XINLUDA®
WWW.XINLUDA.COM 信路达

Stand-alone CAN controller

XD1000 DIP28
XL1000 SOP28

1 FEATURES

- ∞ Pin compatibility to the PCA82C200 stand-alone CAN controller
- ∞ Electrical compatibility to the PCA82C200 stand-alone CAN controller
- ∞ PCA82C200 mode (BasicCAN mode is default)
- ∞ Extended receive buffer (64-byte FIFO)
- ∞ CAN 2.0B protocol compatibility (extended frame passive in PCA82C200 compatibility mode)
- ∞ Supports 11-bit identifier as well as 29-bit identifier
- ∞ Bit rates up to 1 Mbits/s
- ∞ PeliCAN mode extensions:
  - – Error counters with read/write access
  - – Programmable error warning limit
  - – Last error code register
  - – Error interrupt for each CAN-bus error
  - – Arbitration lost interrupt with detailed bit position
  - – Single-shot transmission (no re-transmission)
  - – Listen only mode (no acknowledge, no active error flags)
  - – Hot plugging support (software driven bit rate detection)
  - – Acceptance filter extension (4-byte code, 4-byte mask)
  - – Reception of 'own' messages (self reception request)
- ∞ 24 MHz clock frequency
- ∞ Interfaces to a variety of microprocessors
- ∞ Programmable CAN output driver configuration
- ∞ Extended ambient temperature range (–40 to +125 ℃).

2 GENERAL DESCRIPTION

The XD1000/XL1000 is a stand-alone controller for the Controller Area Network (CAN) used within automotive and general industrial environments. It is the successor of the PCA82C200 CAN controller (BasicCAN) from Philips Semiconductors. Additionally, a new mode of operation is implemented (PeliCAN) which supports the CAN 2.0B protocol specification with several new features.

3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE	
	NAME	DESCRIPTION
XD1000	DIP28	plastic dual in-line package; 28 leads (600 mil)
XL1000	SOP28	plastic small outline package; 28 leads; body width 7.5 mm

BLOCK DIAGRAM

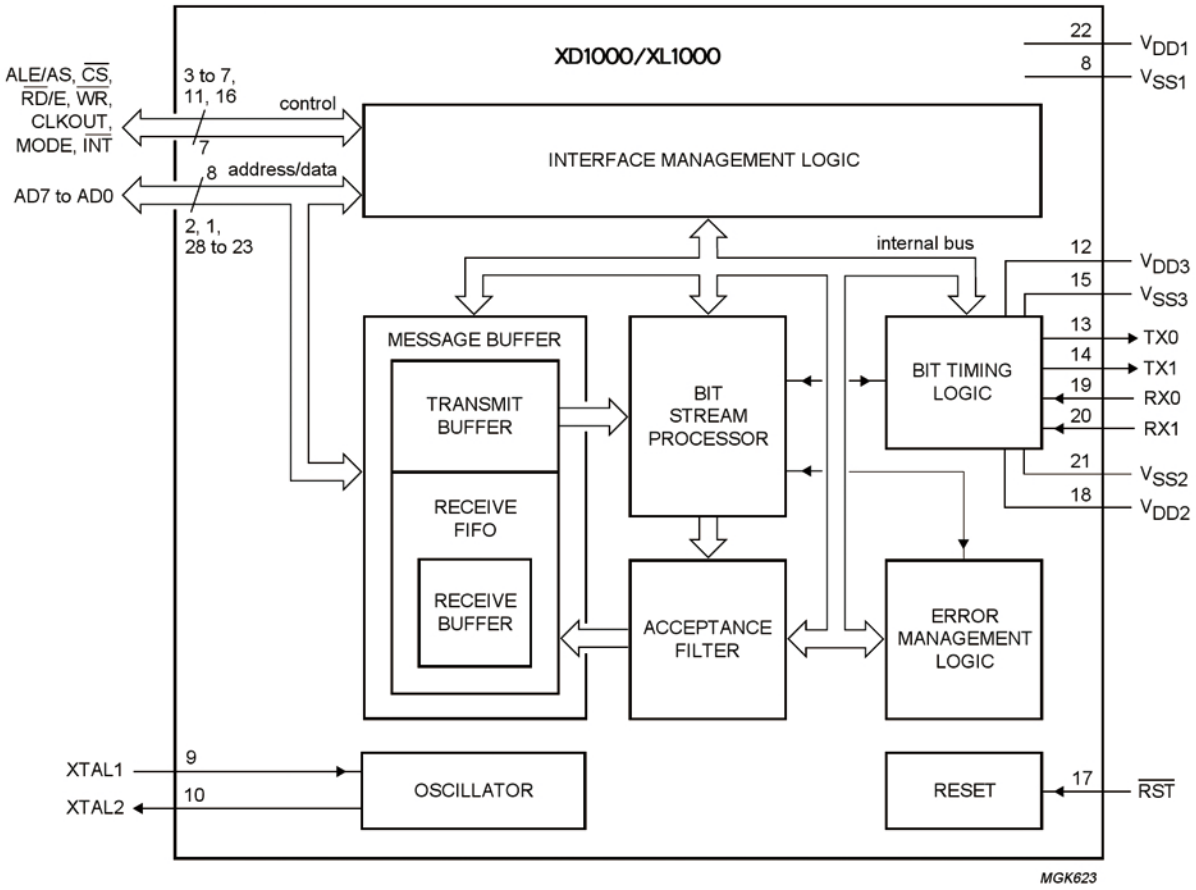

Fig.1 Block diagram.

PINNING

SYMBOL	PIN	DESCRIPTION
AD7 to AD0	2, 1, 28 to 23	multiplexed address/data bus
ALE/AS	3	ALE input signal (Intel mode), AS input signal (Motorola mode)
$\overline{CS}$	4	chip select input, LOW level allows access to the XD1000/XL1000
$\overline{RD}$/E	5	$\overline{RD}$ signal (Intel mode) or E enable signal (Motorola mode) from the microcontroller
$\overline{WR}$	6	$\overline{WR}$ signal (Intel mode) or RD/$\overline{WR}$ signal (Motorola mode) from the microcontroller
CLKOUT	7	clock output signal produced by the XD1000/XL1000 for the microcontroller; the clock signal is derived from the built-in oscillator via the programmable divider; the clock off bit within the clock divider register allows this pin to disable
V_{SS1}	8	ground for logic circuits
XTAL1	9	input to the oscillator amplifier; external oscillator signal is input via this pin; note 1
XTAL2	10	output from the oscillator amplifier; the output must be left open-circuit when an external oscillator signal is used; note 1
MODE	11	mode select input 1 = selects Intel mode 0 = selects Motorola mode
V_{DD3}	12	5 V supply for output driver
TX0	13	output from the CAN output driver 0 to the physical bus line
TX1	14	output from the CAN output driver 1 to the physical bus line
V_{SS3}	15	ground for output driver
$\overline{INT}$	16	interrupt output, used to interrupt the microcontroller; $\overline{INT}$ is active LOW if any bit of the internal interrupt register is set; $\overline{INT}$ is an open-drain output and is designed to be a wired-OR with other $\overline{INT}$ outputs within the system; a LOW level on this pin will reactivate the IC from sleep mode
$\overline{RST}$	17	reset input, used to reset the CAN interface (active LOW); automatic power-on reset can be obtained by connecting $\overline{RST}$ via a capacitor to V_{SS} and a resistor to V_{DD} (e.g. C = 1 ∝F; R = 50 k⏐)
V_{DD2}	18	5 V supply for input comparator
RX0, RX1	19, 20	input from the physical CAN-bus line to the input comparator of the XD1000/XL1000; a dominant level will wake up the XD1000/XL1000 if sleeping; a dominant level is read, if RX1 is higher than RX0 and vice versa for the recessive level; if the CBP bit (see Table 49) is set in the clock divider register, the CAN input comparator is bypassed to achieve lower internal delays if an external transceiver circuitry is connected to the XD1000/XL1000; in this case only RX0 is active; HIGH is interpreted as recessive level and LOW is interpreted as dominant level
V_{SS2}	21	ground for input comparator
V_{DD1}	22	5 V supply for logic circuits

Note

1. XTAL1 and XTAL2 pins should be connected to V_{SS1} via 15 pF capacitors.

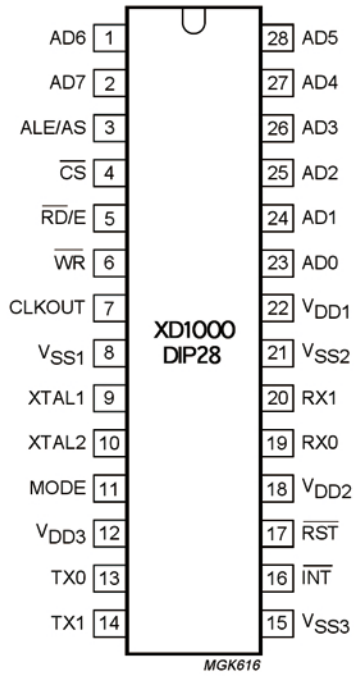

Fig.2 Pin configuration (DIP28).

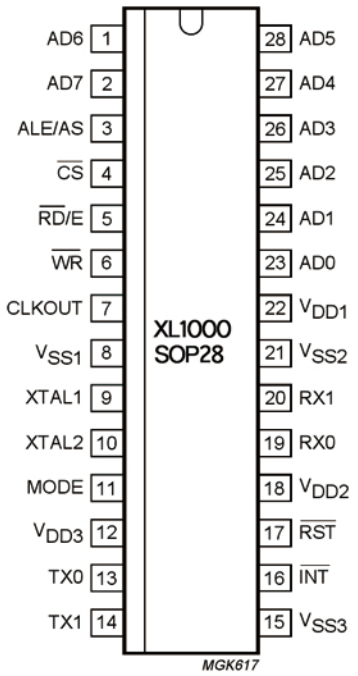

Fig.3 Pin configuration (SOP28).

Table 1 BasicCAN address allocation; note1

CAN ADDRESS	SEGMENT	OPERATING MODE		RESET MODE	
		READ	WRITE	READ	WRITE
0	control	control	control	control	control
1		(FFH)	command	(FFH)	command
2		status	–	status	–
3		interrupt	–	interrupt	–
4		(FFH)	–	acceptance code	acceptance code
5		(FFH)	–	acceptance mask	acceptance mask
6		(FFH)	–	bus timing 0	bus timing 0
7		(FFH)	–	bus timing 1	bus timing 1
8		(FFH)	–	output control	output control
9		test	test; note 2	test	test; note 2
10	transmit buffer	identifier (10 to 3)	identifier (10 to 3)	(FFH)	–
11		identifier (2 to 0), RTR and DLC	identifier (2 to 0), RTR and DLC	(FFH)	–
12		data byte 1	data byte 1	(FFH)	–
13		data byte 2	data byte 2	(FFH)	–
14		data byte 3	data byte 3	(FFH)	–
15		data byte 4	data byte 4	(FFH)	–
16		data byte 5	data byte 5	(FFH)	–
17		data byte 6	data byte 6	(FFH)	–
18		data byte 7	data byte 7	(FFH)	–
19		data byte 8	data byte 8	(FFH)	–
20	receive buffer	identifier (10 to 3)	identifier (10 to 3)	identifier (10 to 3)	identifier (10 to 3)
21		identifier (2 to 0), RTR and DLC	identifier (2 to 0), RTR and DLC	identifier (2 to 0), RTR and DLC	identifier (2 to 0), RTR and DLC
22		data byte 1	data byte 1	data byte 1	data byte 1
23		data byte 2	data byte 2	data byte 2	data byte 2
24		data byte 3	data byte 3	data byte 3	data byte 3
25		data byte 4	data byte 4	data byte 4	data byte 4
26		data byte 5	data byte 5	data byte 5	data byte 5
27		data byte 6	data byte 6	data byte 6	data byte 6
28		data byte 7	data byte 7	data byte 7	data byte 7
29		data byte 8	data byte 8	data byte 8	data byte 8
30		(FFH)	–	(FFH)	–
31		clock divider	clock divider; note 3	clock divider	clock divider

Notes

1. It should be noted that the registers are repeated within higher CAN address areas (the most significant bits of the 8-bit CPU address are not decoded: CAN address 32 continues with CAN address 0 and so on).
2. Test register is used for production testing only. Using this register during normal operation may result in undesired behaviour of the device.
3. Some bits are writeable in reset mode only (CAN mode and CBP).

(1) RESET VALUES

Detection of a 'reset request' results in aborting the current transmission/reception of a message and entering the reset mode. On the '1-to-0' transition of the reset request bit, the CAN controller returns to the operating mode.

Table 2 Reset mode configuration; notes 1 and 2

REGISTER	BIT	SYMBOL	NAME	VALUE	
				RESET BY HARDWARE	SETTING BIT CR.0 BY SOFTWARE OR DUE TO BUS-OFF
Control	CR.7	–	reserved	0	0
	CR.6	–	reserved	X	X
	CR.5	–	reserved	1	1
	CR.4	OIE	Overrun Interrupt Enable	X	X
	CR.3	EIE	Error Interrupt Enable	X	X
	CR.2	TIE	Transmit Interrupt Enable	X	X
	CR.1	RIE	Receive Interrupt Enable	X	X
	CR.0	RR	Reset Request	1 (reset mode)	1 (reset mode)
Command	CMR.7	–	reserved	note 3	note 3
	CMR.6	–	reserved		
	CMR.5	–	reserved		
	CMR.4	GTS	Go To Sleep		
	CMR.3	CDO	Clear Data Overrun		
	CMR.2	RRB	Release Receive Buffer		
	CMR.1	AT	Abort Transmission		
	CMR.0	TR	Transmission Request		
Status	SR.7	BS	Bus Status	0 (bus-on)	X
	SR.6	ES	Error Status	0 (ok)	X
	SR.5	TS	Transmit Status	0 (idle)	0 (idle)
	SR.4	RS	Receive Status	0 (idle)	0 (idle)
	SR.3	TCS	Transmission Complete Status	1 (complete)	X
	SR.2	TBS	Transmit Buffer Status	1 (released)	1 (released)
	SR.1	DOS	Data Overrun Status	0 (absent)	0 (absent)
	SR.0	RBS	Receive Buffer Status	0 (empty)	0 (empty)
Interrupt	IR.7	–	reserved	1	1
	IR.6	–	reserved	1	1
	IR.5	–	reserved	1	1
	IR.4	WUI	Wake-Up Interrupt	0 (reset)	0 (reset)
	IR.3	DOI	Data Overrun Interrupt	0 (reset)	0 (reset)
	IR.2	EI	Error Interrupt	0 (reset)	X; note 4
	IR.1	TI	Transmit Interrupt	0 (reset)	0 (reset)
	IR.0	RI	Receive Interrupt	0 (reset)	0 (reset)

REGISTER	BIT	SYMBOL	NAME	VALUE	
				RESET BY HARDWARE	SETTING BIT CR.0 BY SOFTWARE OR DUE TO BUS-OFF
Acceptance code	AC.7 to 0	AC	Acceptance Code	X	X
Acceptance mask	AM.7 to 0	AM	Acceptance Mask	X	X
Bus timing 0	BTR0.7	SJW.1	Synchronization Jump Width 1	X	X
	BTR0.6	SJW.0	Synchronization Jump Width 0	X	X
	BTR0.5	BRP.5	Baud Rate Prescaler 5	X	X
	BTR0.4	BRP.4	Baud Rate Prescaler 4	X	X
	BTR0.3	BRP.3	Baud Rate Prescaler 3	X	X
	BTR0.2	BRP.2	Baud Rate Prescaler 2	X	X
	BTR0.1	BRP.1	Baud Rate Prescaler 1	X	X
	BTR0.0	BRP.0	Baud Rate Prescaler 0	X	X
Bus timing 1	BTR1.7	SAM	Sampling	X	X
	BTR1.6	TSEG2.2	Time Segment 2.2	X	X
	BTR1.5	TSEG2.1	Time Segment 2.1	X	X
	BTR1.4	TSEG2.0	Time Segment 2.0	X	X
	BTR1.3	TSEG1.3	Time Segment 1.3	X	X
	BTR1.2	TSEG1.2	Time Segment 1.2	X	X
	BTR1.1	TSEG1.1	Time Segment 1.1	X	X
	BTR1.0	TSEG1.0	Time Segment 1.0	X	X
Output control	OC.7	OCTP1	Output Control Transistor P1	X	X
	OC.6	OCTN1	Output Control Transistor N1	X	X
	OC.5	OCPOL1	Output Control Polarity 1	X	X
	OC.4	OCTP0	Output Control Transistor P0	X	X
	OC.3	OCTN0	Output Control Transistor N0	X	X
	OC.2	OCPOL0	Output Control Polarity 0	X	X
	OC.1	OCMODE1	Output Control Mode 1	X	X
	OC.0	OCMODE0	Output Control Mode 0	X	X
Transmit buffer	–	TXB	Transmit Buffer	X	X
Receive buffer	–	RXB	Receive Buffer	X; note 5	X; note 5
Clock divider	–	CDR	Clock Divider Register	00000000 (Intel); 00000101 (Motorola)	X

Notes

1. X means that the value of these registers or bits is not influenced.
2. Remarks in brackets explain functional meaning.
3. Reading the command register will always reflect a binary '11111111'.
4. On bus-off the error interrupt is set, if enabled.
5. Internal read/write pointers of the RXFIFO are reset to their initial values. A subsequent read access to the RXB would show undefined data values (parts of old messages). If a message is transmitted, this message is written in parallel to the receive buffer but no receive interrupt is generated and the receive buffer area is not locked. So, even if the receive buffer is empty, the last transmitted message may be read from the receive buffer until it is overridden by the next received or transmitted message.
   Upon a hardware reset, the RXFIFO pointers are reset to the physical RAM address '0'. Setting CR.0 by software or due to the bus-off event will reset the RXFIFO pointers to the currently valid FIFO start address which is different from the RAM address '0' after the first release receive buffer command.

(2) Control Register (CR)

The contents of the control register are used to change the behaviour of the CAN controller. Bits may be set or reset by the attached microcontroller which uses the control register as a read/write memory.

Table 3 Bit interpretation of the control register (CR); CAN address 0

BIT	SYMBOL	NAME	VALUE	FUNCTION
CR.7	–	–	–	reserved; note 1
CR.6	–	–	–	reserved; note 2
CR.5	–	–	–	reserved; note 3
CR.4	OIE	Overrun Interrupt Enable	1	enabled; if the data overrun bit is set, the microcontroller receives an overrun interrupt signal (see also status register; Table 5)
			0	disabled; the microcontroller receives no overrun interrupt signal from the XD1000/XL1000
CR.3	EIE	Error Interrupt Enable	1	enabled; if the error or bus status change, the microcontroller receives an error interrupt signal (see also status register; Table 5)
			0	disabled; the microcontroller receives no error interrupt signal from the XD1000/XL1000
CR.2	TIE	Transmit Interrupt Enable	1	enabled; when a message has been successfully transmitted or the transmit buffer is accessible again, (e.g. after an abort transmission command) the XD1000/XL1000 transmits a transmit interrupt signal to the microcontroller
			0	disabled; the microcontroller receives no transmit interrupt signal from the XD1000/XL1000

(3) TRANSMIT BUFFER LAYOUT

The global layout of the transmit buffer is shown in Table 7. The buffer serves to store a message from the microcontroller to be transmitted by the XD1000/XL1000. It is subdivided into a descriptor and data field. The transmit buffer can be written to and read out by the microcontroller in operating mode only. In reset mode a 'FFH' is reflected for all bytes.

Table 4 Layout of transmit buffer

CAN ADDRESS	FIELD	NAME	BITS							
			7	6	5	4	3	2	1	0
10	descriptor	identifier byte 1	ID.10	ID.9	ID.8	ID.7	ID.6	ID.5	ID.4	ID.3
11		identifier byte 2	ID.2	ID.1	ID.0	RTR	DLC.3	DLC.2	DLC.1	DLC.0
12	data	TX data 1	transmit data byte 1							
13		TX data 2	transmit data byte 2							
14		TX data 3	transmit data byte 3							
15		TX data 4	transmit data byte 4							
16		TX data 5	transmit data byte 5							
17		TX data 6	transmit data byte 6							
18		TX data 7	transmit data byte 7							
19		TX data 8	transmit data byte 8							

(3.1) *Identifier (ID)*

The identifier consists of 11 bits (ID.10 to ID.0). ID.10 is the most significant bit, which is transmitted first on the bus during the arbitration process. The identifier acts as the message's name. It is used in a receiver for acceptance filtering and also determining the bus access priority during the arbitration process. The lower the binary value of the identifier the higher the priority. This is due to a larger number of leading dominant bits during arbitration.

(3.2) *Remote Transmission Request (RTR)*

If this bit is set, a remote frame will be transmitted via the bus. This means that no data bytes are included within this frame. Nevertheless, it is necessary to specify the correct data length code which depends on the corresponding data frame with the same identifier coding.

If the RTR bit is not set, a data frame will be sent including the number of data bytes as specified by the data length code.

(3.3) *Data Length Code (DLC)*

The number of bytes in the data field of a message is coded by the data length code. At the start of a remote frame transmission the data length code is not considered due to the RTR bit being at logic 1 (remote). This forces the number of transmitted/received data bytes to be logic 0. Nevertheless, the data length code must be specified correctly to avoid bus errors if two CAN controllers start a remote frame transmission with the same identifier simultaneously.

The range of the data byte count is 0 to 8 bytes and is coded as follows:

DataByteCount = 8 × DLC.3 + 4 × DLC.2 + 2 × DLC.1 + DLC.0

For reasons of compatibility no data length code >8 should be used. If a value >8 is selected, 8 bytes are transmitted in the data frame with the data length code specified in DLC.

(3.4) *Data field*

The number of transferred data bytes is determined by the data length code. The first bit transmitted is the most significant bit of data byte 1 at address 12.

(4) RECEIVE BUFFER

The global layout of the receive buffer is very similar to the transmit buffer described in Section (3) The receive buffer is the accessible part of the RXFIFO and is located in the range between CAN address 20 and 29.