ANALOG DEVICES

AN-1001
APPLICATION NOTE

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

Checksum Calculations

by Enrico Del Mastro

This application note describes how the various checksums generated by the ADM106x development software are calculated and, when appropriate, discusses the locations in EEPROM where these calculated checksums are stored.

CALCULATING THE CONFIGURATION REGISTER CHECKSUMS (0x00 TO 0xFF)

The ADM106x configuration registers contain the data that the user is currently working with (referred to as volatile memory). In the event of a loss of power, all of this data will be lost.

The calculation of the checksum is only concerned with the registers which can be configured by the user, such as the fault detector registers, output registers, DAC registers, general-purpose registers, and ADC limit registers.

To calculate the checksum for the configurable locations, the data from all locations in the volatile memory is read from the device. However, only the data from the following locations is used in the calculation of the checksum:

- 0x00 to 0x7C (but not including 0x7C)
- 0x80 to 0x83 (but not including 0x83)
- 0x91 to 0x93 (but not including 0x93)

The data from the above locations is added together and the sum total of the data from these locations is then inverted and then AND'ed with 0xFFFF (see Figure 2).

CALCULATING THE CONFIGURATION EEPROM CHECKSUMS (0xF8; 0x00 TO 0xF8; 0xFF)

The configuration EEPROM registers contain the data that the user has saved from the controlling registers into the EEPROM (nonvolatile memory). Therefore, in the event of a loss of power all this data will not be lost.

The calculation of this checksum is only concerned with the registers which can be configured by the user, such as the fault detector registers, output registers, DAC registers, general-purpose registers, and ADC limit registers.

To calculate the checksum for the configurable EEPROM locations, the data from only 148 of the 256 locations in the 0xF8 nonvolatile memory space is read from the device. However, only the data from the following locations is used in the calculation of the checksum:

- 0xF8; 0x00 to 0xF8; 0x7C (but not including 0xF8; 0x7C)
- 0xF8; 0x80 to 0xF8; 0x83 (but not including 0xF8; 0x83)
- 0xF8; 0x91 to 0xF8; 0x93 (but not including 0xF8; 0x93)

The data from the above locations is added together and the sum total of the data from these locations is then inverted and then AND'ed with 0xFFFF (see Figure 3).

In addition, when data is saved to EEPROM, the checksum calculations are also stored in the EEPROM of the device and can be read by the user at any time. The locations of these stored checksums is shown in Figure 1 These calculations can be viewed in the text file which is generated by the ADM106x development software. Note in the example shown here that xx stands for the checksum data.

```
<F8;88;xx> CONFIGURATION EEPROM CHECKSUM LSB
<F8;89;xx> CONFIGURATION EEPROM CHECKSUM MSB
<F8;8A;xx> USER EEPROM CHECKSUM LSB
<F8;8B;xx> USER EEPROM CHECKSUM MSB
<F8;8C;xx> SE CHECKSUM LSB
<F8;8D;xx> SE CHECKSUM 2ND BYTE
<F8;8E;xx> SE CHECKSUM MSB
```

08011-001

Figure 1. Checksum Data Stored within the Text File Generated by the Development Software

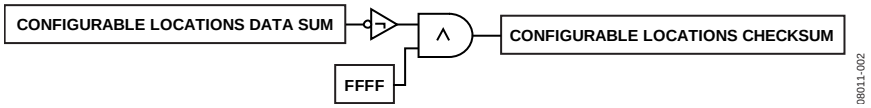

Figure 2. Configurable Locations Checksum Calculation

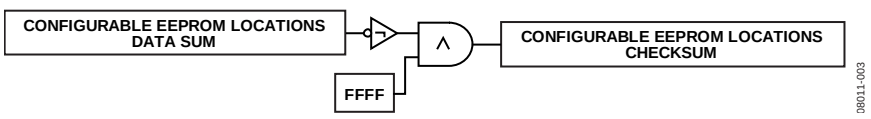

Figure 3. Configurable EEPROM Locations Checksum Calculation

CALCULATING THE USER EEPROM CHECKSUMS (0xF9; 0x00 TO 0xF9; 0xFF)

The user EEPROM registers contain the data that the user has saved. These 256 locations are for the user to store data, such as revision IDs. This data has no impact on the function of the ADM106x device.

To calculate the checksum for the user EEPROM locations, the data from all 256 locations in the 0xF9 nonvolatile memory space is read from the device.

The data from the locations 0xF9; 0x00 to 0xF9; 0xFF are added together and the sum total of the data from these locations is then inverted and then AND'ed with 0xFFFF.

CALCULATING THE SEQUENCE ENGINE EEPROM CHECKSUMS (0xFA; 0x00 TO 0xFB; 0xFF)

The sequence engine EEPROM registers contain the data which the user has saved from the sequencing engine editor in the ADM106x development software.

To calculate the checksum for the sequence engine EEPROM registers, the data from all locations in the 0xFA; 0x00 to 0xFB; 0xFF is read from the device.

The data from the above locations are added together and the sum total of the data from these locations is then inverted and then AND'ed with 0xFFFFF.

CALCULATING THE INTEL HEXADECIMAL CHECKSUMS (0xF8; 0x00 TO 0xFB; 0xFF)

When an Intel hexadecimal file is generated, the checksum is calculated by adding all the end of line twos complement checksums, excluding the 0xFF on the last line of the file and excluding the checksums in the area of EEPROM from 0xF8; 0xA0 to 0xF8; 0xFF. This area is the locked space in EEPROM. which contains all the trim and configuration registers specific to the device. These are set up in the test process at Analog Devices, Inc. This space is locked to prevent the user from inadvertently erasing this data and rendering the ADM106x device useless.

DISPLAYED CHECKSUMS ON SOFTWARE STARTUP

Figure 4 shows the display box that appears on the bottom right of the top level ADM106x evaluation software.

On startup the software looks to the registers in the EEPROM space shown in Figure 1 and displays this data. If no data is available in the EEPROM, then a calculation of the configuration registers and the Intel hexadecimal file only is performed and the results are displayed.

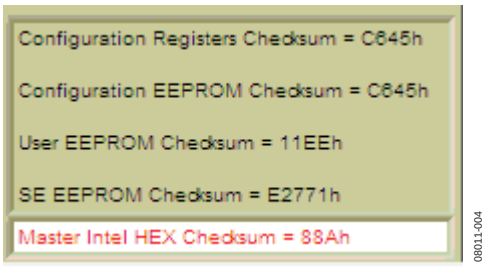

Figure 4. Example of ADM106x Software Top-Level Checksum Display Box

AN08011-0-4/09(0)