# Application Note

## MPEG Messages

By Steve Proffitt

### SERIAL CONTROL COMMANDS

When using the MPEG code supplied from Crystal, the system designer has the option of communicating to the CS4920A via the serial control port (SCP). There are several defined messages in addition to the MPEG algorithm which provide control and information concerning the decompression process. Note MPEG is used throughout this document to mean MPEG 1 and MPEG 2 layers I and II.

The CS4920A's SCP is always configured as a slave, therefore the master, such as a microcontroller, must initiate all messages, including those messages for which the CS4920A will be providing information to the master.

The basic message format can be seen in Figure 1. Each message has a unique byte for the command. All messages require two data bytes to follow the command byte, however not all message data bytes contain valid information.

#### Sending Messages to the CS4920A

The syntax for the serial control commands sent to the CS4920A by the host has two basic forms. The first form is an eight bit command field, also described as the command byte, followed by a 16-bit data field. This form is shown in Figure 1. The second form is a superset of the first, in that the command has the 8-bit command field, followed by the 16-bit data field. However, the command continues with multiple 24-bit data fields. The second form is shown in Figure 2. Every 24 bits is considered to be a single message to the CS4920A.

After a reset, the microcode is in a default state. The basic format for sending commands will be discussed first. The microcode expects all messages to be sent twice. This is the case after any reset.

Every serial command has a unique command byte to identify the message. Page 6 has a list of the available commands and their respective command bytes. The messages are sent (to the CS4920A) most significant bit first.

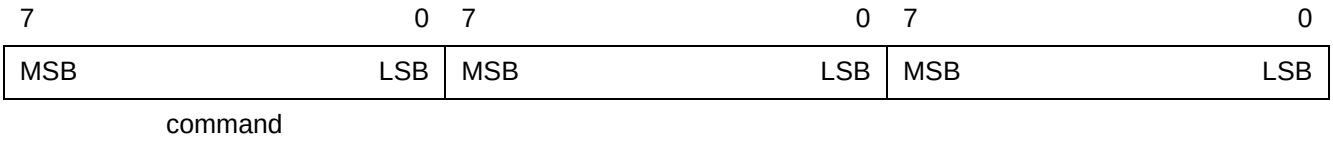



**Figure 1. MPEG Message Format**

*Cirrus Logic, Inc.*
*Crystal Semiconductor Products Division*
*P.O. Box 17847, Austin, Texas 78760*
*(512) 445 7222 FAX: (512) 445 7581*
*http://www.crystal.com*

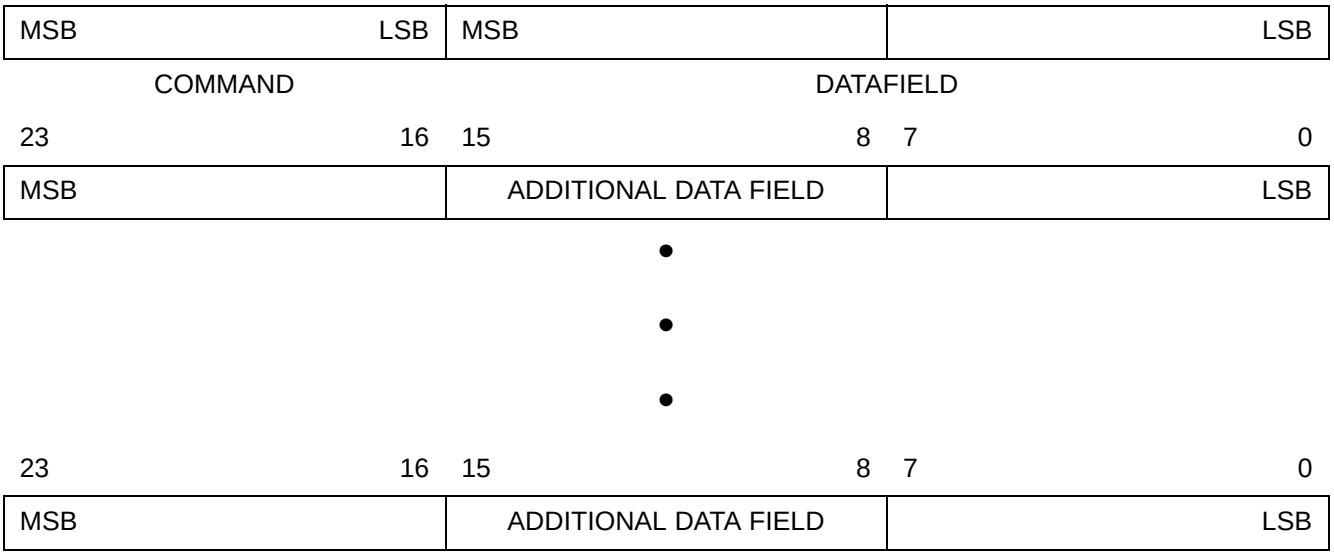



**Figure 2.**

To the CS4920A microcode, every 24 bits is considered a message. In the case of commands with additional 24-bit data fields, each additional 24 bits is considered a message by the microcode. In the *default state* the microcode expects every *message* to be sent twice, this is called double-messaging mode. The user can put the microcode into single-messaging mode using the MSG_MSG_FORMAT command.

In the default state these messages must be sent twice before the message takes effect. The reason for sending the messages twice is twofold. First, if a bit error occurs in one of the messages, the message will not execute, thus preventing a possible error in playback. Second, with two messages being sent the CS4920A can recover from a missing byte caused by a rejection of one of the transmitted bytes. The message sent first can not be recovered but any message following will be received correctly. A rejection of a byte would occur if the CS4920A was unable to extract a byte from the SCPIN register before another byte was shifted into the control port's shift register.

It may be difficult to detect an error in communication, especially when operating in SPI mode. There are however, a few methods for detecting and correcting communication errors.

The easiest way to detect a communication error is when operating in $I^2C$ mode. The failure of the CS4920A to ACK the host indicates that a data byte was not received. An appropriate course of action would be to send the byte again until the byte was ACKed. The master could also choose to send a STOP condition and send the byte at a later time. If the CS4920A still fails to accept the byte, the master has no choice but to assume the communication path is down and must use the $\overline{\text{RESET}}$ signal to re-initialize the CS4920A.

Another way of determining the functionality of the SCP interface is to send a command to the CS4920A that requires a reply. The most common command for this function is MSG_PLL_UNLOCK. Once the command is sent, the CS4920A should respond by driving the $\overline{\text{REQ}}$ signal active low indicating to the master it needs servicing. The host should then respond by reading the message to verify the response it correct. A valid response is a great indicator that the communica-

tion path is functioning normal.

If the CS4920A fails to respond there is one last option to reset the handler without resetting the chip. If the CS4920A is operating in double-messaging mode the following sequence can be used to flush the SCP input buffer:

0xA1 0xA2 0xA3 0xA1 0xA2 0xA3

The above sequence can be viewed as one command (where the message is 0xA1 0xA2 0xA3). The command byte is an invalid command and it is also assumed the bytes sent will not match any other bytes sent to the CS4920A in another command. Once this sequence is sent, a MSG_PLL_UNLOCK command can be used to verify the communication path as described above. If a valid reply is not received at this point a hardware $\overline{\text{RESET}}$ is recommended.

Finally, the CS4920A does have a means to indicate to the host that a collision of bytes has occurred while receiving bytes from the master. MSG_PLL_UNLOCK is used to enable this feature. If collision checking is enabled, the CS4920A will send a message to the master (MSG_COLLISION) once the collision is detected. This message is particularly useful when operating in SPI mode because there is no hardware mechanism to detect if a byte was not received.

After receiving MSG_COLLISION, the master can attempt to re-send the command which is believed to have failed. (Note the CS4920A will transmit the last valid byte it received before the collision in MSG_COLLISION). It is also recommended that the master verifies the integrity of the communication path by the method described previously.

**Receiving Messages From the CS4920A**

There are several commands that require the CS4920A to respond with a message. Some of the responses are one time replies, while other responses occur at a time dependent on the audio decoding process. In either case, the CS4920A will send NO messages unless the master sends a message first.

The CS4920A uses the $\overline{\text{REQ}}$ signal (pin 3) to indicate to the host that there is a valid byte in the SCPOUT register. This signal is active low. In many designs, $\overline{\text{REQ}}$ is used as an interrupt to the host.

When operating in $I^2C$ mode it may be inconvenient to use the $\overline{\text{REQ}}$ signal for commands which require a simple reply. The use of $\overline{\text{REQ}}$ can be avoided under the following conditions:

1. The SCL/SCK frequency must be less than 400 kHz.
2. The master must delay a minimum of 256 sample periods before attempting to read the reply of a command sent.
3. No features may enabled which allows the CS4920A to send unsolicited messages.

As long as the system adheres to the above constraints, the use of $\overline{\text{REQ}}$ is not required. It cannot be stressed enough that any robust system should be capable of reading messages from the CS4920A.

The master MUST be able to read multiple byte messages in order to read from the CS4920A. In $I^2C$ mode, this means that the master must be able to read at least three bytes of data between START and STOP conditions. In SPI mode, this means that the master must be able to read at least three bytes while $\overline{\text{CS}}$ is active low. Failure to read data in this fashion will result in lost data.

Bytes are sent from the CS4920A in a continuous stream for a given message. If the reply is longer than three bytes, more than three bytes must be read during a given transfer. (A transfer is defined as the movement of data either between START and STOP conditions ($I^2C$) or while $\overline{\text{CS}}$ is low (SPI)).

Figure 3 shows the basic operation of the $\overline{\text{REQ}}$ signal. While the figure does not show an actual three byte read, it does indicate how $\overline{\text{REQ}}$ behaves and

how data bytes are transferred. In the typical case of message replies from the CS4920A $\overline{REQ}$ will remain low until near the end of the third byte.

It is possible for the CS4920A to send more than one message in a transfer. The CS4920A can queue up to 32 three byte messages. However, it should be understood that all the bytes would need to be read during one transfer. In other words, the master should continue to read the SCP until the $\overline{REQ}$ signal goes high.

If unsolicited messages are expected, the use of $\overline{REQ}$ is considered mandatory. Without $\overline{REQ}$, there is no way for the master to know when the SCP has valid data because the unsolicited message could be placed in the port at anytime.

It is important to understand the operation of $\overline{REQ}$ when using unsolicited messages. $\overline{REQ}$ is guaranteed to remain low (once it has gone low) until the second to last rising edge of SCL/SCK of a byte transfer. For $I^2C$, this is the rising edge for the last bit of the byte being transferred. For SPI, this is the rising edge for the second to last bit of the byte being transferred.

$\overline{REQ}$ will come high at this point, if there is no data currently in the SCPOUT register. It is guaranteed to stay high until the next rising edge of SCL/SCK, even if data is written into the SCPOUT register before this time. Therefore, to guarantee a capture of the end of a data transfer ($\overline{REQ}$ going low to high), $\overline{REQ}$ should be sampled on the falling edge immediately following the rising edge of the second to last SCL/SCK for a byte transfer.

The end of a data transfer signals to the host to send a STOP condition or, in the case of SPI, raise $\overline{CS}$. If the master cannot read the state of REQ on the appropriate edge there is still an opportunity to handle the end of data transfer.

It is often the case that the status of $\overline{REQ}$ can only be sampled after the read of a byte is complete. In this case, if $\overline{REQ}$ is still low after the last byte of a message, the master should treat this as a multiple message read. (Note: it is assumed that the frequency of the SCL/SCK signal is less than 900 kHz). The next byte read should be the command byte of the next message. If this byte is a 0x00, then the master knows that $\overline{REQ}$ went high then low again before $\overline{REQ}$ could be sampled. The master would simply discard that byte and continue the message read. The next byte to be read will be the command byte of the next message.

The above is possible for the following reasons. First, a command byte 0x00 is not a valid command byte for the CS4920A, so it will not confuse the message handler on the host. The shift register of the SCP shifts in zeros as the data is being shifted out. Therefore, a 0x00 will be read if another byte is read when there is no valid data to be sent. Second, as long as a STOP condition ($I^2C$) or $\overline{CS}$ high (SPI) is not performed, the CS4920A will remain in read mode. Therefore, when the 0x00 byte is shifted out, the data in the SCPOUT register will be loaded. Thus, the next byte to be read will be the command byte of the next message.

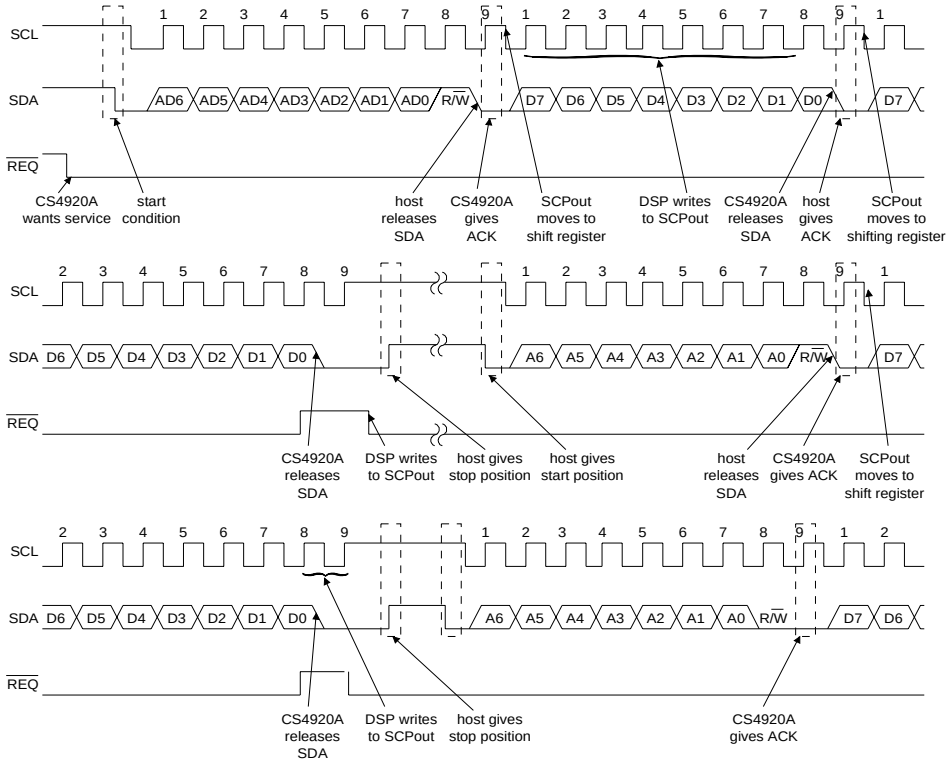



**Figure 3. Operation of REQ in I2C mode**

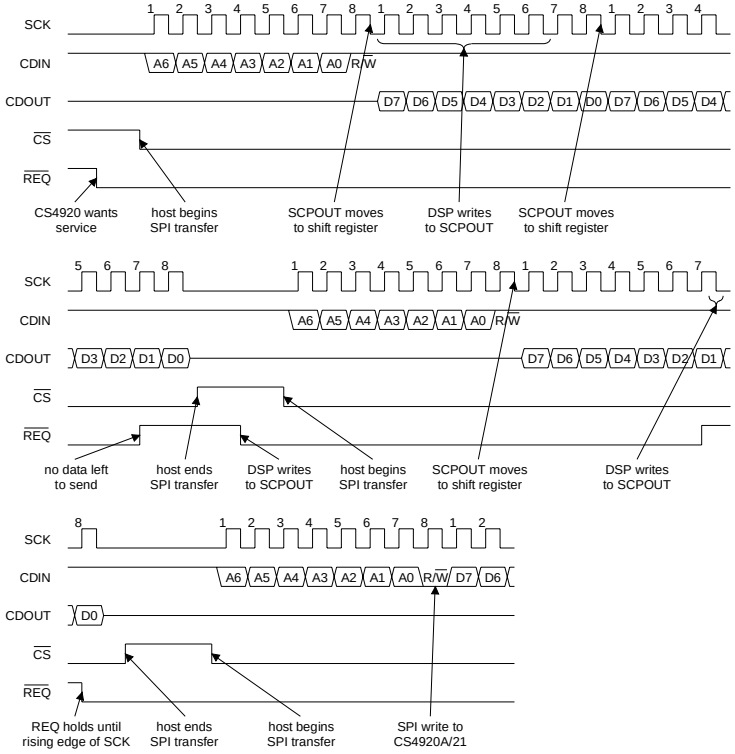



**Figure 3.4. Operation of REQ in SPI mode**

| Message Mnemonic | Command Code | Message Direction | Page # |
|---|---|---|---|
| Initialization | | | |
| MSG_ASI_DATATYPE | 14h | input | 8 |
| MSG_MSG_FORMAT | 15h | input | 9 |
| MSG_CM1_N | 19h | input | 10 |
| MSG_CM1_M | 1Ah | input | 10 |
| MSG_CM0_Q | 1Bh | input | 11 |
| MSG_CM0_CTRL | 1Ch | input | 12 |
| MSG_LINT | 1Eh | input | 13 |
| MSG_ASIC | 1Fh | input | 14 |
| MSG_SCPCN | 20h | input | 15 |
| MSG_XMTCN | 21h | input | 16 |
| MSG_CM1_INIT | 22h | input | 17 |
| MSG_NUM_SYNC | 07h | input | 18 |
| MSG_HFS_SET | 25h | input | 18 |
| Volume Control | | | |
| MSG_VOLL | 00h | input | 19 |
| MSG_VOLR | 01h | input | 19 |
| MSG_MUTE | 02h | input | 20 |
| Status Messages | | | |
| MSG_HEADER | 03h | input w/ response | 21 |
| MSG_PLL_UNLOCK | 08h | input w/ response | 22 |
| MSG_COLLISION | 29h | output | 23 |
| MSG_IO_STATUS | 24h | input w/ response | 24 |
| CRC Control | | | |
| MSG_CRC | 04h | input w/ response | 27 |
| MSG_RESET_CRC | 05h | input | 27 |
| MSG_BYPASS_CRC | 06h | input | 28 |

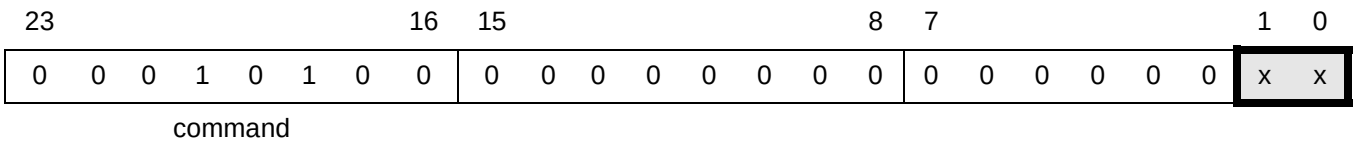
| Message Mnemonic | Command Code | Message Direction | Page # |
|---|---|---|---|
| Control Messages | | | |
| MSG_ADATA | 09h | input w/ response | 29 |
| MSG_POWER_DN | 0Ah | input | 30 |
| MSG_SOFT_RESET | 0Fh | input | 30 |
| MSG_ADATA_EN | 10h | input | 31 |
| MSG_UNSOLICITED_ADATA | 11h | output | 31 |
| MSG_GAME_MODE | 12h | input | 32 |
| MSG_CHANNEL_SWAP | 1Dh | input | 33 |
| MSG_PAUSE | 23h | input | 34 |
| MSG_BEEP | 27h | input w/ response* | 35 |
| MSG_EBS | 28h | input | 36 |
| Data Flow Control | | | |
| MSG_FIFO_THRESHOLD | 0Bh | input | 37 |
| MSG_FIFO_CHK_EN | 0Ch | input | 38 |
| MSG_FIFO_LEVEL | 0Dh | input w/ response | 39 |
| MSG_FIFO_LOW | 0Eh | output | 39 |
| Synchronization Control | | | |
| MSG_PCR_INPUT | 13h | input | 40 |
| MSG_PCR_DISABLE | 16h | input | 41 |
| MSG_PCR_READ | 18h | input w/ response | 42 |
| MSG_PTS_READ | 26h | input w/ response | 43 |

Message: MSG_ASI_DATATYPE

Description: Audio data type to be input to ASI port. The command byte is followed by 14 bits of zero followed by 2 bits which describe the audio data type.

Protocol: The master must send the command byte first, followed by 2 data byte fields containing the audio data type. Refer to "Sending Messages to the CS4920A" on page 1.

The CS4920A MPEG microcode is capable of playing MPEG 1 and MPEG 2 files in audio frames only format and audio packet format.




| bit 1 | bit 0 | | |
|---|---|---|---|
| x | x | = 00 | :MPEG layer I and layer II audio frames |
| x | x | = 01 | :MPEG 2 audio PES packets |
| x | x | = 11 | :MPEG1 audio packets |

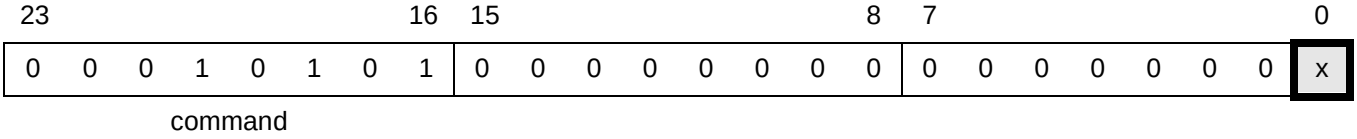
Reply: None.

Default: MPEG layer I and layer II audio frames.

Message: MSG_MSG_FORMAT

Description: MSG_MSG_FORMAT selects between double messaging and single messaging for sending messages to the SCP. The command byte is followed by 15 bits of zero followed by one bit to select the type of messaging. The default condition after any type of reset is double messaging. Therefore the master must send this command twice when selecting single messages. For more information on double messaging refer to "Sending Messages to the CS4920A" on page 1.

Protocol: The master must send the command byte first, followed by two data byte fields containing information to select the type of messaging. Refer to "Sending Messages to the CS4920A" on page 1.




x = 0 :Select double messaging

x = 1 :Select single messaging

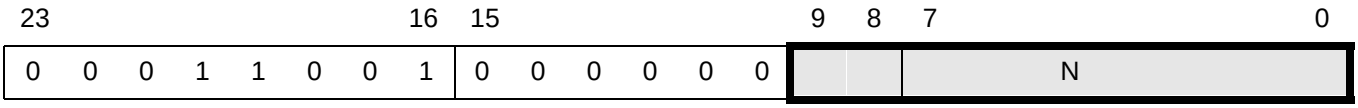
Reply: None.

Default: Double messaging.

Message: MSG_CM1_N

Description: MSG_CM1_N allows the user to change the value of N in CM1 register. The command byte is followed by 6 bits of zero followed by the 10-bit value of N.

Protocol: The master must send the command byte first, followed by 2 data byte fields containing the 10-bit value of N. When using microcode to play MPEG, the M and N values set by MSG_CM1_N and MSG_CM1_M will be overwritten. The value of N will not be written into the CM1 register until the completion of the MSG_CM1_M command. MSG_CM1_M must immediately follow MSG_CM1_N. Refer to "Sending Messages to the CS4920A" on page 1.




bit 0 ... bit 9 = N value of CM1.

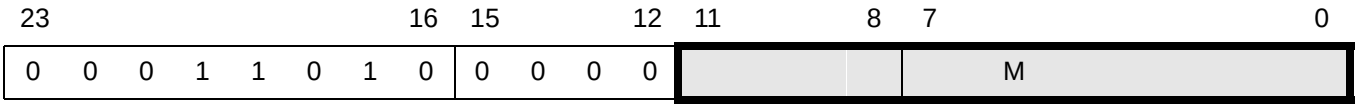
Reply: None.

Default: N/A.

Message: MSG_CM1_M

Description: MSG_CM1_M is used to set the value of M in the CM1 register. The command byte is followed by 4 bits of zero followed by the 12-bit value of M.

Protocol: The master must send the command byte first, followed by 2 data byte fields containing the 12-bit value of M. When using microcode to play MPEG, the M and N values set by MSG_CM1_N and MSG_CM1_M will be overwritten. MSG_CM1_N command must precede the MSG_CM1_M command. The master cannot send any other command before MSG_CM1_M other than MSG_CM1_N. Refer to "Sending Messages to the CS4920A" on page 1.




bit 0 ... bit 15 = M value of CM1.

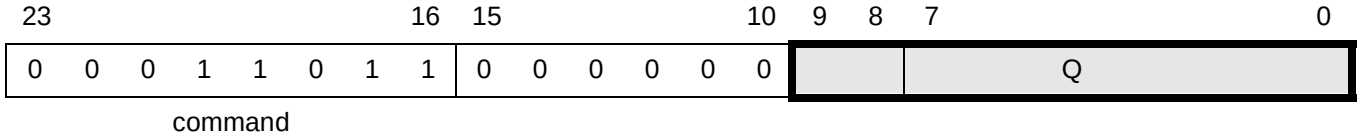
Reply: None.

Default: N/A.

Message: MSG_CM0_Q

Description: MSG_CM0_Q is used for setting the value of Q in the CM0 register. The command byte is followed by 6 bits of zero followed by the 10-bit value of Q. The value of Q is actually 1 less than divide value for CLOCKOUT. For further information about Q value, see the clock manager section of the CS4920A data sheet.

Protocol: The master must send the command byte first, followed by 2 data byte fields containing the 10-bit value of Q. MSG_CM0_Q must be followed by MSG_CM0_CTRL to make MSG_CM0_Q valid. Refer to "Sending Messages to the CS4920A" on page 1.



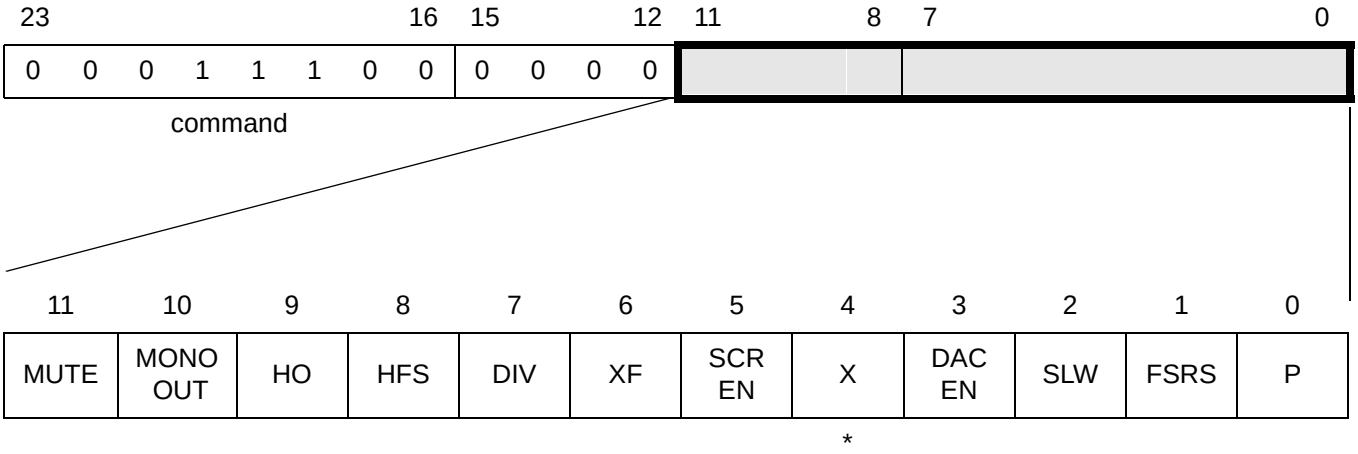

Reply: None.

Default: Q = 1.

Message: MSG_CM0_CTRL

Description: MSG_CM0_CTRL enables the master to write to the upper bits of the CM0 register. The command byte is followed by 4 bits of zero followed by the upper bits of the CM0 register.

Protocol: The master must send the command byte first, followed by 2 data byte fields containing the 12 bits to be written into the upper portion of the CM0 register. When using the MPEG microcode to play MPEG files, values written into the P, DAC EN, HFS, or MONO OUT bits will not remain. The microcode will set P, CAD EN, HGS, and MONO OUT based on the MPEG audio stream. MSG_CM0_CTRL completes the two part message to write to the CM0 register. MSG_CM0_CTRL must be preceded by a MSG_CM0_Q with no other messages coming between the two. Refer to "Sending Messages to the CS4920A" on page 1. When playing MPEG audio, this message may only be sent prior to MSG_CM1_INIT




bit 0 ... bit 11 = CTRL values to be loaded in CM0

*Note: bit 4 is a don't care

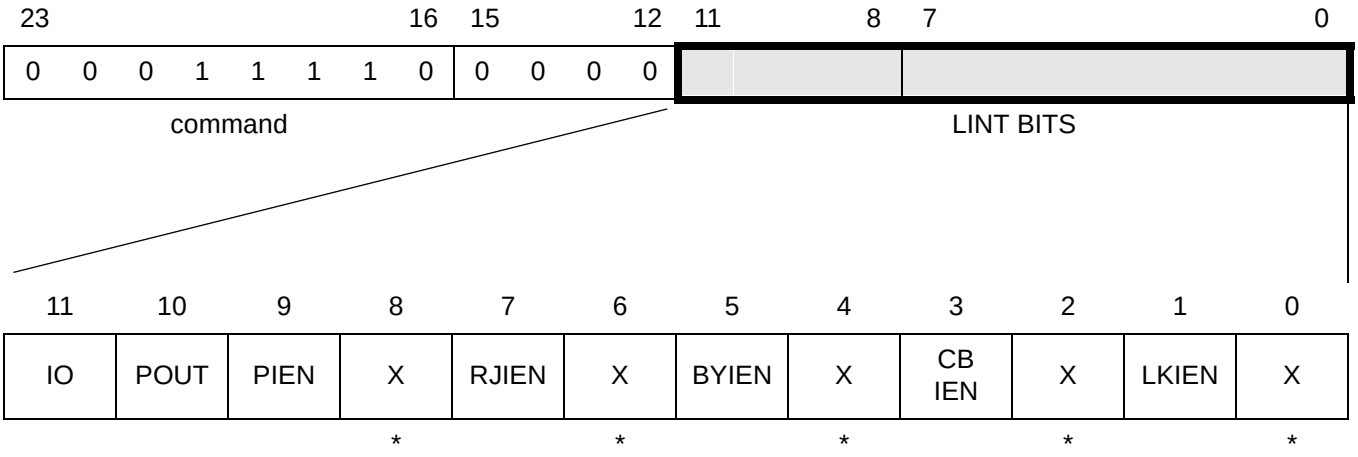
Reply: None.

Default: All bits are zero except DIV = 1.

Message: MSG_LINT

Description: MSG_LINT enables the host to write to the LINT register. The command byte is followed by 4 bits of zero followed by the 12 bits of the LINT

Protocol: The master must send the command byte first, followed by two data byte fields containing information for the LINT register. Refer to "Sending Messages to the CS4920A" on page 1.




bit 0 ... bit 11 = values to be loaded into LINT

*Note: bits 0, 2, 4, 6, and 8 are don't cares.

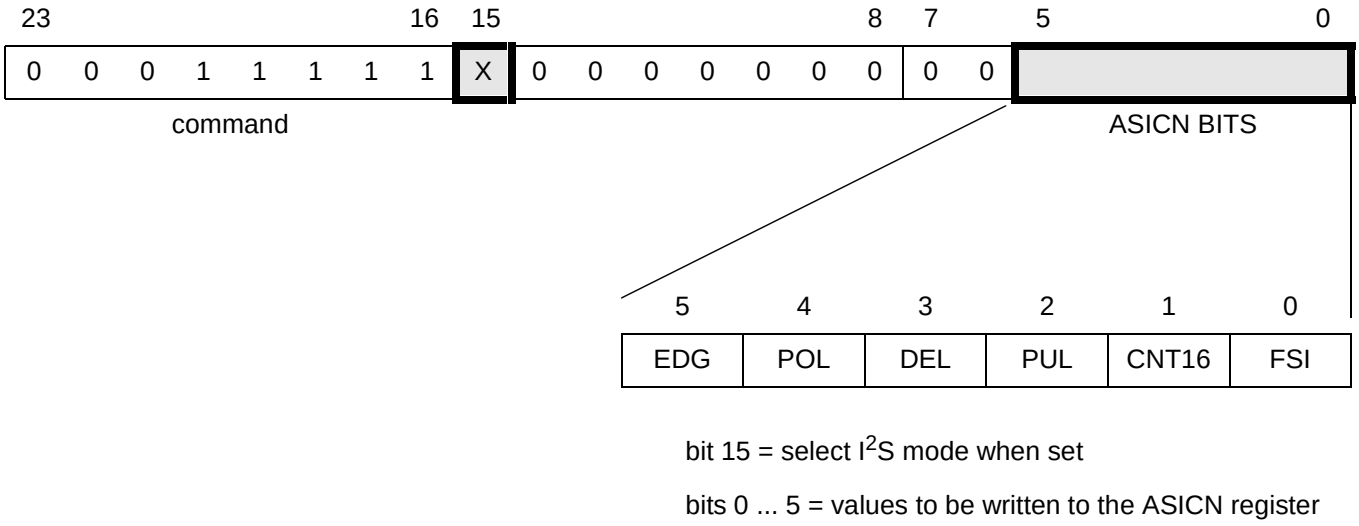
Reply: None.

Default: All bits are zero except LKIEN = 1.

Message: MSG_ASIC

Description: MSG_ASIC enables the host to write to the ASICN register. The command byte is followed by 1 bit to select $I^2S$ mode followed by 9 bits of zero followed by the 6 bits of ASICN.

Protocol: The master must send the command byte first, followed by two data byte fields containing information for the ASICN register and selection of $I^2S$ mode of the ASI port. The default mode of the ASI port is 24-bit, falling edge. Refer to "Sending Messages to the CS4920A" on page 1.



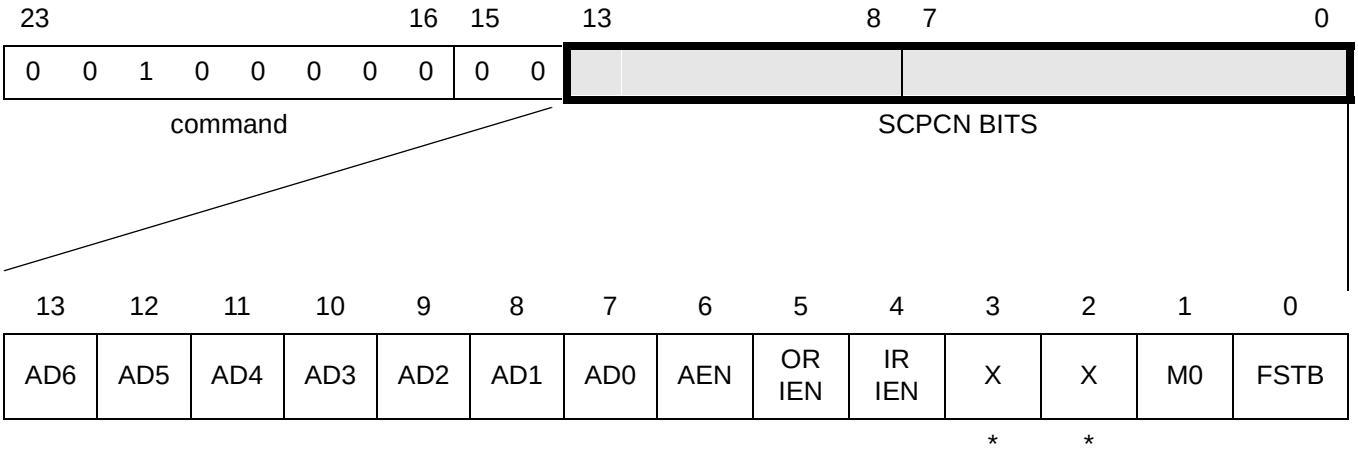

Reply: None.

Default: EDG = 0, POL = 1, DEL = 0, PUL = 1, CNT16 = 0, FSI = 0, and not $I^2S$ mode.

Message: MSG_SCPCN

Description: MSG_SCPCN enables the host to write to the SCPCN register. The command byte is followed by 2 bits of zero followed by the 14 bits of the SCPCN.

Protocol: The master must send the command byte first followed by two data byte fields containing information for the SCPCN register. Refer to "Sending Messages to the CS4920A" on page 1.




bit 0 ... bit 13 = values to be loaded into SPCN

*Note: bits 2 and 3are don't cares.

Reply: None.

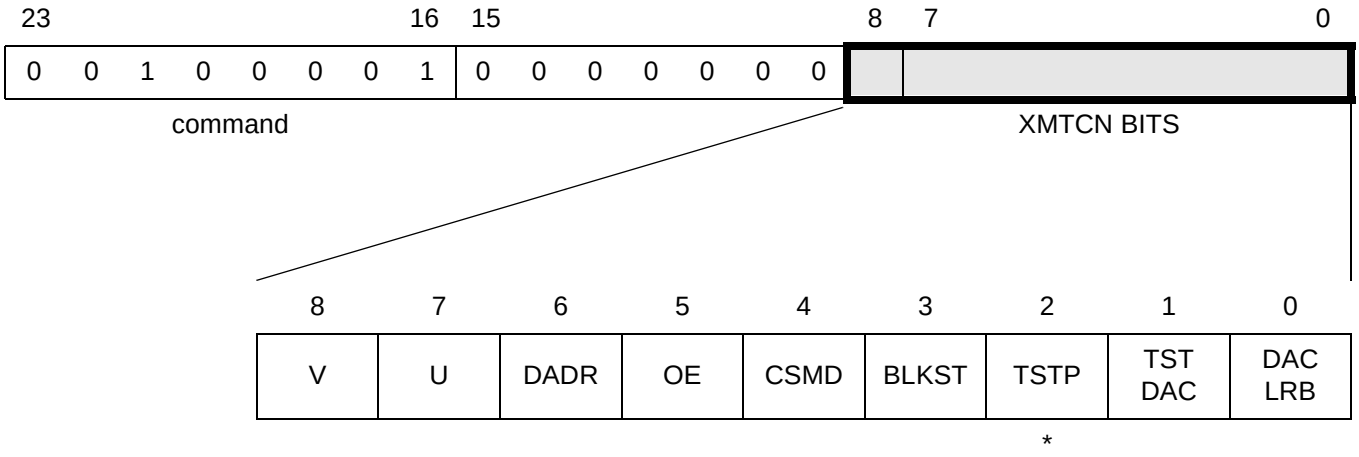
Default: If $I^2C$ AD6..AD0 = 0, AEN = 0, ORIEN = 1, IRIEN = 1, M0 = 0 and FSTB = 1.

If SPI AD6..AD0 = 0, AEN = 0, ORIEN = 1, IRIEN = 1, M0 = 1 and FSTB = 1.

Message: MSG_XMTCN

Description: MSG_XMTCN enables the host to write to the XMTCN register. The command byte is followed by 7 bits of zero followed by the 9 bits of the XMTCN.

Protocol: The master must send the command byte first, followed by tow data byte fields containing information for the XMTCN register. Refer to "Sending Messages to the CS4920A" on page 1.




bit 0 .. bit 8 = values to be loaded in XMTCN

*Note: bit 2 must be a zero for the CS4920A to operate

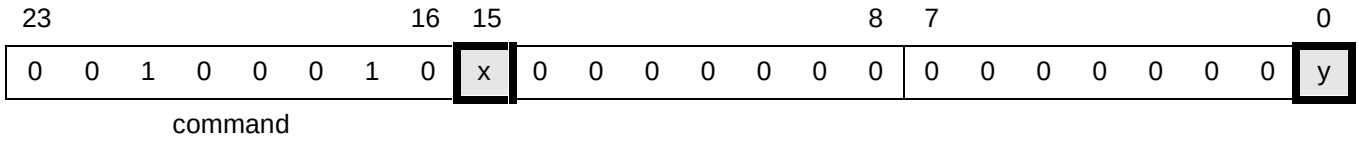
Reply: None.

Default: All bits are zero except DADR = 1 and OE = 1.

Message: MSG_CM1_INIT

Description: Initialize the CM1 register values to be used for MPEG audio. The command byte is followed by 15 bits of zero followed by 1 bit to select default values. When default values are not wanted the command continues after the 1 bit to select default values with 18 more bytes. The 18 following bytes contain the values to be loaded into a CM1 register look-up table.

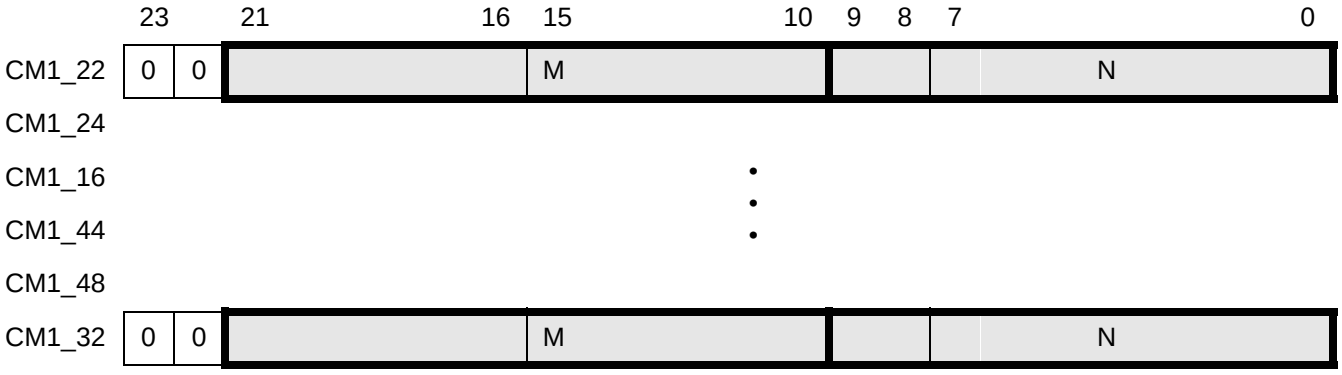
Protocol: The master must send the command byte first, followed by two data byte fields containing information on which values to use for a CM1 register look-up table. When default values are selected, (CM1 values when clkin = 27 MHz) the master is finished sending the command. When the default values are not selected the command continues with 6 more messages. Each of the 6 messages start with 2 bits of zero followed by the 12-bit value of M followed by the 10-bit value of N. The first of these 6 messages contain the CM1 register information. For Fs = 22.050 kHz, the second message for Fs = 24 kHz, the third message for Fs = 16 kHz, the fourth message for Fs = 44.1 kHz, the fifth message for Fs = 48 kHz, and the sixth message for Fs = 32 kHz. Refer to "Sending Messages to the CS4920A" on page 1.




If x = 1, 4920A will use the HO bit for Fs changes.

If y = 0, 4920A will use default values for clkin = 27 MHz

If y = 1, the following data should follow




Note the order is critical. The CM1 register value for Fs = 22 kHz must immediately follow the CM1_INIT message. The others should follow, maintaining the order depicted above, finishing with the CM1 register value for Fs = 32 kHz. This message is mandatory for decoding to begin.

Reply: None.

Default: N/A. This command must be sent.

Message: MSG_NUM_SYNC

Description: Number of SYNC words. The command is followed by two bytes containing the number of SYNC words. This message is used to set the number of SYNC words which must be found in an MPEG audio stream to establish an in frame condition.

Protocol: The master must send the command byte first, followed by two data bytes containing the SYNC word count. Refer to "Sending Messages to the CS4920A" on page 1.

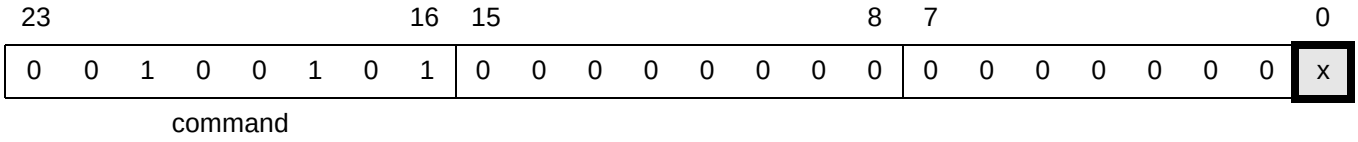

Reply: None.

Default: Number of SYNC words is set to 2

Message: MSG_HFS_SET

Description: MSG_HFS_SET

Protocol: The master must send the command byte first, followed by two data byte fields containing information to set or clear the HFS bit. Refer to "Sending Messages to the CS4920A" on page 1.




x = 0 :Clear HFS

x = 1 :Set HFS

Reply: None.

Default: HFS is cleared.

Warning: Setting HFS will double the DSP speed. Caution must be taken not to run the DSP over 25 MHz.

Message: MSG_VOLL

Description: Left channel volume attenuation. The command is followed by a 16 bit value ranging from 0000h to 8000h. A value of 8000h is the full scale value. A value of 0000h is zero volume. Values above 8000h will be ignored.

Protocol: The master must send the command byte first, followed by 2 data byte fields containing the 16 bit volume attenuation value, most significant bit first. Refer to "Sending Messages to the CS4920A" on page 1.


Reply: None.

Default: Full scale.

Message: MSG_VOLR

Description: Right channel volume attenuation. The command is followed by a 16 bit value ranging from 0000h to 8000h. A value of 8000h is the full scale value. A value of 0000h is zero volume. Values above 8000h will be ignored.

Protocol: The master must send the command byte first, followed by 2 data byte fields containing the 16 bit volume attenuation value, most significant bit first. Refer to "Sending Messages to the CS4920A" on page 1.

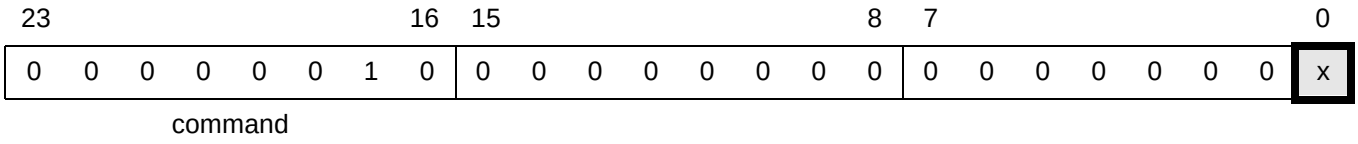

Reply: None.

Default: Full scale.

Message: MSG_MUTE

Description: Mute control of DACs. The command is followed two bytes containing control information. This message is used to activate or mute the DAC outputs.

Protocol: The master must send the command byte first, followed by two data byte fields containing the control information. Refer to "Sending Messages to the CS4920A" on page 1.




x = 0 :Enable DACs

x = 1 :Mute

Reply: None.

Default: Mute off.

Message: MSG_HEADER

Description: MPEG header information. The command is followed two bytes containing request information. This message is used retrieve present MPEG header information.

Protocol: The master must send the command byte first, followed by two data byte fields containing the request information. The CS4920A's reply will depend upon the information requested. The values returned in the data bytes correspond directly to the bit representation and meaning as defined in the ISO/IEC 11172-3 section 2.4.2.3. All data information returned will be right aligned in the data fields. All unused bits will be zero. Refer to both "Sending Message to the CS4920A" and "Receiving Messages from the CS4920A"on pages 1 and 3.

| 23 | | | | | | | 16 | 15 | | | | | | | 8 | 7 | | | 4 | 3 | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | x | x | x |
| command | | | | | | | | | | | | | | | | | | | | | | | |

| bit 3 | bit 2 | bit 1 | bit 0 | | |
|---|---|---|---|---|---|
| x | x | x | x | =0000 | :MPEG ID |
| x | x | x | x | =0001 | :Layer Information |
| x | x | x | x | =0010 | :Protection Bit |
| x | x | x | x | =0011 | :Bit Rate |
| x | x | x | x | =0100 | :Sample Frequency |
| x | x | x | x | =0101 | :Padding Bit |
| x | x | x | x | =0110 | :Private Bit |
| x | x | x | x | =0111 | :Mode Bit |
| x | x | x | x | =1000 | :Mode Extension Bit |
| x | x | x | x | =1001 | :Copyright Bit |
| x | x | x | x | =1010 | :Original/Home |
| x | x | x | x | =1011 | :Emphasis |

Reply:

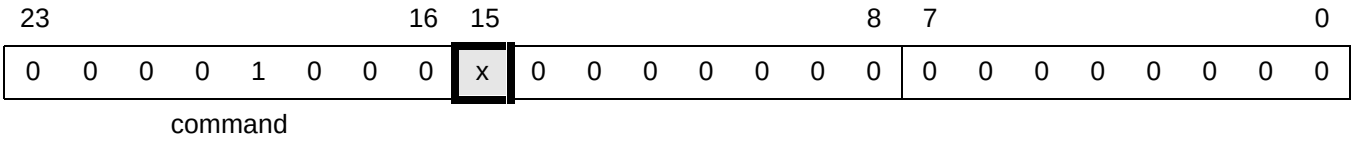
| 23 | | | | | | | 16 | 15 | | | | | | | 8 | 7 | | | 4 | 3 | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| command | | | | | | | | | | | | | | | | | | | | | | | |

Default: N/A.

Message: MSG_PLL_UNLOCK

Description: Phase locked loop unlock condition. The command is followed by two don't care data bytes. This message's primary purpose is to check for an unlock condition in the clock manager.

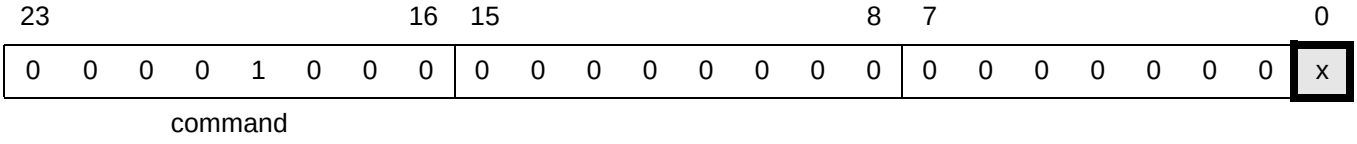
Protocol: The master must send the command byte first, followed by two zero bytes. The CS4920A will return the command byte followed by two data bytes. The least significant bit of the second data byte indicates the condition of the PLL. A zero means the PLL is locked and a one means the PLL is unlocked. The message also allows the host to enable/ disable the unsolicited MSG_COLLISION (command byte 0x29) message. Refer to both "Sending to the CS4920A" and "Receiving Messages from the CS4920A" on pages 1 and 3.




x = 0 :Disable unsolicited MSG_COLLISION

x = 1 :Enable unsolicited MSG_COLLISION

Reply:




x = 0 :PLL locked

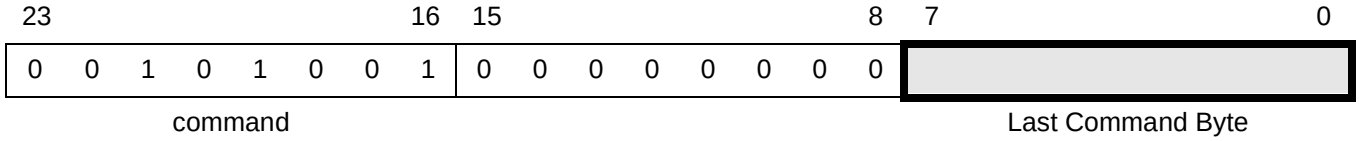
x = 1 :PLL unlocked

Default: N/A.

Message: MSG_COLLISION

Description: Unsolicited response message of the CS4920A when an overflow of the input to SCP interface has occurred. MSG_COLLISION is an enabled feature via MSG_PLL_UNLOCK command. The unsolicited response is the command byte followed a don't care byte followed by the command byte of the message that was overwritten.

Protocol: The host will respond to the CS4920A's request for a read operation by reading the command byte followed by a zero byte followed by the command byte of the message that was overwritten in the SCP. Refer to "Receiving Messages from the CS4920A" on page 1.



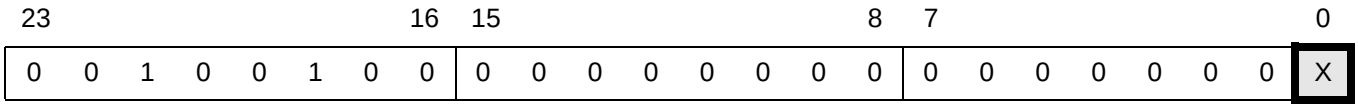

Reply: This message is an output from the CS4920A, and therefore there is no reply.

Default: Unsolicited message is not activated.

Message: MSG_IO_STATUS

Description: Return current status of ASIC, SCPCN, CM0, CM1, XMTCN, and LINT registers or the status of the MPEG microcode's internal control flags. The command byte is followed by 15 bits of zero followed by a select bit for IO registers or control flags. The reply is six messages (18 bytes) for the IO registers or three messages (9 bytes) for the control flags which both start with the command byte followed by the bits of the ASIC register (right justified). The IO registers reply will continue after the ASIC with SCPCN register message (left justified). The Clock Manager registers follow with CM0 and CM1 respectively (both are right justified). The remaining two messages are the XMTCN register (left justified) and the LINT register. The Silicon Revision are in the 4 most significant bits and the 12 least significant bits are the LINT.The control flags reply will continue after the ASIC with six more bytes of flag information. See the flag description section below for details about each flag. The majority of flags are one bit fields.

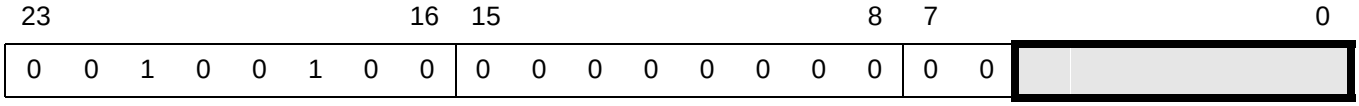
Protocol: The master must send the command byte first, followed 15 bits of zero followed by a select bit. Refer to "Sending Messages to the CS4920A" and "Receiving Messages from the CS4920A" on pages 1 and 3.



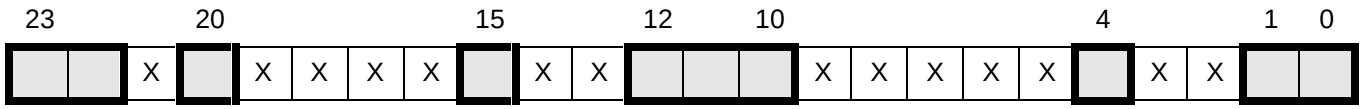

x = 0 :IO_Registers

x = 1 :Control Flags

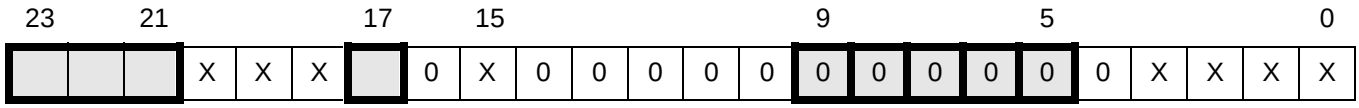
The 9 byte reply for Control Flags:




Control Register 1:




Control Register 2:




X = Don't Care

MSG_IO_STATUS (continued)

Control flags 1 bit definition:

| | |
|---|---|
| bit 23: | $I^2S$ mode enabled |
| bit 22: | Audio Frame Lock |
| bit 20: | hardware handshaking mode via XF pin enabled |
| bit 15: | A/V synchronization enabled |
| bit 12&11: | 00 = MPEG audio frames mode; |
| | 01 = MPEG2 PES data mode; |
| | 11 MPEG1 Packet mode |
| bit 10: | Single message mode enabled |
| bit 6: | Game mode enabled |
| bit 4: | CM1 values are initialized |
| bit 1: | Ancillary data is currently being extracted |
| bit 0: | Ancillary data unsolicited messages enabled |

Control flags 2 bit definition:

| | |
|---|---|
| bit 23: | Pause mode enabled |
| bit 22&21: | output channel swap mode |
| | 00 - no switch; 01 - switch Left and Right |
| | 10 - Left to both; 11 - Right to both |
| bit 17: | Unsolicited collision message enabled |
| bit 9: | EBS is on during channel change |
| bit 8: | EBS in progress |
| bit 7: | Channel change procedure in progress |
| bit 6: | PTS and PCR out of sync |
| bit 5: | Beep in progress |

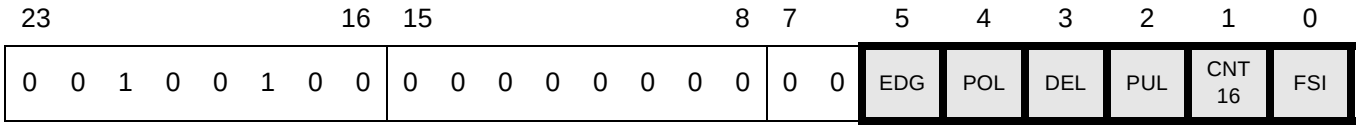
(when not stated bit = 1 implies statement is True)

Default for Control Flags: all flags are zero

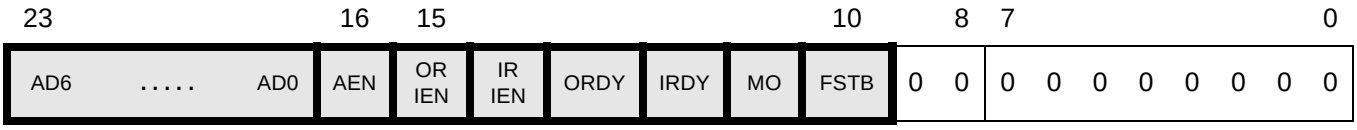
MSG_IO_STATUS (continued)

The 18 byte reply for IO Registers:

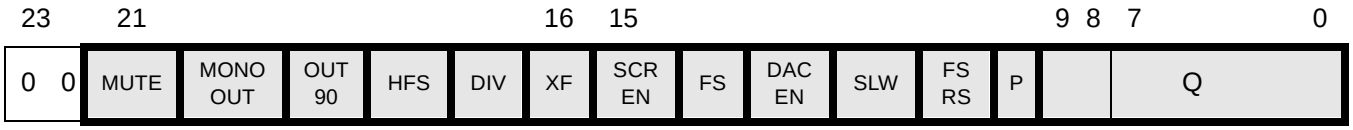
ASIC




SCPCN

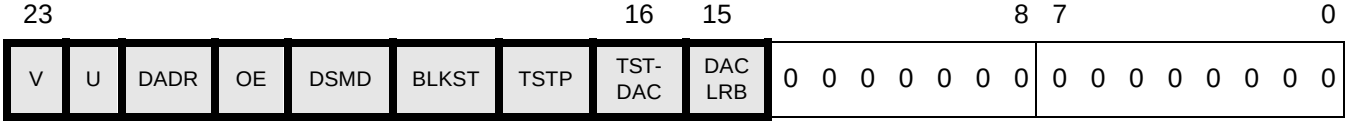



CM0

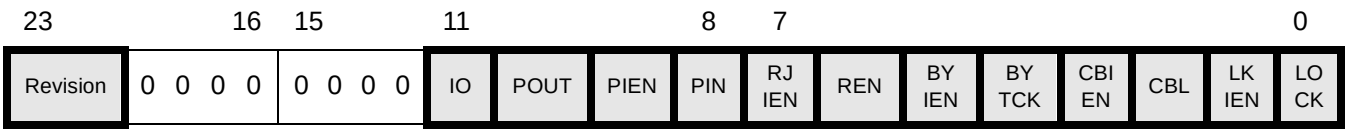



CM1


XMTCN




LINT




Default for IO registers:N/A.

Message: MSG_CRC

Description: Cyclical Redundancy Code counter. The command is followed two don't care data bytes. This message is used to determine the amount of CRC errors that have occurred over a given period.

Protocol: The master must send the command byte first, followed by two don't care data bytes. The CS4920A will return the command byte followed by a 16 bit, right aligned, unsigned value which indicates the amount of CRC errors since the last reset of the counter. Refer to both "Sending Messages to the CS4920A" and "Receiving Messages from the CS4920A" on pages 1 and 3.

| 23 | | | | | | | 16 | 15 | | | | | | | 8 | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

command

Reply:

| 23 | | | | | | | 16 | 15 | | | | | | | 8 | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | | | | | | | | | | | | | | | | |

Default: N/A.

Message: MSG_RESET_CRC

Description: Reset Cyclical Redundancy Code counter. The command is followed two don't care data bytes. This message is used to reset the CRC error counter to zero.

Protocol: The master must send the command byte first, followed by two don't care data bytes. Refer to "Sending Messages to the CS4920A" on page 1.

| 23 | | | | | | | 16 | 15 | | | | | | | 8 | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

command

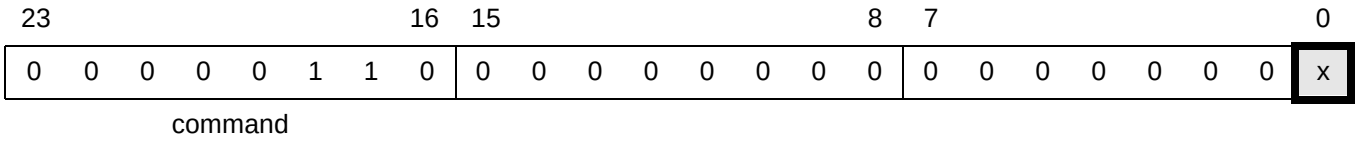
Reply None.

Default: N/A.

Message: MSG_BYPASS_CRC

Description: Bypass Cyclical Redundancy Code check. The command is followed by two bytes containing control information. This message is used to enable/disable the CRC check.

Protocol: The master must send the command byte first, followed by two data bytes containing the control information. Refer to "Sending Messages to the CS4920A" on page 1.




x = 0 :Enable CRC checking

x = 1 :Bypass CRC checking

Reply: None.

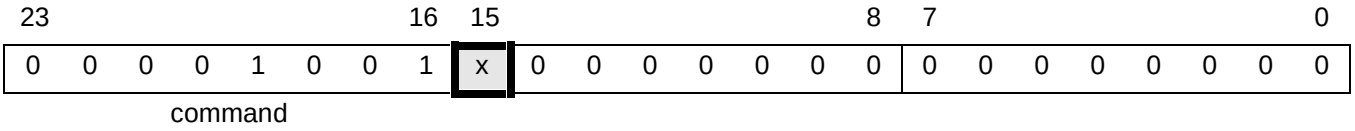
Default: Bypass CRC checking.

Message: MSG_ADATA

Description: Ancillary data. The command is followed by two don't care data bytes. This message is used to retrieve ancillary data stored in a 48 byte FIFO buffer in the CS4920A. The data must be retrieved periodically to ensure no data is lost.

Protocol: The master must send the command byte first, followed by two data bytes which determine the format of the reply. By setting bit 15 of the command message to 1, the CS4920A response message will contain the number of ancillary data bits extracted from the audio frame in the reply message data field. When bit 15 of the command message is zero, the CS4920A will respond with the number of ancillary data words. The reply from the CS4920A will be the command byte followed by two data bytes which will contain the information about the amount of the ancillary data and if the read was valid. The ancillary data words will immediately follow this first message. The master must clock out of the SCP the number of data words determined from the data field of the first message.

If user requested a response in # of words then the amount of ancillary data is a 14-bit unsigned value between 0 and 16. If user requested response in # of bits then the amount of ancillary data is a 14-bit unsigned value between 0 and 384. In either case the CS4920A must transfer data on a word boundary. If the # of bits is not an integer multiple of 24 then the last word transferred will have the valid data bits left aligned. Refer to both "Sending Messages to the CS4920A" and "Receiving Messages from the CS4920A" on pages 1 and 3.

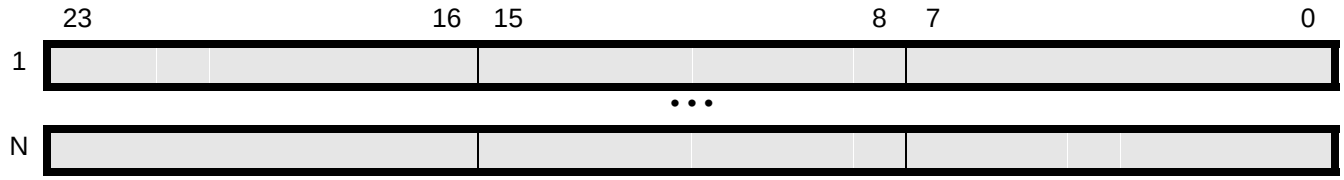



x = 0 :CS4920A should respond in # of words
x = 1 :CS4920A should respond with # of bits

Reply:


x = 0 :valid read
x = 1 :error (DATA transfer in progress)

Ancillary data:




N - determined from first message (amount of ancillary data).

Default: N/A.

Message: MSG_POWER_DN

Description: Power Down. The command is followed by two don't care data bytes. This message is used to place the CS4920A in power down mode. A hardware reset is required to restart the chip.

Protocol: The master must send the command byte first, followed by two don't care data bytes. Refer to "Sending Messages to the CS4920A" on page 1.

| 23 | | | | | | | 16 | 15 | | | | | | | 8 | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | command | | | | | | | | | | | | | | | | | | | | |

Reply: None.

Default: N/A.

Message: MSG_SOFT_ RESET

Description: Soft ware reset. The command is followed by two don't care data bytes. This message is used to issue an internal software reset.

Protocol: The master must send the command byte first, followed by the two don't care data bytes. A delay of 2 ms will be necessary for the CS4920A to acquire phase lock with CLKIN. After the delay, normal operation may resume. The software reset message will clear all registers and reset all pointers to buffers. Refer to "Sending Messages to the CS4920A" on page 1.

| 23 | | | | | | | 16 | 15 | | | | | | | 8 | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | command | | | | | | | | | | | | | | | | | | | | |

Reply: None.

Default: N/A.

Message: MSG_ADATA_EN

Description: Unsolicited ancillary data enable. This command is used to enable the CS4920A to send a MSG_UNSOLICITED_ADATA when new ancillary is available. The command is followed by two control bytes. The LSB of the second byte must be sent to logic one to enable this feature. The default state is disabled. This feature, once enabled, will stay enabled until a command is sent to disable it, or a hardware or software reset is issued.

Protocol: The master must send the command byte first, followed by the two control bytes. Refer to "Sending Messages to the CS4920A" on page 1.

| 23 | | | | | | | 16 | 15 | | | | | | | 8 | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x |
| | | | command | | | | | | | | | | | | | | | | | | | | |

x = 0 :Disable

x = 1 :Enable

Reply: None.

Default: Disabled

.

Message: MSG_UNSOLICITED_ADATA

Description: Unsolicited ancillary data. This command is unsolicited by the master. When the CS4920A reaches the end of the audio data in a particular MPEG audio frame, it will extract up to the first 48 bytes immediately following the end of this data. It will then send this message to the master, if the MSG_ADATA_EN has enabled this feature. The master should respond to this $\overline{\text{REQ}}$. Once the master receives this unsolicited message it should respond with a MSG_ADATA to the CS4920A requesting the ancillary data. The MSG_UNSOLICITED_ADATA, if enabled, will be sent once per frame.

Protocol: The master must read the command byte, followed by two don't care data bytes. Refer to "Receiving Messages from the CS4920A" on page 1

| 23 | | | | | | | 16 | 15 | | | | | | | 8 | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | | | command | | | | | | | | | | | | | | | | | | | | |

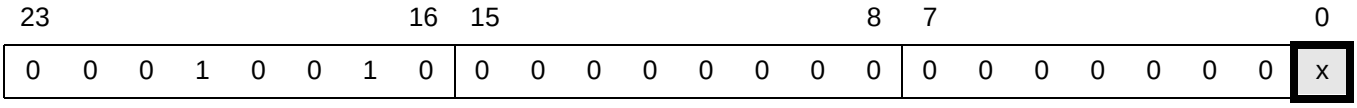
Reply: This message is an output from the CS4920A, and therefore there is not a reply.

Default: N/A.

Message: MSG_GAME_MODE

Description: Game mode. This command is followed by two control bytes. This command is used to bypass the ancillary data channel. Once a particular frame has been extracted the next sync word, as defined by ISO 11172-3, is searched for by the algorithm. The LSB of this message must be set to a logic one to enable this feature. The default state is disabled. This feature, once enabled, will remain enabled until either a hardware/software reset, or a disable message is issued. Note this mode will cause skips in audio playback if the ancillary data channel has a valid sync word. This mode is only necessary if the encoded data does not follow the ISO 11172 definition of the padding bit during playback of 44.1 kHz.

Protocol: The master must send the command byte first, followed by the two control bytes. Refer to "Sending Messages to the CS4920A" on page 1.




x = 0 :Disable

x = 1 :Enable

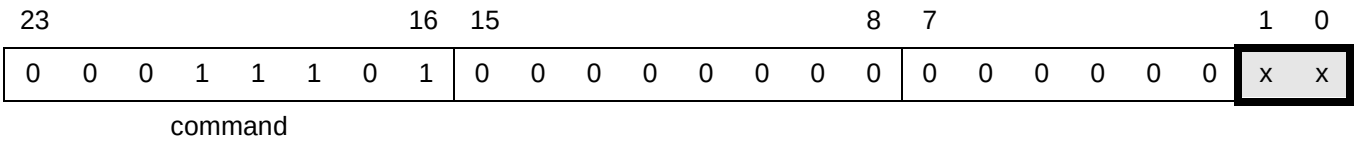
Reply: None.

Default: Disabled.

Message: MSG_CHANNEL_SWAP

Description: Swap the left and right channels, play the left channel out of both DACs, or play the right channel out of both DACs. The command byte is followed by 14 bits of zero followed by 2 bits to select channel output mode. This message is not valid for PCM mode.

Protocol: The master must send the command byte first, followed by 2 data byte fields containing the 2 bits to select channel output mode. Refer to "Sending Messages to the CS4920A" on page 1.




| bit 1 | bit 0 | | |
|---|---|---|---|
| x | x | = 00 | :Normal playback |
| x | x | = 01 | :Switch left and right |
| x | x | = 10 | :Left on both |
| x | x | = 11 | :Right on both |

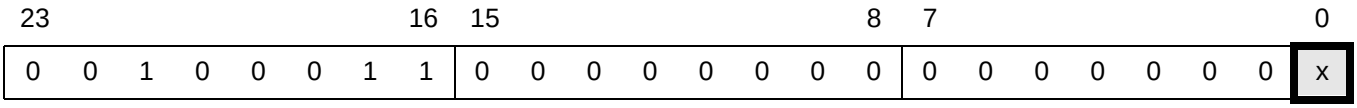
Reply: None.

Default: Normal playback.

Message: MSG_PAUSE

Description: Stop and start the decoding process of an MPEG stream. The command byte is followed by 15 bits of zero followed by 1 bit to select pause or play. In pause mode, the 33-bit PCR counter is internally read and stored while the audio processor is placed into a wait state. The 33-bit PCR counter can be loaded using MSG_PCR_INPUT (see page 40) while in pause mode. When the part is placed in play mode, the PCR counter is updated (if the counter is enabled).

Protocol: The master must send the command byte first, followed by 2 data byte fields containing the bit to select pause or play. Refer to "Sending Messages to the CS4920A" on page 1.




x = 0 :Play

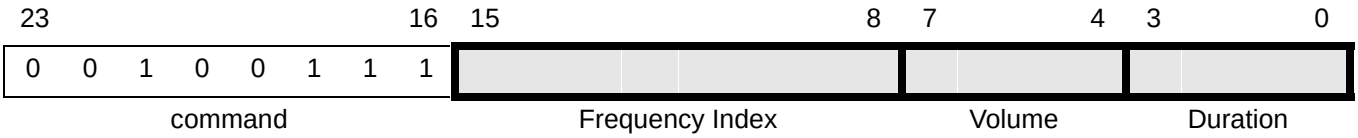
x = 1 :Pause

Default: Play.

Message: MSG_BEEP

Description: Beep generator message. The command is followed by two bytes containing frequency, volume, and duration information. The message is used to create single frequency tones while decoding MPEG audio. The beep will be played in place of the decoded MPEG audio.

Protocol: The master must send the command byte first, followed by two data byte fields containing, the frequency, volume, and duration information. The first data byte contains the frequency selection information. The frequency selection (0x00 - 0x17) is based on the table given below. The second data byte contains both the volume and duration information. The first nibble of the second byte is the volume (0x0 - 0xF) and the second nibble is the duration (0x0 - 0xF). After the completion of the beep, the CS4920A will respond with a 24 bit message containing the command byte (0x27) followed by 16 bits of zero. This response message signals that the CS4920A can accept a new beep command. Refer to both "Sending Message to the CS4920A" and "Receiving Messages from the CS4920A" on pages 1 and 3.




Frequency Selections: (0x00-0x17)

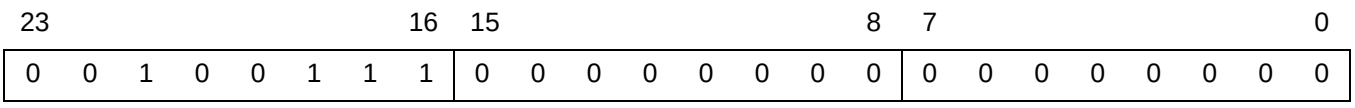
| | | | | | |
|---|---|---|---|---|---|
| Index | 0x00 | 0x01 | 0x02 | 0x03 | 0x04 |
| Tone (Hz) | 220 | 233 | 247 | 262 | 277 |
| Index | 0x05 | 0x06 | 0x07 | 0x08 | 0x09 |
| Tone (Hz) | 294 | 311 | 330 | 349 | 370 |
| Index | 0x0A | 0x0B | 0x0C | 0x0D | 0x0E |
| Tone (Hz) | 392 | 415 | 440 | 466 | 494 |
| Index | 0x0F | 0x10 | 0x11 | 0x12 | 0x13 |
| Tone (Hz) | 523 | 554 | 587 | 622 | 659 |
| Index | 0x14 | 0x15 | 0x16 | 0x17 | |
| Tone (Hz) | 698 | 740 | 784 | 831 | |

Volume: (0x0-0xF)

0x0 = -6 dB Full Scale
0x1 = -9 dB Full Scale
0x2 = -12 dB Full Scale
• • •
0xF = -51 dB Full Scale

Duration: (0x0-0xF) Selections in 100 ms intervals from 0-1.5 seconds

Reply:

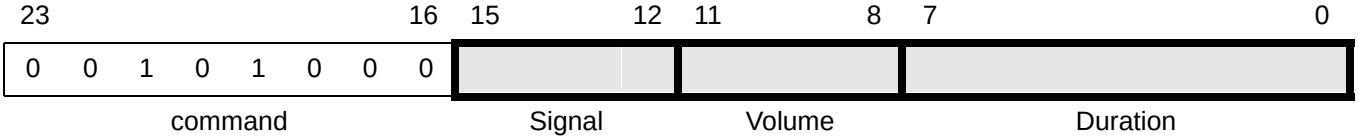



Default: Beep off.

Message: MSG_EBS

Description: Emergency Broadcast Signal generator. The command is followed by two data bytes to select signal, volume, and duration.

Protocol: The master must send the command byte first, followed by two data byte fields containing information about the signal, volume, and duration. The first byte after the command byte contains signal and volume selection information. The first nibble of this first data byte selects the type of signal to be generated. There is currently only one type of signal: summation of a 853 Hz and a 960 Hz tone. Additional selections may be added later. The second nibble contains volume information. The second data byte contains duration. Refer to “Sending Messages to the CS4920A” on page 1.




Signal: Currently must always be set to 0x0

Volume: (0x0-0xF)

0x0 = 0 dB Full Scale

0x1 = -3 dB Full Scale

0x2 = -6 dB Full Scale

• • •

0xF = -45 dB Full Scale

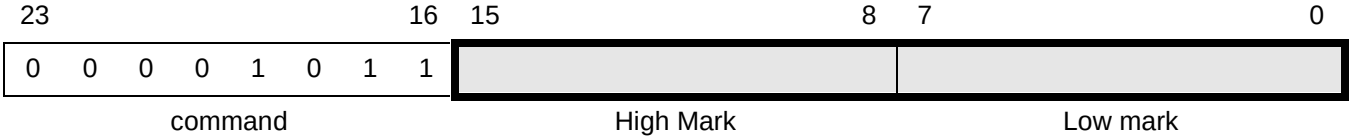
Duration: (0x0-0xAE) Selections in 1 second intervals from 0-174 seconds

Default: EBS tone off.

Message: MSG_FIFO_THRESHOLD

Description: Input FIFO threshold level setting. The command byte is followed by a High Mark setting (when using the XF pin to transfer audio data) followed by a Low Mark setting. Low Mark and High Mark are 8 bit unsigned values and are in units of words (1 word = 3 bytes = 24 bits). The Low Mark defaults to 128 while the High Mark defaults to zero. When not using the High Mark, set High Mark to less than or equal to the Low Mark. A High Mark of greater than 250 words is not recommended.

Protocol: The master must send the command byte first, followed by 2 data byte fields containing the 8-bit unsigned values of High Mark and Low Mark. Refer to "Sending Messages to the CS4920A" on page 1.



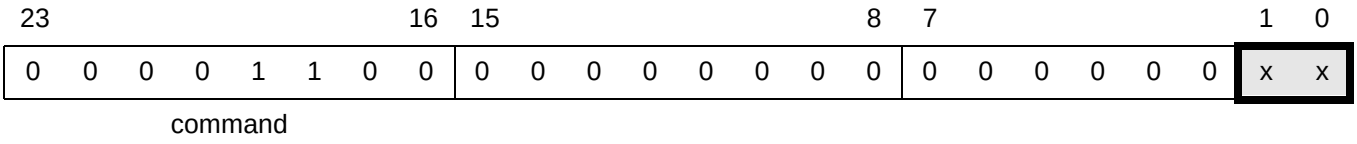

Reply: None.

Default: High Mark = 0, Low Mark = 128.

Message: MSG_FIFO_CHECK_EN

Description: Input FIFO threshold check enable. The command byte is followed by 14 bits of zero followed by 2 bits to select/deselect either FIFO checking that uses MSG_FIFO_LOW or uses the XF pin. This command must be sent every time the MSG_FIFO_LOW command is received to re-enable FIFO checking. This command only needs to be sent once to enable FIFO checking when using the XF pin to transfer audio data.

Protocol: The master must send the command byte first, followed by 2 data byte fields containing 2 bits to enable or disable input FIFO checking. Refer to "Sending Messages to the CS4920A" on page 1.




| bit 1 | bit 0 | | |
|---|---|---|---|
| x | x | = 00 | :Disable FIFO checking |
| x | x | = 01 | :Enable FIFO checking with message MSG_FIFO_LOW |
| x | x | = 10 | :Disable FIFO checking |
| x | x | = 11 | :Enable FIFO checking with XF usage |

Reply: None.

Default: Disabled.

Message: MSG_FIFO_LEVEL

Description: Input FIFO level. The command is followed by two don't care data bytes. This message is used to retrieve the current level, in number of words, in the input FIFO. (1 word = 24 bits)

Protocol: The master must send the command byte first, followed by two don't care data bytes. The CS4920A will return the command byte followed by two data bytes containing a 16 bit unsigned value of the number of words currently in the input FIFO. Note the MSB of the value immediately follows the command byte. Refer to both "Sending Messages to the CS4920A" and "Receiving Messages from the CS4920A" on pages 1 and 3.

| 23 | | | | | | | 16 | 15 | | | | | | | 8 | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

command

Reply:

| 23 | | | | | | | 16 | 15 | | | | | | | 8 | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | | | | | | | | | | | | | | | | |

command — 16 bit unsigned value of the number of words in the input FIFO

Default: N/A

Message: MSG_FIFO_LOW

Description: FIFO low. This command is unsolicited by the master. When the CS4920A's input FIFO falls below the threshold level set by MSG_FIFO_THRESHOLD and if the FIFO check is enabled by the MSG_FIFO_CHK message, then the CS4920A will request to send this message. Once this message is sent, the master must send the FIFO_CHK_EN message to re-enable the FIFO check. Refer to "Receiving Messages from the CS4920A" on page 1.

Protocol: The master must read the command byte, followed by two don't care bytes.

| 23 | | | | | | | 16 | 15 | | | | | | | 8 | 7 | | | | | | | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

command

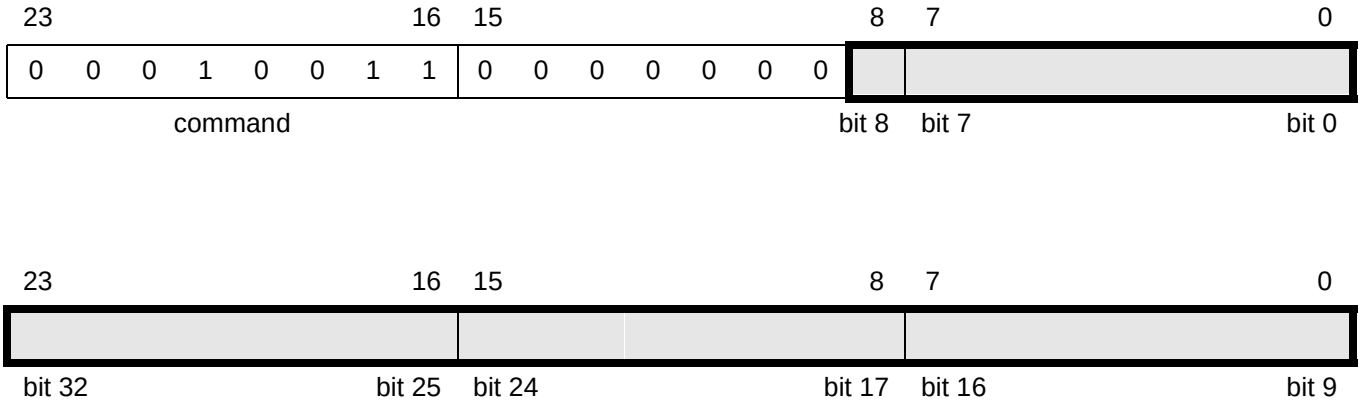
Reply: This message is an output from the CS4920A, and therefore there is not a reply.

Default: N/A.

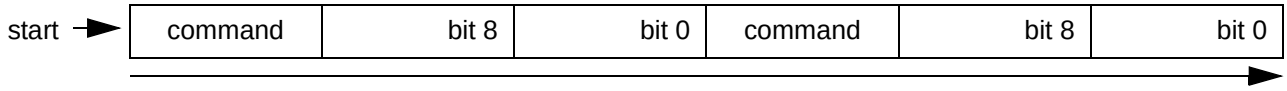
Message: MSG_PCR_INPUT

Description: Message to send PCR 33 bit value. This command is unusual in that it will be sent as two messages. The first message contains the lower bits of the PCR, the second message contains the upper bits of the PCR. Sending this message enables the PCR/PTS synchronization feature. See MSG_PCR_EN to disable the PCR/PTS synchronization feature.

Protocol: The master must send the command byte first. Bit 8 of the message definition corresponds to bit 8 as defined in the ISO/IEC 13818-1 specification for the PCR. Bit 32 is the most significant bit of the 33 bit PCR and bit 0 is the least significant bit. Following the command byte are two data bytes which contain bit 8 through bit 0, aligned as shown below. Unused bits should be set to a zero. The second message contains bits 32 through 9 of the 33 bit PCR. It's recommended that the reader review the section "Sending Messages to the CS4920A" on page 1, due to the fact that this command contains two messages.




The byte order is shown below, for when MSG_FORMAT is set for double messages.




MSG_FORMAT byte order (if set for double messages.)

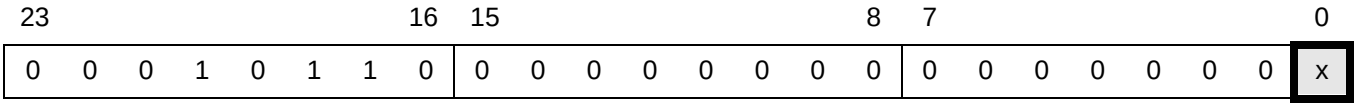
Reply: None

Default: Synchronization is off.

Message: MSG_PCR_DISABLE

Description: MSG_PCR_DISABLE disables PCR/PTS synchronization* and initiates a channel change sequence. The channel change sequence starts with MSG_PCR_DISABLE which will turn off PCR/PTS synchronization and auto-mute the DACs. MSG_PCR_INPUT is required to turn on PCR/PTS synchronization and auto-unmute the DACs. PCR/PTS synchronization may be disabled without initiating a channel change sequence by clearing the LSB of the message. The command byte is followed by 15 bits of zero and 1 bit for selection of channel change.

Protocol: The master must send the command byte first, followed by 15 bits of zero and 1 bit for selection of channel change. When the selection bit is 1 a channel change sequence is initiated. When the selection bit is 0 the PCR/PTS synchronization is turned off.




x = 0 :No channel change

x = 1 :Initiate channel change

Reply: None

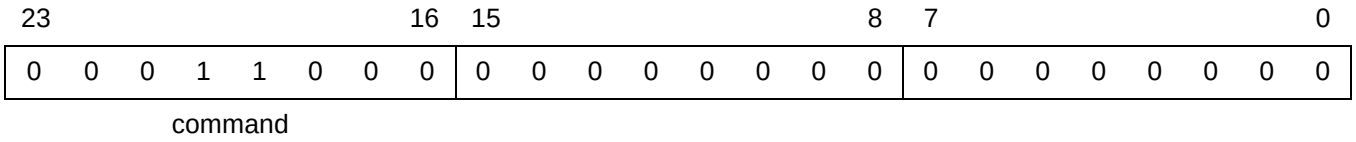
*Note: Synchronization using the 33-bit PCR counter is activated when using the command MSG_PCR_INPUT.

Default: Disabled.

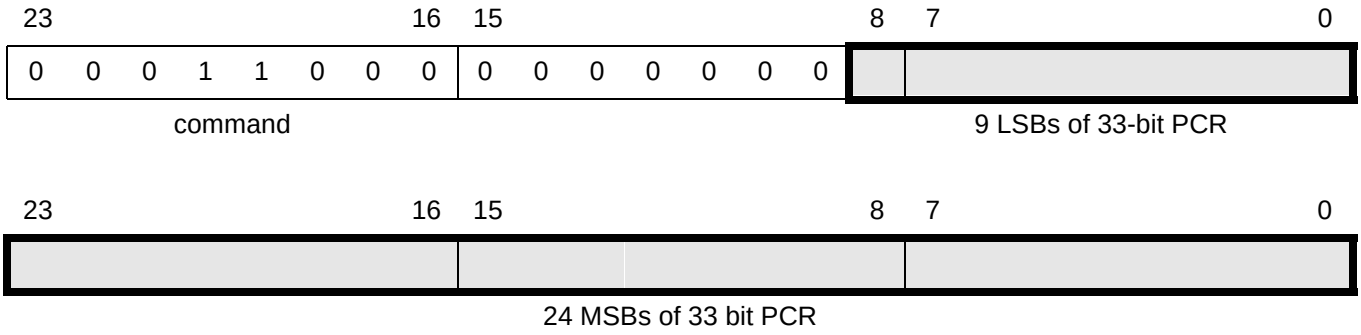
Message: MSG_PCR_READ

Description: MSG_PCR_READ enables the host to query the CS4920A for the current value of the internal 33-bit counter of the CS4920A. The command byte is followed by two don't care data bytes.

Protocol: The master must send the command byte first, followed by two don't care bytes. Upon receiving the MSG_PCR_READ command the CS4920A will respond with two 24-bit word values. The first word contains the command byte followed by 7 binary zeros followed by the nine least significant bits of the internal 33-bit counter. The second word contains the remaining 24 most significant bits of the 33-bit counter. Refer to both "Sending Messages to the CS4920A" and "Receiving Messages from the CS4920A" on pages 1 and 3.




Reply:

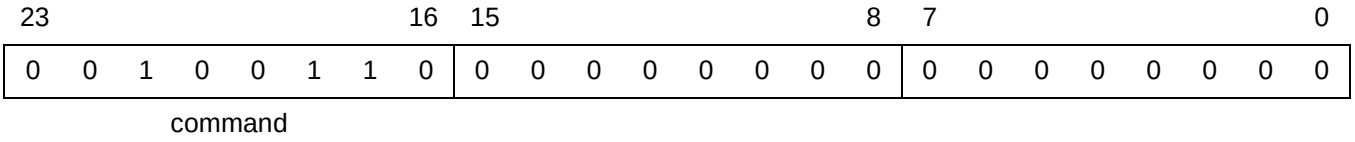



Default: N/A.

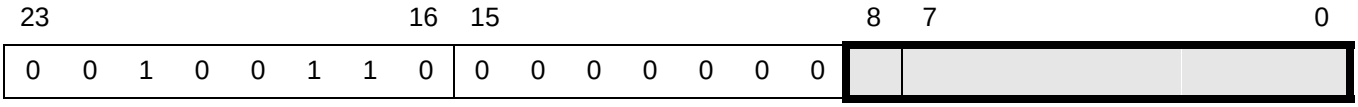
Message: MSG_PTS_READ

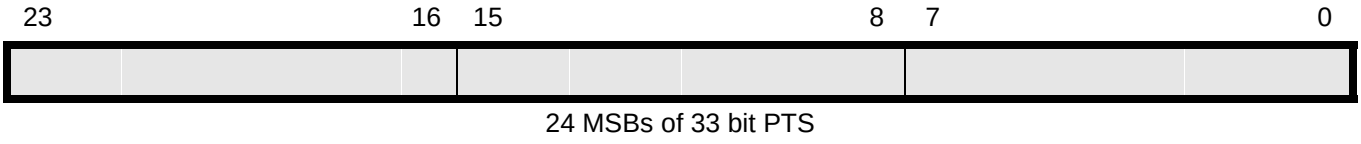
Description: MSG_PTS_READ allows the host to query the CS4920A for the value of the most recent PTS (Presentation Time Stamp) from the MPEG1 audio or MPEG2 audio packet stream. The command byte is followed by two don't care data bytes.

Protocol: The master must send the command byte first followed by two don't care bytes. Upon receiving the MSG_PTS_READ command, the CS4920ACS4920A will respond with two 24-bit word values. The first word contains the command byte followed by the 7 binary zeros followed by the time least significant bits of the most recent PTS extracted. The second word contains the 24 most significant bits of the most recent PTS extracted. Refer to both "Sending Messages to the CS4920A" and "Receiving Messages from the CS4920A" on page 1 and 3.




Reply:




SMART
Analog
™