CAN IP (precan_opb), v1.1

- With OPB interface

Product Specification

Prevas AB

PO Box 4 (Legeringsgatan 18)
SE-721 03 Västerås
Sweden

Phone: +46 21 360 19 00
Fax: +46 21 360 19 16
Email: erik.olofsson@prevas.se
URL: www.prevas.se

Features

- Compliant with ISO 11898-1, CAN 2.0B protocol.
- Supports bus speeds above 1 Mbit/s
- Overload frames on CAN bus are recognized and handled but not transmitted by this core
- TX and RX FIFOs with configurable depth of up to 64 messages each.
- RX message filtering. Number of filters can be set from 0-2 by generic parameter.
- Loop mode for diagnostic purpose
- Maskable interrupts
- Readable error counters
- Single clock, fully synchronous design.
- Seamless interface (slave port) to On-Chip Peripheral Bus (OPB) bus for easy integration with Xilinx MicroBlaze™ and IBM PowerPC™ in Xilinx devices.

The Prevas CAN Controller IP Core provides a flexible solution, which may be implemented in all Xilinx Spartan™-3, -3A, -3E and Virtex™-II, -II Pro, -4, -5 device families. See examples in the table below. The slice count depends on the selected FIFO depths and number of message filters. The figures shown are min-max values.

Table 1: Example implementation statistics

Family	Example Device	Fmax (MHz)	Slices (min-max)	IOBs (external	I/O (total)	BRAMs	Design Tools
Spartan™-3x	XC3S200A-5	94	633-838	2	105	2	ISE 10.1.03i
Spartan™-3x	XA3S500E-4Q	89	637-828	2	105	2	ISE 10.1.03i
Virtex™-4	XC4VLX15-12	142	636-834	2	105	2	ISE 10.1.03i

CORE Facts	
Provided with Core	
Documentation	User's Manual Data sheet (this document)
Design File Formats	VHDL Source RTL
Constraint Files	None
Verification	VHDL test bench, delivered with core. CAN protocol compliance tested, according to ISO16845, using the XA3S1600E ECU development board from Si-Gate
Instantiation templates	None
Reference designs & application notes	None
Additional Items	Xilinx XPS peripheral description files (.pao, .mpd)
Simulation Tool Used	
Modelsim v6.1c	
Support	
Support provided by Prevas AB	

Applications

The Prevas CAN Controller IP core targets many CAN communication applications like:

- Automotive networks
- Industrial control networks
- Sensor monitoring and actuator control
- Other embedded systems with CAN capabilities

General Description

The CAN Controller IP implements to the ISO 11898-1, CAN 2.0B protocol. It takes care of all transfer layer protocol tasks like message framing, message arbitration, error signaling and fault confinement and automatic retransmission due to tx faults or loss of arbitration. TX/RX FIFOs of configurable depth and message filters provides message buffering and filtering and decreases the load on the local CPU.

Max bus speed for CAN networks according to the standard is 1 Mbit/s. This CAN controller can however support bus speeds above 1 Mbit/s for special applications.

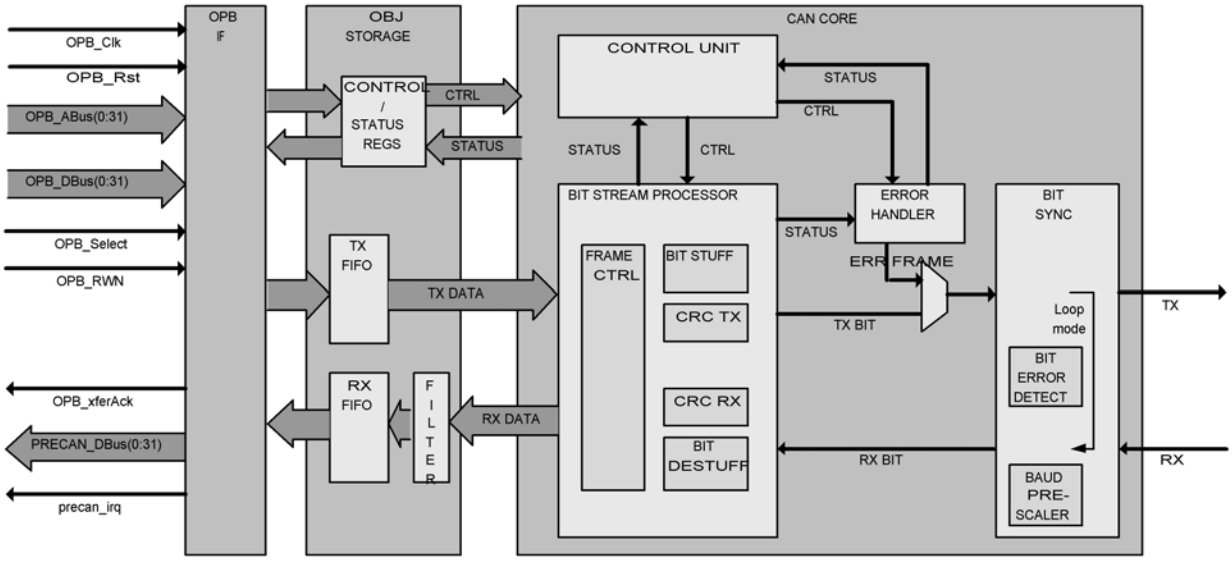

Figure1. Prevas CAN Controller IP core Block diagram

Functional Description

The figure above shows a high level block diagram and the major data flow of the CAN Controller. The TX and RX signals should be connected to an external CAN transceiver chip. A description of the sub-modules follows.

CAN core

The CAN core is responsible for all transfer layer functions of the CAN protocol and consist primarily of a control unit, bit-stream processor, error handler and bit synchronization unit.

Control unit

The control unit is supervisor of the other modules in the CAN core and handles the overall node behavior depending on the node state, i.e. if the node is in init mode, error active, error passive or bus off, and if the node is receiver or transmitter of the current message.

Bit stream processor

The bits stream processor (BSP) takes care of message data serialization and de-serialization, framing and format tasks. During transmission, these tasks are:

- Insert fixed form bits in the message frame.
- Insert stuff bits
- Calculate CRC and append CRC bits to the outgoing message stream.
- Perform bus arbitration
- Monitor the transmitted bits to detect bit errors.
- Automatic retransmission of the message in the case of a transmission error or if arbitration was lost

During reception of a message, similar tasks are performed in an opposite way. There are also several other error checks:

- Remove and check value of fixed form bits to detect form errors
- Remove stuff bits and check for stuffing errors
- Calculate CRC of the incoming message and compare with received CRC
- Acknowledge successfully received messages by transmitting a dominant bit in the ACK slot of the CAN frame.

Error handler

Any errors detected by the BSP are signaled to the error handler, which is responsible for fault confinement and transmission of error frames. It updates its transmit and receive error counters in accordance with the ISO 11898-1, CAN 2.0B standard and, based on the value of these counters, determines the error state of the controller; active, passive or bus off.

Bit synchronization
The bit synchronization module has the following functions:

- Clock pre-scaling.
- Synchronizing the CAN core to the traffic on the CAN bus.
- Place TX bits on the bus with the correct timing
- Calculate the sampling point and provide a sample clock to the rest of the CAN core.
- In loop mode, the core is “disconnected” from the CAN bus, i.e. only recessive bits are transmitted and the bits transmitted by this node is instead routed back to the receive input. Incoming data from the CAN bus is ignored.
- Bit error detection

The synchronization process compensates for propagation delays and oscillator frequency differences between the transmitting and receiving nodes.

Configuration and status registers
The configuration and status registers is the interface for an external micro-controller. Refer to the user manual for description of the registers.

All registers are 32-bit wide, addressed at word boundaries and represented in big-endian format (Bit 0 is MSB, bit 31 is LSB).

TX/RX FIFO
Separate storage buffers for transmit and receive message are provided in a FIFO structure. The depth of each buffer is individually configurable through generic parameters and may be up to 64 messages each.

RX filter
IDs of incoming messages are compared with user-defined ID acceptance masks. If there is a match, the message is stored in the RX buffer. If no match is found, the message is just acknowledged and then discarded. The number of acceptance filters is defined with a generic parameter which may be set from 0 to 2.

If no acceptance filter is used, all successfully received messages are stored in the RX buffer.

User interface

The user interface of the CAN controller is a subset of the OPB signals as defined in the Xilinx document “OPB Usage in Xilinx FPGAs”, march 2002.

These signals provide a seamless interface (slave port) for easy integration with Xilinx MicroBlaze™ and IBM PowerPC™ in Xilinx devices.

The controller is regarded as a slave device on the OPB bus. The table below describes the signal interface.

Signal	Dir	Description
OPB_Clk	IN	System clock (Minimum 24 MHz)
OPB_Rst	IN	Active high reset
OPB_ABus[0:31]	IN	Address bus
OPB_DBus[0:31]	IN	Write data bus
OPB_select	IN	Active high module select
OPB_RWN	IN	Transfer type: 1 = Read cycle 0 = Write cycle
OPB_BE[0:3]	IN	Not used, all registers are addressed at word boundaries
OPB_seqAddr	IN	Not used
PRECAN_DBus[0:31]	OUT	Read data bus
PRECAN_xferAck	OUT	Active high transfer acknowledge
PRECAN_retry	OUT	Not used, tied to GND
PRECAN_errAck	OUT	Not used, tied to GND
PRECAN_toutSup	OUT	Not used, tied to GND
can_rx	IN	RX data from CAN transceiver
can_tx	OUT	TX data to CAN transceiver
precan_irq	OUT	Interrupt line. Goes high when an interrupt is generated.

Table 2: Core I/O signals

Design parameters
A number of generic parameters can be used to tailor the design to specific application needs for optimum performance. The value of these parameters has some impact on the resource utilization of the IP, see table 1.

Design Services

Prevas also offers core integration, core customisation and other design services.

Parameter	Default	Description
C_BASEADDR	X"00000000"	Base address for the peripheral
C_HIGHADDR	X"0000007F"	High address for the peripheral
C_RX_FIFO_DEPTH	4	The desired depth of the RX buffer in number of messages. Settings 2-64 are valid.
C_TX_FIFO_DEPTH	4	The desired depth of the TX buffer in number of messages. Settings 2-64 are valid.
C_NR_OF_FILTERS	1	The desired number of receive filters. Settings 0-2 are valid.

Table 3: Core generic parameters

Core Modifications

More features may be added to the core at request and additional cost.

Customers may also modify the RTL source code at own responsibility.

Verification Methods

The CAN Controller core's functionality has been extensively tested with a test bench, which is also delivered with the core. The functionality has also been verified in accordance with the "ISO16845:2004 Road vehicles - Controller area network (CAN) - Conformance Test Plan". This compliance test has been performed by C&S Group, Germany.

Ordering Information

This product is available from Prevas AB under terms of the SignOnce IP License. See www.prevas.se for pricing or contact Prevas for additional information about this product.

Prevas AB
PO Box 4 (Legeringsgatan 18)
SE-721 03 Västerås,
Sweden

Phone: +46 21 360 19 00
Fax: +46 21 360 19 16
Email: erik.olofsson@prevas.se
URL: www.prevas.se

Prevas AB cores are purchased under a Licence Agreement, copies of which are available on request. Prevas AB retains the right to make changes to these specifications at any time, without notice. All trademarks, registered trademarks, or service marks are the property of their respective owners.

Related Information

Xilinx Programmable Logic

For information on Xilinx programmable logic or development system software, contact your local Xilinx sales office, or:

Xilinx, Inc. 2100 Logic Drive
San Jose, CA 95124
Phone:+1 408-559-7778
URL: www.xilinx.com