set via 'Global Target Mask'	Assigns a pattern provider on a per-channel basis; Pattern Selector identifies both the provider (PRBS, Preset pattern, User Pattern, Pattern Sequencer, …) and specific source (Preset ID, User Slot ID, …)
Instruction Sequencer Pattern Provider	0x06	RW	Reg	A/B	-	Pattern Selector (Section 8.2.1)	-	Assigns a pattern provider to the Instruction Sequencer. Pattern selector identifies both the provider (User Pattern, Pattern Sequencer) and specific source (User Slot ID, Sequencer Slot). Note that only User Patterns and Pattern Sequencers are supported.
Output Amplitude	0x10	RW	Dual	A/B	800000uV	Dual Access, 32-bit word per channel: Voltage (¤V)	[Channel Bitfield] Set via 'Global Target Mask'	Peak-to-peak differential Tx amplitude, expressed in microvolts.
Tx Amplitude Offset Mode 1	0x14	RW	Dual	A/B	-	Dual Access, 32-bit word per channel: Voltage (uV)	[Channel Bitfield] Set via 'Global Target Mask'	Offset value applied to tx amplitude for 'Tx Mode 1'. Data is per channel and applies to all data rates. Output amplitude is updated immediately.

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Tx Amplitude Cal Offset Mode 1	0x15	W	Data	A	-	32-bit word per channel: Voltage (uV)	-	Allows programming of the offset calibration data for 'Tx Mode 1'. Data block must be 128bytes. 32 Channels x 4 bytes per offset. Note that system must be power cycled after writing.
Tx Amplitude Slope Mode 1	0x16	RW	Dual	A/B	-	Dual Access, 32-bit word per channel: Voltage (uV)	[Channel Bitfield] Set via 'Global Target Mask'	Slope value applied to tx amplitude for 'Tx Mode 1'. Data is per channel and applies to all data rates. Output amplitude is updated immediately.
Tx Amplitude Cal Slope Mode 1	0x17	W	Data	A	-	32-bit word per channel: Fixed point notation <<28. ie int(float(2^28))	-	Allows programming of the slope calibration data for 'Tx Mode 1'. Data block must be 128bytes. 32 Channels x 4 bytes per offset. Note that system must be power cycled after writing.
Tx Amplitude Offset Mode 2	0x18	RW	Dual	A/B	-	Dual Access, 32-bit word per channel: Voltage (uV)	[Channel Bitfield] Set via 'Global Target Mask'	Offset value applied to tx amplitude for 'Tx Mode 2'. Data is per channel and applies to all data rates. Output amplitude is updated immediately.
Tx Amplitude Cal Offset Mode 2	0x19	W	Data	A	-	32-bit word per channel: Voltage (uV)	-	Allows programming of the offset calibration data for 'Tx Mode 2'. Data block must be 128bytes. 32 Channels x 4 bytes per offset. Note that system must be power cycled after writing.
Tx Amplitude Slope Mode 2	0x1A	RW	Dual	A/B	-	Dual Access, 32-bit word per channel: Fixed point notation <<28. ie int(float(2^28))	[Channel Bitfield] Set via 'Global Target Mask'	Slope value applied to tx amplitude for 'Tx Mode 2'. Data is per channel and applies to all data rates. Output amplitude is updated immediately.
Tx Amplitude Cal Slope Mode 2	0x1B	W	Data	A	-	32-bit word per channel: Fixed point notation <<28. ie int(float(2^28))	-	Allows programming of the slope calibration data for 'Tx Mode 2'. Data block must be 128bytes. 32 Channels x 4 bytes per offset. Note that system must be power cycled after writing.

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Pre-Emphasis Tap Weights	0x36	RW	Dual	A/B	0	Dual Access, 32-bit word per channel: Three signed 1-byte tap weights packed into each word: [pre, post1, post2, 0]	[Channel Bitfield] Set via 'Global Target Mask'	Sets Tx pre-emphasis tap weights; each tap weight is an 8-bit signed integer.
Tx Reference Clock Source	0x3C	RW	Reg	A/B	0	System Ref Clock (0) / Extracted (1)	-	"System Ref Clock" (0) will use the system reference clock, configurable by the System Reference Clock Source register; "Extracted" (1) will use either a forwarded or a recovered clock, as appropriate for the currently loaded personality
Tx Fine Phase Delay	0x50	RW	Reg	A/B	0	Set phase delay in femto seconds.	[Group Bitfield] Set via 'Global Target Mask'	Allows fine phase adjustment to the 8 phase groups. There are 4 groups per port. Valid phase range is -500000 fs to 500000 fs.
Tx Fine Phase Cal Data	0x051	W	Data	A	-	Fine phase in femto seconds. 32-bit word per value.	-	Allows programming of the fine phase calibration data. Data block must be 544 bytes. 8 Phase Groups x 17 Cal Rates x 4bytes per skew. Note that system must be power cycled after writing.
Tx Coarse/Bit Delay	0x52	RW	Dual	A/B	0	Dual Access, 32-bit word per channel: Signed Bit delay (UI)	[Channel Bitfield] Set via 'Global Target Mask'	Allows injection of a coarse delay, in UI only), on a per-channel basis; the applied phase ranges from -1024 to 1023 UI.
Tx Coarse Delay Cal Data	0x53	W	Data	A	-	Coarse skew. 32-bit word per value.	-	Allows programming of the coarse skew calibration data. Data block must be 2176 bytes. 32 Channels x 17 Cal Rates x 4 bytes per skew. Note that system must be power cycled after writing.
Jitter Injection: Jitter Amplitude	0x60	RW	Dual	A/B	0	Dual Access, 32-bit word per channel: Jitter amplitude (ps)	[Group Bitfield] Set via 'Global Target Mask'	Requests a amplitude of jitter to inject on each group. Allowable values of jitter frequency/amplitude are TBD; changes take effect on next jitter start

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Jitter Injection: Jitter Frequency	0x62	RW	Dual	A/B	38,147	Dual Access, 32-bit word per channel: Jitter frequency (Hz)	[Group Bitfield] Set via 'Global Target Mask'	Controls the frequency of jitter injected on each group; allowable values of jitter frequency/ amplitude are TBD; changes take effect on next jitter start
Random Jitter Amplitude	0x64	RW	Dual	A/B		Dual Access, 32-bit word per channel: Jitter Amplitude (ps)	[Group Bitfield] Set via 'Global Target Mask'	Requests a amplitude of jitter to inject on each group. Allowable values of jitter frequency/amplitude are TBD; changes take effect on next jitter start
Jitter Start	0x66	RW	Dual	A/B	0	Dual Access, single byte per channel: Trigger ID (Section)	[Group Bitfield] Set via 'Global Target Mask'	Starts jitter with most recently provided parameters, based on Trigger ID; this allows an immediate start, a triggered start on pins, or no start, on each group
Jitter Stop	0x68	W	Reg	A/B	-	Group Bitfield: Stop (1), or no Change (0)	-	Stops a previously started jitter injection operation
Duty Cycle Distortion Offset	0x70	RW	Reg	A/B	4	Value of DCD offset	[Channel Bitfield] Set via 'Global Target Mask'	Controls the duty cycle distortion offset on a per channel basis. Valid values are 0 – 6.
Tx Pattern Status	0x82	R	Reg	A/B	0	-	-	Returns channel mask indicating the status of each channels pattern generator. 1 - Pattern Generator Active, 0 - Pattern Generator Stopped.
Tx Force Zero	0x84	W	Reg	A/B	0	Normal Operation (0), Zero Data (1)	[Channel Bitfield] Set via 'Global Target Mask'	0 specifies normal operation and 1 zeroes out the data on that channel. Note that this mask is applied after the fanout.
Jitter Compensation Offset	0x90	RW	Dual	A/B	-	Dual Access, 32-bit word per channel.	[Channel Bitfield] Set via 'Global Target Mask'	Offset value applied to jitter. Data is per channel and applies to all data rates.

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Jitter Compensation Offset Cal	0x91	W	Data	A	-	32-bit word per channel: Voltage (uV)	-	Allows programming of the jitter offset calibration data. Data block must be 32bytes. 8 Groups x 4 bytes per offset. Note that system must be power cycled after writing.
Jitter Compensation Linear Factor	0x92	RW	Dual	A/B	-	Dual Access, 32-bit word per channel: Fixed point notation <<28. ie int(float(2^28))	[Channel Bitfield] Set via 'Global Target Mask'	Linear factor value applied to jitter. Data is per channel and applies to all data rates.
Jitter Compensation Linear Factor Cal	0x93	W	Data	A	-	32-bit word per channel: Fixed point notation <<28. ie int(float(2^28))	-	Allows programming of the jitter linear factor calibration data. Data block must be 32bytes. 8 Groups x 4 bytes per offset. Note that system must be power cycled after writing.
Jitter Compensation Squared Factor	0x94	RW	Dual	A/B	-	Dual Access, 32-bit word per channel: Fixed point notation <<28. ie int(float(2^28))	[Channel Bitfield] Set via 'Global Target Mask'	Square factor value applied to jitter. Data is per channel and applies to all data rates.
Jitter Compensation Squared Factor Cal	0x95	W	Data	A	-	32-bit word per channel: Fixed point notation <<28. ie int(float(2^28))	-	Allows programming of the jitter square factor calibration data. Data block must be 32bytes. 8 Groups x 4 bytes per offset. Note that system must be power cycled after writing.
Jitter Compensation Cube Factor	0x96	RW	Dual	A/B	-	Dual Access, 32-bit word per channel: Fixed point notation <<28. ie int(float(2^28))	[Channel Bitfield] Set via 'Global Target Mask'	Cube factor value applied to jitter. Data is per channel and applies to all data rates.
Jitter Compensation Cube Factor Cal	0x97	W	Data	A	-	32-bit word per channel: Fixed point notation <<28. ie int(float(2^28))	-	Allows programming of the jitter cube factor calibration data. Data block must be 32bytes. 8 Groups x 4 bytes per offset. Note that system must be power cycled after writing.
Measurement And Test	**0x06**							

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
BERT Data Pattern Provider	0x04	RW	Dual	A/B	K28.5	Dual Access (Section 7.5), 32-bit word per channel: Pattern Selector (Section 8.3.1)	[Channel Bitfield] Set via 'Global Target Mask'	Assigns a BERT reference pattern provider on a per-channel basis; similar to the Pattern Source's Data Pattern Provider register
BERT Mask	0x06	RW	Dual	A/B	0	Dual Access, 32bits per channel.	[Channel Bitfield] Set via 'Global Target Mask'	Sets the BERT mask that will be applied when BERT compare mode is set to "BERT with generalized bit mask (1)"
BERT Compare Mode	0x08	RW	Dual	A/B	0	Dual Access, single byte per channel: Standard BERT (0) / BERT with generalized bit mask (1) / BERT with Non-Transition (2) / BERT with worst-case transition (3) / BERT with Single Edge transition (4) / BERT with Pre bit 0 (5) / BERT with Pre bit 1 (6)	[Channel Bitfield] Set via 'Global Target Mask'	Selects the comparison mode for the BERT on a per channel basis.
BERT Count Mode	0x10	RW	Dual	A/B	0	Dual Access, single byte per channel: Single-shot (0) / Continuous (1)	[Channel Bitfield] Set via 'Global Target Mask'	Selects single-shot or continuous count mode
BERT Single-Shot Duration	0x12	RW	Reg	A/B	10^6	Duration, in bits	-	Sets the duration of all single-shot BERTs; Truncated to a multiple of 32 bits;
BERT Sync Start	0x20	RW	Dual	A/B	0	Dual Access, single byte per channel: Trigger ID (Section 8.3.1)	[Channel Bitfield] Set via 'Global Target Mask'	Starts based on Trigger ID; BERT sync is a busy process, returning control to the module only once it's completed.
BERT Sync Result	0x22	R	Reg	A/B	0	Channel Bitfield: Success (1) or Failure (0)	-	Reflects the status of the most recently completed sync operation; each bit of the bitfield represents the sync status for a single channel, with Ch 1 as the LSB
BERT Sync Error Threshold	0x24	RW	Dual	A/B	3	Dual Access, 32 bits per channel: sync error limit	[Channel Bitfield] Set via 'Global Target Mask'	Maximum bit errors for successful BERT or Strobe sync. Valid range: 3 to 2^{32}-1

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
BERT Start	0x30	RW	Dual	A/B	0	Dual Access, single byte per channel: Trigger ID (Section 8.3.1)	[Channel Bitfield] Set via 'Global Target Mask'	Starts based on Trigger ID; BERT start is a busy process, returning control to the module only once it's completed.
BERT Error Count	0x32	R	Dual	A/B	0	Dual Access, 32-bit word per channel: Error count, in bits	[Channel Bitfield] Set via 'Global Target Mask'	Reads as the number of bit errors that occurred in the most recently-completed BERT; max reported error count is $2^{32} - 1$, and the count saturates at that level; only valid after an explicit stop for Continuous BERT, or completion of all bits in a Single-Shot BERT
BERT Received Bit Count On First Error	0x34	R	Data	A/B	0	Channel data block (Section 8.1.4), 64-bit word per channel: Bit count stored as a 64-bit integer	-	Reads as the number of bits that were successfully received before the first bit error was observed; cleared to 0 on start of a new BERT; only valid after an explicit stop for Continuous BERT, or completion of all bits in a Single-Shot BERT
BERT Received Bit Count	0x36	R	Data	A/B	0	Channel data block, 64-bit word per channel: Bit count stored as a 64-bit integer	-	Reads as the number of bits received in the most recently completed BERT; cleared to 0 on start of a new BERT; only valid after an explicit stop for Continuous BERT, or completion of all bits in a Single-Shot BERT
BERT Received Bit Count on Overflow	0x38	R	Data	A/B	0	Channel data block, 64-bit word per channel: Bit count stored as a 64-bit integer	-	Reads as the number of bits received before 2^{32} errors or more were observed in the most recently completed BERT, or as zero if no overflow occurred; cleared to 0 on start of a new BERT; only valid after an explicit stop for Continuous BERT, or completion of all bits in a Single-Shot BERT
BERT Stop	0x3A	W	Reg	A/B	-	Channel Bitfield: Stop (1) / No Change (0)	-	Stops a BERT running in Continuous mode

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Strobe Sync Start	0x40	RW	Dual	A/B	0	Dual Access, single byte per channel: Trigger ID(Section 8.3.1)	[Channel Bitfield] Set via 'Global Target Mask'	Starts based on Trigger ID; this allows an immediate start or no start, on each channel; this automatically places the module in Reference Clock mode: CDR is disabled; strobe sync is used to initiate controlled-phase measurements, such as bathtub plots and mask tests – see the programming examples in Section 9 for more on this; strobe sync is a busy process, returning control to the module only once it's completed
Strobe Sync Result	0x42	R	Reg	A/B	0	Channel Bitfield: Success (1) or Failure (0)	-	Reflects the status of the most recently completed strobe sync operation
Sync Offset	0x43	R	Dual	A/B	0	Dual Access, 32-bit word per channel: Sync offset, in bits	[Channel Bitfield] Set via 'Global Target Mask'	Reflects the offset, in bits, that was applied to the reference data pattern to yield a successful sync in the most recently completed Strobe Sync or Bert Sync operation; the higher the offset, the further forward in time the reference data pattern was advanced; values are undefined for channels that did not sync. For PRBS, returns the seed value.
BERT Scan Start	0x50	W	Data	A/B	-	Block of three two signed 32-bit integers, and one unsigned 32-bit integer in order: [Start (ps), End (ps), Step Size (ps)]	[Channel Bitfield] Set via 'Global Target Mask'	Writing to this address starts bathtub measurements on targeted channels in a true parallel manner; the direction of the sweep is determined by the start and end location, and not by the sign of the step size. The step size should always be positive. Measurements will be performed based on the single shot duration specified in 0x0612. BERT scans may not be performed while BERT is in continuous mode.

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
BERT Scan Results	0x52	R	Data	A/B	-	Block of ((Abs(EndIdx – StartIdx)/StepSize)+1) 32-bit integers: Error count, in bits. (Section 7.6.1)	[Channel Bitfield] Set via 'Global Target Mask'	Reads as an array containing the number of bit errors that occurred at each index setting specified in the most recently completed BERT Scan Start; max reported error count is 2^{32}-1, and the count saturates at that level;
Reset BERT Scan Read	0x54	W	Reg	A/B	-	-	[Channel Bitfield] Set via 'Global Target Mask'	Resets the read location of BERT Scan Results [0x0652] to read from the beginning of the data.
BERT Scan Length	0x56	R	Reg	A/B	0	Block length, in bytes	[Channel Bitfield] Set via 'Global Target Mask'	Returns the length of the BERT scan last performed on the channel identified by Target; useful as a check, and when reading back BERT scan results, as this is the length of data block to read
Analog Capture	0x60	W	Data	A/B	-	Block of two signed integers, in order: [Minimum Threshold(uV), Maximum Threshold (uV)]	[Channel Bitfield] Set via 'Global Target Mask'	Writing to this address starts an analog capture on targeted channels in a true parallel manner.
Analog Capture Results	0x62	R	Data	A/B	-	Block of 4096, 32-bit integers.	[Channel Bitfield] Set via 'Global Target Mask'	Read results of analog capture. Only a single channel result may be read at a time.
Reset Analog Capture Read	0x64	W	Reg	A/B	-	-	[Channel Bitfield] Set via 'Global Target Mask'	Resets the read location of Analog Capture Results [0x0662] to read from the beginning of the data.
Bert Bit Position	0x78	RW	Dual	A/B	-	Dual Access, 32-bit word per channel	[Channel Bitfield] Set via 'Global Target Mask'	Used to set bert bit position when 'BERT Compare Mode' is set to 'single edge'. Position must be less than the currently sourced rx pattern.

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
BERT VScan Start	0x80	W	Data	A/B	-	Block of two signed 32-bit integers, in order: [Start (¤V), End (¤V)]	[Channel Bitfield] Set via 'Global Target Mask'	Writing to this address starts vertical bathtub measurements on targeted channels in a true parallel manner; the direction of the sweep is determined by the start and end location. Measurements will be performed based on the single shot duration specified in 0x0612. Vertical BERT scans may not be performed while BERT is in continuous mode.
BERT VScan Results	0x82	R	Data	A/B	-	Block of ((Abs(End– Start)/10000)+1) 32-bit integers: Error count, in bits. (Section 7.6.1)	[Channel Bitfield] Set via 'Global Target Mask'	Reads as an array containing the number of bit errors that occurred at each index setting specified in the most recently completed Vertical BERT Scan Start; max reported error count is 2^{32}-1, and the count saturates at that level;
Reset VScan Read	0x84	W	Reg	A/B	-	-	[Channel Bitfield] Set via 'Global Target Mask'	Resets the read location of VScan Results [0x0682] to read from the beginning of the data.
BERT VScan Length	0x86	R	Reg	A/B	0	Block length, in bytes	[Channel Bitfield] Set via 'Global Target Mask'	Returns the length of the Vertical BERT scan last performed on the channel identified by Target; useful as a check, and when reading back BERT scan results, as this is the length of data block to read
Triggers And Flags	**0x07**							
User Flag LED	0x30	W	Reg	A	0	LED on (1) / LED off (0)	-	Enables and disable the user LED.
Flag State	0x32	RW	Reg	A	0	Hi-Z (0), Level High (1), Level Low (2)	[Flag Bitfield] Set via 'Global Target Mask'	Allows setting or reading of 5 flag pins. When writing, the following values are valid. Hi-Z(0), High(1), Low(2). A flag bitfield of 0x1F will set the output state on all pins. When operating as a flag input the flag state should first be set to Hi-Z. When reading a single bit must be set in the bitfield.

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Clock Configuration	**0x08**							
System Reference Clock Source	0x10	RW	Reg	A->B	Internal (1)	External (0), Internal (1)	-	Changes are applied on a write to the "Commit Clock Changes" register
System Reference Clock Frequency	0x12	RW	Reg	A->B	250000 kHz	Frequency, expressed in increments of 1kHz, and stored as a 32-bit integer	-	Changes the system reference clock frequency. Value only applies when System reference clock source is external. Changes are applied on a write to the "Commit Clock Changes" register.
Clock Out A Mode	0x14	RW	Reg	A->B	LVDS	Number specifying output format. CMOS(0), LVPECL(1), LVDS(2), HCSL(3), CML(4)	-	Changes the Clock A output format. Change only applies following a write to the "Commit Clock Changes" register.
Clock Out A Frequency	0x16	RW	Reg	A->B	100000 Khz	Frequency, expressed in increments of 1kHz, and stored as a 32 bit integer	-	Changes the Clock Out A frequency. Value only applies on a write to the "Commit Clock Changes" register.
Clock Out B Mode	0x18	RW	Reg	A->B	LVDS	Number specifying output format. CMOS(0), LVPECL(1), LVDS(2), HCSL(3), CML(4)		Changes the Clock B output format. Change only applies following a write to the "Commit Clock Changes" register.
Clock Out B Frequency	0x20	RW	Reg	A->B	100000 Khz	Frequency, expressed in increments of 1kHz, and stored as a 32 bit integer	-	Changes the Clock Out B frequency. Value only applies on a write to the "Commit Clock Changes" register.
SSC Enable	0x22	RW	Reg	A->B	Disabled	Disabled (0), Enabled (1)	-	Changes are applied on a write to the "Commit Clock Changes" register
SSC Frequency	0x24	RW	Reg	A->B	Disabled, 0	Frequency, expressed in increments of 1 Hz. Minimum frequency 31.5kHz. Maximum frequency is 63kHz.	-	Value only applies on a write to the "Commit Clock Changes" register.
SSC Down Spread	0x26	RW	Reg	A->B	Disabled, 0	Down spread percentage. Percentage expressed multiplied by 100. Ex, 1% would be written as 100.	-	Value only applies on a write to the "Commit Clock Changes" register.
Data Rate	0x30	RW	Data	A->B	8Ghz	Frequency, expressed in increments of 0.1 Hz, and stored as a 64-bit integer	-	Sets the data rate of the module; changes are applied when "Commit Clock Changes" register is written to

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Data Rate Calibrated Result	0x31	R	Data	A/B	8Ghz	Frequency, expressed in increments of 0.1 Hz, and stored as a 64-bit integer	-	Returns the actual data rate obtained as a result of the most recently committed Data Rate; At startup, reflects the default data rate; this value is updated on a write to the "Commit Clock Changes" register
Pattern Rate Ratio	0x40	R	Reg	A/B	1	Oversampling ratio	-	Pattern rate ratio specifies oversampling ratio set within the device. The value read defines the following: 1: 1 to 1 oversampling ratio (No oversampling) 2: 2 to 1 oversampling ratio. 4: 4 to 1 oversampling ratio. 8: 8 to 1 oversampling ratio. 16: 16 to 1 oversampling ratio. NOTE: Pattern Rate Ratio is updated upon "Commit Clock Changes".
Pattern Rx Rate Ratio	0x41	R	Reg	A/B	1	Rx Oversampling Ratio	-	Pattern rx rate ratio specifies oversampling ratio set within the device. The value read defines the following: 1: 1 to 1 oversampling ratio (No oversampling) 2: 2 to 1 oversampling ratio. 4: 4 to 1 oversampling ratio. 8: 8 to 1 oversampling ratio. 16: 16 to 1 oversampling ratio. NOTE: Pattern Rx Rate Ratio is updated upon "Commit Clock Changes".

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Commit Clock Changes	0x80	W	Reg	A->B	-	-	-	Changes to the above registers have no effect until this Commit register is written to. Side effects: • Many registers are returned to default status • All BERT counters and sync results cleared • Raw capture data los • Rx reference phase set to default
Speed Grade	0x94	R	R	A/B	-	Speed Grade	-	Returns module speed grade indicating the max supported data rate. 0=4.0Gbps. 1=8.0Gbps, 2=12.5Gbps, 3=12.8Gbps
Calibration	**0x09**							
Max Data Rate	0x26	R	Data	A/B	-	Max Frequency, expressed in increments of 0.1 Hz, and stored as a 64-bit integer	-	Returns the highest achievable data rate.
User Pattern Memory Size	0x50	R	Data	A/B	-	Size of pattern memory in bytes. Stored as 64-bit integer.	-	Returns the full size of the user pattern memory in bytes.
Rx User Pattern Memory Size	0x52	R	Data	A/B	-	Size of pattern memory in bytes. Stored as 64-bit integer.	-	Returns the full size of the rx user pattern memory in bytes.
Disable Tx Alignment Data	0x54	W	Reg	A/B	0	0 - Default, 1 - Disable Tx Alignment Data	-	Set to 1 to disable the tx alignment data. Tx alignment is updated immediately.
Tx Amplitude State	0x56	RW	Reg	A/B	0	State 0 (0), State 1 (1), State 2 (2), State 3 (3)	-	State 0: use the mode 1 flash data. Data in regs 0x0514, 0x0516, 0x0518, and 0x051A are ignored. State 1: disable the use of mode 1 flash data. Data in regs 0x0514 and 0x0516 are used. State 2: use the mode 2 flash data. Data in regs 0x0514, 0x0516, 0x0518, and 0x051A are ignored. State 3: disable the use of mode 2 flash data. Data in regs 0x0518 and 0x051A are used.

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Rx Threshold State	0x58	RW	Reg	A/B	0	State 0 (0), State 1 (1)	-	State 0: use the flash data for rx threshold compensation. Data written to 0x0414, 0x0416, 0x0418, 0x041A, 0x041C, and 0x041E are all ignored. State 1: disable the use of the rx threshold flash data. Data in 0x0414, 0x0416, 0x0418, 0x041A, 0x041C, and 0x041E are used.
Jitter State	0x5A	RW	Reg	A/B	0	State 0 (0), State 1 (1)	-	State 0: use the flash data for jitter amplitude compensation. Data written to 0x0690, 0x692, 0x0694, and 0x0696 are all ignored. State 1: disable the use of the jitter flash data. Data in 0x0690, 0x692, 0x0694, and 0x0696 are used.
Diagnostic Control	**0xFE**							
Enable Internal Serial Loopback	0x20	RW	Reg	A/B	0	Channel Bitfield: Enabled (1) / Disabled (0)	-	Enables internal serial loopback mode on a per-channel basis. When written, if any channel has CDR enabled the Rx alignment data will become invalid.
Raw Data Capture Target	0x22	RW	Reg	A/B	1	Channel Bitfield	-	The channel from which to capture data in a raw data capture; a single channel must be selected
Raw Data Capture Start Condition	0x24	RW	Reg	A/B	0	Disabled (0) / Start on pattern start (1) / Start on first BERT error (2)	[Channel Bitfield] Set via 'Global Target Mask'	The condition under which a raw data capture starts; The channel selected in Target determines which BERT or Patten Source to use as the start trigger; only a single channel may be selected as the trigger channel
Raw Data Capture Memory	0x26	R	Data	A/B	Undefined	Raw capture memory, length = 16,384 bits (2048 bytes)	-	Reads as the data captured in the most recently completed raw data capture operation; capture length is fixed at 16,384 bits (2048 bytes)

	Cmpt / Subaddr	R/W	Type	Port	Default	Data / Data Block Contents	Target	Description
Raw Data Capture Status	0x27	R	Reg	A/B	0 (stopped)	Stopped (0) / Running or pending (1)	-	Current state of raw capture logic. 1 means either running or waiting for pattern start to run
Raw Data Capture Stop Condition	0x28	RW	Reg	A/B	-	FF = immediate stop	-	Arms stop trigger when FE24 is in mode 2. Starts capture if not already running. Pattern triggers can be any channel.
Hardware Status	0x32	R	Reg	A/B	0	A hardware status code (0 is ok)	-	Bitfield indicating the status of various hardware elements throughout the DJ60HS (Table 17).
Hardware Status Mask	0x34	RW	Reg	A/B	0xFFFFFFFF	Hardware Status Error Mask	-	Sets which Hardware Status error bits will be used in triggering the Hardware Status Error pin. By default, all errors are monitored. Logic 0 would mean a status bit is masked and therefore will not generate a Hardware status pin assertion. Errors will always be reported in the Hardware and Global Status register.
Clear Hardware Status	0x36	W	Reg	A/B	-	32-bit Hardware Bitfield	-	Clears selected bits in the Hardware Status register [0xFE32]. Logic 1 indicates the Error bit will be cleared.

Table 9: SV3 Address Map – component addresses are bolded

7 Access Modes

7.1 Simple Registers

Values that fit in 32 bits and support only register reads/writes. e.g. PRBS Polynomial Order exposes a single word, ignoring the Target field and rejecting data block transactions.

7.2 Targeted Registers

A collection of several single-word elements that are associated with a group of Targets, such as channels, channel banks, or user pattern slots. The individual elements are addressed using the 'Global Target Mask' command. Data block transactions are rejected. e.g. User Data Pattern Length reads as the length of the user data pattern identified by the 'Global Target Mask' – there is one Length element per user pattern slot.

7.3 Simple Data Blocks

These blocks of data ignore the Target field / 'Global Target Mask' and reject register transactions. e.g. Module Temperature reads the temperature from several sensors and exposes them as a single data block.

7.4 Targeted Data Blocks

These are used where larger blocks of data are to be associated with a group. As with targeted registers, the elements are identified by the 'Global Target Mask'. Register accesses are rejected. e.g. Pattern Sequence Memory allows storage of Pattern Sequencer Programs in a set of slots, where the slot to address is identified using the Target field.

7.5 Dual-Access Addresses

These addresses accept both register and data transfers, allowing flexible access to groups of settings. They appear as either a register or a data block depending on the access mode:

7.5.1 Register Read of Dual-Access Address

This behaves identically to a targeted register. Performing a register read on Threshold Voltage, for example, with 'Global Target Mask' set to the channel bitfield 0x04 (third bit on), will read the threshold voltage for Channel 3. Target must identify a single element in this mode – for a Target which is a bitfield, there must be a single bit set.

7.5.2 Register Write of Dual-Access Address

A register write takes the content of the Data field and applies it to all the entries identified in the Target field. 'Global Target Mask' may address multiple elements in this mode. For example, performing a register write of 0 to Threshold Voltage, with 'Global Target Mask' set to the channel bitfield 0x1F, will write the threshold voltage 0 to Channels 1 through 5.

7.5.3 Data Read of Dual-Access Address

A data read always reads all of the entries in a group and returns them as a data block. The 'Global Target Mask' field is ignored. For example, performing a data read of Threshold Voltage will yield all 8 settings in a Channel Data Block (Section 8.1.4).

7.5.4 Data Write of Dual-Access Address

A data write accepts a data block containing entries for all the elements of a group, but only applies those addressed in the 'Global Target Mask' field. The 'Global Target Mask' field may identify multiple elements in this mode. For example, performing a data write to Threshold Voltage, with a data block containing the values 00,11,22,33,44,55,66,77, and with the 'Global Target Mask' bitfield set to 0x81, will write the threshold voltages 00 and 77 to Channels 1 and 8, respectively, ignoring the data block entries for Channels 2 through 7.

7.6 Multi-Transfer Data Block

These are used where larger blocks of data are to be associated with a group. As with targeted registers, the elements are identified by the Target field. Register accesses are rejected. The Target field of these registers only accept single index values.

7.6.1 Data Read of Multi-Transfer Data Block

Rather than reading an entire data block at once, which may not be possible depending on interface being used, data blocks can be read back in smaller chunks. This feature is performed automatically on specified registers. The length of

data transfers may be set to any length less than 128kBytes. Each successive read yields the next section of data. All read lengths should be a multiple of 4 bytes.

For example, if you wish to read a data block of 64kBytes from channel 1 the following will read back the block in four chunks of size 16 kBytes (0x4000 bytes):

1. Data read, Target = 0x01, DataField = 0x00004000, yields bytes 0x00000-0x03FFF
2. Data read, Target = 0x01, DataField = 0x00004000, yields bytes 0x04000-0x07FFF
3. Data read, Target = 0x01, DataField = 0x00004000, yields bytes 0x08000-0x0BFFF
4. Data read, Target = 0x01, DataField = 0x00004000, yields bytes 0x0C000-0x0FFFF

8 Data Types

8.1 Channel Addressing

8.1.1 Channel Bitfield

These simply map one bit to each channel, with the LSB always corresponding to the lowest-numbered channel. So Bit 0 corresponds to Channel 1, Bit 1 to Channel 2, and so on. This is used in reporting synchronization success, for example, where a value in Bit 0 means Channel 1 successfully synchronized.

When passed as the target of a register read operation, channel bitfields must have only one bit set, as only one entry can be read at a time in a register read.

8.1.2 Single-Ended Channel Bitfield

Like the channel bitfield, except this maps a bit to each of a set of single-ended channels, such that the first bit corresponds to Channel 1 negative, the second to Channel 1 positive, the third to Channel 2 negative, and so on.

8.1.3 Channel Bank Bitfield

Maps a single bit to each of the module's two channel banks. The first bit maps to the first bank, and the second bit maps to the second bank.

8.1.4 Channel Data Blocks

A data block which maps at least a single byte, and possibly a wider word, to each channel. The width of each entry in the data block is specified in the register map. The first entry in the data block always corresponds to Channel 1, the second to Channel 2, and so on. Channel Data Blocks are generally exposed as part of Dual-Type registers (Section 7.5).

8.1.5 Single-ended Channel Data Blocks

This is similar to a Channel Data Block, except there are 16 entries in a single-ended channel data block, one for each single-ended channel, in the same order as in the Single-Ended Channel Bitfield: Channel 1 negative, Channel 1 positive, Channel 2 negative, and so on….

8.1.6 *Channel Bank Data Blocks*

Similar to Channel Data Blocks, these registers map two entries to the module's two channel banks, where the first entry maps to the first Bank, and the second to the second. The width and interpretation of the entries is provided in the address map.

8.2 Pattern Configuration

A set of pattern providers is available to drive the 32 Tx data patterns and the 32 BERT reference patterns, as shown in. Providers are the green objects to the left, and are identified by a Pattern Provider ID. Within each provider lies the possibility of several *Sources*, such as each of the PRBS generator, or the user slots in a user pattern provider. Details are provided below.

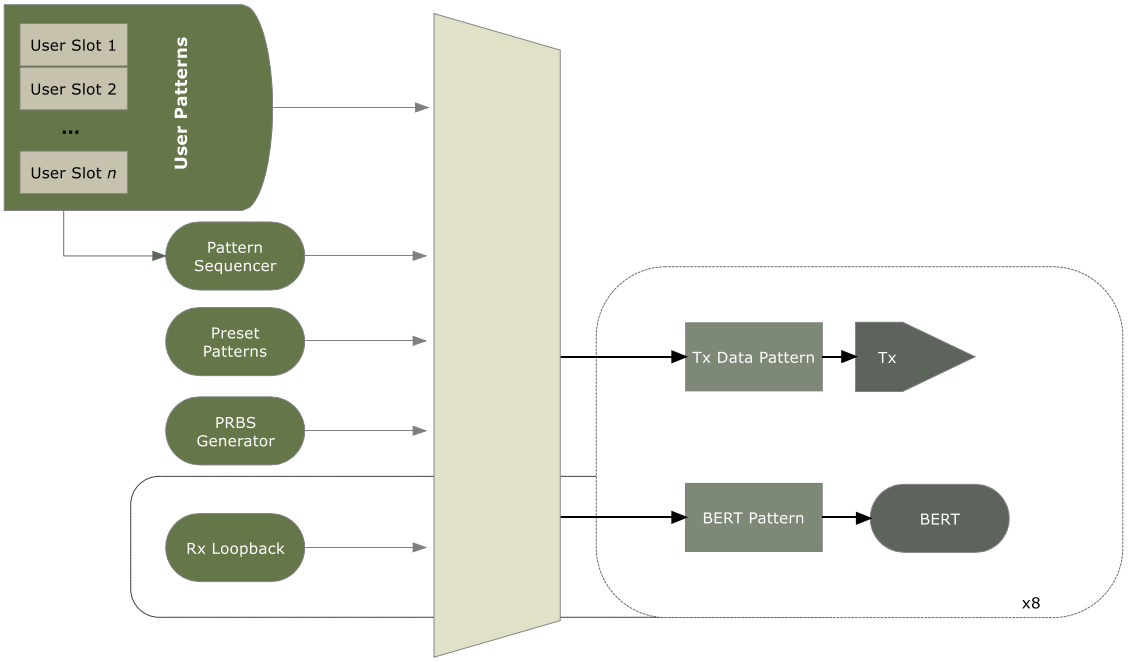


Figure 3: The Pattern Sources and Consumers Model

8.2.1 Pattern Selector

Pattern selectors are used to identify both the provider (e.g. PRBS) and specific source (e.g. User Pattern Slot ID). The 4-byte Selector is split up as follows:

Field	Provider ID	Source ID	Total
Byte cnt	2	2	**4 bytes**
Byte addr	3:2	1:0	

Table 10: Pattern Selector Fields

Each field is described below. Note that there is presently a single PRBS generator available globally, so the Source ID for PRBS is always written as zero. The Pattern Selector for a PRBS would be 0x0001 0000. And the User Pattern in Slot #3: 0x0004 0003.

Provider ID	Meaning
1	PRBS
2	Rx Loopback
3	Pattern sequencer
4	User Pattern
Others	Reserved

Table 11: Pattern Provider ID Map

Provider	Source ID	Meaning
Sequencer	0-15	Pattern Sequence Slot IDs
User	0-127	User Pattern Slot IDs
Others	Others	Reserved

Table 12: Source ID Map

8.2.2 Preformatted Data Patterns

User data patterns are transmitted to the module as a raw block of packed binary data. A 2048-bit data pattern is packed into 2048/8 bytes, for example, with the first bit packed into the MSB of the first byte of the block, the next bit to follow, and so forth for 256 bytes.

8.2.3 User Pattern Slot ID

User patterns are uploaded into slots, and later recalled from those slots, using unique Pattern Slot IDs. There are 1024 slots available, and each slot has its own ID, in the range 0 to 1023. User Pattern Slot IDs are used when uploading to, or downloading from, the User Data Pattern Memory. Similarly, the Source ID field of a Pattern Selector accepts a User Pattern Slot ID, when the Pattern Provider is set to User Pattern. Note that User Patterns can only be a multiple of 8 bits.

8.2.4 Writing to User Pattern Memory

User pattern memory is automatically managed so that a total of up to 1 GB of data pattern can be stored across the 1024 slots. The downside of this memory management is that uploading a pattern to memory can cause already-uploaded patterns to be moved to keep them contiguous. If a pattern gets moved while it is being sourced, either directly as a User

Pattern or through a Pattern Sequencer, then the output pattern will be corrupted until it is restarted. The same applies to an overwritten pattern: if you source the pattern in Slot 1, then overwrite the contents of that slot, the output pattern will be incorrect until you restart it.

The recommended use of user pattern memory is therefore to stop sourcing User Patterns and Pattern Sequencers before uploading to pattern memory.

8.2.5 Clearing a user pattern

Writing a pattern of length zero to a slot will empty it, freeing any memory that was previously occupied by that slot. When writing a pattern of length zero, a data transfer is still required – even though a length of zero is indicated, any length of transfer is accepted by the command processor without raising errors, and the contents of that transfer are simply ignored.

8.2.6 Scrolling Readback from User Pattern Memory

Rather than reading an entire user pattern at once, user patterns can be read back in smaller chunks. This feature is enabled by first performing a normal data read, with the data size set to any length desired, then repeating that read command with the uppermost bit of the Target bitfield set. Each read yields the next section of the user data pattern, until the entire pattern is read back. All reads must be a multiple of 512 bits (64 bytes) long.

For example, if a user pattern of length 128 kBytes (1MBit) has been loaded into Slot 2, the following will read back the pattern in four chunks of size 32 kBytes (0x8000 bytes):

1. Data read, Target = 0x01, Length = 0x8000, yields bytes 0x00000-0x07FFF
2. Data read, Target = 0x81, Length = 0x8000, yields bytes 0x08000-0x0FFFF
3. Data read, Target = 0x81, Length = 0x8000, yields bytes 0x10000-0x17FFF
4. Data read, Target = 0x81, Length = 0x8000, yields bytes 0x18000-0x1FFFF

Note that reading with the uppermost bit of Target cleared will always reset to reading from byte 0 of the pattern.

8.2.7 Writing User Pattern Memory in Blocks

Patterns that exceed 32KB need to be written as a series of blocks. First a memory space must be allocated for the entire pattern length [0x0314] and then the data pattern must be written to this allocated space [0x0310]. This feature is enabled by performing the user pattern memory command with the uppermost bit of the Global Target Mask [0x0232] set. Each write then stores the next section of the user data pattern, until the entire pattern is written. Note, user patterns can only be written in blocks following memory allocation [0x0314]. All pattern lengths must be a multiple of 4 bytes.

For example, if a user pattern of length 64 kBytes is to be written the following commands will perform the write to Slot 2 in 4 blocks of 16 kB.

1. Register Write, Set user slot via Global Target. Command = 0x0232, Target = 0x00, Data = 0x00000002
2. Register Write, Allocate Pattern Space. Command = 0x0314, Target = 0x00, Data = 0x10000
3. Register Write. Set repeated write bit and user slot via Global Target. Command = 0x0232, Target = 0x00, Data = 0x80000002
4. Data Write, Command = 0x0310, Target=0x00, Data = 0x4000, Pattern Bytes
5. Data Write, Command = 0x0310, Target=0x00, Data = 0x4000, Pattern Bytes
6. Data Write, Command = 0x0310, Target=0x00, Data = 0x4000, Pattern Bytes
7. Data Write, Command = 0x0310, Target=0x00, Data = 0x4000, Pattern Bytes

8.2.8 Pattern sequencer programs

Pattern sequencer programs are expressed as a block of up to twelve 4-byte *pattern commands* and a four-byte external loop count. The format is summarized in Table 13.

Byte Range	Content	Meaning
3:0	Pattern Command 0	Entry slot - not affected by external loop count.
7:4	Pattern Command1	
...	...	
47:44	Pattern Command 11	Exit slot - not affected by external loop count.
<51:48>	External Loop Count	Indicates number of times program will loop.

Table 13: Pattern Sequencer Format

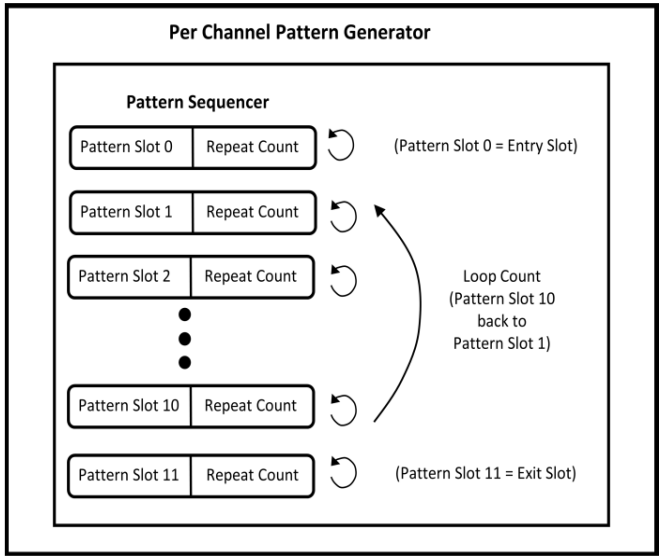
Each 4-byte pattern command follows this format:

Field	Repetition count	Padding	User Pattern Slot ID	
Byte cnt	2	1	1	4 bytes
Byte addr	2	1	0	

Table 14: The 2-byte pattern command format

Repetition count is 2 bytes, and so valid repetition counts are from 0 to 65535. There are two special cases of values. Writing a repetition count of 0 causes the pattern slot to be skipped, writing a value of 65535 will cause the pattern slot to repeat infinitely. Any other count causes the pattern to run the requested number of times – i.e. a value of 1 causes the pattern to run exactly once, 2 causes it to run twice, and so on.

Valid External Loop Counts are from 1 to 65535. Writing the maximum value of 65535 in the loop count will cause this outer loop to repeat infinitely. Please note that the external loop count only affects Pattern Slot 1 to Slot 10. Pattern slots 0 and 11 are the entry and exit slot and do not loop based on the external loop count.



Block diagram of the Pattern Sequencers within each Pattern Generator.

8.2.9 Pattern Sequence Slot ID

Similar to the scheme with User Patterns, Pattern Sequences are uploaded into slots with unique IDs. These unique IDs are then used to identify the pattern sequences when uploading, downloading, and as the Source ID field of a Pattern Selector, when the Pattern Provider is set to Pattern Sequencer. There are 16 Pattern Sequence slots, numbered from 0 to 15.

8.2.10 PRBS Generator

There is a onboard PRBS generator for each channel which can be configured via the PRBS Polynomial Order registers. The available orders are 5, 7, 9, 11, 13, 15, 21, 23 and 31.

8.3 Triggers and Flags

8.3.1 Trigger ID

Each triggerable component accepts as a parameter a Trigger ID. BERTs, pattern sources and jitter injection, for example, all start based on the Trigger ID written to their Start registers.

Table 15 summarizes the available Trigger IDs: "no start" (0x00), "start now" (0xFF), and "start on trigger event going active" (others). Note that "no start" is not the same as "stop" – processes that can be stopped have separate stop registers.

Trigger ID	Meaning
0x00	No start
0x01 - 0x02	Trigger Pins #1 to #2
0xFF	Start immediately

Table 15: Trigger IDs

8.4 Fixed point formats

In some registers, fixed point notation is used to represent fractional numbers. The format is specified in three parts: signed/unsigned, bits dedicated to the whole number, and bits dedicated to the fractional number. The latter two are shown as <whole>:<fractional>.

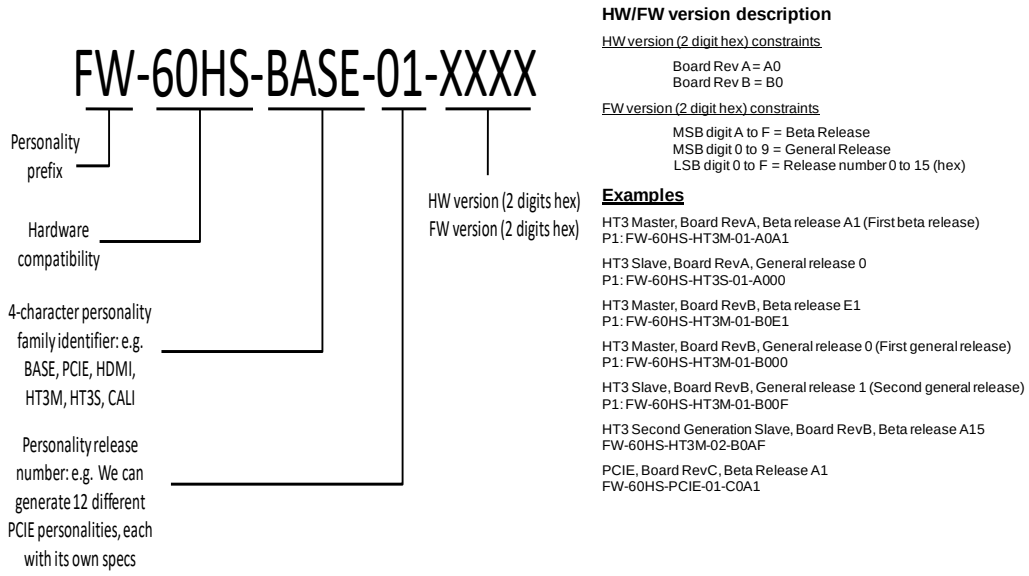
For example, an unsigned fixed point 8:24 format means an unsigned number is represented with 8 whole number bits and 24 fractional bits. In this format, the number 0x02 8000000 decodes to 2 ½, and the number 0xFF 400000 to 255 ¼. The minimum fraction that can be expressed in this format is 2^{-24}.

8.5 ID Strings

8.5.1 Personality ID String

This is a NULL-terminated (and NULL-padded) string with a length of 24 bytes. It contains the personality part number stripped of its dashes and padded, at the end, with NULL characters. The personality part numbering scheme is documented elsewhere.

Below is the String numbering definition with examples:



From the stated example, FW-60HS-IESP-01-B000 would appear in the Personality ID String without dashes and with null-padding at the end, as in:

'F'	'W'	'6'	'0'	'H'	'S'	'I'	'E'	'S'	'P'	'0'	'1'	'B'	'0'	'0'	'0'	0	0	0	0	0	0	0	0

8.6 Error Codes

The various error codes that appear throughout the interface are summarized in the tables below.

Global Error Status bit	Meaning
0	Generic invalid command
1	Generic parameter out of range
2	Voltage parameter out of range
3	Transfer too short: too few bits were included in the most recent SPI transfer
4	Hardware issue: indicates one of a number of hardware-related issues, including overtemperature, undervoltage, and loss of clock lock
6	Trigger Timeout
7	Digital CDR lock failed
31	Licence Error

Table 16: Global Error Codes

Hardware status bit	Meaning
0	Clock Reconfiguration Complete
1	System Clock Locked
2	Secondary Clock Locked
3	Temperature Ok
6	Memory Calibration Complete
7	Memory Initialization Complete
8	Memory Locked
Others	Reserved

Table 17: Hardware Status Bits – in all cases, a 0 means OK, and a 1 indicates an error

9 Programming Examples

9.1 Pattern Source

A PRBS-7 pattern source on all 32 channels with parameterized pre-emphasis. Prerequisite is that the module be in its default state.

Operation	Transaction / comments	Addr	Target	Data / Data Block
Set Global Target Mask	Write reg: Configuration Manager: Global Target Mask. Written to SPI Port A	0x2 32	-	0xFFFFF Select channels 1-16 for modification.
Set Global Target Mask	Write reg: Configuration Manager: Global Target Mask. Written to SPI Port B	0x2 32	-	0xFFFFF Select channels 17-32 for modification.
Set output amplitude to 1200 mV	Write dual as reg: Tx Configuration: Output Amplitude. Written to SPI Port A	0x05 10	-	0x00124F80 (Amplitudes expressed in ▫V, channels 1-16)
Set output amplitude to 1200 mV	Write dual as reg: Tx Configuration: Output Amplitude. Written to SPI Port B	0x05 10	-	0x00124F80 (Amplitudes expressed in ▫V, channels 17-32)
Set pre-emphasis tap weights	Write dual as reg: Tx Configuration: Pre-emphasis Settings. Written to SPI Port A	0x05 36	-	0x01020300 Sets pre-cursor to 1, post-cursor #1 to 2, and post-cursor #2 to 3, , channels 1-16
Set pre-emphasis tap weights	Write dual as reg: Tx Configuration: Pre-emphasis Settings. Written to SPI Port B	0x05 36	-	0x01020300 Sets pre-cursor to 1, post-cursor #1 to 2, and post-cursor #2 to 3, , channels 17-32
Set Tx data pattern	Write dual as reg: Tx Configuration: Tx Data Pattern Provider. Written to SPI Port A	0x05 04	-	0x0001 0000 (Pattern Selector for PRBS, channels 1-16)

Set Tx data pattern	Write dual as reg: Tx Configuration: Tx Data Pattern Provider. Written to SPI Port B	0x05 04	-	0x0001 0000 (Pattern Selector for PRBS, channels 17-32)
Set PRBS Order to 7	Write reg: Pattern Source: PRBS Polynomial Order. Written to SPI Port A	0x03 30	-	7 (PRBS Polynomial Order)
Set PRBS Order to 7	Write reg: Pattern Source: PRBS Polynomial Order. Written to SPI Port B	0x03 30	-	7 (PRBS Polynomial Order)
Set pattern to run immediately	Write dual as reg: Pattern Source: Pattern Start. Written to SPI Port A	0x03 04	-	0xFF (Start immediately, all channels)
Set pattern to run immediately	Write dual as reg: Pattern Source: Pattern Start. Written to SPI Port B	0x03 04	-	0xFF (Start immediately, all channels)

9.2 Data Rate Change

Demonstrates changing the operating rate of the SV3D. In this example we use polling to determine when the clock change is complete.

Operation	Transaction / comments	Addr	Target	Data
Set Data Rate	Write data: Clock Configuration: Data Rate. Written to SPI Port A	0x08 30	-	0x00000003 7e11d600 (set data rate to 1500.0 Mbps)
Set Data Rate	Write data: Clock Configuration: Data Rate. Written to SPI Port B	0x08 30	-	0x00000003 7e11d600 (set data rate to 1500.0 Mbps)
Poll on Completion	Read reg: Module Control: Global Status. Written to SPI Port A Command is repeated until expected status is returned. A response of '05 05 05 05 05 05 05 05' indicates the system is busy. '07 07 07 07 07 07 07 07' indicates it is available.	0x01 02	-	0x00000000
Poll on Completion	Read reg: Module Control: Global Status. Written to SPI Port B Command is repeated until expected status is returned.	0x01 02	-	0x00000000