

## PRBS (according ITU-T O.150) and Bit-Sequence Tester : VHDL-Modules

---

A PRBS (Pseudo Random Binary Sequence) is a binary PN (Pseudo-Noise) signal. The sequence of binary 1's and 0's exhibits certain randomness and autocorrelation properties.

Bit-sequences like PRBS are used for testing transmission lines and transmission equipment because of their randomness properties.

Simple bit-sequences are used to test the DC compatibility of transmission lines and transmission equipment.

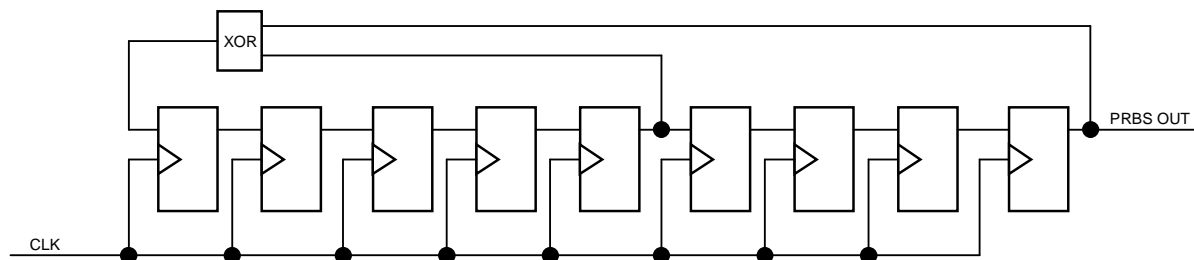
The different types of PRBS and the suggested data-rates for the different PRBS types are described in the ITU-T standards O.150, O.151, O.152 and O.153.

PRBS type	Standard	Suggested Datarate [kbit/sec]	Feedback tap
$2^9 - 1$	ITU-T O.150 / O.153	up to 14.4	$5^{\text{th}} + 9^{\text{th}}$
$2^{11} - 1$	ITU-T O.150 / O.152 / O.153	64, $n \cdot 64$ ( $n=1..31$ ), 48 to 168	$9^{\text{th}} + 11^{\text{th}}$
$2^{15} - 1$	ITU-T O.150 / O.151	1544, 2048, 6312, 8448, 32064, 44736	$14^{\text{th}} + 15^{\text{th}}$
$2^{20} - 1$	ITU-T O.150 / O.151	1544, 6312, 32064, 44736	$17^{\text{th}} + 20^{\text{th}}$
$2^{20} - 1$	ITU-T O.150 / O.153	up to 72	$3^{\text{rd}} + 20^{\text{th}}$ (note 1)
$2^{23} - 1$	ITU-T O.150 / O.151	34368, 44736, 139264	$18^{\text{th}} + 23^{\text{rd}}$
$2^{29} - 1$	ITU-T O.150		$27^{\text{th}} + 29^{\text{th}}$
$2^{31} - 1$	ITU-T O.150		$28^{\text{th}} + 31^{\text{st}}$

Note 1 = an output bit is forced to be a ONE whenever the previous 14 bits are all ZERO.

PRBS bit-pattern are generated in a linear feed-back shift-register. This is a shift-register with a xored-feedback of the output-values of specific flip-flops to the input of the first flip-flop.

Example : PRBS-Generation of the sequence  $2^9 - 1$  :



At start time all flip-flops are set to '1'.

## PRBS (according ITU-T O.150) and Bit-Sequence Tester : VHDL-Modules

---

The PRBS and bit-sequence tester consists of two modules :

**PRBS and bit-sequence generator**  
**PRBS and bit-sequence receiver**

Both modules are plain VHDL based without any special units (e.g. RAM-blocks from a FPGA vendor).

The modules are synchronous designs with clock and clock-enable inputs. A control port enables the possibility to select different PRBS sequences or bit pattern (See page 4). The generator is able to transmit sequences with selectable error rates from  $10^{-1}$  to  $10^{-12}$ . It has a transmit single bit error input too.

The receiver has signaling outputs for synchronization state, synchronization loss, bit error and clock error.

**Synchronization state :**

The output goes to high when :

- in PRBS mode  $2 \cdot X$  ( $X = \text{PRBS shift register length [9..31]}$ ) error free bits are received
- in bit pattern mode 20 error free bits are received

The output goes to low if the error rate exceeds 0.2. To detect this level, the bit errors during the last 128 received bits are memorized. If there are more than 25 errors within these 128 bits, the output goes low.

**Synchronization loss :**

When the synchronization upon the received bit sequence is lost, the output goes high for one clock period (plus clock enable).

**Bit error :**

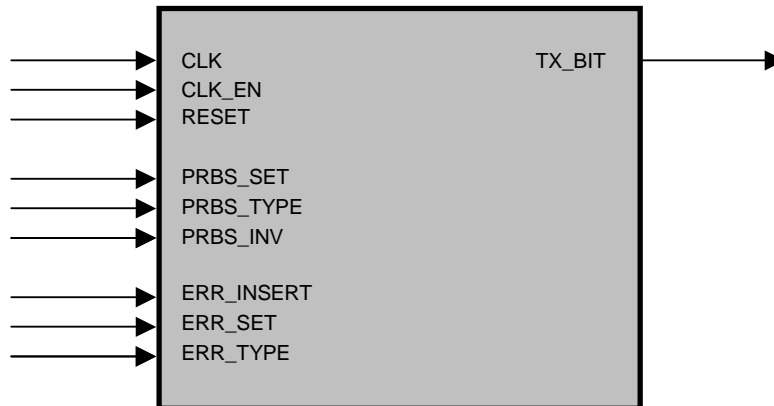
This output goes high, when the receiver is synchronized and a bit error in the received bit sequence is detected. This signaling output has a delay of 128 bit times, because massive bit errors who causes a synchronization loss must not be reported as bit errors.

**Clock error :**

This output goes high, when the receiver is synchronized and a clock error (bit slip : bit lost or bit inserted) in the received bit sequence is detected.

*The clock error functionality is not implemented at the moment.*

## PRBS-Generator-Module



## PRBS-Generator : VHDL-Entity

```
entity PRBS_TX_SER is
  port (
    CLK          : in  std_logic;           -- synchron clock
    RESET        : in  std_logic;           -- asynchron reset
    CLK_EN       : in  std_logic;           -- clock enable
    PRBS_SET     : in  std_logic;           -- set new PRBS / bit pattern
    PRBS_TYPE    : in  std_logic_vector (3 downto 0); -- type of PRBS / bit pattern
    PRBS_INV     : in  std_logic;           -- invert PRBS pattern
    ERR_INSERT   : in  std_logic;           -- manual error insert
    ERR_SET      : in  std_logic;           -- set new error type
    ERR_TYPE     : in  std_logic_vector (3 downto 0); -- error type
    TX_BIT       : out std_logic           -- tx serial output
  );
end PRBS_TX_SER;
```

## **PRBS-Generator : Interface Description**

### **CLK**

Base clock for the PRBS generator. The whole PRBS logic of the generator works with this clock.

### **RESET**

Asynchronous reset for the whole internal logic of the PRBS generator.

### **CLK\_EN**

Clock enable signal for the PRBS generator.

### **PRBS\_SET** (synchronous to CLK)

When PRBS\_SET is high the generator reads the values on the inputs PRBS\_TYPE and PRBY\_INV.

### **PRBS\_TYPE** (synchronous to CLK)

Input vector for selection of the PRBS sequence or of the bit pattern. This vector is read by the generator when PRBS\_SET is high.

PRBS_TYPE vector	PRBS sequence / bit pattern	Note
0 0 0 0	$2^9 - 1$	DC free
0 0 0 1	$2^{11} - 1$	DC free
0 0 1 0	$2^{15} - 1$	DC free
0 0 1 1	$2^{20} - 1$	DC free
0 1 0 0	$2^{20} - 1$	DC free
0 1 0 1	$2^{23} - 1$	DC free
0 1 1 0	$2^{29} - 1$	DC free
0 1 1 1	$2^{31} - 1$	DC free
1 0 0 0	all '0' : "00000000"	DC only
1 0 0 1	all '1' : "11111111"	DC only
1 0 1 0	alternating '0' and '1' : "01010101"	DC free
1 0 1 1	alternating '00' and '11' : "00110011"	DC free
1 1 0 0	one '0' and seven '1' : "01111111"	With DC component
1 1 0 1	one '1' and seven '0' : "10000000"	With DC component
1 1 1 0	two '0' and six '1' : "01110111"	With DC component
1 1 1 1	two '1' and six '0' : "10001000"	With DC component

### **PRBS\_INV** (synchronous to CLK)

Input signal for selection of the polarity of the PRBS sequence. A high signal on this input inverts the PRBS bit sequence on TX\_BIT. This signal is read by the generator when PRBS\_SET is high. The polarity of the bit pattern is not affected by PRBS\_INV.

## PRBS (according ITU-T O.150) and Bit-Sequence Tester : VHDL-Modules

---

### ERR\_INSERT (synchronous to CLK)

Input signal for inserting an error into the actual generated PRBS bit or pattern bit. A rising edge on this input signal (mentioned with CLK) generated exact one defective bit.

### ERR\_SET (synchronous to CLK)

When ERR\_SET is high the generator reads the value on the input ERR\_TYPE.

### ERR\_TYPE (synchronous to CLK)

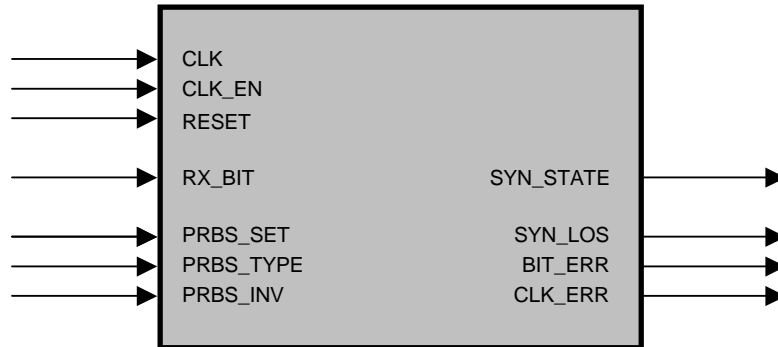
Input vector for selection of transmitted error rate (if desired). This vector is read by the generator when ERR\_SET is high.

ERR_TYPE vector	Transmitted error rate	Error distance at 100 Mbit/sec
0 0 0 0	0	-
0 0 0 1	$10^{-1}$	100 ns
0 0 1 0	$10^{-2}$	1 us
0 0 1 1	$10^{-3}$	10 us
0 1 0 0	$10^{-4}$	100 us
0 1 0 1	$10^{-5}$	1 ms
0 1 1 0	$10^{-6}$	10 ms
0 1 1 1	$10^{-7}$	100 ms
1 0 0 0	$10^{-8}$	1 sec
1 0 0 1	$10^{-9}$	10 sec
1 0 1 0	$10^{-10}$	100 sec
1 0 1 1	$10^{-11}$	17 min
1 1 0 0	$10^{-12}$	167 min = 2,8 h
1 1 0 1	variable : $10^{-3}$ to $10^{-6}$	(not implemented)
1 1 1 0	variable : $10^{-3}$ to $10^{-12}$	(not implemented)
1 1 1 1	variable : $10^{-9}$ to $10^{-12}$	(not implemented)

### TX\_BIT (synchronous to CLK)

Output of the PRBS generator for the PRBS sequence alternatively for the bit pattern.

## PRBS-Receiver-Module



## PRBS-Receiver : VHDL-Entity

```
entity PRBS_RX_SER is
  port (
    CLK      : in  std_logic;           -- synchron clock
    RESET    : in  std_logic;           -- asynchron reset
    CLK_EN   : in  std_logic;           -- clock enable
    RX_BIT   : in  std_logic;           -- rx serial input
    PRBS_SET : in  std_logic;           -- set new PRBS / bit pattern
    PRBS_TYPE : in  std_logic_vector (3 downto 0); -- type of PRBS / bit pattern
    PRBS_INV : in  std_logic;           -- invert PRBS pattern
    SYN_STATE : out std_logic;          -- synchronisation state output
    SYN_LOS  : out std_logic;          -- sync loss signaling output
    BIT_ERR  : out std_logic;          -- biterror signaling output
    CLK_ERR  : out std_logic;          -- clockerror signaling output
  );
end PRBS_RX_SER;
```

## **PRBS-Receiver : Interface Description**

### **CLK**

Base clock for the PRBS receiver. The whole PRBS logic of the receiver works with this clock.

### **RESET**

Asynchronous reset for the whole internal logic of the PRBS receiver.

### **CLK\_EN**

Clock enable signal for the PRBS receiver.

### **RX\_BIT** (synchronous to CLK)

Input of the receiver for the received PRBS sequence or for the received bit pattern.

### **PRBS\_SET** (synchronous to CLK)

When PRBS\_SET is high the receiver reads the values on the inputs PRBS\_TYPE and PRBY\_INV.

### **PRBS\_TYPE** (synchronous to CLK)

Input vector for selecting the PRBS sequence or of the bit pattern. This vector is read by the receiver when PRBS\_SET is high.

<b>PRBS_TYPE vector</b>	<b>PRBS sequence / bit pattern</b>	<b>Note</b>
0 0 0 0	$2^9 - 1$	DC free
0 0 0 1	$2^{11} - 1$	DC free
0 0 1 0	$2^{15} - 1$	DC free
0 0 1 1	$2^{20} - 1$	DC free
0 1 0 0	$2^{20} - 1$	DC free
0 1 0 1	$2^{23} - 1$	DC free
0 1 1 0	$2^{29} - 1$	DC free
0 1 1 1	$2^{31} - 1$	DC free
1 0 0 0	all '0' : "00000000"	DC only
1 0 0 1	all '1' : "11111111"	DC only
1 0 1 0	alternating '0' and '1' : "01010101"	DC free
1 0 1 1	alternating '00' and '11' : "00110011"	DC free
1 1 0 0	one '0' and seven '1' : "01111111"	With DC component
1 1 0 1	one '1' and seven '0' : "10000000"	With DC component
1 1 1 0	two '0' and six '1' : "01110111"	With DC component
1 1 1 1	two '1' and six '0' : "10001000"	With DC component

### **PRBS\_INV** (synchronous to CLK)

Input signal for selecting the polarity of the PRBS sequence. A high signal on this input inverts the PRBS bit sequence received on RX\_BIT. This signal is read by the receiver when PRBS\_SET is high. The polarity of the bit pattern is not affected by PRBS\_INV.

## PRBS (according ITU-T O.150) and Bit-Sequence Tester : VHDL-Modules

---

### **SYN\_STATE** (synchronous to CLK)

Output signal of the module for the synchronization state of the PRBS or bit pattern receiver.

The output goes to high when :

- in PRBS mode  $2 \cdot X$  ( $X = \text{PRBS shift register length [9..31]}$ ) error free bits are received
- in bit pattern mode 20 error free bits are received

The output goes to low if the error rate exceeds 0.2. To detect this level, the bit errors during the last 128 received bits are memorized. If there are more than 25 errors within these 128 bits, the output goes low.

### **SYN\_LOS** (synchronous to CLK)

Output signal of the module for synchronization lost indication.

When the synchronization upon the received bit sequence is lost, SYN\_STATE goes low.

At this time SYN\_LOS goes high for one period of CLK (plus CLK\_EN).

### **BIT\_ERR** (synchronous to CLK)

Output signal of the module for signaling a bit error in the received bit sequence when the receiver is synchronized (SYN\_STATE is high). This signaling output has a delay of 128 bit times (see SYN\_STATE description), because massive bit errors who causes a synchronization loss must not be reported as bit errors.

### **CLK\_ERR** (synchronous to CLK)

Output signal of the module for signaling a clock error (bit slip : bit lost or bit inserted) in the received bit sequence when the receiver is synchronized (SYN\_STATE is high).

*Not implemented at the moment.*

```
*****
* This document file is provided "as is" and WITHOUT any express or implied *
* warranties, that this document file is                                     *
* 1. free from any claims of infringement,                                   *
* 2. the merchantability or fitness for a particular purpose.               *
*****
```