ANALOG DEVICES

Technical notes on using Analog Devices DSPs, processors and development tools
Contact our technical support at dsp.support@analog.com and at dsptools.support@analog.com
Or visit our on-line resources http://www.analog.com/ee-notes and http://www.analog.com/processors

CAN Configuration Procedure for ADSP-21992 DSPs

Contributed by Petre M. and Stephen F.

Rev 1 – January 22, 2004

Introduction

ADSP-21992 DSPs contain a Controller Area Network (CAN) module. This document supplements the *ADSP-2199x DSP Hardware Reference Manual [1]* and shows how to configure the CAN module on ADSP-21992 DSPs. The Appendix includes an updated version of the Peripheral Interrupt Sources table to include the CAN module interrupt sources (Table 6). For information on CAN module registers, refer to the *ADSP-2199x DSP Hardware Reference Manual*.

CAN Module

Key features of the CAN module on ADSP-21992 DSPs include:

- Conforms to the CAN V2.0B standard
- Supports standard (11-bit) and extended (29-bit) identifiers
- Supports data rates of up to 1Mbit/sec (and higher)
- 16 configurable mailboxes (all receive or transmit)
- Dedicated acceptance mask for each mailbox
- Data filtering (first 2 bytes) can be used for acceptance filtering
- Error status and warning registers
- Transmit priority by identifier
- Universal Counter Module
- Readable receive and transmit counters

The CAN module is an asynchronous, low bit rate serial interface intended for use in applications with bit rates under 1 Mbit/ sec. The CAN protocol incorporates a data CRC check, message error tracking and fault node confinement as means to improve network reliability to the level required for control applications.

The CAN module architecture is based around a 16-entry mailbox RAM. The mailbox is accessed sequentially by the CAN serial interface or the host CPU. Each mailbox consists of eight 16-bit data words. The data is divided into fields, which includes a message identifier, a time stamp, a byte count, up to eight bytes of data, and several control bits. Each node monitors the messages being passed on the network. If the identifier in the transmitted message matches an identifier in one of it's mailboxes, the module knows that the message was meant for it, passes the data into its appropriate mailbox, and signals the host of its arrival with an interrupt.

The CAN network itself is a single, differential pair line. All nodes continuously monitor this line. There is no clock wire. Messages are passed in one of four standard message types (frames). Synchronization is achieved by an elaborate sync scheme performed in each CAN receiver. Message arbitration is accomplished one bit at a time. A dominant polarity is established for the network. All nodes are allowed to start transmitting at the same time following a frame sync pulse.

As each node transmits a bit, it checks to see if the bus is the same state that it transmitted. If it is, it continues to transmit. If not, then another node has transmitted a dominant bit, so the first node knows it has lost the arbitration and it stops transmitting. Arbitration continues, bit by bit, until only one node is left transmitting.

Because the electrical characteristics of each network connection are very stringent, the CAN interface is typically divided into two parts: a controller and a transceiver. This allows a single controller to support different drivers and CAN networks. The ADSP-21992 DSP CAN module represents only the controller part of the interface. This module's network I/O is a single transmit line and a single receive line, which communicate to a line transceiver.

Configuring the CAN Module

After power-up, the CAN module must be configured. This operation is accomplished by specifying whether the CANTX pin is to be used as open drain output or push pull and initializing the bit configuration registers.

By default, the CANTX pin of ADSP-21992 is open drain after reset, supporting single-wire network topologies. As more likely an off-chip CAN transceiver device is connected to implement the physical layer the CANTX_CTRL bit of the SYSREG0 register (IOPG=0xE) is usually set.

Bit	Name	Function	Type	Reset State
0	CANTX_CTRL	0 - CANTX is open drain 1 – CANTX is push pull	Read/Write	0
15-1	Reserved		Read	0

Table 1: SYSREG0 register

After Power-Up, the CAN module must be configured. This entails setting the module in configuration mode, initializing the bit configuration registers CANBCR0 (Table 2), CANBCR1 (Table 3), and then enabling normal mode. If the CAN module is not properly configured, it cannot be connected to the CAN bus line. In this mode, the length of the bit (baud rate), the location of the sample point inside the bit, and the synchronization jump width (SJW) are defined.

Bit	Name	Function	Type	Reset State
9-0	BRP	Baud Rate Prescaler	Read/Write	0
15-10	Reserved		Read	0

Table 2: Bit Configuration Register 0 (CANBCR0)

Bit	Name	Function	Type	Reset State
3-0	TSEG1	Time Segment 1	Read/Write	0
6-4	TSEG2	Time Segment 2	Read/Write	0
7	SAM	Sampling	Read/Write	0
9-8	SJW	Synchronization Jump Width	Read/Write	0
15-10	Reserved		Read	0

Table 3: Bit Configuration Register 1 (CANBCR1)

Figure 1 presents the structure of one bit implemented in this CAN module:

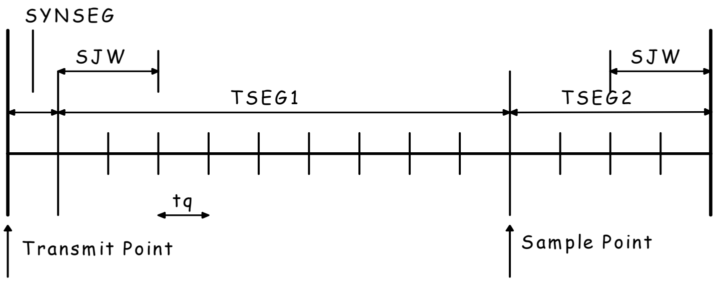

Figure 1: Bit timing structure

t_q represents the period of one quantum and is derived from the clock cycle that powers the CAN module. For the ADSP-21992 DSP, this is the peripheral clock (HCLK), and the prescale rate is BRP. The period of one bit is therefore:

$$TQ_{bit} = [1 + (1 + TSEG1) + (1 + TSEG2)] \cdot t_q$$

$$t_q = \frac{BRP + 1}{HCLK}$$

Equation 1. Bit timing

Table 4 presents the values that are allowed for each parameter.

Name	Bit Length	Min. Value	Max. Value	Definition
TSEG1	4	0	15	Time segment1
TSEG2	3	0	7	Time segment2
SAM	1	0	1	Sample point
SJW	2	0	3	Synchronization jump width
BRP	10	0	1023	Baud rate prescaler

Table 4: Bit timing configuration parameters

TSEG1 must always be greater than or equal to TSEG2. SJW must be less than or equal to TSEG2. New restrictions are presented in Table 5.

BRP	Time quanta, t_q	Minimum length of TSEG2
0	$\frac{1}{HCLK}$	3
1	$\frac{2}{HCLK}$	2
>1	$\frac{BRP+1}{HCLK}$	1

Table 5: Restrictions between BRP and TSEG2

The sample point is the point in time at which the CAN bus level is read and interpreted as the value of that respective bit. Its location is at the end of TSEG1. If SAM is 0, the CAN bus is sampled only once and the read value is assigned to the bit. If SAM is 1, the CAN bus is sampled three times (three time sampling mode) and the majority determines the resulting value. Note that three time sampling mode is not allowed when BRP is less than 4.

For example, suppose the communication on the CAN bus must take place at the maximum bit rate allowed by the specification (1 Mbit/s). Assume the DSP core of ADSP-21992 is operating at 160 MHz, the maximum rate; therefore, the peripheral clock is 80 MHz.

$$\frac{1}{1MHz} = \left[1 + \left(1 + TSEG1\right) + \left(1 + TSEG2\right)\right] \cdot \frac{BRP + 1}{80MHz}$$

$$80 = \left(3 + TSEG1 + TSEG2\right) \cdot \left(BRP + 1\right)$$

Equation 2. CAN Bit Timing Equation

From a large variety of answers, one may choose:

TSEG1=12, TSEG2=5, BRP=3, SJW=3, SAM=3,

CANBCR0=0x3, CANBCR1=0x03DC

Perform the following procedure to configure the CAN module:

1) Set CCR (bit 7 in the CANMCR register) (i.e., CANMCR=0x0080). This enables configuration mode.
2) Test CCA (bit 7 of the CANGSR register) until it becomes 1. You must wait until the CAN module is in configuration mode.
3) Initialize the CANBCR0, CANBCR1, and eventual acceptance mask registers (CANMBAMxH, CANMBAMxL, x=0,..,F)
4) Clear CCR (bit 7 in the CANMCR register) (ie.e, CANMCR=0x0). This cancels configuration mode and enters normal mode.
5) Test CCA (bit 7 of CANGSR register) until it becomes 0. You must wait until the CAN module is in normal mode.
6) Wait for the CAN module to power up (i.e., 11 consecutive recessive bits must be detected on the CAN bus line).

Conclusion

This document has provided a brief description of the CAN module on the ADSP-21992 and a configuration procedure for this CAN module.

References

[1] *ADSP-2199x DSP Hardware Reference*. Preliminary Edition 0, 2003. Analog Devices, Inc.

[2] *ADSP-21992 Mixed Signal DSP Controller with CAN Data Sheet*. Rev.0, Analog Devices, Inc.

[3] *CAN specification 2.0B.* 1991, Robert Bosch GmbH.

Appendix – Updated ADSP-21992 Peripheral Interrupt Sources Table

Peripheral Interrupt ID	IPR Register Bits	Interrupt Name	Interrupt Source & Description
0	IPR0 [3:0]	SPORT0_RX_IRQ	SPORT Receive Interrupt
1	IPR0 [7:4]	SPORT0_TX_IRQ	SPORT Transmit Interrupt
2	IPR0 [11:8]	SPI_IRQ	SPI Receive/Transmit Interrupt
3	IPR0 [15:12]		Reserved
4	IPR1 [3:0]		Reserved
5	IPR1 [7:4]		Reserved
6	IPR1 [11:8]		Reserved
7	IPR1 [15:12]		Reserved
8	IPR2 [3:0]	PWMSYNC_IRQ	PWM Synchronization Interrupt
9	IPR2 [7:4]	PWMTRIP_IRQ	PWM Shutdown Interrupt
10	IPR2 [11:8]		Reserved
11	IPR2 [15:12]		Reserved
12	IPR3 [3:0]	EIU0TMR_IRQ	EIU Loop Timer Interrupt
13	IPR3 [7:4]	EIU0LATCH_IRQ	EIU Latch Interrupt
14	IPR3 [11:8]	EIU0ERR_IRQ	EIU Error Interrupt
15	IPR3 [15:12]	ADC0_IRQ	ADC End of Conversion Interrupt
16	IPR4 [3:0]		Reserved
17	IPR4 [7:4]		Reserved
18	IPR4 [11:8]		Reserved
19	IPR4 [15:12]		Reserved
20	IPR5 [3:0]	TMR0_IRQ	General Purpose Timer 0 Interrupt
21	IPR5 [7:4]	TMR1_IRQ	General Purpose Timer 1 Interrupt
22	IPR5 [11:8]	TMR2_IRQ	General Purpose Timer 2 Interrupt
23	IPR5 [15:12]	MEMDMA_IRQ	Memory DMA Interrupt
24	IPR6 [3:0]	FIOA_IRQ	Flag IO Interrupt A
25	IPR6 [7:4]	FIOB_IRQ	Flag IO Interrupt A
26	IPR6 [11:8]	AUXSYNC_IRQ	Auxiliary PWM Sync. Interrupt
27	IPR6 [15:12]	AUXTRIP_IRQ	Auxiliary PWM Trip Interrupt
28	IPR7 [3:0]	CAN_RX_IRQ	CAN Mailbox Receive Interrupt
29	IPR7 [7:4]	CAN_TX_IRQ	CAN Mailbox Transmit Interrupt
30	IPR7 [11:8]	CAN_ERR_IRQ	CAN Error Condition Interrupt
31	IPR7 [15:12]		Reserved

Table 6: Peripheral Interrupt Sources

Document History

Revision	Description
Rev 1 – January 22, 2004 *by Petre M. and Stephen F.*	Initial Release