Using the XC4000 Readback Capability

XAPP 015.000 Application Note By WOLFGANG HÖFLICH

Summary

This Application Note describes the XC4000 Readback capability and its use. Topics include: initialization of the Readback feature, format of the configuration and Readback bitstreams, timing considerations, software support for reading back LCA devices, and Cyclic Redundancy Check (CRC).

Xilinx Family	***Demonstrates***
XC4000	XC4000 Readback Capability

Purpose

Every LCA device shipped by Xilinx is tested using the device Readback capability. All CLBs and IOBs are configured and read back using extensive test patterns to guarantee 100% functionality of the LCA device.

An LCA device can be read back at any time after configuration.The Readback data consists of the configuration data and, optionally, the current state of the CLBs and IOBs.

When is a Readback Necessary or Useful?

The Xilinx devices are 100% pretested and the XC4000 series LCA devices can use Cyclic Redundancy Checking (CRC) on the configuration bitstream to check the integrity of the bitstream loaded into the LCA configuration memory.

In the configuration bitstream, there are four error-check bits for each data frame transmitted into the LCA device. Using this technique, the LCA device detects invalid data bits and aborts the configuration process. The INIT status pin is pulled Low, signaling that an error occurred during loading of the configuration memory.

Therefore, Readback is useful only in few cases.

- Verifying the configuration in a very unstable environment,
- Reading back the internal state of the RAM, CLBs and IOBs during the LCA development phase,
- In high-reliability applications that require in-system functional analysis and verification,
- For Xilinx internal testing

For examples of how to use Readback in your application, contact Xilinx.

Readback Highlights

The Readback features and the user interface of the XC4000 devices are significantly improved over the XC2000/XC3000 devices.

The Readback operation does not interfere with the LCA operation. After a valid Readback request, the current state of LCA internal nodes can be captured into a special shift register. Then the data can be transferred out of the device using a user-defined clock signal.

The following LCA internal configuration data and circuit nodes are available for Readback (Figure 1).

- Configuration memory bits that define the logic configration of CLBs, IOBs, and the LCA interconnects.
- X and Y output pins of CLB Function Generators.
- XQ and YQ output pins of CLB flip-flops,
- OQ output pins of IOB flip-flops,
- I1 and I2 input pins of IOBs

A mask file (<design_name>.LL), generated with the MakeBits program, contains information about the location of the user data bits in the Readback bitstream and the names of the signals connected.

The user can implement comparison logic in CLBs to perform the comparison with data stored in the configuration PROM. This technique does not work if any CLB is used as RAM, since changing the RAM contents alters the data in the configuration memory. In this case, an additional mask PROM is needed to disable the comparison of Readback bitstream locations that represent the RAM data.

The Readback speed is 10 kHz min, 1 MHz max. See the timing diagrams at the end of this application note.

The XC4000 family features a Boundary-Scan instruction that initiates a Readback sequence using the standard IEEE 1149.l/JTAG Boundary-Scan ports.

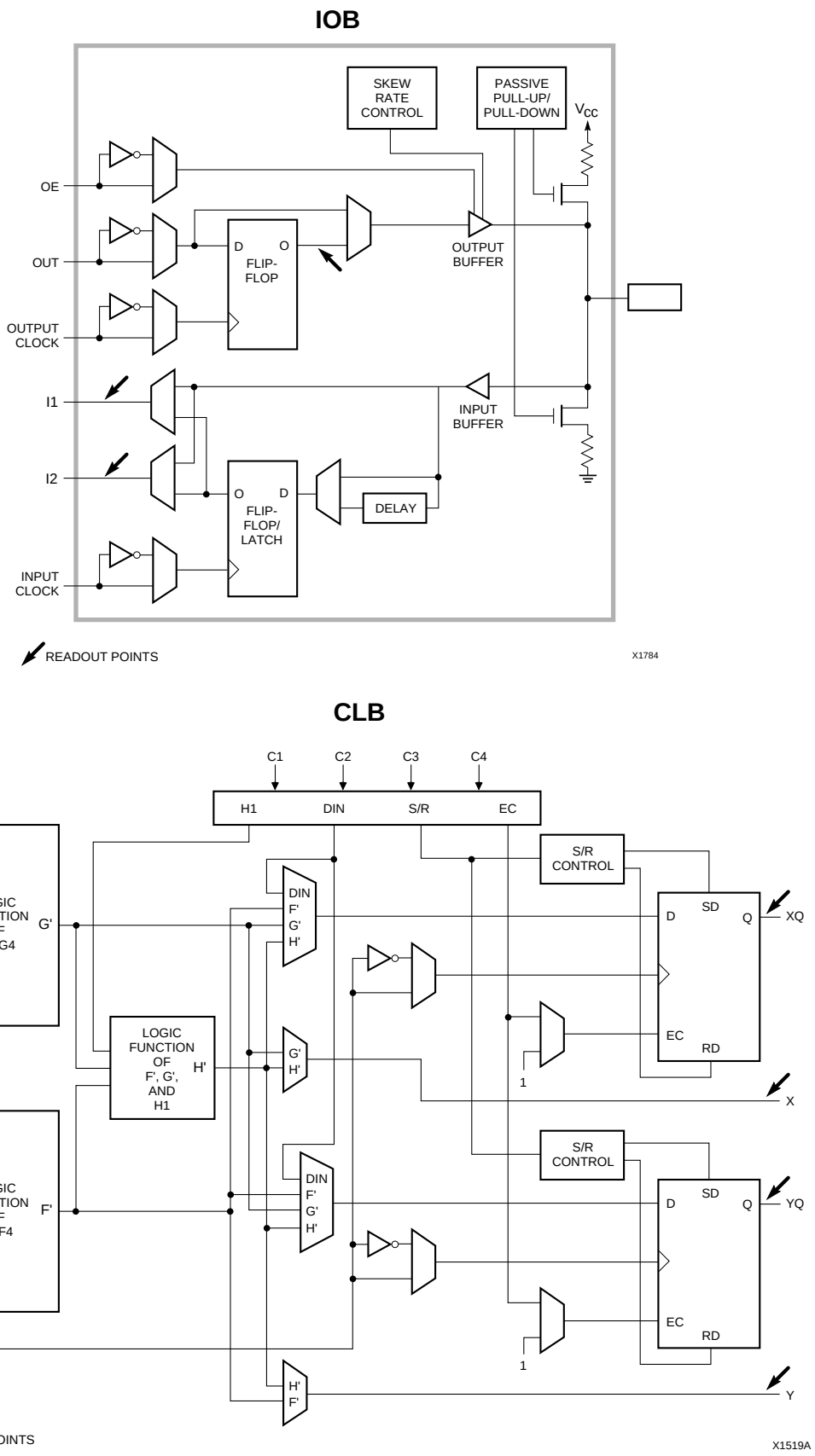


Figure 1. Readback Capture Enable

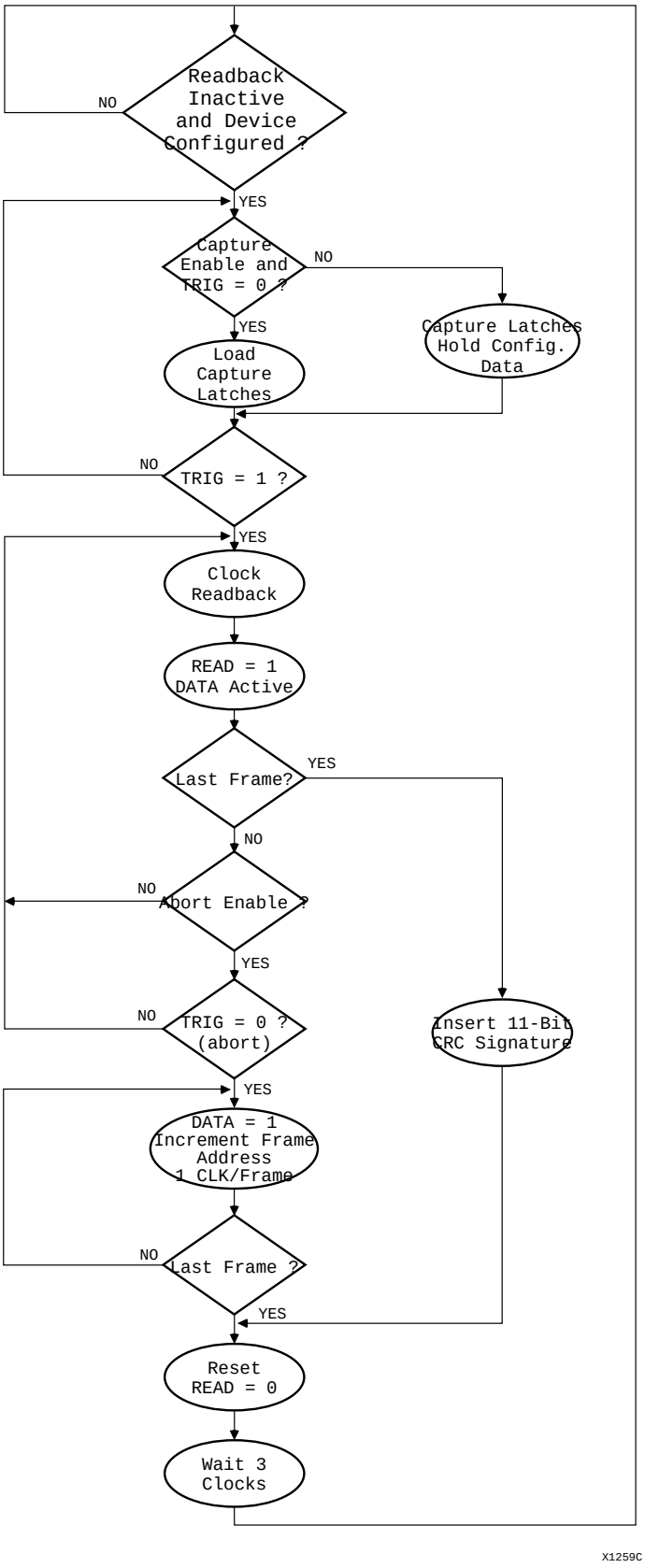


Figure 2. Readback State Diagram

Daisy chaining LCA devices for Readback is not possible. Each device must be read back individually.

The XChecker Universal Download Cable and Logic Probe handles configuration and Readback of XC2000, XC3000, and XC4000 FPGA families. In addition, it displays selected LCA internal nodes on screen.

Performing a Readback

Readback State Diagram

An LCA-internal state machine controls the Readback process. See Figure 2 for the Readback state diagram. For an explanation of the terms used, see below.

Readback Primitive

The XC4000 LCA device has a dedicated primitive that handles all of the Readback functions. It is located in the lower left and right corners of the LCA device and has two inputs and two outputs (Figure 3).

The Readback primitive can access general-purpose interconnects. Therefore, the four signals – rdclk.I, rdbk.TRIG, rdbk.RIP, and rdbk.DATA – can connect to the user I/Os and to CLBs as follows.

- rdclk.I – The Clock input can be connected to any device input pin, or any CLB output. If it is not connected to a user net, it connects to the device CCLK input pin, if the appropriate option is selected in the bitstream-generator MakeBits program.

- rdbk.TRIG – A Low-to-High transition on the TRIG input starts a Readback sequence. The minimum required pulse width is one rdclk.I cycle. A valid trigger causes the current value of certain nodes to be latched into an LCA internal holding register. If ReadAbort was selected as an option in MakeBits, a High-to-Low on the TRIG input aborts the Readback. In this case, additional clocks must be provided until rdbk.RIP signals the end of a Readback. The rdbk.TRIG cannot be reasserted until at least three clock periods after the previous Readback has been terminated correctly.

- rdbk.RIP (Readback-In-Progress) – A High on this output indicates that a Readback is being performed. RIP goes active one Readback clock cycle after a valid Readback trigger has occurred. It goes Low with the last data

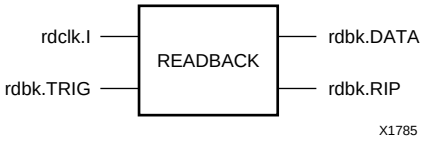


Figure 3. The Readback Primitive

bit shifted out of the LCA device. In the case of a Readback abort, RIP remains active until the Readback sequence is terminated correctly.

- rdbk.DATA – The Readback data is available on the DATA output of the Readback primitive. Each rising edge on rdclk.I shifts one data bit from the LCA-internal holding register to the DATA output. The data bitstream is explained below. There is an option to disable the user data bits in the Readback bitstream.

Note that in XC3000 devices, the input pin M0/RTRIG is used as a Readback Trigger pin and M1/RDATA as a Readback Data pin. In XC4000, the M0 pin can be used as an input pin, the M1 pin as a 3-state output.

Also, XC3000 has a MakeBits option to inhibit Readback. In XC4000, conventional Readback is possible if the Readback primitive is used in the design, or if a Boundary-Scan Readback is performed.

Readback Initialization

There are three ways of preparing an LCA design for Readback.

- Using the Readback primitive on the schematic.
- Activating Readback from the XACT Design Editor.
- Performing a Readback during a Boundary-Scan operation.

Readback from the schematic level
In the Xilinx Design Interface Libraries, there is a Readback primitive that can be called up into the schematic like any other library primitive. Simply connect the inputs and outputs of the Readback primitive to your user nets as desired. See Figure 4 for an example.

Note: If the CLK input is not connected to any net, the Place-and-Route software connects it to the CCLK input pin, if the appropriate ClkSelect=Cclk was selected in the MakeBits program.

Readback from the XDE
In XDE, the Readback primitive is located in the lower left and lower right corners of the device. It is activated if the

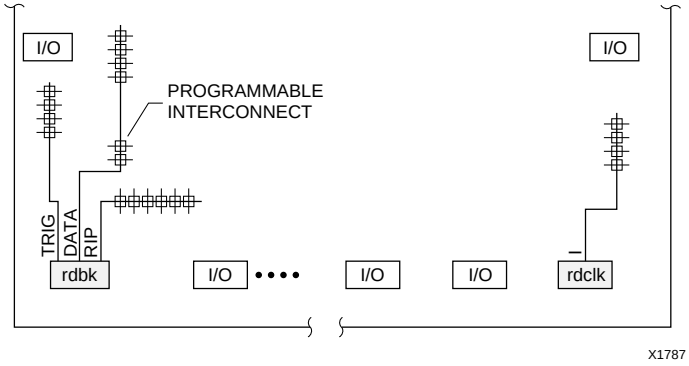


Figure 5. The XACT Readback Primitive

rdbk.TRIG and the rdbk.DATA signals are connected. The rdclk.I pin is connected to the CCLK pin, if not connected otherwise. See Figure 5.

Readback during a Boundary-Scan
No changes are required to prepare a design for Readback through the Boundary-Scan port. Contact Xilinx for additional information.

Configuration and Readback Bitstreams

The XC4000 Configuration Bitstream

Figure 6 shows the format of the XC4000 configuration bitstream, as generated by the XACT MakeBits program. The bitstream consists of header and program data. The header consists of four dummy bits, the preamble code, the configuration-program-length count, and an additional four dummy bits. The program data is divided into frames consisting of a Start bit (0), the data field, and four error check bits (eeee). The bitstream ends with eight or more postamble bits (01111XXX). The exact number of the bits in the bitstream is determined by the 24-bit program-length count.

The XC4000 Readback Bitstream

The Readback bitstream contains configuration information as well as the state of internal user logic. The Readback bitstream starts with five dummy bits. The Readback data frame has the same format as the configuration data frame which eases a bit-by-bit comparison between

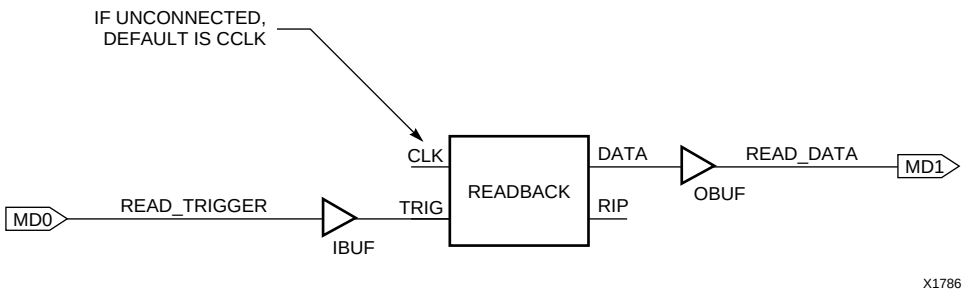


Figure 4. Readback Symbol on the Design Schematic

Readback and configuration data. Each data frame consists of a Start bit (0), the Data field, and four Stop bits (1111). The bitstream ends with 11 CRC bits, Figure 7.

Both the configuration data and the internal-logic data are included in the Readback bitstream. In the Readback bitstream, the configuration data bits are not inverted with respect to the configuration bitstream. The user-logic data bits, however, are inverted with respect to their values during Readback capture.

The read-back configuration data may differ from the original data downloaded into the device if CLB RAM is used in the design. The RAM data is stored in the F- and G-function tables of the CLB.

The first two bits of the first Readback data frame are variable; they are non-user, non-configuration bits. Their input state is dependent on the configuration speed and the configuration error-check mode of the LCA device. The last seven bits of the last Readback data frame are always ones.

If Readback capture of user data is disabled in the MakeBits program, logic Highs replace the user data. Note that the RAM data is not part of the captured user logic data; it is contained in the read-back configuration data.

The bitstream ends with eleven bits of a CRC signature appended. If ReadCapture is disabled and the design does not use any CLB RAM, this signature will be constant in successive Readbacks. See below for more information on the Polynomial Cyclic Redundancy Check CRC-16.

Software Support for Readback

The user can set Readback options with the MakeBits program.The following MakeBits options are relevant for Readback of XC4000 devices.

ReadCapture:

Settings: Enable, Disable
Default: Disable

This option determines whether the state of internal user logic is included in the Readback bitstream. If ReadCapture is disabled, the user data is replaced by ones.

ReadAbort:

Settings: Enable, Disable
Default: Disable

ReadAbort enables the level-sensitive signal rdbk.TRIG to abort the Readback. A High-to-Low transition stops the Readback. Additional clocks must be supplied to terminate the Readback correctly. As a minimum, the number of data frames contained in the device plus three must be sent as additional clocks. During this period, the Readback data is High. The rdbk.RIP signal indicates the completion of a Readback process.

ClkSelect:

Settings: CCLK, RDBK (user supplied)
Default: CCLK

The rdclk.I pin can be connected to any user net or to the CCLK I/O pin. With this option, the user can choose between the alternatives.

MakeBits features an option used to create a “logic allocation” file (<design_name>.LL) that contains information on which bit in the Readback bitstream corresponds to which signal in the design. This ASCII mask file indicates the offset from the beginning of the Readback bitstream, the frame number, the offset within a frame, and names of user signals in the Readback bitstream. Figure 8 shows an example.

Readback Timing

Minimum Readback frequency is 10 kHz; maximum Readback frequency is 1 MHz. The rdclk.I High time and Low time are each 0.5 µs min. See Table 1 for additional preliminary Readback switching characteristics.

Cyclic Redundancy Check (CRC) for LCA Configuration and Readback

Concept of the Cyclic Redundancy Check

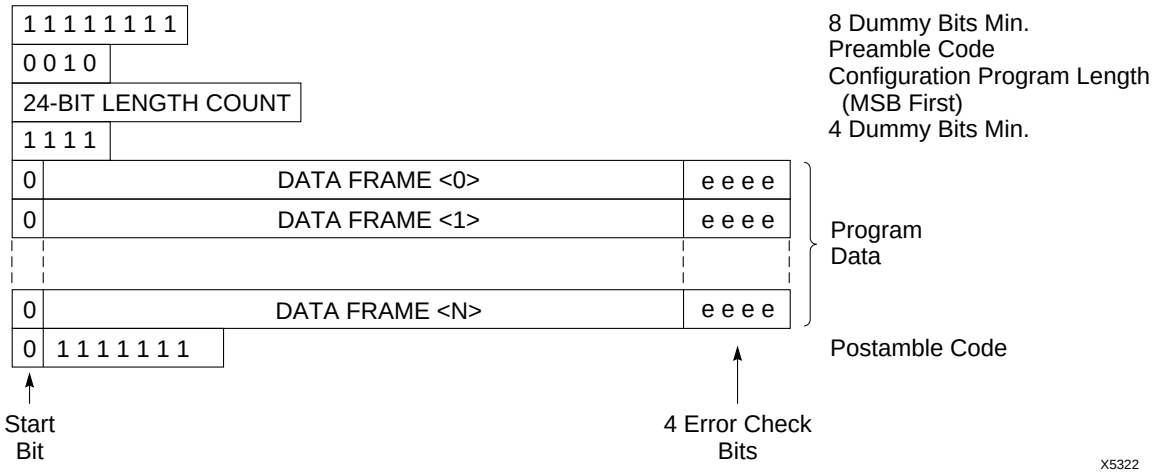
The Cyclic Redundancy Check is a method of error detection in data transmission applications. Generally, the transmitting system performs a calculation on the serial bitstream. The result of this calculation is tagged onto the data stream as additional check bits. The receiving system performs an identical calculation on the bitstream and compares the result with the received checksum. CRC Checksum Compare is often referred to as Signature Analysis.

CRC During LCA Configuration

Each data frame of the LCA configuration bitstream has four error bits at the end. See Figure 6. If a frame data error is detected during the loading of the LCA device, the configuration process with a potentially corrupted bitstream is terminated. The LCA pulls the INIT pin Low and goes into a Wait state.

CRC During LCA Readback

During an LCA Readback, 11 bits of the 16-bit checksum are appended to the end of the Readback data stream. The checksum is computed using the CRC-16 CCITT polynomial (Figure 9).The LCA checksum consists of the 11 most significant bits of the 16-bit code. A change in the checksum indicates a change in the Readback bitstream. Statistically, one in 2048 errors might go undetected.



Device	XC4002A	XC4003A	XC4003/3H	XC4004A	XC4005A	XC4005/5H	XC4006	XC4008	XC4010/10D	XC4025
Gates	2,000	3,000	3,000	4,000	5,000	5,000	6,000	8,000	10,000	20,000
CLBs	64	100	100	144	196	196	256	324	400	1,024
(Row x Col)	(8 X 8)	(10 X 10)	(10 X 10)	(12 X 12)	(14 X 14)	(14 X 14)	(16 X 16)	(18 X 18)	(20 X 20)	(32 X 32)
IOBs	64	80	80/160	96	112	112 (192)	128	144	160	256
Flip-Flops	256	360	360/300	480	616	616 (392)	768	936	1,120	2,560
Horizontal TBUF Longlines	16	20	20	24	28	28	32	36	40	64
TBUFs/Longlines	10	12	12	14	16	16	18	20	22	34
Bits per Frame	102	122	126	142	162	166	186	206	226	346
Frames	310	374	428	435	502	572	644	716	788	1,220
Program Data	31,628	45,636	53,936	62,204	81,332	94,960	119,792	147,504	178,096	422,128
PROMs size (bits)	31,668	45,676	53,976	62,244	81,372	95,000	119,832	147,544	178,136	422,168

XC4000, 4000H: Bits per Frame = (10 x number of Rows) + 7 for the top + 13 for the bottom + 1 + 1 start bit + 4 error check bits

Number of Frames = (36 x number of Columns) + 26 for the left edge + 41 for the right edge + 1

XC4000A: Bits per Frame = (10 x number of Rows) + 6 for the top + 10 for the bottom + 1 + 1 start bit + 4 error check bits

Number of Frames = (32 x number of Columns) + 21 for the left edge + 32 for the right edge + 1

Program Data = (Bits per Frame x Number of Frames) + 8 postamble bits

PROM Size = Program Data + 40

The user can add more “one” bits as leading dummy bits in the header, or, if CRC = off, as trailing dummy bits at the end of any frame, following the four error check bits, but the Length Count value **must** be adjusted for all such extra "one" bits, even for leading extra ones at the beginning of the header.

Figure 6. XC4000 Configuration Bitstream Format

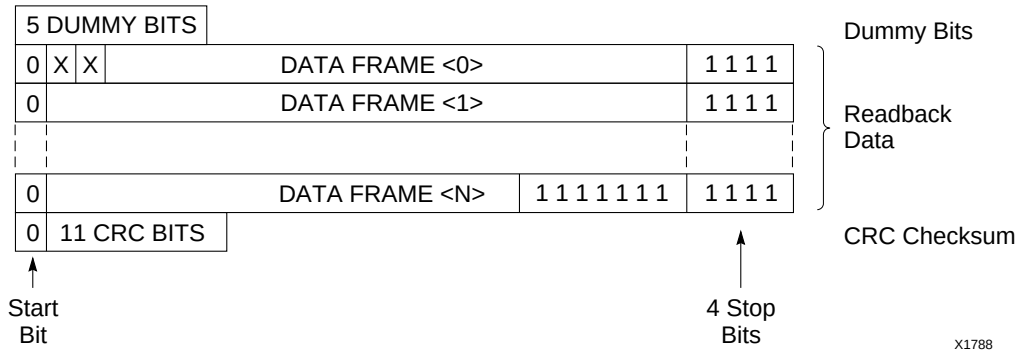
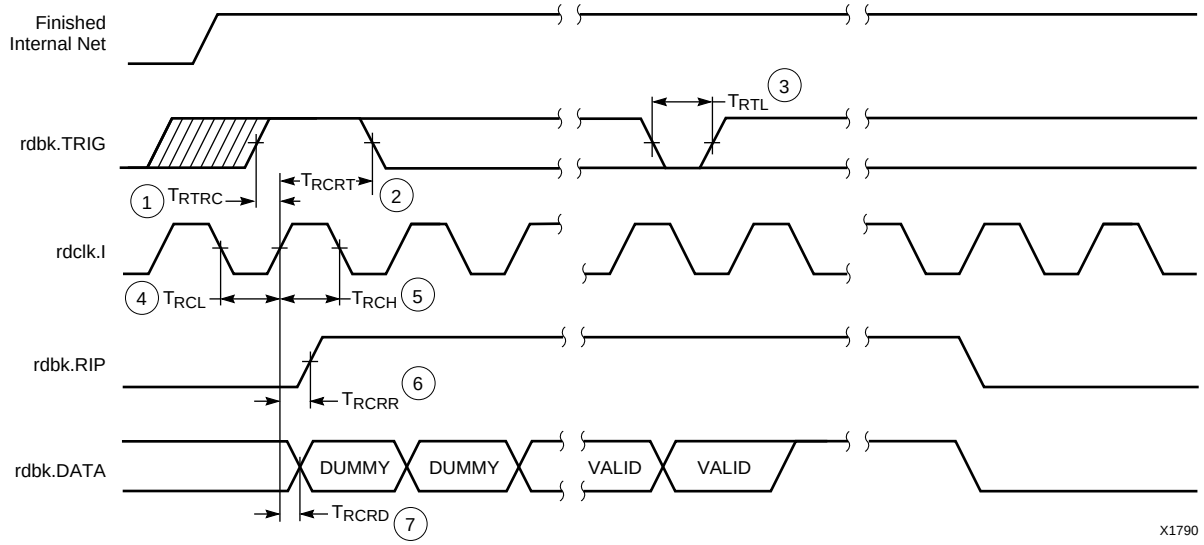


Figure 7. XC4000 Readback Bitstream

```
;  Offset   Column  (Frame)  Row (Frame  Offset)    Description
        21                1                 100        P57 I1
        32                1                  90        U37 I1
        41                1                  79        P60 U1
         .                .                   .          .
         .                .                   .          .
         .                .                   .          .
     36640              303                  23         CD YQ
     36650              303                  13         BD YQ
     37044              307                 103         LD XQ  CFG/TOGGLE
     37054              307                  93         KD XQ  CFG/RDATA_REG/Q9
     37064              307                  83         JD XQ  CFG/RDATA_REG/Q1
     37074              307                  73         ID XQ  CFG/RDATA_REG/Q2
     37084              307                  63         HD XQ  REFDATA_REG/Q5
     37095              307                  52         FD XQ
     37105              307                  42         ED XQ
         .                .                   .          .
         .                .                   .          .
         .                .                   .          .
```

Figure 8. Sample Logic Allocation File

Table 1. Readback Switching Characteristics



	Description	Symbol		Limits		
				Min	Max	Units
rdbk.TRIG	rdbk.TRIG setup	1	T_R			s
	rdbk.TRIG hold	2	T_R			s
	rdbk.TRIG Low to abort Readback	3	T_R			s
rdclk.I	rdbk.DATA delay	7	T_R			s
	rdbk.RIP delay	6	T_R			s
	High time	5	T_R			s
	Low time	4	T_{RCL}			s

Notes:
1. Timing parameters apply to all speed grades.
2. If rdbk.TRIG is High prior to Finished, Finished will trigger the first Readback.

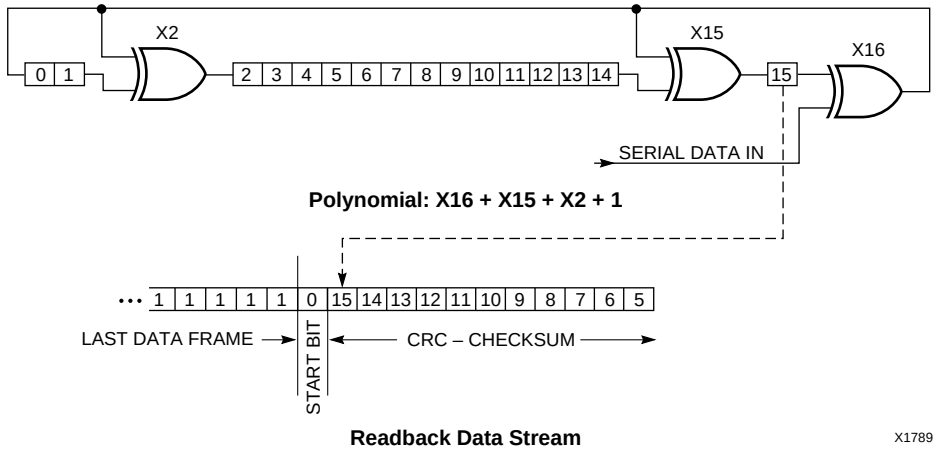


Figure 9. Circuit for Generating the CRC-16