HI-3220

High Density 16Rx / 8Tx or 8Rx / 4Tx ARINC 429 Protocol IC with optional Integrated Line Receivers

December 2018

GENERAL DESCRIPTION

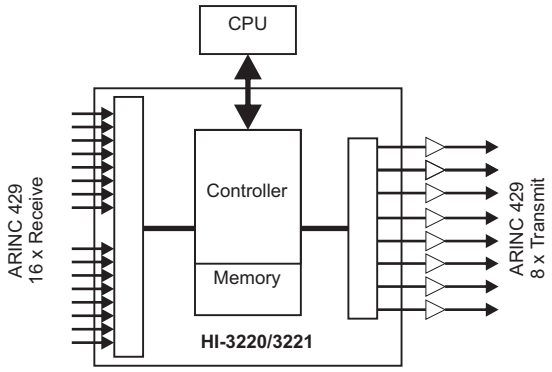
The HI-3220 from Holt Integrated Circuits is a family of single chip CMOS high density application data management ICs capable of managing, storing and forwarding avionics data messages between sixteen ARINC 429 receive channels and eight ARINC 429 transmit channels. Options for eight receive and four transmit channels are also available in a compact, cost effective 7mm x 7mm QFN package footprint.

The ARINC 429 buses may operate independently - the IC can be programmed to automatically re-format, re-label, re-packetize and re-transmit data from ARINC 429 receive buses to ARINC 429 transmit buses. Alternatively, a host CPU can send and receive data on multiple buses. 32KB of on-board memory allows received data to be logically organized and automatically updated as new ARINC 429 labels are received via ARINC 429 label mailboxes. A 64 message deep Receive FIFO is also available for each receive channel. Each transmitter has the option to independently enable a programmable scheduler on the ARINC 429 bus, or to have the CPU directly send data. An optional auto-initialization feature allows configuration information to be up-loaded from an external EEPROM on reset to facilitate rapid start-up or operation without a host CPU.

The HI-3220, HI-3221 and HI-3222 have integrated ARINC 429 line receivers, capable of RTCA/DO-160 level 3 lightning compliance with external 40kΩ resistors. HI-3225 and HI-3226 utilize external line receivers, for example Holt's octal HI-8458 family. The HI-3222 and HI-3223 are 8-channel receive, 4-channel transmit options available in smaller package footprints, with HI-3223 having the same slope control feature as HI-3220. Transmit outputs interface directly to external HI-8592, HI-8596, or integrated lightning protected HI-8597 ARINC 429 line drivers. All parts use a 40 MHz 4-wire SPI (Serial Peripheral Interface) host connection.

HI-3220 and HI-3225 are available in an 80-pin PQFP package, whereas HI-3221 and HI-3226 are available in a 72-pin 10mm x 10mm QFN package. The HI-3222 (8-ch. Rx, 4-ch. Tx) is available in a very compact 48-pin 7mm x 7mm QFN.

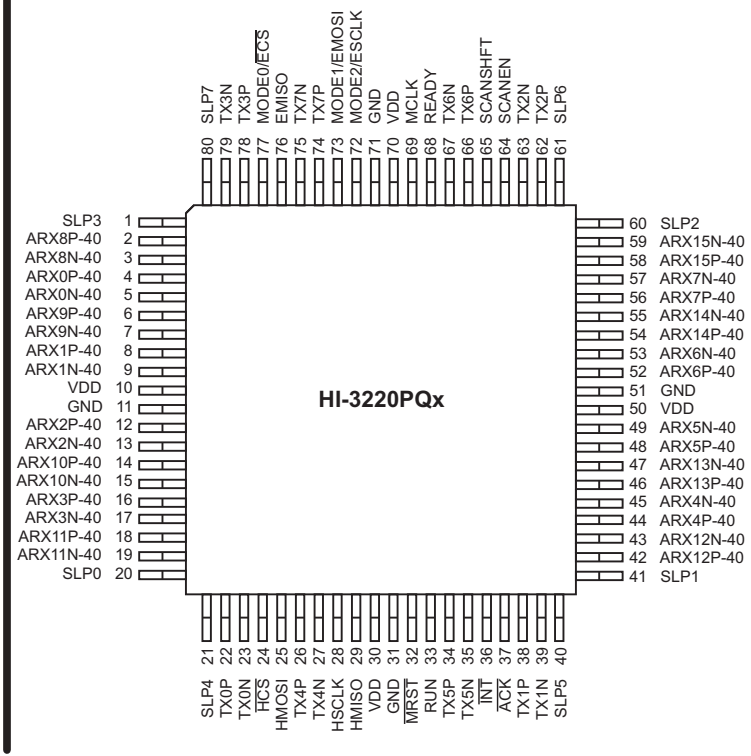
APPLICATION



FEATURES

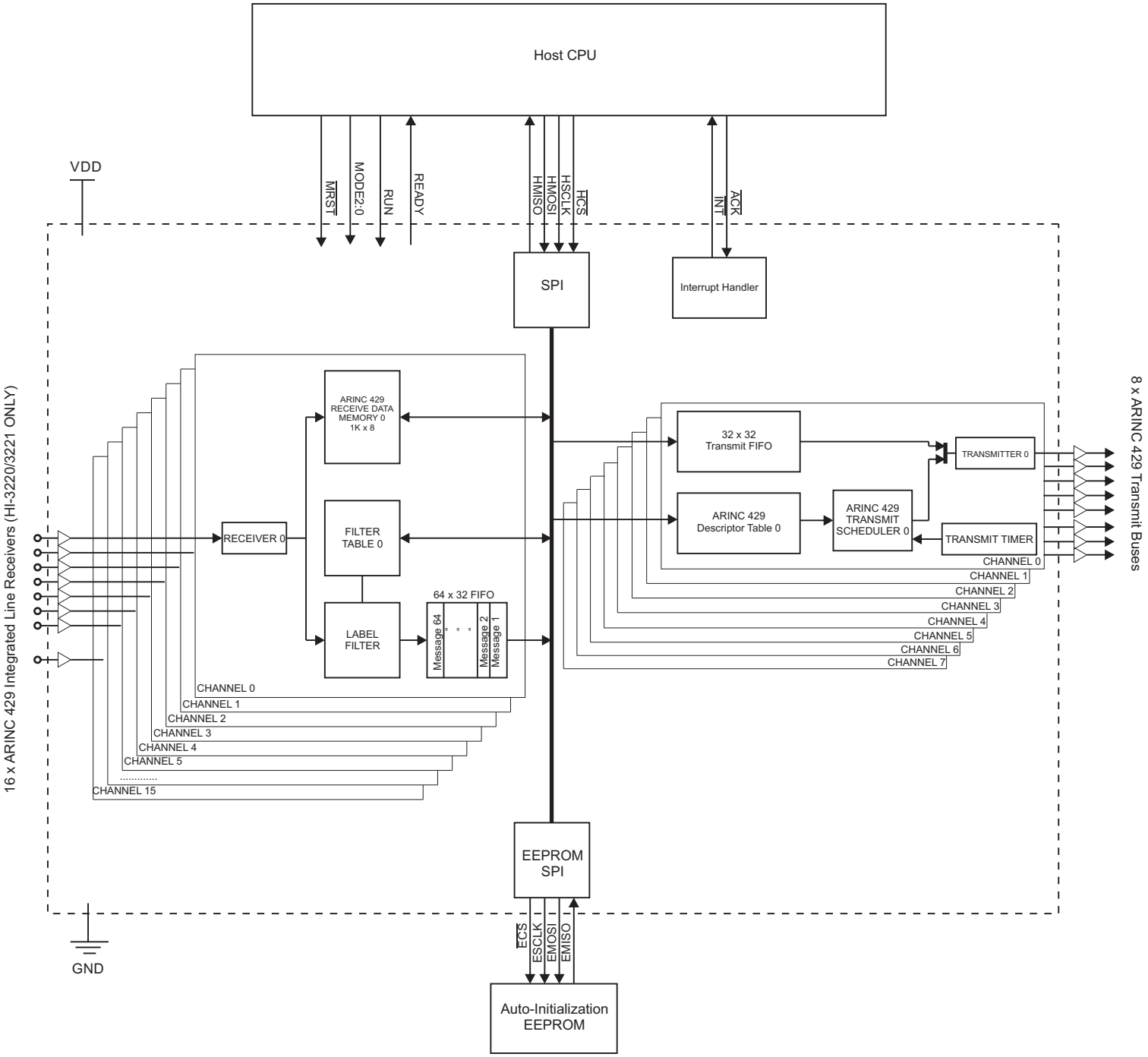
- Fully compliant to ARINC 429 Specification
- Sixteen ARINC 429 Receive channels with optional integrated line Receivers
- Eight ARINC 429 Transmit channels
- 8-channel Rx, 4-channel Tx version available in compact 48-pin 7mm x 7mm QFN
- 32 KB on chip user-configurable data storage memory
- Programmable receive data filtering
- Non-annunciated fault protection
- Option for programmable transmission schedulers for periodic ARINC 429 broadcasting or CPU directed transmission
- Auto-initialization feature allows power-on configuration or independent operation without CPU
- 40 MHZ SPI CPU interface
- Transmitter rate and tri-state control outputs
- Supports 100kbs / 50kbs / 12.5kbs data rates

PIN CONFIGURATION



80 - Pin Plastic Quad Flat Pack (PQFP)

BLOCK DIAGRAM



PRODUCT OPTIONS

PART NUMBER	PACKAGE	LINE RECEIVERS	TRANSMIT CHANNELS
HI-3220	80-pin QFP	16 on-chip	8 Transmitters & SLP7:0 outputs
HI-3221	72-pin QFN	16 on-chip	8 Transmitters. No SLP7:0 outputs
HI-3222	48-pin QFN	8 on-chip	4 Transmitters. No SLP outputs
HI-3223	52-pin QFP or 64-pin QFN	8 on-chip	4 Transmitters & SLP3:0 outputs
HI-3225	80-pin QFP	16 External	8 Transmitters & SLP7:0 outputs
HI-3226	72-pin QFN	16 External	8 Transmitters. No SLP7:0 outputs

PIN DESCRIPTIONS (HI-3220, HI-3221, HI-3222, HI-3223)

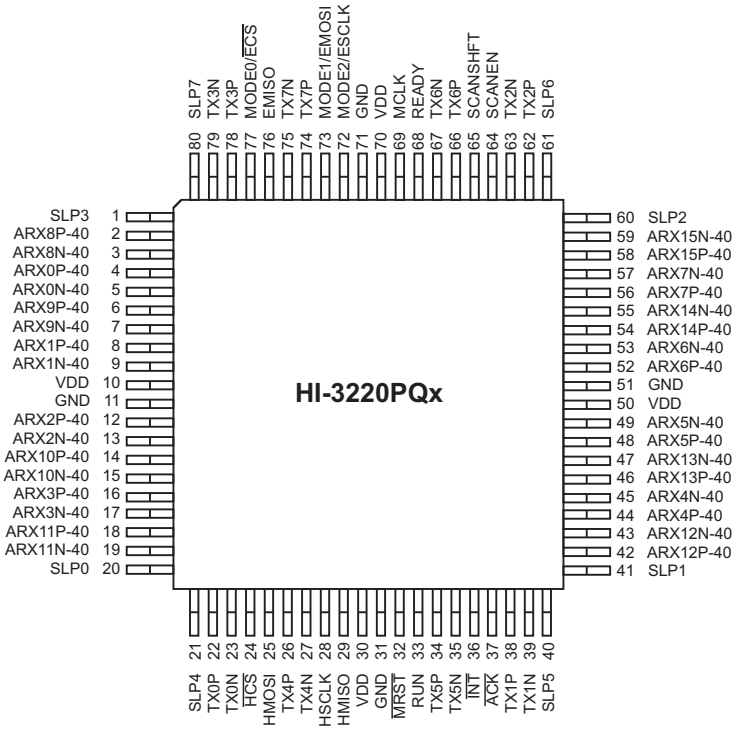
Signal	Function	Description	Internal Pull-Up/Down
$\overline{ACK}$	INPUT	Interrupt Acknowledge. Active low.	Pull-Up
$\overline{INT}$	OUTPUT	Interrupt. Active low, open-drain.	Pull-Up
ARX0N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 0. Requires external 40KOhm resistor.	
ARX0P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 0. Requires external 40KOhm resistor.	
ARX1N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 1. Requires external 40KOhm resistor.	
ARX1P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 1. Requires external 40KOhm resistor.	
ARX2N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 2. Requires external 40KOhm resistor.	
ARX2P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 2. Requires external 40KOhm resistor.	
ARX3N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 3. Requires external 40KOhm resistor.	
ARX3P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 3. Requires external 40KOhm resistor.	
ARX4N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 4. Requires external 40KOhm resistor.	
ARX4P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 4. Requires external 40KOhm resistor.	
ARX5N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 5. Requires external 40KOhm resistor.	
ARX5P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 5. Requires external 40KOhm resistor.	
ARX6N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 6. Requires external 40KOhm resistor.	
ARX6P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 6. Requires external 40KOhm resistor.	
ARX7N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 7. Requires external 40KOhm resistor.	
ARX7P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 7. Requires external 40KOhm resistor.	
ARX8N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 8. Requires external 40KOhm resistor.	
ARX8P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 8. Requires external 40KOhm resistor.	
ARX9N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 9. Requires external 40KOhm resistor.	
ARX9P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 9. Requires external 40KOhm resistor.	
ARX10N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 10. Requires external 40KOhm resistor.	
ARX10P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 10. Requires external 40KOhm resistor.	
ARX11N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 11. Requires external 40KOhm resistor.	
ARX11P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 11. Requires external 40KOhm resistor.	
ARX12N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 12. Requires external 40KOhm resistor.	
ARX12P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 12. Requires external 40KOhm resistor.	
ARX13N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 13. Requires external 40KOhm resistor.	
ARX13P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 13. Requires external 40KOhm resistor.	
ARX14N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 14. Requires external 40KOhm resistor.	
ARX14P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 14. Requires external 40KOhm resistor.	
ARX15N-40	ANALOG INPUT	ARINC 429 Receive bus negative line input for channel 15. Requires external 40KOhm resistor.	
ARX15P-40	ANALOG INPUT	ARINC 429 Receive bus positive line input for channel 15. Requires external 40KOhm resistor.	
TX0N	OUTPUT	ARINC 429 Tx channel 0 negative data output to line driver	
TX0P	OUTPUT	ARINC 429 Tx channel 0 positive data output to line driver	
TX1N	OUTPUT	ARINC 429 Tx channel 1 negative data output to line driver	
TX1P	OUTPUT	ARINC 429 Tx channel 1 positive data output to line driver	
TX2N	OUTPUT	ARINC 429 Tx channel 2 negative data output to line driver	
TX2P	OUTPUT	ARINC 429 Tx channel 2 positive data output to line driver	
TX3N	OUTPUT	ARINC 429 Tx channel 3 negative data output to line driver	
TX3P	OUTPUT	ARINC 429 Tx channel 3 positive data output to line driver	
TX4N	OUTPUT	ARINC 429 Tx channel 4 negative data output to line driver	
TX4P	OUTPUT	ARINC 429 Tx channel 4 positive data output to line driver	
TX5N	OUTPUT	ARINC 429 Tx channel 5 negative data output to line driver	
TX5P	OUTPUT	ARINC 429 Tx channel 5 positive data output to line driver	
TX6N	OUTPUT	ARINC 429 Tx channel 6 positive data output to line driver	
TX6P	OUTPUT	ARINC 429 Tx channel 6 positive data output to line driver	
TX7N	OUTPUT	ARINC 429 Tx channel 7 positive data output to line driver	
TX7P	OUTPUT	ARINC 429 Tx channel 7 positive data output to line driver	
SLP0	OUTPUT	Digital slope control signal for external line driver channel 0 (HI-3220 and HI-3223)	
SLP1	OUTPUT	Digital slope control signal for external line driver channel 1 (HI-3220 and HI-3223)	
SLP2	OUTPUT	Digital slope control signal for external line driver channel 2 (HI-3220 and HI-3223)	
SLP3	OUTPUT	Digital slope control signal for external line driver channel 3 (HI-3220 and HI-3223)	
SLP4	OUTPUT	Digital slope control signal for external line driver channel 4 (HI-3220 and HI-3223)	
SLP5	OUTPUT	Digital slope control signal for external line driver channel 5 (HI-3220 and HI-3223)	
SLP6	OUTPUT	Digital slope control signal for external line driver channel 6 (HI-3220 and HI-3223)	
SLP7	OUTPUT	Digital slope control signal for external line driver channel 7 (HI-3220 and HI-3223)	

PIN DESCRIPTIONS (HI-3220, HI-3221, HI-3222, HI-3223) continued

Signal	Function	Description	Internal Pull-Up/Down
MODE0/$\overline{ECS}$	I/O	MODE0 configuration input sampled at reset / SPI chip select output for initialization EEPROM	See Note below
EMISO	INPUT	SPI serial data input from auto-inialization EEPROM	Pull-Down
MODE1/EMOSI	I/O	MODE1 configuration input sampled at reset / SPI serial data output to initialization EEPROM	See Note below
MODE2/ESCLK	I/O	MODE2 configuration input sampled at reset / SPI clock for auto-initialization EEPROM	See Note below
GND	POWER	Chip 0V supply. All four pins must be connected.	
VDD	POWER	3.3V power supply. All four pins must be powered.	
$\overline{HCS}$	INPUT	Host chip select. Data is shifted into HMOSI and out of HMISO when $\overline{HCS}$ is low	Pull-Up
HMISO	OUTPUT	Host CPU SPI interface serial data output	Pull-Down
HMOSI	INPUT	Host CPU SPI interface serial data input	Pull-Down
HSCLK	INPUT	Host SPI Clock. Data is shifted into or out of the SPI interface using HSCLK	Pull-Down
MCLK	INPUT	Master and reference clock for ARINC 429 bus bit timing. 50MHz +/- 0.1%	
$\overline{MRST}$	INPUT	Master Reset to HI-322X Active Low. 225 ns minimum pulse width.	Pull-Down
READY	OUTPUT	READY goes high when post-RESET initialization is complete	
RUN	INPUT	Master enable signal for ARINC 429 transmit schedulers	Pull-Down
SCANEN	INPUT	Factory test only. Connect to GND.	Pull-Down
SCANSHFT	INPUT	Factory test only. Connect to GND.	Pull-Down

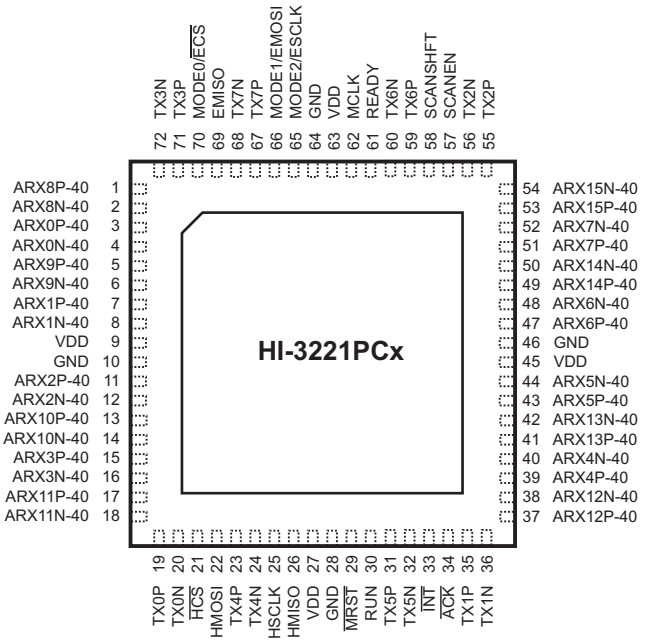
NOTE: When not using an external EEPROM, the Mode0, Mode1 and Mode2 pins must be connected to either VDD or GND through a 1K - 3.3K resistor, depending on the desired power up mode. These pins do not have internal pull-up or pull-down resistors and need to be connected to a known state to prevent possible initialization of an un-intended mode.

PIN CONFIGURATIONS

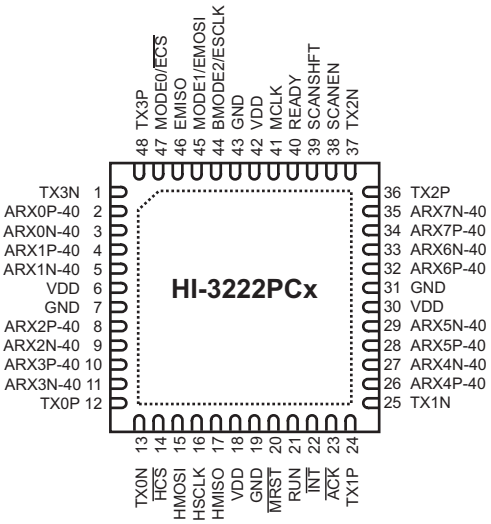


80 - Pin Plastic Quad Flat Pack (PQFP)

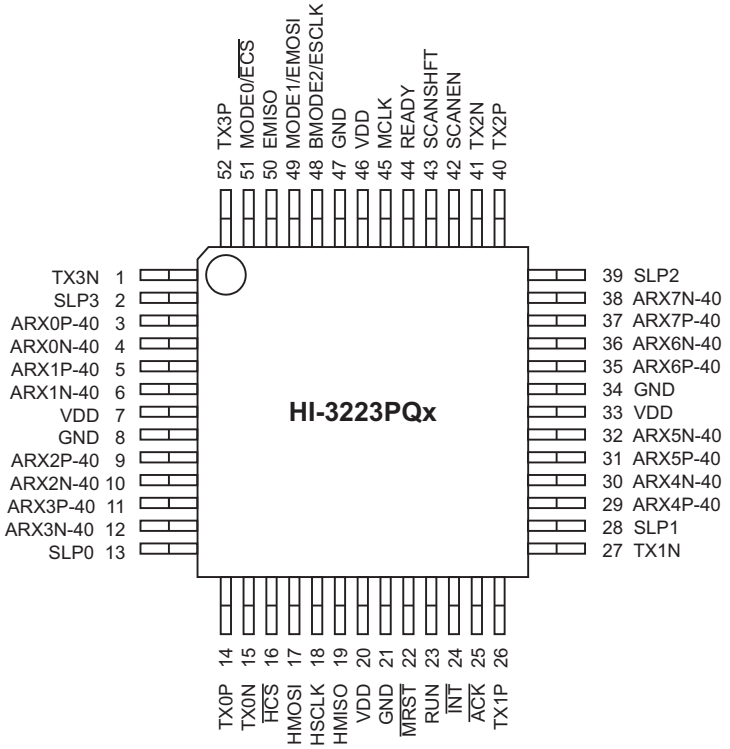
PIN CONFIGURATIONS (continued)



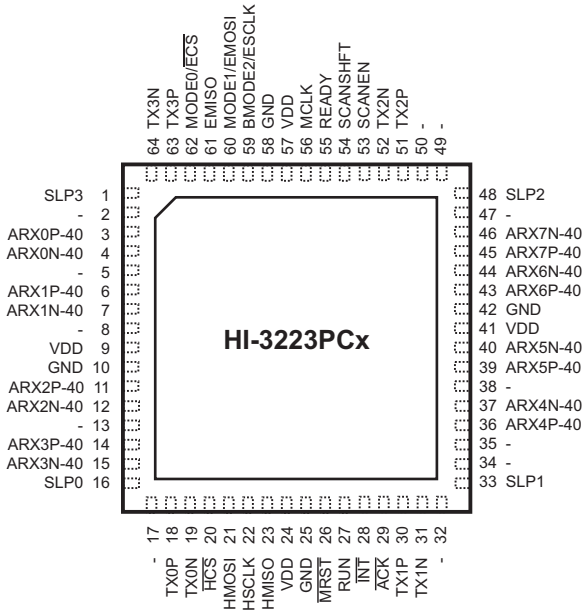
72 - Pin Plastic QFN (Top View)



48 - Pin Plastic QFN (Top View)



52 - Pin Plastic Quad Flat Pack (PQFP)



"-" denotes "Not Connected internally"

64 - Pin Plastic QFN (Top View)

PIN DESCRIPTIONS (HI-3225, HI-3226)

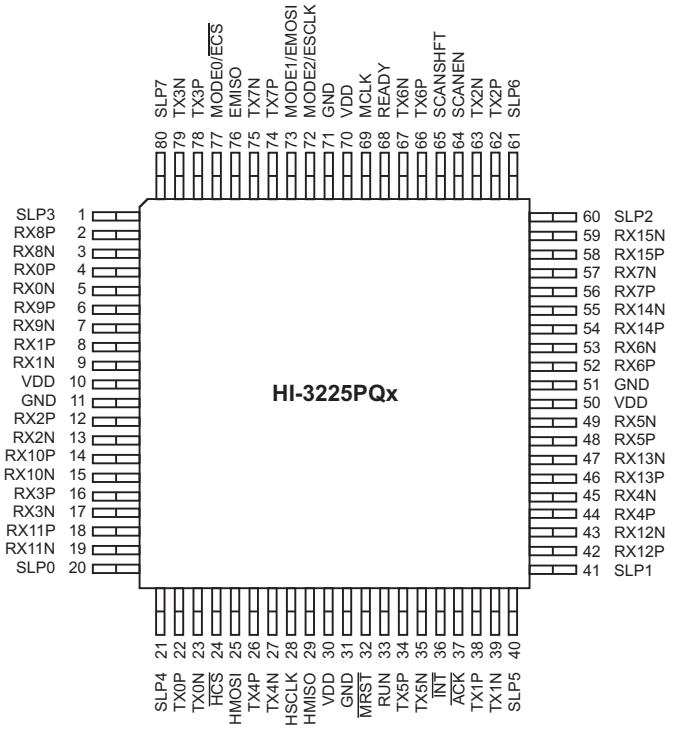
Signal	Function	Description	Internal Pull-Up/Down
$\overline{ACK}$	INPUT	Interrupt Acknowledge. Active low.	Pull-Up
$\overline{INT}$	OUTPUT	Interrupt. Active low, open-drain.	Pull-Up
RX0N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 0.	Pull-down
RX0P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 0.	Pull-down
RX1N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 1.	Pull-down
RX1P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 1.	Pull-down
RX2N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 2.	Pull-down
RX2P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 2.	Pull-down
RX3N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 3.	Pull-down
RX3P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 3.	Pull-down
RX4N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 4.	Pull-down
RX4P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 4.	Pull-down
RX5N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 5.	Pull-down
RX5P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 5.	Pull-down
RX6N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 6.	Pull-down
RX6P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 6.	Pull-down
RX7N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 7.	Pull-down
RX7P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 7.	Pull-down
RX8N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 8.	Pull-down
RX8P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 8.	Pull-down
RX9N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 9.	Pull-down
RX9P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 9.	Pull-down
RX10N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 10.	Pull-down
RX10P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 10.	Pull-down
RX11N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 11.	Pull-down
RX11P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 11.	Pull-down
RX12N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 12.	Pull-down
RX12P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 12.	Pull-down
RX13N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 13.	Pull-down
RX13P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 13.	Pull-down
RX14N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 14.	Pull-down
RX14P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 14.	Pull-down
RX15N	INPUT	ARINC 429 Digital input from line receiver negative signal for channel 15.	Pull-down
RX15P	INPUT	ARINC 429 Digital input from line receiver positive signal for channel 15.	Pull-down
TX0N	OUTPUT	ARINC 429 Tx channel 0 negative data output to line driver	
TX0P	OUTPUT	ARINC 429 Tx channel 0 positive data output to line driver	
TX1N	OUTPUT	ARINC 429 Tx channel 1 negative data output to line driver	
TX1P	OUTPUT	ARINC 429 Tx channel 1 positive data output to line driver	
TX2N	OUTPUT	ARINC 429 Tx channel 2 negative data output to line driver	
TX2P	OUTPUT	ARINC 429 Tx channel 2 positive data output to line driver	
TX3N	OUTPUT	ARINC 429 Tx channel 3 negative data output to line driver	
TX3P	OUTPUT	ARINC 429 Tx channel 3 positive data output to line driver	
TX4N	OUTPUT	ARINC 429 Tx channel 4 negative data output to line driver	
TX4P	OUTPUT	ARINC 429 Tx channel 4 positive data output to line driver	
TX5N	OUTPUT	ARINC 429 Tx channel 5 negative data output to line driver	
TX5P	OUTPUT	ARINC 429 Tx channel 5 positive data output to line driver	
TX6N	OUTPUT	ARINC 429 Tx channel 6 positive data output to line driver	
TX6P	OUTPUT	ARINC 429 Tx channel 6 positive data output to line driver	
TX7N	OUTPUT	ARINC 429 Tx channel 7 positive data output to line driver	
TX7P	OUTPUT	ARINC 429 Tx channel 7 positive data output to line driver	
SLP0	OUTPUT	Digital slope control signal for external line driver channel 0 (HI-3225)	
SLP1	OUTPUT	Digital slope control signal for external line driver channel 1 (HI-3225)	
SLP2	OUTPUT	Digital slope control signal for external line driver channel 2 (HI-3225)	
SLP3	OUTPUT	Digital slope control signal for external line driver channel 3 (HI-3225)	
SLP4	OUTPUT	Digital slope control signal for external line driver channel 4 (HI-3225)	
SLP5	OUTPUT	Digital slope control signal for external line driver channel 5 (HI-3225)	
SLP6	OUTPUT	Digital slope control signal for external line driver channel 6 (HI-3225)	
SLP7	OUTPUT	Digital slope control signal for external line driver channel 7 (HI-3225)	

PIN DESCRIPTIONS (HI-3225, HI-3226) continued

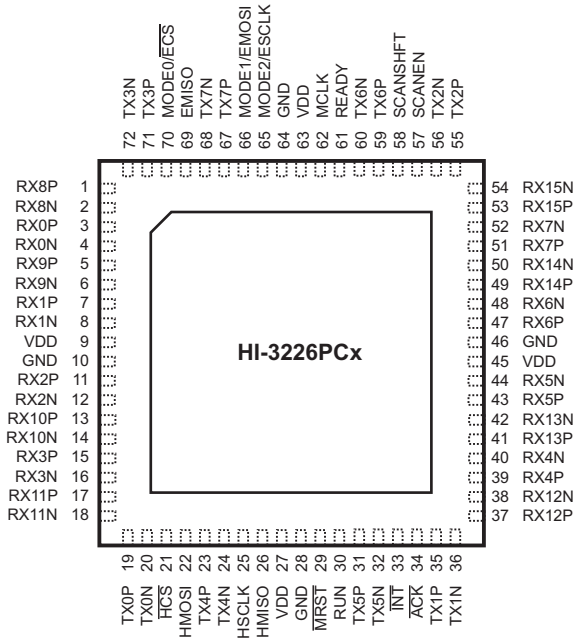
Signal	Function	Description	Internal Pull-Up/Down
MODE0/$\overline{ECS}$	I/O	MODE0 configuration input sampled at reset / SPI chip select output for EEPROM initialization	See Note below
EMISO	INPUT	SPI serial data input from auto-inialization EEPROM	Pull-Down
MODE1/EMOSI	I/O	MODE1 configuration input sampled at reset / SPI serial data output for EEPROM initialization	See Note below
MODE2/ESCLK	I/O	MODE2 configuration input sampled at reset / SPI clock for EEPROM auto-initialization	See Note below
GND	POWER	Chip 0V supply. All four pins must be connected.	
VDD	POWER	3.3V power supply. All four pins must be powered.	
$\overline{HCS}$	INPUT	Host chip select. Data is shifted into HMOSI and out of HMISO when $\overline{HCS}$ is low	Pull-Up
HMISO	OUTPUT	Host CPU SPI interface serial data output	Pull-Down
HMOSI	INPUT	Host CPU SPI interface serial data input	Pull-Down
HSCLK	INPUT	Host SPI Clock. Data is shifted into or out of the SPI interface using HSCLK	Pull-Down
MCLK	INPUT	Master and reference clock for ARINC 429 bus bit timing. 50 MHZ +/- 0.1%	
$\overline{MRST}$	INPUT	Master Reset to HI-322X Active Low. 225 ns minimum pulse width.	Pull-Down
READY	OUTPUT	READY goes high when post-RESET initialization is complete	
RUN	INPUT	Master enable signal for ARINC 429 transmit schedulers	Pull-Down
SCANEN	INPUT	Factory test only. Connect to GND.	Pull-Down
SCANSHFT	INPUT	Factory test only. Connect to GND.	Pull-Down

NOTE: When not using an external EEPROM, the Mode0, Mode1 and Mode2 pins must be connected to either VDD or GND through a 1K - 3.3K resistor, depending on the desired power up mode. These pins do not have internal pull-up or pull-down resistors and need to be connected to a known state to prevent possible initialization of an un-intended mode.

PIN CONFIGURATIONS



80 - Pin Plastic Quad Flat Pack (PQFP)



72 - Pin Plastic QFN (Top View)

APPLICATION OVERVIEW

The HI-3220 is a flexible device for managing ARINC 429 communications and data storage in many avionics applications. The device architecture centers around a 32K x 8 static RAM used for data storage, data filtering tables and table-driven transmission schedulers. Once configured, the device can operate autonomously without a host CPU, negating the need for software development or DO-178 certification. Configuration data may be uploaded into the device from an external EEPROM, following system reset.

The device supports up to sixteen ARINC 429 receive channels. Analog line receivers are on-chip (HI-3220, HI-3221 and HI-3222), or off-chip (HI-3225 and HI-3226). Received data is stored in on-chip RAM organized by channel number and label. The data table continually updates as new labels arrive. Programmable interrupts and filters alert the host subsystem to labels of interest.

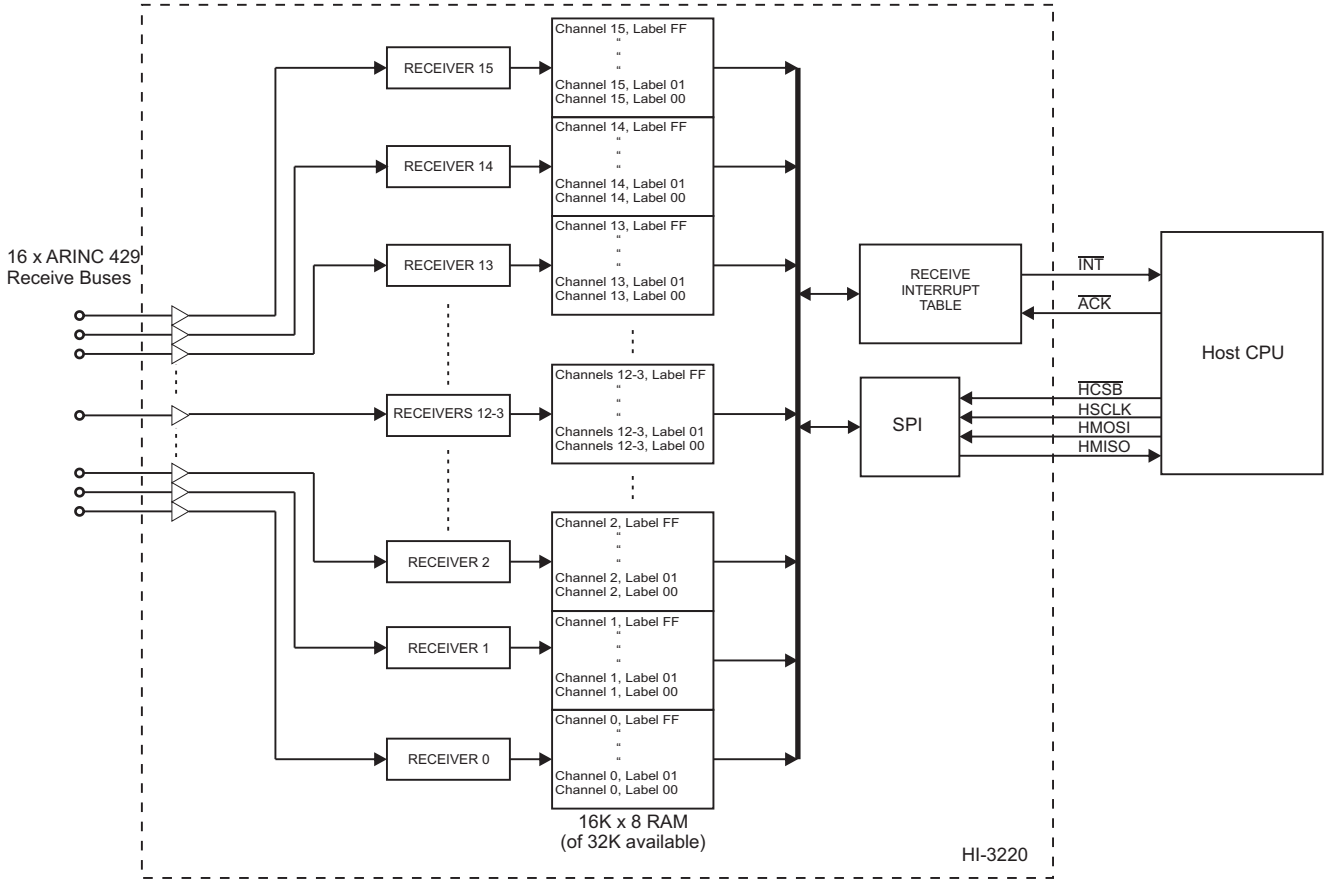
Each ARINC 429 receive channel also includes a 64 message deep FIFO allowing selected label data to be queued for subsequent host access.

The HI-3220 supports up to eight independent ARINC 429 transmit channels. Transmission may be controlled entirely by an external CPU, or autonomously by programming one or more of the eight on-chip ARINC 429 transmit schedulers. These allow periodic transmission to occur without CPU. Source data for transmission may be selected from RAM based tables of constants and / or from the received channel data. Powerful options exist for constructing ARINC 429 labels as well as controlling their timing and conditional transmission.

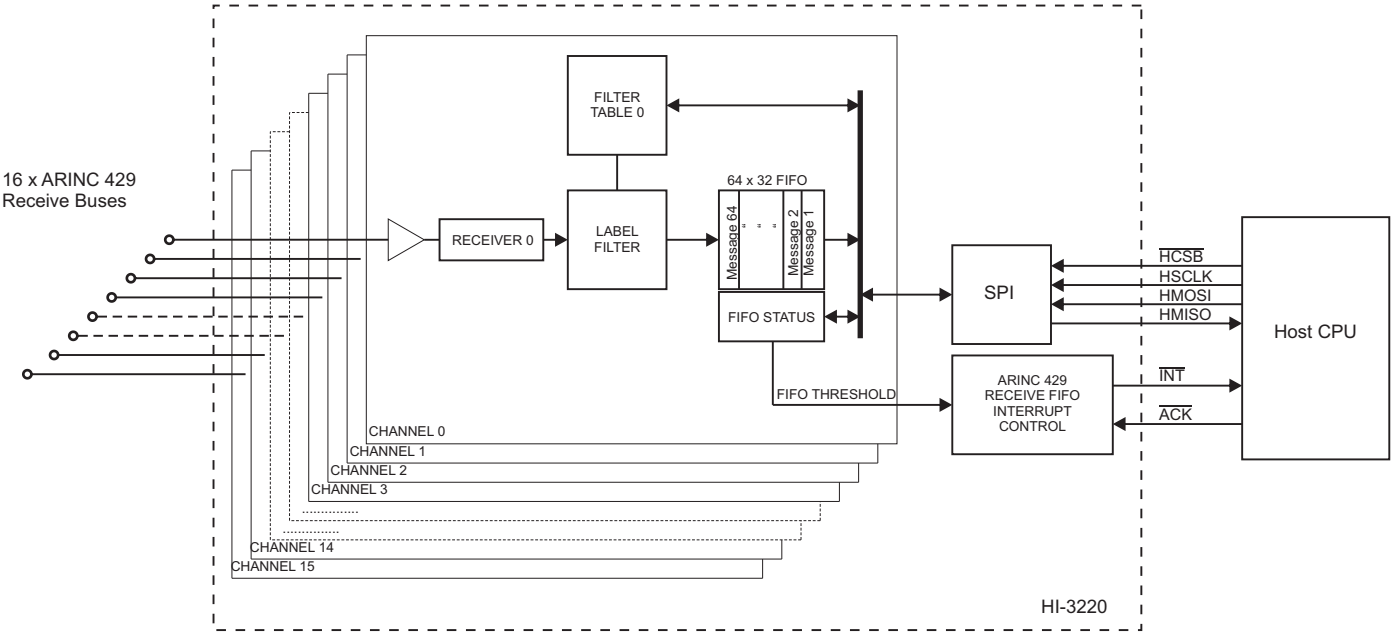
Even when running under the control of schedulers, the host CPU may insert new labels for transmission at will.

The following examples show five possible configurations of how the HI-3220 may be used:

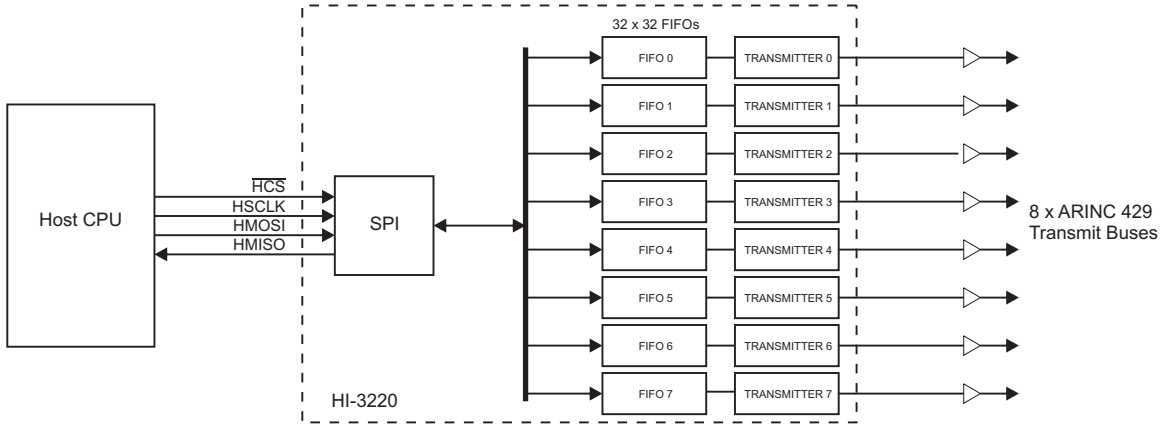
Example 1. ARINC 429 Data reception using on-chip RAM



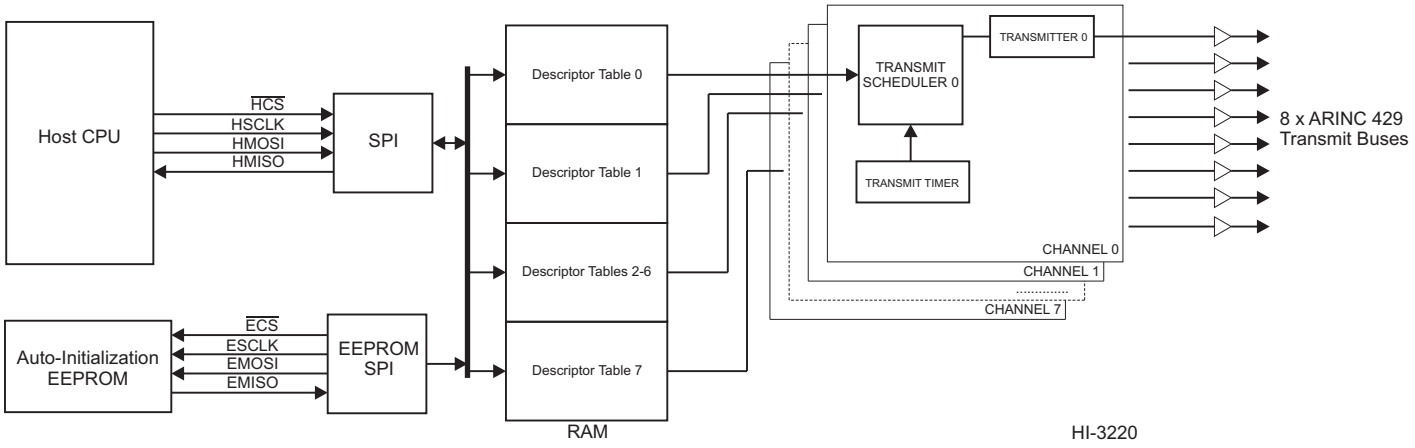
Example 2. ARINC 429 Data reception using on-chip filters and FIFOs



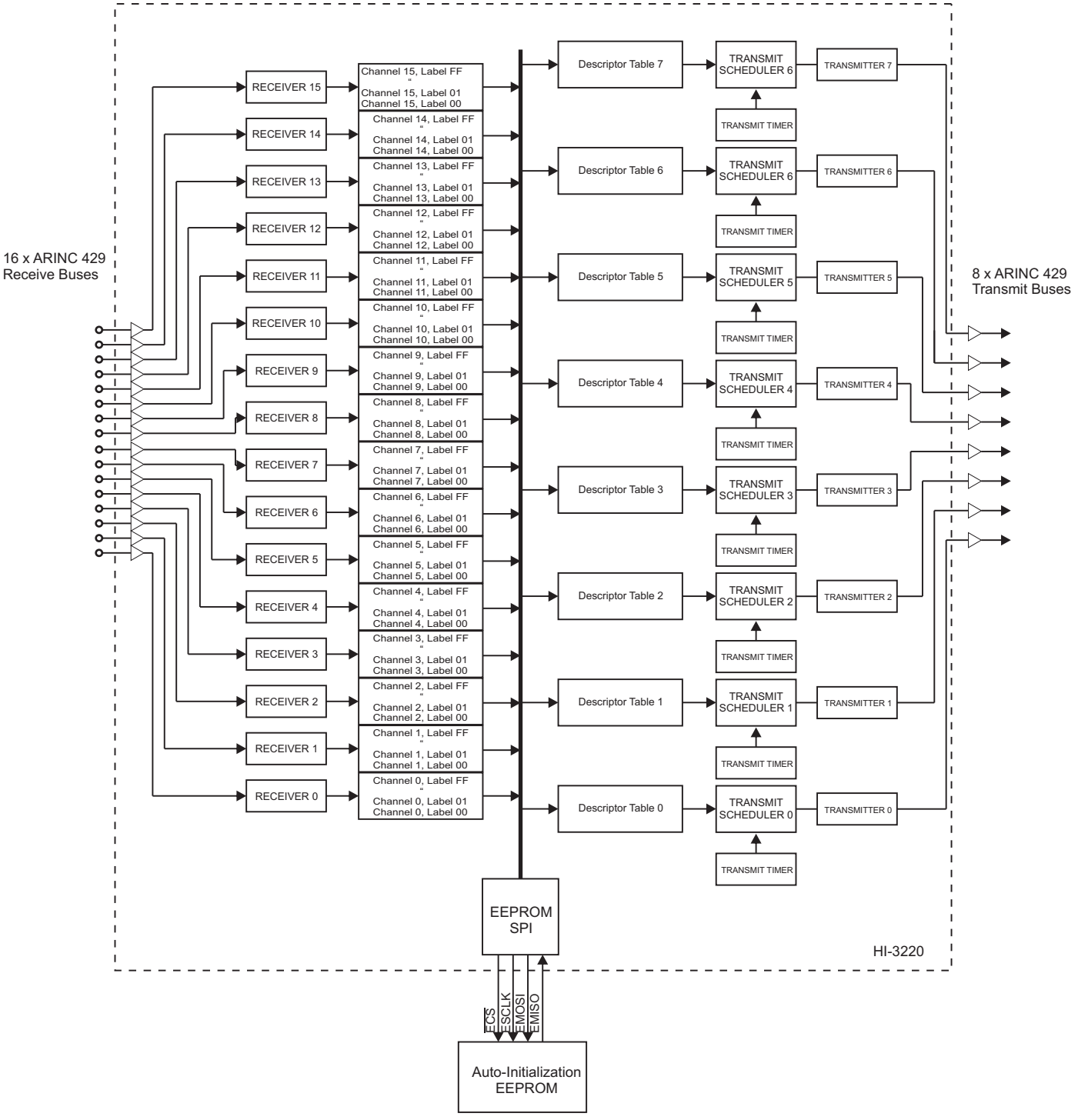
Example 3. ARINC 429 Data transmission directly from CPU



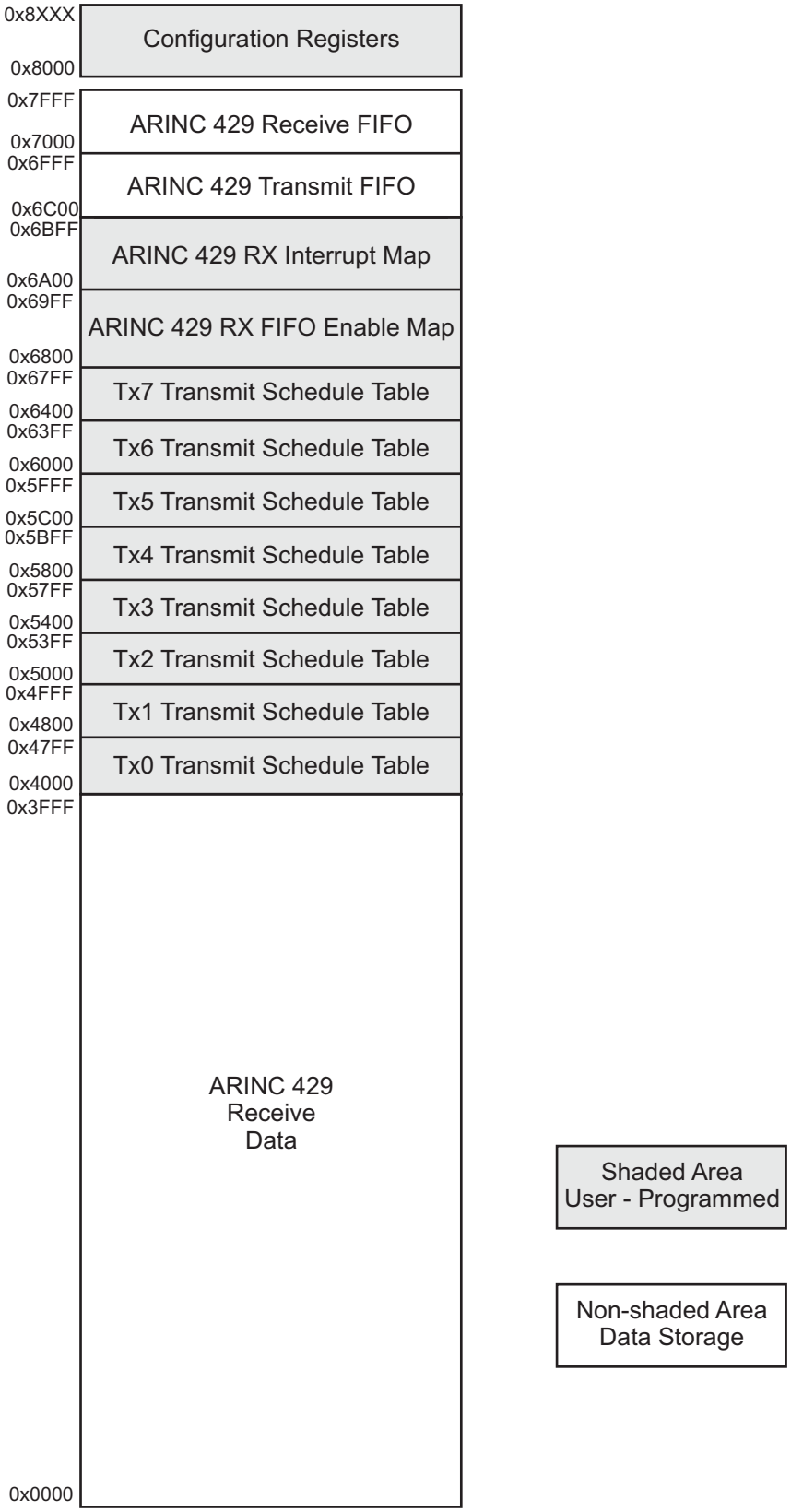
Example 4. ARINC 429 Data transmission using on-chip schedulers



Example 5. Autonomous ARINC 429 Data Concentrator / Repeater



HI-3220 MEMORY MAP



HI-3220 REGISTER MAP

ADDRESS	R/W	REGISTER	MNEMONIC	DESCRIPTION
0x8000	R/W	MASTER CONTROL REGISTER	MCR	HI-3220 global configuration (write address)
0x8001	R	MASTER CONTROL REGISTER	MCR	HI-3220 global configuration (read address)
0x8002	R	MASTER STATUS REGISTER	MSR	Indicates HI-3220 current status
0x8003	R	Not used		
0x8004	R*	PENDING INTERRUPT REGISTER	PIR	Indicates Interrupt type
0x8005	R	Not used		
0x8006	R*	RECEIVE PENDING INTERRUPT LOW	RPIRL	Defines channel(7-0) with pending Interrupt
0x8007	R	Not used		
0x8008	R*	RECEIVE PENDING INTERRUPT HIGH	RPIRH	Defines channel(15-8) with pending Interrupt
0x8009	R	Not used		
0x800A	R*	PENDING INTERRUPT REGISTER	PIR	Indicates Interrupt type
0x800B	R*	RECEIVE PENDING INTERRUPT LOW	RPIRL	Defines channel(7-0) with pending Interrupt
0x800C	R*	RECEIVE PENDING INTERRUPT HIGH	RPIRH	Defines channel(15-8) with pending Interrupt
0x800D	R	RECEIVE FIFO THRESHOLD FLAG (LOW)	FTFL	Shows which Rx FIFOs hold > (thresh) words
0x800E	R	RECEIVE FIFO THRESHOLD FLAG (HIGH)	FTFH	Shows which Rx FIFOs hold > (thresh) words
0x800F	R	TRANSMITTER FIFO THRESHOLD FLAGS	TFR	Shows which Tx FIFOs hold < (thresh) words
0x8010	R	RECEIVE INTERRUPT ADDRESS 0	IAR0	ARINC 429 Interrupt Vector channel 0
0x8011	R	RECEIVE INTERRUPT ADDRESS 1	IAR1	ARINC 429 Interrupt Vector channel 1
0x8012	R	RECEIVE INTERRUPT ADDRESS 2	IAR2	ARINC 429 Interrupt Vector channel 2
0x8013	R	RECEIVE INTERRUPT ADDRESS 3	IAR3	ARINC 429 Interrupt Vector channel 3
0x8014	R	RECEIVE INTERRUPT ADDRESS 4	IAR4	ARINC 429 Interrupt Vector channel 4
0x8015	R	RECEIVE INTERRUPT ADDRESS 5	IAR5	ARINC 429 Interrupt Vector channel 5
0x8016	R	RECEIVE INTERRUPT ADDRESS 6	IAR6	ARINC 429 Interrupt Vector channel 6
0x8017	R	RECEIVE INTERRUPT ADDRESS 7	IAR7	ARINC 429 Interrupt Vector channel 7
0x8018	R	RECEIVE INTERRUPT ADDRESS 8	IAR8	ARINC 429 Interrupt Vector channel 8
0x8019	R	RECEIVE INTERRUPT ADDRESS 9	IAR9	ARINC 429 Interrupt Vector channel 9
0x801A	R	RECEIVE INTERRUPT ADDRESS 10	IAR10	ARINC 429 Interrupt Vector channel 10
0x801B	R	RECEIVE INTERRUPT ADDRESS 11	IAR11	ARINC 429 Interrupt Vector channel 11
0x801C	R	RECEIVE INTERRUPT ADDRESS 12	IAR12	ARINC 429 Interrupt Vector channel 12
0x801D	R	RECEIVE INTERRUPT ADDRESS 13	IAR13	ARINC 429 Interrupt Vector channel 13
0x801E	R	RECEIVE INTERRUPT ADDRESS 14	IAR14	ARINC 429 Interrupt Vector channel 14
0x801F	R	RECEIVE INTERRUPT ADDRESS 15	IAR15	ARINC 429 Interrupt Vector channel 15
0x8020	R/W	RECEIVE CONTROL REGISTER 0	RXC0	Configures ARINC 429 receive channel 0
0x8021	R/W	RECEIVE CONTROL REGISTER 1	RXC1	Configures ARINC 429 receive channel 1
0x8022	R/W	RECEIVE CONTROL REGISTER 2	RXC2	Configures ARINC 429 receive channel 2
0x8023	R/W	RECEIVE CONTROL REGISTER 3	RXC3	Configures ARINC 429 receive channel 3
0x8024	R/W	RECEIVE CONTROL REGISTER 4	RXC4	Configures ARINC 429 receive channel 4
0x8025	R/W	RECEIVE CONTROL REGISTER 5	RXC5	Configures ARINC 429 receive channel 5
0x8026	R/W	RECEIVE CONTROL REGISTER 6	RXC6	Configures ARINC 429 receive channel 6
0x8027	R/W	RECEIVE CONTROL REGISTER 7	RXC7	Configures ARINC 429 receive channel 7
0x8028	R/W	RECEIVE CONTROL REGISTER 8	RXC8	Configures ARINC 429 receive channel 8
0x8029	R/W	RECEIVE CONTROL REGISTER 9	RXC9	Configures ARINC 429 receive channel 9
0x802A	R/W	RECEIVE CONTROL REGISTER 10	RXC10	Configures ARINC 429 receive channel 10
0x802B	R/W	RECEIVE CONTROL REGISTER 11	RXC11	Configures ARINC 429 receive channel 11
0x802C	R/W	RECEIVE CONTROL REGISTER 12	RXC12	Configures ARINC 429 receive channel 12
0x802D	R/W	RECEIVE CONTROL REGISTER 13	RXC13	Configures ARINC 429 receive channel 13
0x802E	R/W	RECEIVE CONTROL REGISTER 14	RXC14	Configures ARINC 429 receive channel 14
0x802F	R/W	RECEIVE CONTROL REGISTER 15	RXC15	Configures ARINC 429 receive channel 15

Fast Access Registers

Memory mapped register access only

* Register is cleared when read (auto clear)

ADDRESS	R/W	REGISTER	MNEMONIC	DESCRIPTION
0x8030	R/W	TRANSMIT CONTROL REGISTER 0	TXC0	Configures ARINC 429 transmit channel 0
0x8031	R/W	TRANSMIT CONTROL REGISTER 1	TXC1	Configures ARINC 429 transmit channel 1
0x8032	R/W	TRANSMIT CONTROL REGISTER 2	TXC2	Configures ARINC 429 transmit channel 2
0x8033	R/W	TRANSMIT CONTROL REGISTER 3	TXC3	Configures ARINC 429 transmit channel 3
0x8034	R/W	TRANSMIT CONTROL REGISTER 4	TXC4	Configures ARINC 429 transmit channel 4
0x8035	R/W	TRANSMIT CONTROL REGISTER 5	TXC5	Configures ARINC 429 transmit channel 5
0x8036	R/W	TRANSMIT CONTROL REGISTER 6	TXC6	Configures ARINC 429 transmit channel 6
0x8037	R/W	TRANSMIT CONTROL REGISTER 7	TXC7	Configures ARINC 429 transmit channel 7
0x8038	R/W	TRANSMIT REPETITION RATE 0	TXRR0	Sets sequence repeat time for ARINC 429 TX0
0x8039	R/W	TRANSMIT REPETITION RATE 1	TXRR1	Sets sequence repeat time for ARINC 429 TX1
0x803A	R/W	TRANSMIT REPETITION RATE 2	TXRR2	Sets sequence repeat time for ARINC 429 TX2
0x803B	R/W	TRANSMIT REPETITION RATE 3	TXRR3	Sets sequence repeat time for ARINC 429 Tx3
0x803C	R/W	TRANSMIT REPETITION RATE 4	TXRR4	Sets sequence repeat time for ARINC 429 Tx4
0x803D	R/W	TRANSMIT REPETITION RATE 5	TXRR5	Sets sequence repeat time for ARINC 429 Tx5
0x803E	R/W	TRANSMIT REPETITION RATE 6	TXRR6	Sets sequence repeat time for ARINC 429 Tx6
0x803F	R/W	TRANSMIT REPETITION RATE 7	TXRR7	Sets sequence repeat time for ARINC 429 Tx7
0x8040	R	TRANSMIT SEQUENCE POINTER 0	TXSP0	Current address of ARINC transmit sequence 0
0x8041	R	TRANSMIT SEQUENCE POINTER 1	TXSP1	Current address of ARINC transmit sequence 1
0x8042	R	TRANSMIT SEQUENCE POINTER 2	TXSP2	Current address of ARINC transmit sequence 2
0x8043	R	TRANSMIT SEQUENCE POINTER 3	TXSP3	Current address of ARINC transmit sequence 3
0x8044	R	TRANSMIT SEQUENCE POINTER 4	TXSP4	Current address of ARINC transmit sequence 4
0x8045	R	TRANSMIT SEQUENCE POINTER 5	TXSP5	Current address of ARINC transmit sequence 5
0x8046	R	TRANSMIT SEQUENCE POINTER 6	TXSP6	Current address of ARINC transmit sequence 6
0x8047	R	TRANSMIT SEQUENCE POINTER 7	TXSP7	Current address of ARINC transmit sequence 7
0x8048	R	Not used		
0x8049	R/W	LOOPBACK	LOOP	Sets Loopback channels
0x804A	R/W	PENDING INTERRUPT ENABLE REGISTER	PIER	Enables Interrupts on $\overline{\text{INT}}$ pin
0x804B	R/W	RECEIVE INTERRUPT ENABLE (LOW)	RIEL	Enables Interrupts (channels 7-0)
0x804C	R/W	RECEIVE INTERRUPT ENABLE (HIGH)	RIEH	Enables Interrupts (channels 15-8)
0x804D	R/W	RECEIVE FIFO INTERRUPT ENABLE (LOW)	FIEL	Enables Receive Interrupts (channels 7-0)
0x804E	R/W	RECEIVE FIFO INTERRUPT ENABLE (HIGH)	FIEH	Enables Receive Interrupts (channels 15-8)
0x804F	R/W	TRANSMIT FIFO INTERRUPT ENABLE	TFIE	Enables Transmit Interrupts
0x8050	R/W	RECEIVE FIFO THRESHOLD VALUE 0	FTV0	Sets flag value for ARINC 429 Receive FIFO 0
0x8051	R/W	RECEIVE FIFO THRESHOLD VALUE 1	FTV1	Sets flag value for ARINC 429 Receive FIFO 1
0x8052	R/W	RECEIVE FIFO THRESHOLD VALUE 2	FTV2	Sets flag value for ARINC 429 Receive FIFO 2
0x8053	R/W	RECEIVE FIFO THRESHOLD VALUE 3	FTV3	Sets flag value for ARINC 429 Receive FIFO 3
0x8054	R/W	RECEIVE FIFO THRESHOLD VALUE 4	FTV4	Sets flag value for ARINC 429 Receive FIFO 4
0x8055	R/W	RECEIVE FIFO THRESHOLD VALUE 5	FTV5	Sets flag value for ARINC 429 Receive FIFO 5
0x8056	R/W	RECEIVE FIFO THRESHOLD VALUE 6	FTV6	Sets flag value for ARINC 429 Receive FIFO 6
0x8057	R/W	RECEIVE FIFO THRESHOLD VALUE 7	FTV7	Sets flag value for ARINC 429 Receive FIFO 7
0x8058	R/W	RECEIVE FIFO THRESHOLD VALUE 8	FTV8	Sets flag value for ARINC 429 Receive FIFO 8
0x8059	R/W	RECEIVE FIFO THRESHOLD VALUE 9	FTV9	Sets flag value for ARINC 429 Receive FIFO 9
0x805A	R/W	RECEIVE FIFO THRESHOLD VALUE 10	FTV10	Sets flag value for ARINC 429 Receive FIFO 10
0x805B	R/W	RECEIVE FIFO THRESHOLD VALUE 11	FTV11	Sets flag value for ARINC 429 Receive FIFO 11
0x805C	R/W	RECEIVE FIFO THRESHOLD VALUE 12	FTV12	Sets flag value for ARINC 429 Receive FIFO 12
0x805D	R/W	RECEIVE FIFO THRESHOLD VALUE 13	FTV13	Sets flag value for ARINC 429 Receive FIFO 13
0x805E	R/W	RECEIVE FIFO THRESHOLD VALUE 14	FTV14	Sets flag value for ARINC 429 Receive FIFO 14
0x805F	R/W	RECEIVE FIFO THRESHOLD VALUE 15	FTV15	Sets flag value for ARINC 429 Receive FIFO 15

ADDRESS	R/W	REGISTER	MNEMONIC	DESCRIPTION
0x8060	R/W	TRANSMIT FIFO THRESHOLD VALUE 0	TFTV0	Sets flag value for ARINC 429 Transmit FIFO 0
0x8061	R/W	TRANSMIT FIFO THRESHOLD VALUE 1	TFTV1	Sets flag value for ARINC 429 Transmit FIFO 1
0x8062	R/W	TRANSMIT FIFO THRESHOLD VALUE 2	TFTV2	Sets flag value for ARINC 429 Transmit FIFO 2
0x8063	R/W	TRANSMIT FIFO THRESHOLD VALUE 3	TFTV3	Sets flag value for ARINC 429 Transmit FIFO 3
0x8064	R/W	TRANSMIT FIFO THRESHOLD VALUE 4	TFTV4	Sets flag value for ARINC 429 Transmit FIFO 4
0x8065	R/W	TRANSMIT FIFO THRESHOLD VALUE 5	TFTV5	Sets flag value for ARINC 429 Transmit FIFO 5
0x8066	R/W	TRANSMIT FIFO THRESHOLD VALUE 6	TFTV6	Sets flag value for ARINC 429 Transmit FIFO 6
0x8067	R/W	TRANSMIT FIFO THRESHOLD VALUE 7	TFTV7	Sets flag value for ARINC 429 Transmit FIFO 7
0x8068	R/W	RX FIFO COUNT 0	RFC0	Current FIFO message count Receiver 0
0x8069	R/W	RX FIFO COUNT 1	RFC1	Current FIFO message count Receiver 1
0x806A	R/W	RX FIFO COUNT 2	RFC2	Current FIFO message count Receiver 2
0x806B	R/W	RX FIFO COUNT 3	RFC3	Current FIFO message count Receiver 3
0x806C	R/W	RX FIFO COUNT 4	RFC4	Current FIFO message count Receiver 4
0x806D	R/W	RX FIFO COUNT 5	RFC5	Current FIFO message count Receiver 5
0x806E	R/W	RX FIFO COUNT 6	RFC6	Current FIFO message count Receiver 6
0x806F	R/W	RX FIFO COUNT 7	RFC7	Current FIFO message count Receiver 7
0x8070	R/W	RX FIFO COUNT 8	RFC8	Current FIFO message count Receiver 8
0x8071	R/W	RX FIFO COUNT 9	RFC9	Current FIFO message count Receiver 9
0x8072	R/W	RX FIFO COUNT 10	RFC10	Current FIFO message count Receiver 10
0x8073	R/W	RX FIFO COUNT 11	RFC11	Current FIFO message count Receiver 11
0x8074	R/W	RX FIFO COUNT 12	RFC12	Current FIFO message count Receiver 12
0x8075	R/W	RX FIFO COUNT 13	RFC13	Current FIFO message count Receiver 13
0x8076	R/W	RX FIFO COUNT 14	RFC14	Current FIFO message count Receiver 14
0x8077	R/W	RX FIFO COUNT 15	RFC15	Current FIFO message count Receiver 15
0x8078	R	BIST CONTROL/STATUS	BISTS	Built-In Self-Test
0x8079	R	BIST FAIL ADDRESS [7:0]	BISTFL	Low-order failing BIST memory address
0x807A	R	BIST FAIL ADDRESS [15:8]	BISTFH	High-order failing BIST memory address and Mode bits
0x807B	R	AUTO-INIT FAIL LS ADDRESS [7:0]	AIFL	Auto-initialization fail address (low-byte)
0x807C	R	AUTO-INIT FAIL MS ADDRESS [15:8]	AIFH	Auto-initialization fail address (high byte)
0x807D	R	Not used		
0x807E	R/W	WATCHDOG CHANNEL MASK [7:0]	WCM	1=masked, 0=active
0x 807F	R/W	WATCHDOG TIMER REGISTER [7:0]	WTR	Defines watchdog time-out period
0x8080	R	TX FIFO COUNT 0	TFC0	Current FIFO message count Transmitter 0
0x8081	R	TX FIFO COUNT 1	TFC1	Current FIFO message count Transmitter 1
0x8082	R	TX FIFO COUNT 2	TFC2	Current FIFO message count Transmitter 2
0x8083	R	TX FIFO COUNT 3	TFC3	Current FIFO message count Transmitter 3
0x8084	R	TX FIFO COUNT 4	TFC4	Current FIFO message count Transmitter 4
0x8085	R	TX FIFO COUNT 5	TFC5	Current FIFO message count Transmitter 5
0x8086	R	TX FIFO COUNT 6	TFC6	Current FIFO message count Transmitter 6
0x8087	R	TX FIFO COUNT 7	TFC7	Current FIFO message count Transmitter 7

HI-3220 SYSTEM CONFIGURATION

Starting at memory address 0x8000, the HI-3220 contains a set of registers that are used to configure the device.

The user needs only to program the HI-3220 configuration registers to completely define the full system operation.

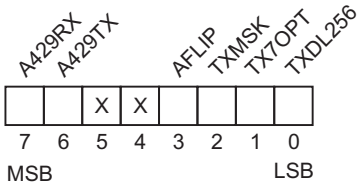
The configuration registers are divided into three categories, as follows;

1. HI-3220 global configuration
2. ARINC 429 Receive channel configuration
3. ARINC 429 Transmit channel configuration

HI-3220 Global Configuration

The following registers define the HI-3220 top-level configuration:

MASTER CONTROL REGISTER
(Address 0x8000 - read/write, 0x8001 - read)



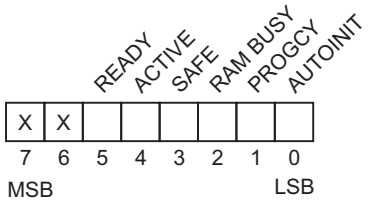
Bit	Name	R/W	Default	Description
7	A429RX	R/W	0	This bit must be set to a “1” to allow the HI-3220 to receive ARINC 429 data on any of the sixteen ARINC 429 receive channels. If set to a zero, the HI-3220 will not respond to any ARINC 429 receive bus, regardless of the state of the ARINC 429 Receive channel Control Registers.
6	A429TX	R/W	0	This bit must be set to a “1” to allow the HI-3220 to transmit ARINC 429 data on any of the eight channels. If set to a zero, the HI-3220 will not output ARINC 429 data and the ARINC 429 transmit sequencers will remain in their reset state. Completes the current TX message* before entering the reset state.
5	-	R/W	0	Not Used
4	-	R/W	0	Not Used
3	AFLIP	R/W	0	When set to a “1”, this bit switches the bit order of the ARINC 429 label byte in both receive and transmit channels. (See ARINC 429 Bit Ordering section)
2	TXMSK	R/W	0	When set to a “1”, this bit prevents the external transmission of self-test loop-back data for any transmitter when set to loopback mode.
1	TX7OPT	R/W	0	Defines the meaning of Receive Status bit 7 in every ARINC 429 receive data block. When TX7OPT = “1”, Status bit 7 is NEWHOST. When TX7OPT =”0”, Status bit 7 is NEWTX7. See ARINC receive data block description for details.
0	TXDL256	R/W	0	When TXDL256 is “0” transmitter 0 and transmitter 1 descriptor length is 128. When TXDL256 is “1” transmitter 0 and transmitter 1 descriptor length is 256.

* A message is defined as one ARINC 429 32-bit message. A frame is defined as multiple messages programmed in a Transmit scheduler.

HI-3220 Operational Status Information

The Master Status Register may be read at any time to determine the current operational state of the HI-3220:

MASTER STATUS REGISTER
(Address 0x8002)



Bit	Name	R/W	Default	Description
7	-	R	0	Not used
6	-	R	0	Not Used
5	READY	R	0	This bit is high, when the READY output pin is high, indicating that the part is able to accept and respond to host CPU SPI commands
4	ACTIVE	R	0	This bit is high after the RUN pin is asserted and the HI-3220 is in normal operating mode.
3	SAFE	R	0	This bit goes high when the part enters safe mode as a result of a Built-in Self-test fail or auto-initialization fail.
2	RAM BUSY	R	0	This is high during the time the RAM Integrity Check is running and RAM is clearing
1	PROGCY	R	0	Indicates that the HI-3220 is currently in the EEPROM programming cycle. Note that READY stays low until the cycle is complete.
0	AUTOINIT	R	0	The HI-3220 is currently loading internal memory, registers and look-up tables from the Auto-initialization EEPROM

ARINC 429 RECEIVE OPERATION

The HI-3220 can receive ARINC 429 messages from up to sixteen ARINC 429 receive buses. Analog line receivers are included on-chip for HI-3220, HI-3221 and HI-3222. Each receiver input line requires only an external 40 kOhm resistor to meet lightning protection DO-160 level 3. The HI-3225 and HI-3226 have digital receiver inputs and are used with off-chip ARINC 429 line receivers.

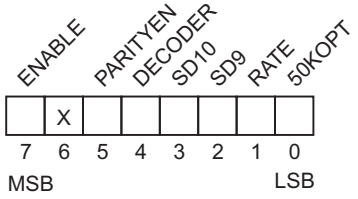
Non-Annunciated Fault Protection

A fault condition may cause one side of the ARINC 429 bus transmission line to become open or shorted to ground. When this condition occurs, it is desirable that the receiver rejects the incoming signal. The receiver thresholds are set above 5.5V (the maximum signal level under this one-wire fault condition), so that signals associated with this non-annunciated fault condition are rejected. This fault protection will also ensure that voltage levels undefined by the ARINC 429 Specification (the voltage range between ±2.5V and ±6.5V) are also rejected.

ARINC 429 Receive Channel Configuration

Each of the sixteen possible ARINC 429 Receive channels is configured using its own Control Register. Register address 0x8020 controls ARINC 429 Receive channel #0, register address 0x8021 controls channel #1 and so on.

RECEIVE CONTROL REGISTER 0 - 15
(Address 0x8020 - 0x802F)



Bit	Name	R/W	Default	Description
7	ENABLE	R/W	0	This bit must be set to a “1” to enable ARINC 429 data reception on this channel.
6	-	R/W	0	Not used
5	PARITYEN	R/W	0	When this bit is a one, the 32nd received ARINC bit is overwritten with a parity flag. The flag bit is set to a zero when the received ARINC word, including its parity bit has an odd number of ones. When PARITYEN is a zero, all 32-bits are received without parity checking.
4	DECODER	R/W	0	When DECODER is a “1”, bits 9 and 10 of ARINC 429 words received on this channel must match the SD9 and SD10 bits in the register. ARINC words received that do not match the SD conditions are ignored.
3	SD10	R/W	0	If DECODER is set to a “1”, then this bit must match the received ARINC word bit 10 for the word to be accepted.
2	SD9	R/W	0	If DECODER is set to a “1”, then this bit must match the received ARINC word bit 9 for the word to be accepted.
1	RATE	R/W	0	Selects the ARINC 429 bit rate for the ARINC 429 receive channel. A “0” selects high-speed (100Kb/s) and a “1” selects low-speed (12.5Kb/s). (See 50KOPT bit for non-standard 50kb/s high-speed operation)
0	50KOPT	R/W	0	When set to a “1”, the RATE bit is ignored and ARINC 429 data rate is set to 50 Kb/s

ARINC 429 Received Data Management

The HI-3220 supports up to sixteen ARINC 429 receive buses using on-chip receivers to handle the protocol validation. The sixteen Receive Control Registers, RXC0 - 15, define the characteristics of each receive channel.

The ARINC 429 receive function of the HI-3220 is activated by setting the A429RX bit in the Master Control Register.

When an ARINC 429 message is received by the HI-3220 on any bus, it is checked for protocol compliance. Messages with incorrect encoding are rejected. **NOTE:** Messages with incorrect parity will be stored.

The HI-3220 allocates 16K bytes of on-chip memory for storing ARINC 429 received data. The memory is organized by channel number and ARINC 429 label value. Four bytes of memory are dedicated to each channel / label to store the 32-bit ARINC 429 message.

A look-up table is used to enable an interrupt on receipt of a new ARINC 429 message. Look-up table bit positions preloaded by the user with a “1” will cause an Interrupt to be generated.

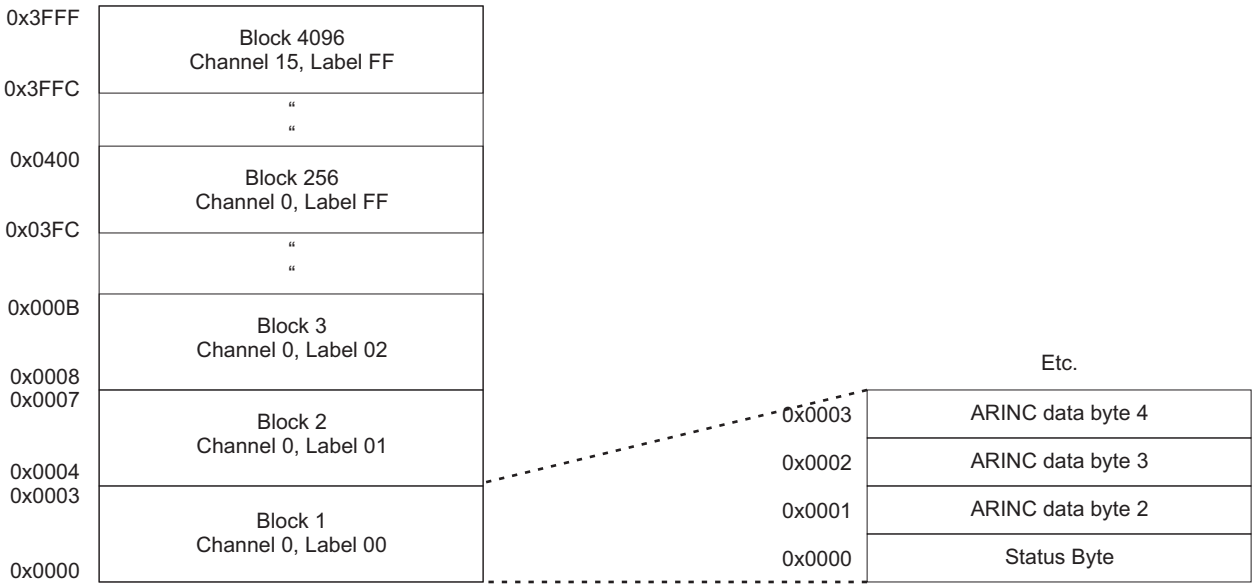
When a message is received that triggers an Interrupt, that channel’s Interrupt bit is set in the Receive Pending Interrupt Register. If this bit is set in the Receive Interrupt Enable Register, the $\overline{INT}$ output pin is asserted. The label number of the ARINC 429 message causing the interrupt is loaded into that channel’s Receive Interrupt Address Register (IAR0 - IAR15).

Because the ARINC Receive Memory is organized by label value, it is not necessary to store the received label value (first eight bits of the ARINC message) in the memory. Instead, the first byte is used to store a status byte.

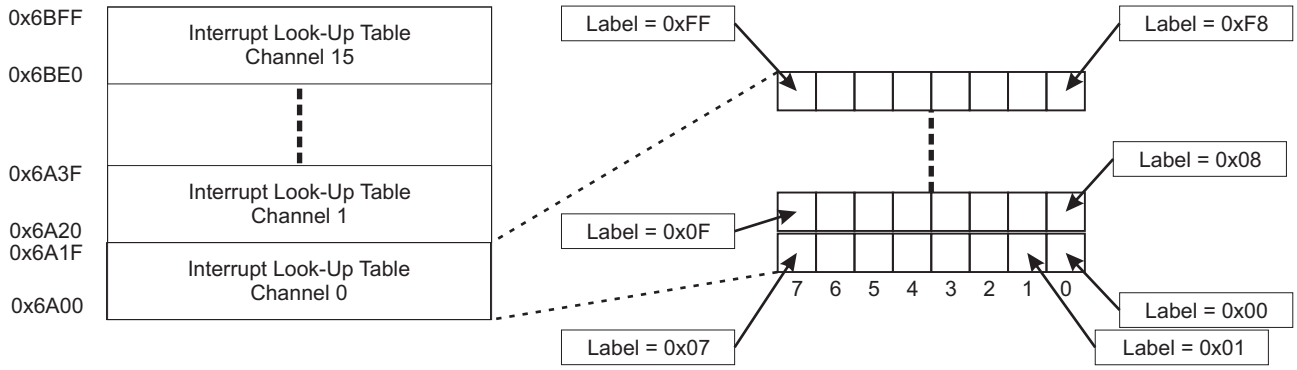
The eight active bits of the status byte are set to “1” when a new ARINC word is stored in the memory. These bits flag the ARINC word as new when the location is interrogated by the host CPU or any of the seven ARINC 429 transmit schedulers. (Note: Transmitter 7 operates differently from the other 7 transmitters. The corresponding “New” bit in the receive status word may be selected between new data for Tx7 or new data for Host CPU as defined by the global TX7OPT bit in the Master control Register.)

The ARINC 429 Receive memory is not filtered so messages will load unconditionally when received if the channel is enabled in the Receive Control Register. Message blocks are single buffered so the host should read the message before a potential overwrite.

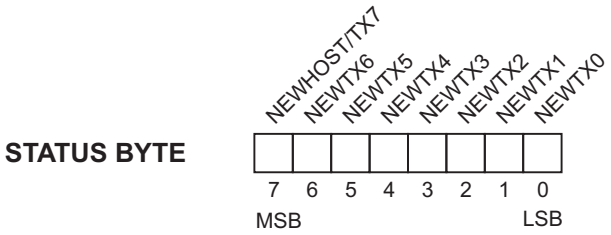
ARINC 429 Received Data Memory Organization



ARINC 429 Received Data Interrupt Look-Up Table



ARINC 429 Received Data Status Byte Definition



Bit	Name	R/W	Default	Description
7	NEWHOST/TX7	R/W	0	The meaning of this bit depends upon the setting of TX7OPT bit in the Master Control Register: When TX7OPT is a '"1", this bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the **host CPU** executes SPI instruction 0xC8 or 0xD0 to read the block. When TX7OPT is a "0", this bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the ARINC 429 **Transmit scheduler #7** reads any bytes from the block.
6	NEWTX6	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #6 reads any bytes from the block.
5	NEWTX5	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #5 reads any bytes from the block.
4	NEWTX4	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #4 reads any bytes from the block.
3	NEWTX3	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #3 reads any bytes from the block.
2	NEWTX2	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when theARINC 429 Transmit scheduler #2 reads any bytes from the block.
1	NEWTX1	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when theARINC 429 Transmit scheduler #1 reads any bytes from the block.
0	NEWTX0	R/W	0	This bit is set when a new ARINC 429 word is received and stored in this block. It is reset when the ARINC 429 Transmit scheduler #0 reads any bytes from the block.

ARINC 429 Received Data Log FIFO

Sixteen FIFOs that are each 64 messages deep.

A 4K x 8 block of memory located between 0x7000 and 0x7FFF is reserved for a set of sixteen ARINC 429 received data FIFOs. There is one FIFO for each ARINC 429 received data channel. Each FIFO can hold up to 64 ARINC 429 32-bit messages.

A look-up table driven filter defines which ARINC 429 messages are stored in each FIFO. The look-up table is initialized by the user with a "1" for each bit position corresponding to a selected channel / label combination. The look-up table is located at memory address 0x6800.

When a new ARINC 429 message is received that meets the programmed conditions for acceptance (Enable look-up table bit = "1"), it is written into the channel's Receive Data FIFO. The contents of the FIFO may be read by the host CPU using dedicated FIFO read SPI Instructions.

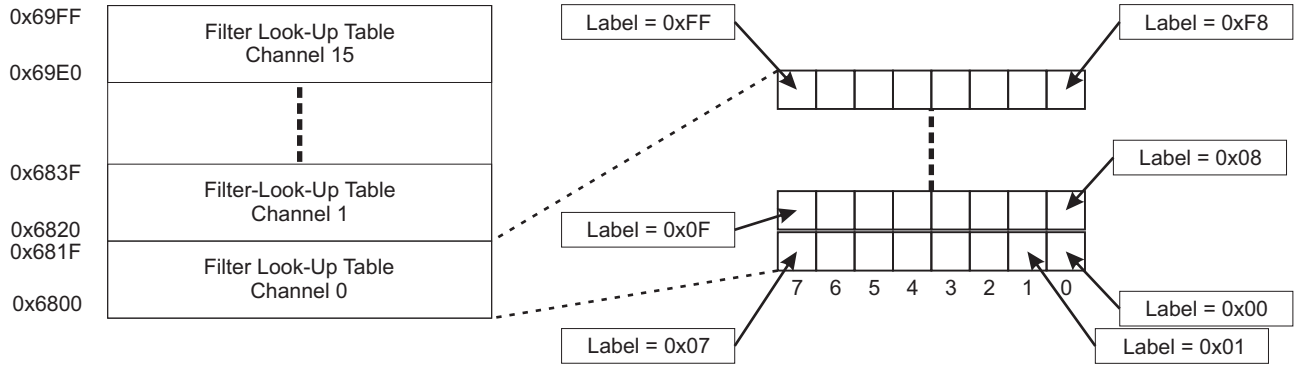
The status of each channel's FIFOs is monitored by a FIFO status register: RECEIVE FIFO THRESHOLD FLAG. Each bit of the register reflects the current status of each FIFO.

The FIFOs are empty following reset. All three status registers are cleared. For each channel, a user-defined RECEIVE FIFO THRESHOLD VALUE register defines the point where the RECEIVE FIFO THRESHOLD FLAG bit is set. That occurs when the number of words in the FIFO exceed the threshold value. Once a FIFO is full, (i.e. contains 64 messages), a received message will be written to the FIFO and the oldest message lost, maintaining a FIFO full count of 64.

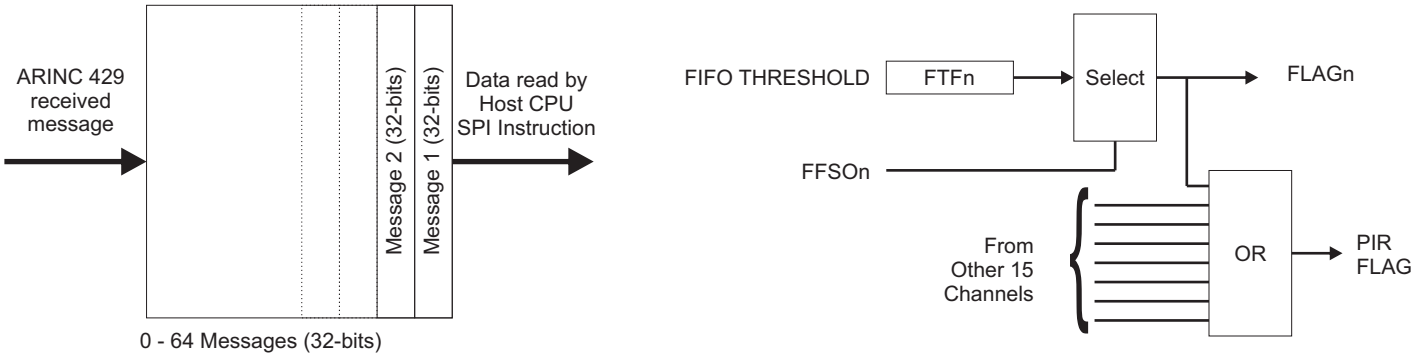
The user may generate an Interrupt by enabling one the FIFO status register bits to assert the FLAG bit in the Pending Interrupt Register. Receive Control Register bit 6 enables the triggering of the FLAG interrupt.

The FIFO feature is particularly useful if the application wishes to accumulate sequential ARINC 429 messages of the same label value before reading them. The regular ARINC 429 receive data memory will, of course, overwrite messages of the same label value if a new message is received before the host CPU extracts the data.

ARINC 429 Received FIFO Data Enable Look-Up Table

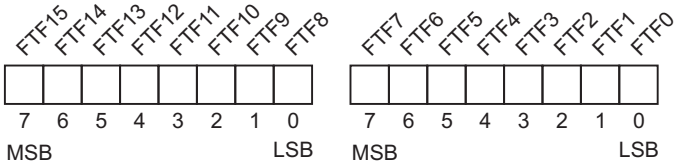


ARINC 429 Received Data FIFO (x16)



ARINC 429 Received Data FIFO Registers

RECEIVE FIFO THRESHOLD REGISTER HIGH / LOW
(Address 0x800E, 0x800D)

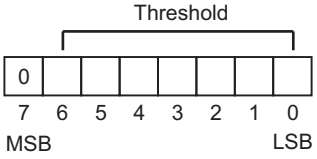


Bit	Name	R/W	Default	Description
15:0	FTF[15:0]	R	0	Bits are set to "1" if the corresponding channel's receive FIFO contains > threshold number of ARINC 429 messages.

Receive FIFO Threshold Value Registers

Each Receiver has its own Receive FIFO Threshold Register

FIFO THRESHOLD VALUE n
(Address 0x8050 - 0x805F)



Default = 0x00

Threshold Value	Description
0000000	Threshold flag is set if at least 1 message is in FIFO (FIFO is not empty) - Default
0000001	Threshold flag is set if more than one message is in the FIFO
0000010	Threshold flag is set if more than two messages are in the FIFO
0000011	Threshold flag is set if more than three messages are in the FIFO
⋮	
0010000	Threshold flag is set if more than sixteen messages are in the FIFO
⋮	
1111111	Threshold flag is set if 64 messages are in the FIFO (FIFO is full)

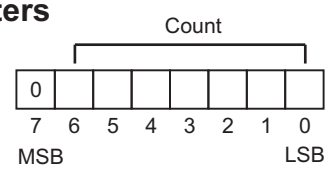
Receive FIFO Interrupt Enable

RECEIVE FIFO THRESHOLD REGISTER HIGH / LOW
(Address 0x804E, 0x804D)

Bit	Name	R/W	Default	Description
15:0	FIE[15:0]	R/W	0	Interrupts are enabled for receive channels where the corresponding bit is set.

Receive FIFO Count Registers

RECEIVE FIFO COUNT(0-15)
(Address 0x8068 - 0x8077)



Default = 0x00

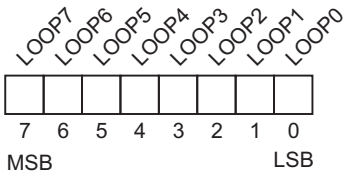
Bit	Name	R/W	Default	Description
7:0	RFCn[7:0]	R/W	0	Reads number of words held in Receive FIFO FIFOs may be emptied by writing 0xA5 to this register.

ARINC 429 Loop-back Self-Test

The HI-3220 includes an ARINC 429 loop-back feature, which allows users to exercise the ARINC 429 transmit and receive channels for self-test purposes. The ARINC 429 Loop-Back register, LOOP, defines which receiver channels are in loop-back mode. When a "1" is programmed in the LOOP bit position for a receiver pair, then their ARINC 429 bus connection to the external pins is broken and instead the input is connected to one of the eight ARINC 429 transmit channels. Transmit channel 0 is connected to receive channels 0 and 1, transmit channel 1 is connected to receive channels 2 and 3, and so on.

When in loop-back mode, incoming ARINC 429 messages are ignored by the HI-3220. When running in loop-back mode the ARINC 429 transmit pins may be disabled by setting the TXMSK bit high in the Master Control Register. This prevents test messages from being output to the external ARINC 429 transmit buses.

SELF-TEST LOOPBACK
(Address 0x8049)



Bit	Name	R/W	Default	Description
7	LOOP7	R/W	0	This bit is set to "1" to loop-back transmit channel 7 to receivers 14 and 15
6	LOOP6	R/W	0	This bit is set to "1" to loop-back transmit channel 6 to receivers 12 and 13
5	LOOP5	R/W	0	This bit is set to "1" to loop-back transmit channel 5 to receivers 10 and 11
4	LOOP4	R/W	0	This bit is set to "1" to loop-back transmit channel 4 to receivers 8 and 9
3	LOOP3	R/W	0	This bit is set to "1" to loop-back transmit channel 3 to receivers 6 and 7
2	LOOP2	R/W	0	This bit is set to "1" to loop-back transmit channel 2 to receivers 4 and 5
1	LOOP1	R/W	0	This bit is set to "1" to loop-back transmit channel 1 to receivers 2 and 3
0	LOOP0	R/W	0	This bit is set to "1" to loop-back transmit channel 0 to receivers 0 and 1

ARINC 429 Bit ordering

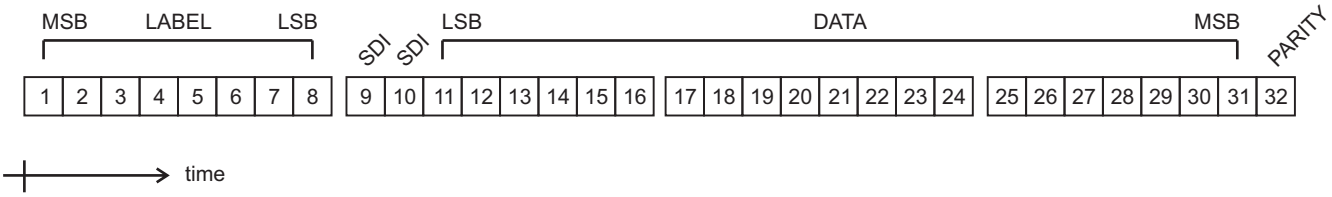
ARINC 429 messages consist of a 32-bit sequence as shown below. The first eight bits that appear on the ARINC 429 bus are the label byte. The next twenty three bits comprise a data field which presents data in a variety of formats defined in the ARINC 429 specification. The last bit transmitted is an odd parity bit.

The HI-3220 stores the received message as four bytes. The bytes are stored in memory in little-endian order. That is to say, the label byte (or status byte) is stored at the lowest memory address, the byte representing received bits 9 through 16 is stored at the next address, the byte representing bits 17 through 24 at the next address and the byte representing bits 25 though 32 at the highest address.

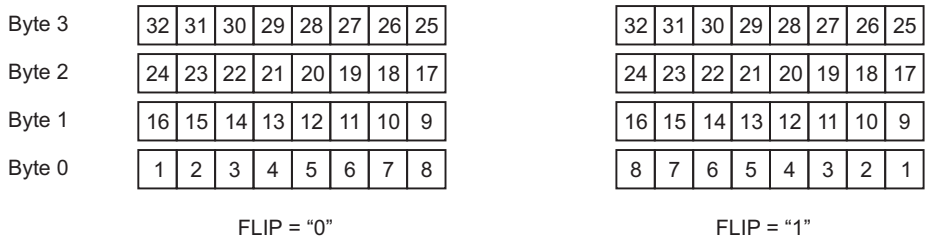
The ARINC 429 specifies the MSB of the label as ARINC bit 1. Conversely, the data field MSB is bit 31. So the bit significance of the label byte and data fields are opposite.

The HI-3220 may be programmed to “flip” the bit ordering of the label byte as soon as it is received and immediately prior to transmission. This is accomplished by setting the FLIP bit to a “1” in the Master Control Register. Note that once the label byte has been flipped, the HI-3220 handles the flipped data byte “post-flip” for the purpose of label interrupt matching, filtering and storage.

ARINC 429 Message as received / transmitted on the ARINC 429 serial bus



ARINC 429 Message as stored in HI-3220 memory



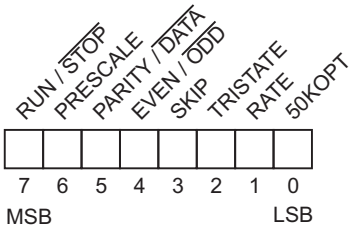
ARINC 429 TRANSMIT OPERATION

The HI-3220 has up to eight on-board ARINC 429 transmit channels which directly drive ARINC 429 differential line drivers such as the Holt HI-8596. ARINC 429 words may be written to the transmitters either directly, using an SPI instruction through a transmit FIFO, or be generated automatically using the eight ARINC 429 message schedulers.

ARINC 429 Transmit Channel Configuration

Each of the eight available ARINC 429 Transmit channels is configured using its own register. Register address 0x8030 controls ARINC 429 Transmit channel #0, register address 0x8031 controls channel #1 and so on. The TXCn registers may be written or read at any time.

TRANSMIT CONTROL REGISTER 0 - 7
(Address 0x8030 - 0x8037)



Bit	Name	R/W	Default	Description
7	RUN / $\overline{\text{STOP}}$	R/W	0	When zero, transmission from this ARINC 429 transmit channel is suspended after the currently transmitting frame of messages is sent. When this bit is taken high, transmission starts at the beginning of the descriptor table for this channel. The RUN bit can be used for one-time frame transmission (see page 25).
6	PRESCALE	R/W	0	This bit sets the LSB value for this transmit channel's Repetition Rate Counter. A "0" set the LSB value to 10 ms, and a "1" selects 1 ms.
5	PARITY / $\overline{\text{DATA}}$	R/W	0	When this bit is a one, the 32nd transmitted ARINC bit is overwritten with a parity flag. When this bit is a zero, all 32-bits are transmitted as data.
4	EVEN / $\overline{\text{ODD}}$	R/W	0	When PARITY / DATA is a "1", this bit defines whether th 32nd transmitted bit is set for Even or Odd Parity. A "1" selects even parity and a "0" selects odd parity.
3	SKIP	R/W	0	When set to a "1", instructs the transmit sequencer to wait for the next Repetition Rate Counter rollover before beginning a new transmission cycle. A "0" causes an immediate restart of the cycle following completion of the prior cycle.
2	TRISTATE	R/W	0	When set to a "1" the TXP and TXN output pins for this transmit channel are both forced to a high state. When used with an ARINC 429 line driver that supports tri-state operation, such as the Holt HI-8596, the ARINC 429 databus is forced to a high-impedance state. No ARINC 429 transmission from the HI-3220 occurs when TRISTATE is set.
1	RATE	R/W	0	Selects the transmission rate for the ARINC 429 transmit channel. A "0" selects high-speed (100Kb/s) and a "1" selects low-speed (12.5Kb/s). High-speed data-rate is switched to 50Kb/s when the 50KOPT bit is set.
0	50KOPT	R/W	0	When this bit is set, the RATE setting is overridden and the ARINC 429 data rate defaults to 50 kb/s

ARINC 429 Transmit Scheduler

Each ARINC 429 transmit channel 0 through 7 has its own transmit controller. The controller is user-programmed to output ARINC labels in a predefined order and repetition rate. A sequence of up to 128 ARINC 429 labels may be transmitted before repeating the sequence.

A descriptor table with up to 128 entries (descriptors) is compiled by the user to define the sequence of ARINC 429 messages transmitted on each channel. When the RUN/$\overline{\text{STOP}}$ bit in the Transmit Control Register is asserted, the controller compiles the first 32-bit ARINC word from the instructions given by the first descriptor and then transmits it. A Transmit Sequence Pointer then increments to the next descriptor in the table and the process is repeated for Descriptor number 2.

ARINC 429 messages continue to be compiled and transmitted until the last descriptor in the table. The end of the table is marked by a special descriptor if not all 128 entries are needed. The Sequence Pointer is then reset to zero.

A Repetition Rate Counter is used to time the start of the next transmission cycle.

The user is responsible for construction of the descriptor table and for setting the Repetition Rate prior to asserting RUN/$\overline{\text{STOP}}$. Facilities exist for immediate cycle repetition and for single-cycle operation.

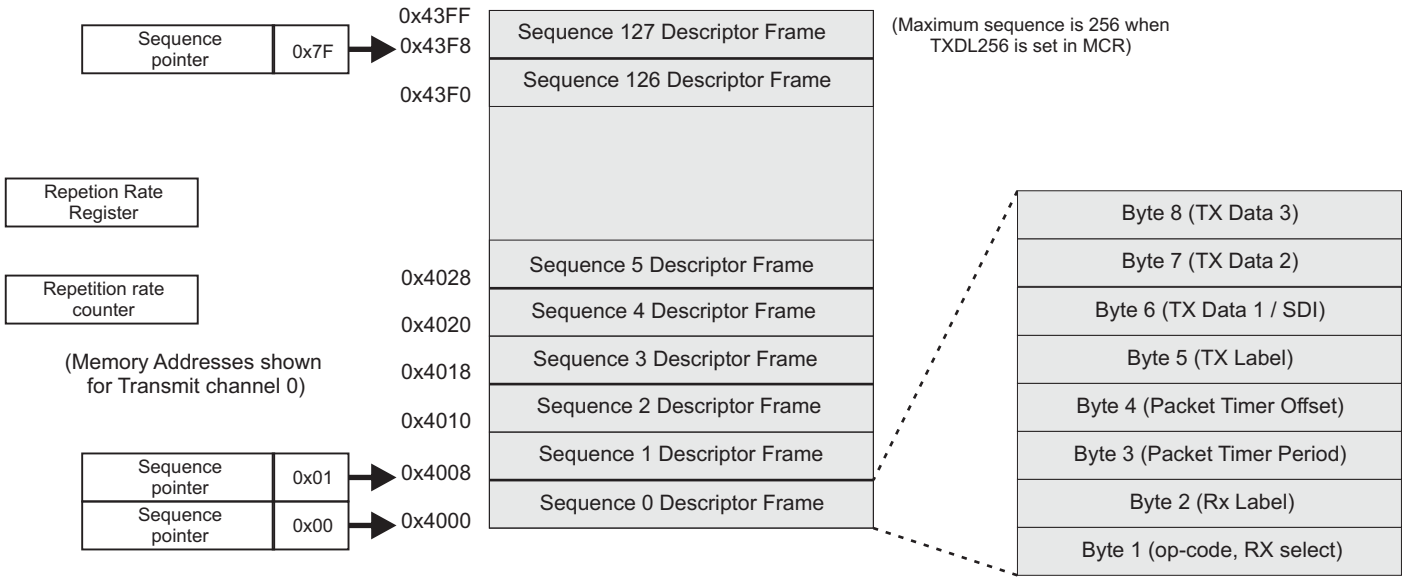
The byte content of each ARINC 429 message transmitted is user defined by the descriptor contents. Data bytes may be sourced from the host CPU / auto-initialization EEPROM (immediate data) or from the ARINC 429 receive memory (indexed). This allows received ARINC data to be re-transmitted on another bus with or without filtering, label byte re-assignment, or data modification. It also allows data from multiple ARINC 429 receive buses to be re-packetized into new ARINC 429 transmitted messages.

Conditional transmission control allows sequenced words to be skipped if no new data is available.

Each transmit channel is independently configured with its own Transmit Control Register, TXCR0-7, as previously described.

During scheduler transmissions de-asserting the RUN bit in the Transmit Control register will stop transmissions after the current frame. To immediately stop after the current message transmission, either de-assert the A429TX bit in the Master Control register or set the RUN input pin low. Keep in mind that setting the RUN input pin low will stop all transmitters after the current message completes.

Transmit Descriptor table

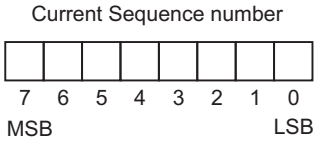


The value of each ARINC 429 label transmitted in the sequence is defined by its eight-byte descriptor. The first two bytes define the source of data for transmission. The next two bytes define transmission timing parameters. The last four bytes contain data to be transmitted as defined by the op-code.

Different op-codes allow the data source to be host CPU populated values, or values from specific locations within the ARINC 429 receive memory.

The construction of Descriptor frame bytes are described in the next section.

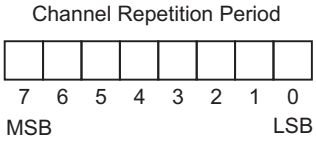
TRANSMIT SEQUENCE POINTERS 0 - 7
(Address 0x8040 - 0x8047)



The transmit sequence pointer is set to zero on Master Reset. Once the Control Register RUN / $\overline{\text{STOP}}$ bit goes high, sequence execution begins at sequence count zero. After the first word is sent, the pointer is incremented by one descriptor (counts descriptor frames).

This continues until the programmed sequence is complete. The sequence pointer is then reset to the beginning of the descriptor table and program execution begins as soon as the channel repetition rate counter time elapses.

REPETITION RATE REGISTER 0 - 7
(Address 0x8038 - 0x803F)



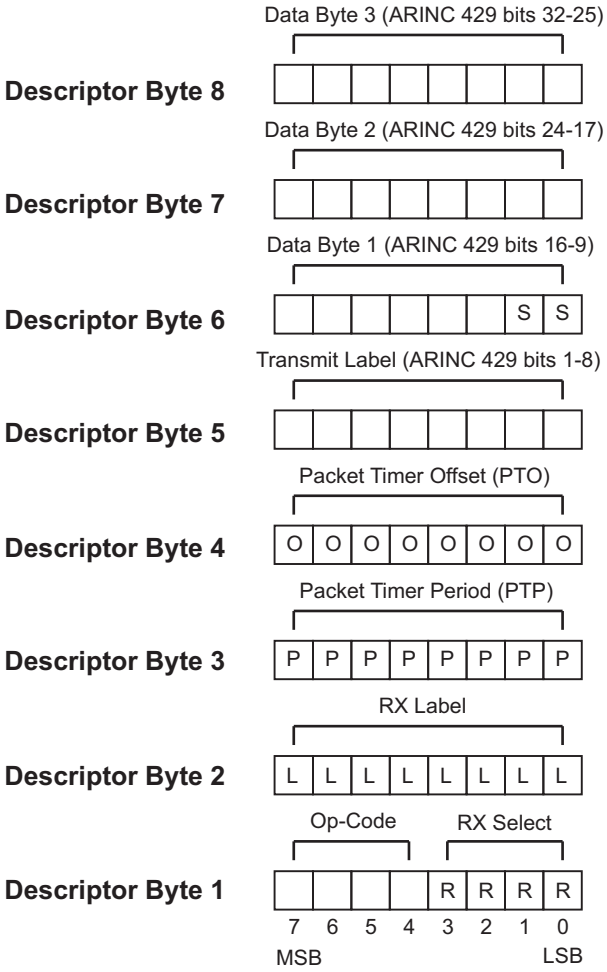
The Repetition rate register value defines the time interval between successive starts of the programmed transmit sequence for each ARINC 429 transmit channel. The value is set in binary, with the LSB representing 10 ms when the PRESCALE bit in this channel's Transmit Control Register is a zero. When the PRESCALE bit is set to a one, the LSB represents 1 ms. Repetition rate time periods may therefore be set from 0 ms to 2.55 seconds

If the repetition rate is shorter than the minimum time needed to transmit all ARINC 429 words in the sequence (but not zero), the transmit sequence will begin again immediately if the Control Register SKIP bit is a zero. If the SKIP bit is a one, the sequencer will wait until the next rollover of the Repetition Rate Counter before starting a new cycle.

One-time Sequence Transmission

To transmit a frame of messages only once (One-time transmission), set the RUN bit high then low in the Transmit Control Register before the current frame ends.

ARINC 429 TRANSMIT DESCRIPTOR



Op-Code	Description
0000	End of sequence. When op-code 0000 is encountered by the sequencer before it reaches sequence number 127, the sequencer resets to zero and begins the next transmission cycle starting at descriptor number 0 as soon as the repetition rate counter rolls over. Note that the descriptor table is cleared following Master Reset, (Mode 0 - Mode 3) so no ARINC 429 transmissions are possible until the sequence table has been configured.
0001	Delay. Delays the transmission of the next ARINC 429 word by the value of the Packet Timer Period byte in milliseconds. Except for PTP, the value of other descriptor bytes are “don’t care”. A delay with PTP=0 is equivalent to a no-op.
0010	Immediate data. The value contained descriptor bytes 5 through 8 is transmitted on the ARINC 429 bus.
0011	Immediate data conditional. Bytes 5 through 8 are transmitted only if the RX packet selected by receive channel “RRRR” and label ”LLLLLLLL” has new data. The NEWTXn bit is sampled to determine if this location has new data. NEWTXn is then reset.
0100	Indexed data. Transmits descriptor byte 5 as the label appended to the 24 bits of data stored at Receive channel “RRRR” and label “LLLLLLLL”.
0101	Indexed data conditional. Same as op-code 0100, but transmits only if new data exists at the specified location as indicated by the NEWTXn bit. NEWTXn is then reset.
0110	Indexed SDI. Same as op-code 0100 except the ARINC 429 SDI bits are replaced with those loaded in descriptor byte 6, bits 10-9 (”SS”).
0111	Indexed SDI conditional. Same as op-code 0101 except the ARINC 429 SDI bits are replaced with those loaded in descriptor byte 6, bits 10-9 (”SS”).
1XXX	Reserved. Do not use.

PACKET TIMER

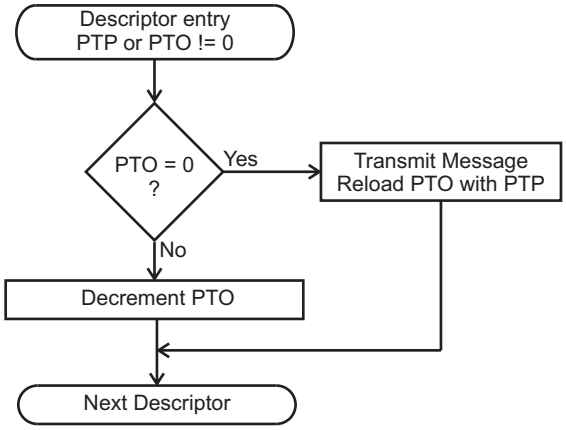
Packet Timer Period (PTP) and Packet Timer Offset (PTO) are used in conjunction with the Repetition Rate counter to make individual ARINC 429 word transmission timing possible. The Repetition Rate Counter is programmable from 0 ms to 2.55 s as defined by the PRESCALE bit and Repetition Rate Register contents.

If PTO is zero, the word message is transmitted and PRO is reloaded with PTP. If PTO is non-zero it is decremented and the word message is skipped.

An initial non-zero value in PTO provides an offset prior to the first transmission. The period between subsequent transmissions is controlled by PTP. A value of all 1's in PTP indicates that the word will be transmitted once (one-shot mode).

See examples 1-4 below.

Descriptor flow using the Packet Timer



Repetition Rate (RR) = Repetition Rate Timer (10ms or 1ms) x Repetition Rate Register value.

Descriptor message rate = (PTP+1) x Repetition Rate.

When PTO is non-zero, the message is skipped until PTO = 0. Each pass of the descriptor block decrements the PTO count.

Example 1.

RR = 80 ms, PTP = 3 and PTO = 1
Transmission in 2nd pass and subsequently every 4th pass (every 320 ms).

Example 2.

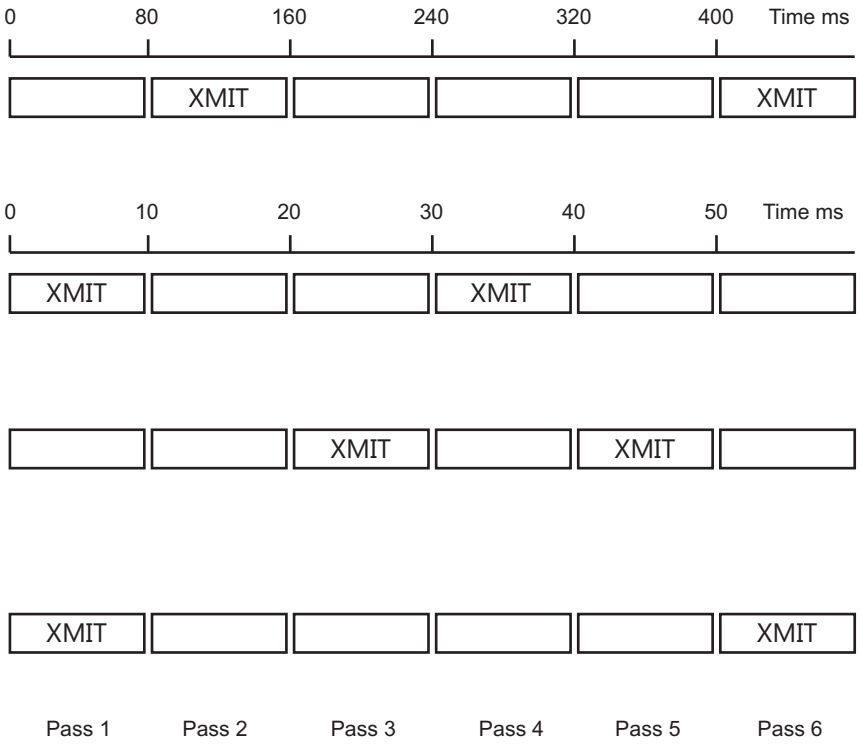
RR = 10 ms, PTP = 2 and PTO = 0
Transmission in 1st pass and subsequently every third pass (every 30 ms).

Example 3.

RR = 10 ms, PTP = 1 and PTO = 2
Transmission in 3rd pass and subsequently every 2nd pass (every 20 ms).

Example 4.

RR = 1 (10 ms), PTP = 4 and PTO = 0
Transmission in 1st pass and subsequently every 5th pass (60 ms).



ARINC 429 Immediate Transmit Option

The Host CPU may instruct the HI-3220 to transmit an ARINC 429 message immediately using a special SPI command. The SPI command selects the transmit channel and provides the four bytes of data to be sent as a 32-bit ARINC 429 message.

A 32-message deep FIFO for each transmitter allows multiple ARINC 429 messages to be queued for transmission reducing host CPU overhead.

Each transmit FIFO may be monitored by a flag which is set when the FIFO contains less than a user-defined number of messages. This threshold value is set in each transmitter's Transmit FIFO Threshold Register. The Flags are accessed via the Transmit Flags register.

If the transmit channel's sequencer is not running (TCR bit RUN/$\overline{\text{STOP}}$ = "0"), or the sequencer is waiting for the repetition rate counter to rollover, then the new ARINC 429 message is transmitted without delay.

The Transmit Interrupt Enable Register is used to generate an interrupt at INT when selected Flags are enabled.

If the transmit sequencer for the selected channel is active, then the new message(s) in the FIFO are transmitted as soon as the current message has been sent. The sequencer then resumes operation at the next location in the queue.

Both the RUN input and the Master Control Register A429TX bit must be high to enable any ARINC 429 transmission.

Host writes to a full FIFO will be ignored.

Table 2 lists the host CPU SPI instruction format.

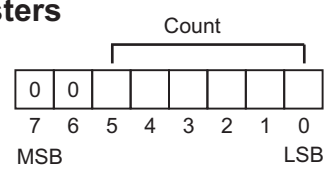
TRANSMIT FIFO THRESHOLD REGISTER (Address 0x800F)

Bit	Name	R/W	Default	Description
7	TFR7	R	1	This bit is set when Transmit FIFO 7 contains less than the programmed threshold number of ARINC 429 messages .
6	TFR6	R	1	This bit is set when Transmit FIFO 6 contains less than the programmed threshold number of ARINC 429 messages .
5	TFR5	R	1	This bit is set when Transmit FIFO 5 contains less than the programmed threshold number of ARINC 429 messages .
4	TFR4	R	1	This bit is set when Transmit FIFO 4 contains less than the programmed threshold number of ARINC 429 messages .
3	TFR3	R	1	This bit is set when Transmit FIFO 3 contains less than the programmed threshold number of ARINC 429 messages .
2	TFR2	R	1	This bit is set when Transmit FIFO 2 contains less than the programmed threshold number of ARINC 429 messages .
1	TFR1	R	1	This bit is set when Transmit FIFO 1 contains less than the programmed threshold number of ARINC 429 messages .
0	TFR0	R	1	This bit is set when Transmit FIFO 0 contains less than the programmed threshold number of ARINC 429 messages .

NOTE: This register may be read using fast-access SPI Command Op-code 00111100 (0x3C).

Transmit FIFO Count Registers

TRANSMIT FIFO COUNT(0-7) (Address 0x8080 - 0x8087)



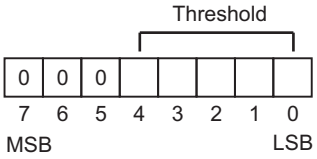
Default = 0x00

Bit	Name	R/W	Default	Description
7:0	TFCn[7:0]	R	0	Read current transmit message count.

Transmit FIFO Threshold Value Registers

Each Transmitter has its own Transmit FIFO Threshold Register

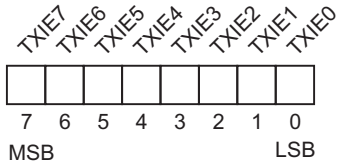
TRANSMIT FIFO THRESHOLD VALUE n
(Address 0x8060 - 0x8067)



Default = 0x00

Threshold Value	Description
00000	Threshold flag is set there are no messages in FIFO (FIFO is empty) (Default)
00001	Threshold flag is set if no more than one message is in the FIFO
00010	Threshold flag is set if no more than two messages are in the FIFO
00011	Threshold flag is set if no more than three messages are in the FIFO
⋮	
10000	Threshold flag is set if no more than sixteen messages are in the FIFO
⋮	
11111	Threshold flag is set if no more than 31 messages are in the FIFO (FIFO is not full)

TRANSMIT FLAG INT ENABLE
(Address 0x804F)



Bit	Name	R/W	Default	Description
7	TXIE7	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 7 FIFO Flag is set. The TXFLAG bit in the Pending Interrupt Register will be set and the $\overline{INT}$ pin will be asserted if the TXFLAGIE bit is set in the Pending Interrupt Enable Register.
6	TXIE6	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 6 FIFO Flag is set. The TXFLAG bit in the Pending Interrupt Register will be set and the $\overline{INT}$ pin will be asserted if the TXFLAGIE bit is set in the Pending Interrupt Enable Register.
5	TXIE5	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 5 FIFO Flag is set. The TXFLAG bit in the Pending Interrupt Register will be set and the $\overline{INT}$ pin will be asserted if the TXFLAGIE bit is set in the Pending Interrupt Enable Register.
4	TXIE4	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 4 FIFO Flag is set. The TXFLAG bit in the Pending Interrupt Register will be set and the $\overline{INT}$ pin will be asserted if the TXFLAGIE bit is set in the Pending Interrupt Enable Register.
3	TXIE3	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 3 FIFO Flag is set. The TXFLAG bit in the Pending Interrupt Register will be set and the $\overline{INT}$ pin will be asserted if the TXFLAGIE bit is set in the Pending Interrupt Enable Register.
2	TXIE2	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 2 FIFO Flag is set. The TXFLAG bit in the Pending Interrupt Register will be set and the $\overline{INT}$ pin will be asserted if the TXFLAGIE bit is set in the Pending Interrupt Enable Register.
1	TXIE1	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 1 FIFO Flag is set. The TXFLAG bit in the Pending Interrupt Register will be set and the $\overline{INT}$ pin will be asserted if the TXFLAGIE bit is set in the Pending Interrupt Enable Register.
0	TXIE0	R/W	0	Setting this bit generates an interrupt when ARINC 429 Transmitter 0 FIFO Flag is set. The TXFLAG bit in the Pending Interrupt Register will be set and the $\overline{INT}$ pin will be asserted if the TXFLAGIE bit is set in the Pending Interrupt Enable Register.

Transmit Watchdog Timer

The Transmit Watchdog Timer (TWDT) is a fail-safe automatic shutdown mechanism that when enabled terminates the automatic transmission of scheduled messages in the event that the host does not refresh the TWDT 8-bit timer value before reaching zero count. The TWDT does not halt any transmit messages currently from the transmit FIFOs. The TWDT is disabled by default after Master Reset.

Two 8-bit registers control the TWDT: The Watchdog Mask Register (WMR), and Watchdog Timer Register (WTR).

Use the following sequence to use the TWDT. The steps must be sequential:

1. Write 0xFF to WMR

2. Write an 8-bit timer value to WTR.

3. Write a non-0xFF value to WMR. Transmit channels with Watchdog Timer enabled must have 0's in the corresponding bit position [TX7:TX0].

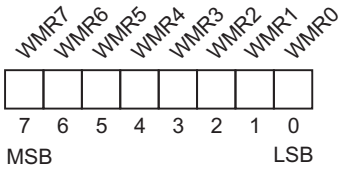
The TWDT is now active and WTR begins decrementing every 20ms.

4. The host must refresh the WTR timer value before it decrements to zero or a Watchdog Timer event will occur and automatic transmit scheduler transmissions will stop after the current frame of message(s) is transmitted as defined in the transmit descriptor tables. Only those channels [TX7:TX0] enabled by corresponding WMR low bits are stopped.

5. Normal ARINC 429 transmissions will resume if a non-zero timer value is written to WTR.

6.Once the TWDT is enabled it cannot be disabled except by a Master Reset or a Software Reset SPI Opcode command.

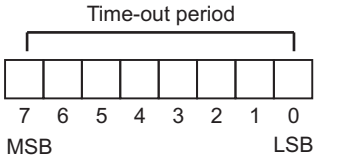
WATCHDOG MASK REGISTER
(Address 0x807E)



Default = 0xFF

Bit	Name	R/W	Default	Description
7	WMR7	R/W	1	Watchdog is enabled for transmitter 7 if WMR7=0 or disabled if WMR7=1.
6	WMR6	R/W	1	Watchdog is enabled for transmitter 6 if WMR6=0 or disabled if WMR6=1.
5	WMR5	R/W	1	Watchdog is enabled for transmitter 5 if WMR5=0 or disabled if WMR5=1.
4	WMR4	R/W	1	Watchdog is enabled for transmitter 4 if WMR4=0 or disabled if WMR4=1.
3	WMR3	R/W	1	Watchdog is enabled for transmitter 3 if WMR3=0 or disabled if WMR3=1.
2	WMR2	R/W	1	Watchdog is enabled for transmitter 2 if WMR2=0 or disabled if WMR2=1.
1	WMR1	R/W	1	Watchdog is enabled for transmitter 1 if WMR1=0 or disabled if WMR1=1.
0	WMR0	R/W	1	Watchdog is enabled for transmitter 0 if WMR0=0 or disabled if WMR0=1.

WATCHDOG TIMER REGISTER
(Address 0x807F)



Default = 0x00

WTR7:0 sets the time-out period for the Watchdog Timer. The LSB is 20ms. The maximum time-out period of 5.1s is programmed by writing 0xFF to the WTR.

RESET AND START-UP MODES

After power-on, the HI-3220 is in an undefined state. The $\overline{\text{MRST}}$ pin must be taken low to begin device initialization. The $\overline{\text{MRST}}$ pin may be asserted at any time. Taking $\overline{\text{MRST}}$ low immediately stops all execution and sets the READY output low, indicating that the part is in the reset state. $\overline{\text{MRST}}$ requires a minimum 225 ns pulse.

On the rising edge of $\overline{\text{MRST}}$ the HI-3220 samples the state of the MODE2-0 input pins. This is the only occasion these inputs are sampled. Once $\overline{\text{MRST}}$ goes high, the MODE2-0 pins become outputs ESCLK, EMOSI and $\overline{\text{ECS}}$. The MODE setting is readily acomplished using three 27KOhm pull-up/down resistors at the three pins. The state of the MODE pins determines one of six possible initialization sequences (Mode 0 through Mode 5). MODE[2:0] = 000 sets Mode 0, MODE[2:0] = 001 sets Mode 1, MODE[2:0] = 101 sets Mode 5, etc. Note Mode 7 is reserved for factory test and must not be used. These six initialization modes allow the user to customize the start-up configuration of the device. See Figure 1, Reset and Start-up Operation Flow Chart.

The total time from $\overline{\text{MRST}}$ rising edge to READY pin high depends on the total number of MCLK cycles (50MHz clock input) and is summarized in Table 2 for each mode.

Once the initialization is complete, the device enters the Idle State when the ready pin goes high. In Idle State, the host CPU may communicate with the HI-3220 memory and registers using the host CPU SPI link. When in the Idle State, the HI-3220 does not transmit or receive any messages on any of the ARINC 429 buses.

To begin data bus operation, the user must transition the RUN input from a low to high state. Immediately following the rising edge of RUN, the part enters the Active State and bus message processing begins.

During initialization, various device configuration tasks are performed according to the Mode selection set at the MODE2:0 input pins. The available options are described below in sub-sections 1-4.

MODE[2:0] status bits can be examined by reading the upper three bits <mmm> in the UPPER BIST FAIL ADDRESS REGISTER 0x807A.

MODE 6 is reserved for RAM BIST.

Software Reset Opcode

The SPI software reset opcode 0xFA5(8+mmm) resets the Configuration registers and Transmit descriptor tables, RX Interrupt Map and RX Enable Map areas according to the chart shown in Figure 1 depending on the mmm bit field (mode selection) used in the SPI opcode. See table 2.

TABLE 1. Start-up Time for each Mode

MODE[2:0] PINS	MODE	MCLK Cycles	Start-up Time*
000	0	8,204	164.080 µs
001	1	555,386	11.107720 ms
010	2	73,745	1.4749 ms
011	3	620,927	12.41854 ms
100	4	53,388	107.760 µs
101	5	9	180 ns
110	6	-	-
111	RESERVED	-	-

* The time (±1 MCLK) from rising edge of $\overline{\text{MRST}}$ to READY pin high.

RESET AND START-UP MODES (cont.)

1. RAM Integrity Check

In Modes 2 and 3, the HI-3220 performs a RAM integrity check. A read/write check is performed on the entire RAM space. An incrementing pattern is written to sequential RAM locations, then this pattern is read and verified. Each RAM location is re-written with the 1s complement of its current contents then this pattern is read and verified. The incrementing pattern followed by its 1s complement ensures that each RAM location can store both a 1 and 0 state. If the RAM integrity check fails, the $\overline{INT}$ pin is asserted and the Pending Interrupt Register RAMFAIL bit is set. The part enters the "Safe" state, in which the HI-3220 is able to accept and respond to Host CPU SPI Instructions, but cannot enter Normal Operating mode until the $\overline{MRST}$ input is taken low to repeat the initialization sequence. The RAMFAIL Interrupt is not maskable.

2. Clear Data Memory

In Modes 0 - 4, the HI-3220 automatically clears all memory locations in the address range 0x0000 to 0x3FFF and 0x6C00 to 0x7FFF. This is the space reserved for ARINC 429 message data. Configuration tables and HI-3220 registers are not affected.

3. Initialize Registers and Clear all memory

In addition to clearing data memory (0x0000 to 0x3FFF and 0xc00 to 0x7FFF), Modes 0, 1, 2, and 3 also clear all configuration and look-up tables (0x4000 to 0x6BFF) as well as setting all configuration registers (0x8000 to 0x807F) to their default states. All registers default to zero unless otherwise noted.

4. Auto-Initialize from EEPROM

The contents of the Auto-Initialization EEPROM are copied into the HI-3220 memory and registers via the EEPROM SPI interface. The part verifies the integrity of the data transfer from the EEPROM by running through a byte-by-byte compare routine and a checksum validation. If a compare error is detected, the AUTOERR bit is set in the Pending Interrupt Register, the $\overline{INT}$ output is asserted. The location of the error is captured in the AUTO-INIT FAIL ADDRESS registers 0x807B (Auto-Init Fail LS address) and 0x807C (Auto-Init Fail MS address) and the part enters the Safe state. If a checksum error is detected, the CHKERR bit is set in the Pending Interrupt Register, the $\overline{INT}$ output is asserted and the part enters the Safe state. The AUTOERR and the CHKERR interrupts are not maskable.

Once initialization is successful, the part enters the Idle state. The host CPU may read and write HI-3220 internal memory and registers in all Modes. If not using the auto-initialization feature, the host CPU should configure the device at this time.

NOTE: Mode 7 is reserved and must not be selected.

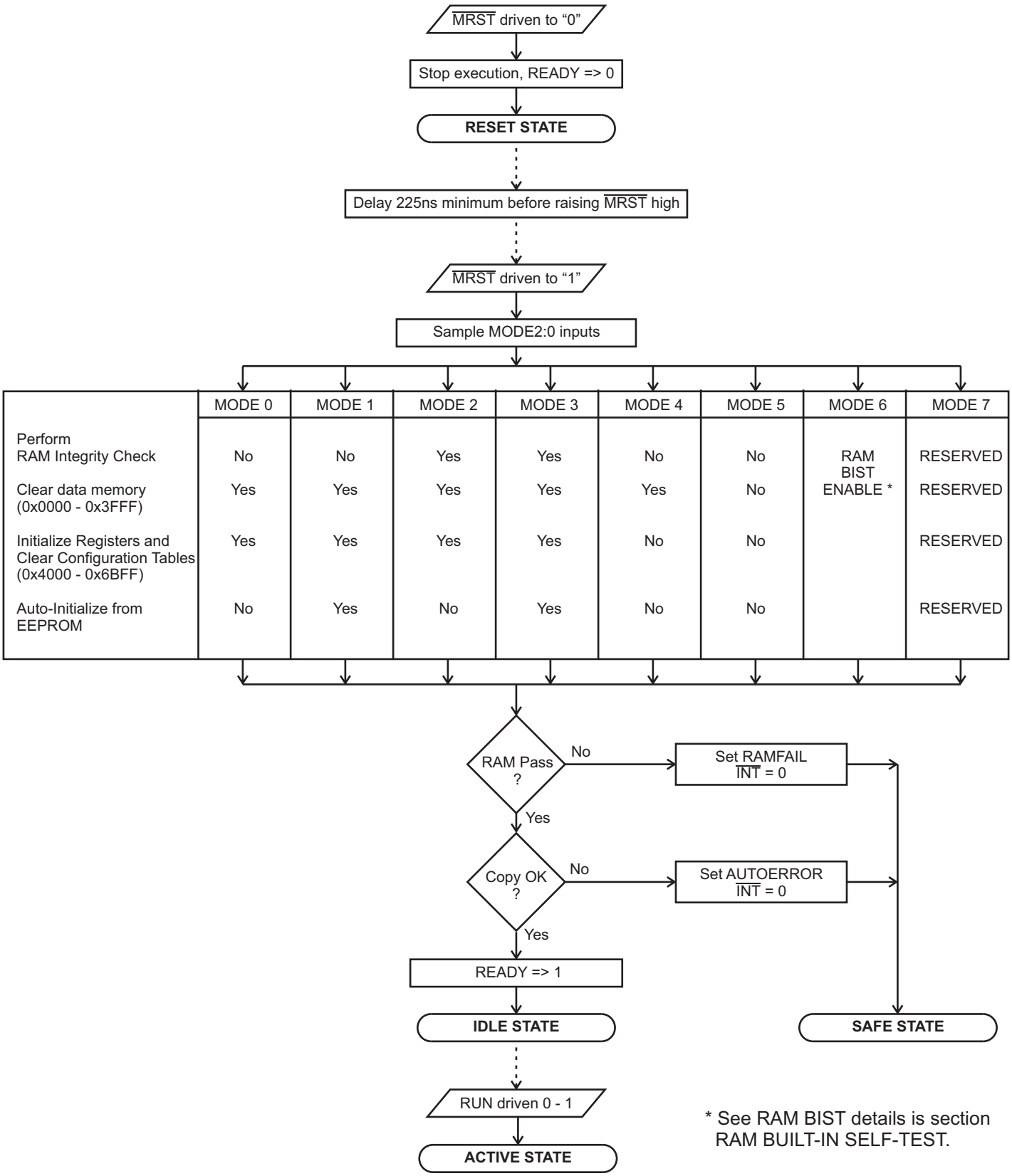


Figure 1. Reset and Start-up Operation flow chart.

INTERRUPTS

The HI-3220 includes a simple, user-selectable Interrupt Handler. Two types of Interrupt are possible - Message Event Driven (ARINC 429 Bus), and Fault Driven.

ARINC 429 Receive Interrupts

As described earlier, the user can elect to generate an interrupt upon receipt of an ARINC 429 message on any combination of the sixteen available channels and for any of the possible 256 label byte (ARINC message bits 1-8) values. Interrupts are enabled when the Receive Interrupt look-up bit is a “1”.

When a message arrives that is flagged to generate an Interrupt, that channel’s bit is set in the Receive Pending Interrupt Register RPIRH, RPIRL. The Interrupt Address Register (IAR) for that channel is updated with the ARINC 429 8-bit label value.

For example, if ARINC Receive channel 7 is enabled for Interrupts when messages with ARINC label 0xD4 arrive, then on receipt of such a message, RPIRL bit 7 is set to a “1” and the value 0xD4 is written to IAR7.

If the corresponding bit in the Receive Interrupt Mask Register is a “1” the $\overline{INT}$ interrupt output will go low and stay low until the $\overline{ACK}$ input pin is driven low. Driving $\overline{ACK}$ low, causes the $\overline{INT}$ pin to return to one.

A special Indexed SPI read instruction is available to allow the host to efficiently retrieve ARINC 429 messages which have Interrupts Enabled (see SPI instruction set section).

Note that if $\overline{ACK}$ is tied low permanently, the $\overline{INT}$ pin will go low for approximately 1 us before returning to one. A host CPU read of the RPIRH or RPIRL register reads the current value and resets it to 0x00.

ARINC 429 Transmit Interrupts

The user can elect to generate an interrupt when sending ARINC 429 messages using the transmit FIFOs. The Transmit Flag Interrupt Enable Register is programmed to define which transmit channels may generate an interrupt. If enabled, each time a bit is set in the Transmit FIFO Threshold Flag Register for an enabled channel, $\overline{INT}$ will be asserted.

Fault Interrupts

There are four fault Interrupt bits in the PIR. Fault Interrupts are not maskable, and their Interrupt Mask bits are fixed at a “1”.

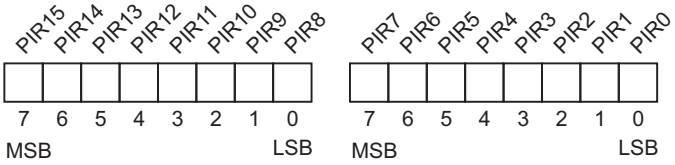
COPYERR is set when the HI-3220 detects a mismatch between RAM and EEPROM after attempting to program the Auto-initialization EEPROM.

AUTOERR is set when the Auto-Initialization EEPROM read verification cycle detects a mismatch between the on-chip memory and EEPROM following auto-initialization.

CHKERR is set when an auto-initialization checksum error is detected.

The RAMFAIL bit is set if the Built-In Self Test sequence fails.

RECEIVE PENDING INTERRUPT REGISTER HIGH / LOW
(Address 0x8008, 0x8006)
(Alternate Address 0x800C, 0x800B)

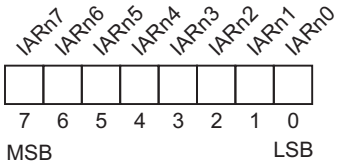


Bit	Name	R/W	Default	Description
15:0	PIR[15:0]	R	0	When a message is received on a given channel that triggers an interrupt, that channel's corresponding bit is set, e.g. if a message received on Rx channel 5 triggers an interrupt, the bit PIR5 will be set. If this bit is unmasked in the Receive Interrupt Mask Register (see below), the $\overline{INT}$ output pin is asserted. **Note:** This register pair is automatically cleared when read.

RECEIVE INTERRUPT ENABLE REGISTER HIGH / LOW
(Address 0x804C, 0x804B)

Bit	Name	R/W	Default	Description
15:0	RIE[15:0]	R/W	0	Each bit in this register, RIE[15:0], is a mask for a corresponding bit in the Receive Pending Interrupt Register (PIR[15:0], described above). Writing a "1" to an IMR bit results in assertion of the $\overline{INT}$ output pin when the corresponding PIR bit is set (ARINC 429 message received). Writing a "0" to an IMR bit will mask the corresponding PIR bit in the Receive Pending Interrupt Register, resulting in non-assertion of the $\overline{INT}$ pin when an ARINC 429 message is received.

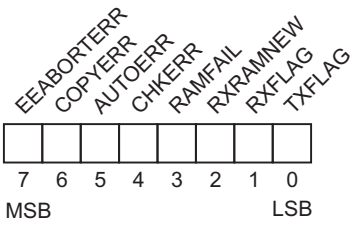
RECEIVE INTERRUPT ADDRESS REGISTERS 0 - 15
(Address 0x8010 - 0x801F)



n = 0 - 15

Bit	Name	R/W	Default	Description
7:0	IAR0[7:0]	R	0	The label number of the ARINC 429 message causing an interrupt is loaded into this register. Each channel has a corresponding Receive Interrupt Address Register (IAR0 - IAR15).

PENDING INTERRUPT REGISTER
(Address 0x8004)
(Alternate Address 0x800A)

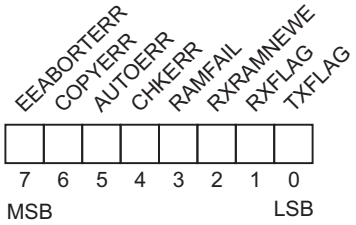


The $\overline{INT}$ will be asserted when any of the bits in this register are set.

Bit	Name	R/W	Default	Description
7	EEABORTERR	R	0	EE Abort programming error
6	COPYERR	R	0	EE copy error. RAM - EEPROM mismatch
5	AUTOERR	R	0	Auto-inititailization RAM read error
4	CHKERR	R	0	Auto-initialization checksum fail
3	RAMFAIL	R	0	Power-On Reset RAM Integrity Check fail
2	RXRAMNEW	R/W	0	Asserted high with any new message reception, all channels
1	RXFLAG	R	0	Logical OR of ARINC 429 Receive FIFO FLAG signals
0	TXFLAG	R	0	A Transmit FIFO Flag is set

NOTE: This register is cleared when read.

PENDING INTERRUPT ENABLE REGISTER
(Address 0x804A)



Bit	Name	R/W	Default	Description
7	EEABORTERR	R	1	EE Abort programming error
6	COPYERR	R	1	COPYERR is not maskable
5	AUTOERR	R	1	AUTOERR is not maskable
4	CHKERR	R	1	CHKERR is not maskable
3	RAMFAIL	R	1	RAMFAIL is not maskable
2	RXRAMNEWE	R/W	0	Enable for RXRAMNEW
1	RXFLAGIE	R/W	0	$\overline{INT}$ pin is asserted if this bit is a “1” and the Pending Interrupt Register RXFLAG bit is set
0	TXFLAGIE	R/W	0	$\overline{INT}$ pin is asserted if this bit is a “1” and the Pending Interrupt Register TXFLAG bit is set

INTERRUPT HANDLING

The static state of $\overline{ACK}$ determines if $\overline{INT}$ is a level or pulse signal.

$\overline{INT}$ Level Mode:

When an interrupt occurs, the $\overline{INT}$ pin is asserted low if the $\overline{ACK}$ pin is high. When $\overline{ACK}$ is asserted low, the $\overline{INT}$ pin returns to a high state. $\overline{ACK}$ should be a negative pulse so it is normally in the inactive state.

$\overline{INT}$ Pulse Mode:

If $\overline{ACK}$ is held low, this causes the $\overline{INT}$ to pulse negative for approximately 1 microsecond when an interrupt occurs.

Host Interrupt Servicing Options

When an interrupt occurs, the host typically reads one or more interrupt pending registers to determine the source of the interrupt. Depending on system requirements a small or large number of interrupts can be supported. If there are several interrupt sources that span across the three pending interrupt registers (PIR, RPIRH and RPIRL) then all three registers need to be read to determine the interrupt source(s). In a simpler system where there may be eight or less receiver interrupts grouped into either the higher or lower RPIRx registers it is possible to detect the interrupt sources by reading a single pending interrupt register. Reading one register is faster than reading three registers so this is a good way to reduce host interrupt processing time. Although the HI-3220 has a very fast SPI port, some SPI masters insert unwanted delays between bytes, so this helps reduce the time the host takes to determine the interrupt source.

Two op-code groups are available for reading pending interrupt registers so host interrupt servicing can be optimized based on the number of system interrupts.

Op-codes for single 8-bit pending register reads

(SPI clocking = 8 Op-code clocks + 8 data clocks)

OP-code	Register Address	Name
0x10	8004	PIR
0x18	8006	RPIRL
0x20	8008	RPIRH

Op-codes for multiple pending interrupt registers

To read all three registers issue an 8-bit op-code 0x28 + 24 data clocks. This returns three consecutive bytes representing PIR + RPIRH + RPIRL. This is the fastest method to read in all three interrupt registers. Another way to use op-codes 0x2C or 0x30 may be beneficial depending on the user requirements. For example , to only read the first and second pending interrupt registers, issue the op-code 0x28 and only 16 data clocks. The two returned bytes represent PIR and RPIRH. If op-code 0x2C was used with 16 data clocks then the two returned bytes represent RPIRH and RPIRL. As with any op-code sequence $\overline{HCS}$ must remain low during the whole time until all the data is fully clocked.

Op-code	Data Bytes	Register Address	Name
0x28	3	800A, 800B, 800C	PIR+RPIRL+RPIRH
0x2C	2	800B, 800C	RPIRL+RPIRH
0x30	1	800C	RPIRH

RAM BUILT-IN SELF-TEST

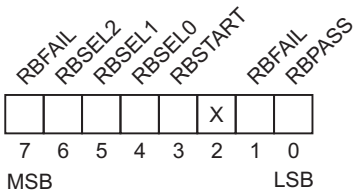
The HI-3220 offers a built-in self-test (BIST) feature which can be used to check RAM integrity. The BIST Control/Status Register is used to control the BIST function. All tests are destructive, overwriting data present before test commencement.

NOTE: To use BIST self-test, the part needs to be in Mode 6 with the RUN pin low.

To use RAM BIST follow these steps:

1. Power up the part in MODE 6. This can be accomplished by having MODE2 and MODE1 input pins high and MODE0 input pin low while powering up the part. Alternatively, configure the part in MODE-6 by executing the SPI Software Reset Opcode in Mode 6 (mmm=6 in Opcode Byte 2).
2. Set the RUN input pin low.
3. Write to the BIST control/status register with the desired RAM test (RBSEL2:0) and the RBSTRT bit high to initiate the test.
4. Wait for READY to be asserted high by polling the READY pin or by examining READY bit-5 in the MSR register using SPI.
5. After READY is asserted high examine bits 0 and 1 for the RAM BIST Fail or Pass result.

BIST CONTROL/STATUS REGISTER (Address 0x8078)



This register controls RAM built-in self-test. Bits 0,1 are Read Only. The remaining bits in this register are Read-Write but can be written only in MODE2:0 = 0x06.

BIST Control Register bits provide a means for the host to perform RAM self-test at other times. Register bits 6:4 select RAM test type. Then bit 3 starts the selected RAM test, and bits 1:0 report a fail/pass result after test completion.

Bit No. *Mnemonic* *Interrupt Type*

7 RBFFAIL RAM BIST Force Failure.
When this bit is asserted, RAM test failure is forced to verify that RAM BIST logic is functional.

6:4 RBSEL2-0 RAM BIST Select Bits 2-0.
This 3-bit field selects the RAM BIST test mode applied when the RBSTART bit is set:

RBSEL2:0	SELECTED RAM TEST
000	Idle
001	Pattern Test, described below
010	Write 0x00 to RAM address range 0x0000 - 0x7FFF
011	Read and verify 0x00 over RAM address range 0x0000 - 0x7FFF
100	Write 0xFF to RAM address range 0x0000 - 0x7FFF
101	Read and verify 0xFF over RAM address range 0x0000 - 0x7FFF
110	Inc / Dec Test performs only steps 5 - 8 of the Pattern Test below
111	Idle

Description of the RAM BIST "PATTERN" test selected when register bits RBSEL2:0 = 001:

1. Write 0x00 to all RAM locations, 0x0000 through 0x7FFF
2. Repeat the following sequence for each RAM location from 0x0000 through 0x7FFF:
   a. Read and verify 0x00
   b. Write then read and verify 0x55

c. Write then read and verify 0xAA
d. Write then read and verify 0x33
e. Write then read and verify 0xCC
f. Write then read and verify 0x0F
g. Write then read and verify 0xF0
h. Write then read and verify 0x00
I. Write then read and verify 0xFF
j. Write 0x00 then increment RAM address and go to step (a)

3. Write 0xFF to all RAM locations, 0x0000 through 0x7FFF

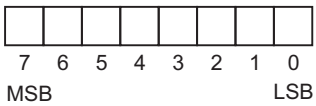
4. Repeat the following sequence for each RAM location from 0x0000 through 0x7FFF:
   a. Read and verify 0xFF
   b. Write then read and verify 0x55
   c. Write then read and verify 0xAA
   d. Write then read and verify 0x33
   e. Write then read and verify 0xCC
   f. Write then read and verify 0x0F
   g. Write then read and verify 0xF0
   h. Write then read and verify 0x00
   I. Write then read and verify 0xFF
   j. Write 0xFF then increment RAM address and go to step (a)

5. Write an incrementing pattern into sequential RAM locations from 0x0000 to 0x7FFF
6. Read each memory location from 0x0000 to 0x7FFF and verify the contents
7. Write 1s complement of each cell's current contents, into each RAM location (same addr range)
8. Read each memory location and verify the contents

3	RBSTRT	RAM BIST Start. Writing logic 1 to this bit initiates the RAM BIST test selected by register bits RBSEL2:0. The RBSTRT bit can only be set in MODE2:0 = 0x04. This bit is automatically cleared upon test completion. Register bits 1:0 indicate fail / pass test result.
2	---------	Not Used.
1	RBFAIL	RAM BIST Fail. Device logic asserts this bit when failure occurs while performing the selected RAM test. This bit is automatically cleared when RBSTRT bit 3 is set. When BIST failure occurs, a clue to the failing RAM address can be read at register addresses 0x8079 and 0x807A. For speed, the RAM BIST concurrently tests four consecutive RAM addresses in parallel. If a test failure occurs, register addresses 0x8079 and 0x807A can be used to determine the four RAM addresses tested.
0	RBPASS	RAM BIST Pass. Device logic asserts this bit when the selected RAM test completes without error. This bit is automatically cleared when RBSTRT bit 3 is set.

LOWER BIST FAIL ADDRESS REGISTER (Address 0x8079)

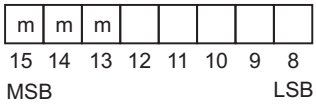
BISTFL



UPPER BIST FAIL ADDRESS AND MODE BITS REGISTER (Address 0x807A)

Bits <15:13> in Register BISTFH are Mode[2:0] Status Bits

BISTFH



HOST SERIAL PERIPHERAL INTERFACE

In the HI-3220, internal RAM and registers occupy a (32K + 128) x 8 address space. The lowest 32K addresses access RAM locations and the remaining addresses access registers. Timing is identical for register operations and RAM operations via the serial peripheral interface, and read and write operations have likewise identical timing.

Host access is only allowed when the part is READY or in SAFE mode. **NOTE:** writes will be blocked and reads will return the Master Status Register value until either of these modes occur.

Serial Peripheral Interface (SPI) Basics

The HI-3220 uses an SPI synchronous serial interface for host access to registers and RAM. Host serial communication is enabled through the Chip Select ($\overline{HCS}$) pin, and is accessed via a three-wire interface consisting of Serial Data Input (HMOSI) from the host, Serial Data Output (HMISO) to the host and Serial Clock (HSCK). All programming cycles are completely self-timed, and no erase cycle is required before write.

The SPI (Serial Peripheral Interface) protocol specifies master and slave operation; the HI-3220 Host CPU interface operates as an SPI slave.

The SPI protocol defines two parameters, CPOL (clock polarity) and CPHA (clock phase). The possible CPOL-CPHA combinations define four possible "SPI Modes." Without describing details of the SPI modes, the HI-3220 operates in the two modes where input data for each device (master and slave) is clocked on the rising edge of HSCK, and output data for each device changes on the falling edge. These are known as SPI Mode 0 (CPHA = 0, CPOL = 0) and SPI Mode 3 (CPHA = 1, CPOL = 1). Be sure to set the host SPI logic for one of these modes.

As seen in Figure 2, the difference between SPI Modes 0 and 3 is the idle state for the HSCK signal. There is no configuration setting in the HI-3220 to select SPI Mode 0 or Mode 3 because compatibility is automatic. Beyond this point, the HI-3220 data sheet only shows the SPI Mode 0 HSCK signal in timing diagrams.

The SPI protocol transfers serial data as 8-bit bytes. Once $\overline{HCS}$ chip select is asserted, the next 8 rising edges on HSCK latch input data into the master and slave devices, starting with each byte's most-significant bit. The HI-3220 SPI can be clocked at 40 MHz. To achieve reliable 40 MHz SPI the hardware should be optimized for low PCB capacitance with a short distance from the device to the host.

Multiple bytes may be transferred when the host holds $\overline{HCS}$ low after the first byte transferred, and continues to clock HSCK in multiples of 8 clocks. A rising edge on $\overline{HCS}$ chip select terminates the serial transfer and reinitializes the HI-3220 SPI for the next transfer. If $\overline{HCS}$ goes high before a full byte is clocked by HSCK, the incomplete byte clocked into the device HMOSI pin is discarded.

In the general case, both master and slave simultaneously send and receive serial data (full duplex) as shown in Figure 2 below. When the HI-3220 is sending data on HMISO during read operations, activity on its HMOSI input is ignored. Figures 3 and 4 show actual behavior for the HI-3220 HMISO output.

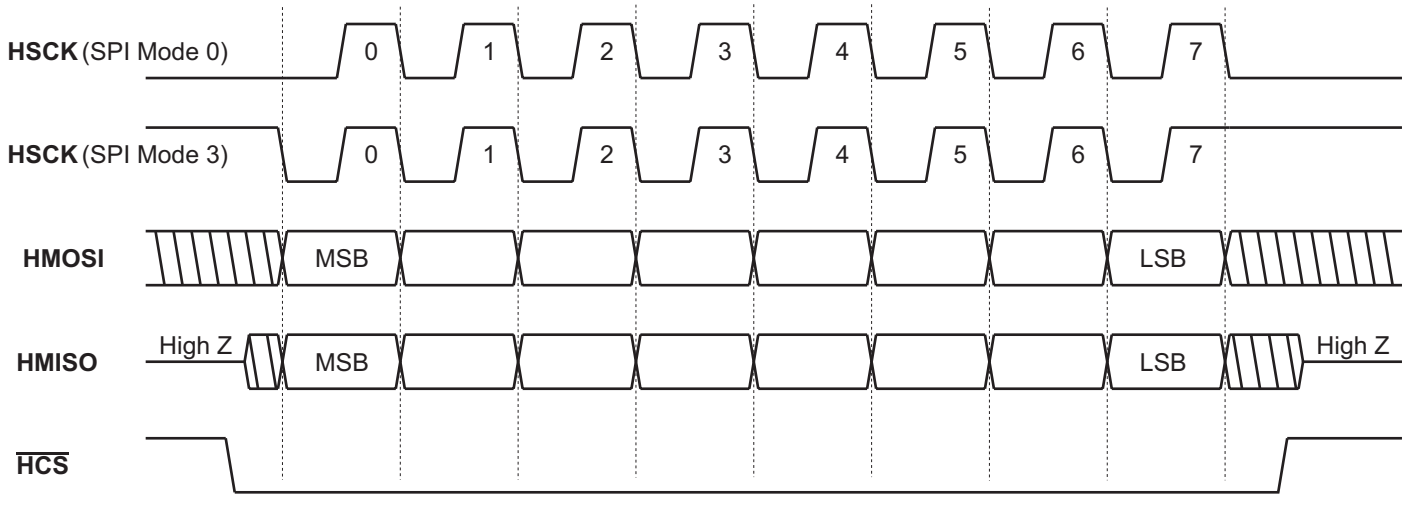


FIGURE 2. Generalized Single-Byte Transfer Using SPI Protocol, HSCK is Shown for SPI Modes 0 and 3

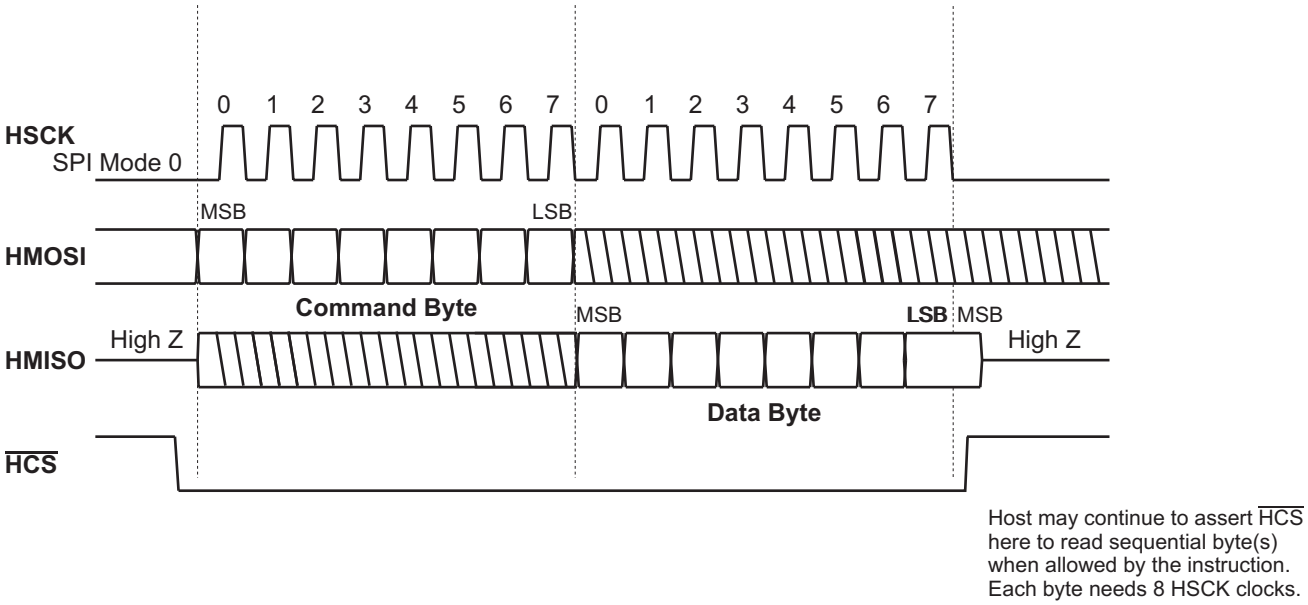


FIGURE 3. Single-Byte Read From RAM or a Register

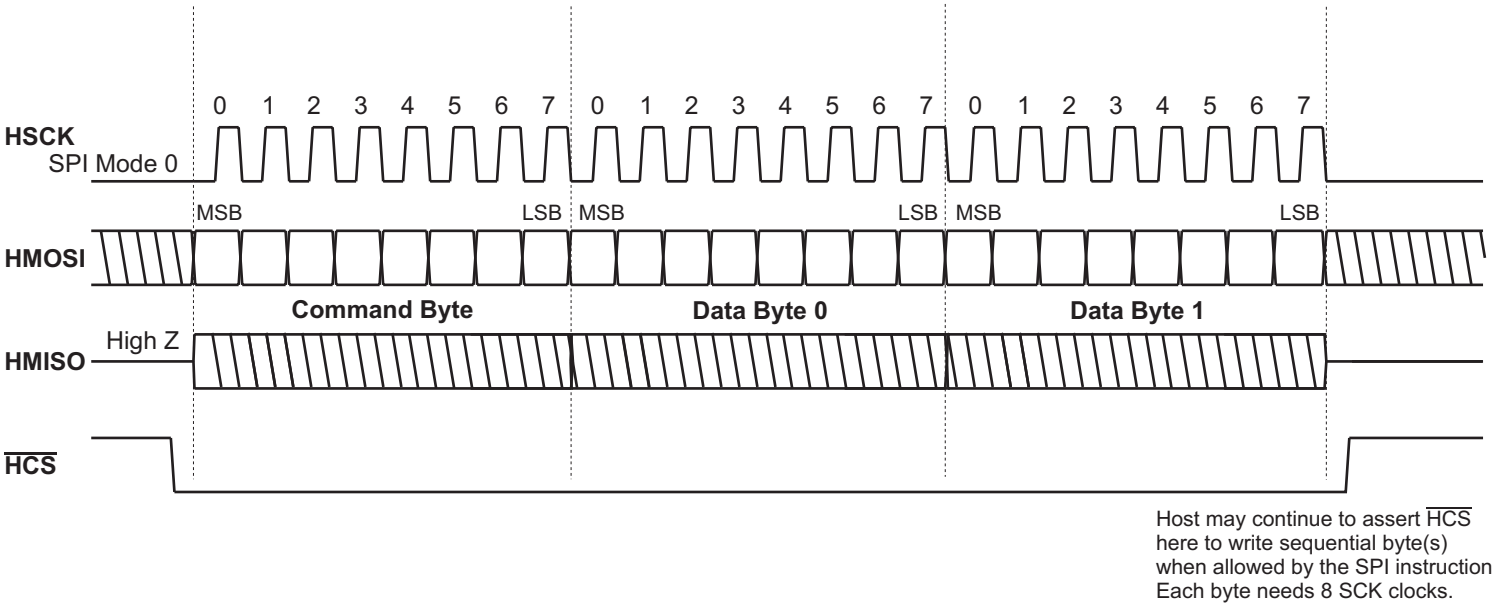


FIGURE 4. 2-Byte Write To RAM or a Register Pair

HI-3220 SPI COMMANDS

Refer to the HI-3220 SPI command set shown in Table 2.

For the HI-3220, each SPI read or write operation begins with an 8- or 16-bit command byte(s) transferred from the host to the device after assertion of $\overline{HCS}$. Since HI-3220 command byte reception is half-duplex, the host discards the dummy byte(s) it receives while serially transmitting the command byte(s).

Fast Access Commands for Registers 0-31

The SPI command set includes directly-addressed read and write commands for registers 0 through 31 (Memory Address 0x8000 to 0x801F). The 8-bit pattern for these commands has the general form

0-R-R-R-R-R-0-0

where RRRRR is the 4-bit register number. All registers within this address range are read-only, with the exception of the Master Control Register, which utilizes two addresses, one for read and one for write operations.

Figures 3 and 4 show read and write timing as it appears for fast-access register operations. The command byte is immediately followed by a data byte comprising the 8-bit data word read or written. For a single register read or write, $\overline{HCS}$ is de-asserted after the data byte is transferred.

Multiple register read or write cycles may be performed by transferring more than one byte before $\overline{HCS}$ is de-asserted. Multiple register access occur in address order starting with the register specified in the SPI instruction.

Note: Register locations not shown in table 2 are "reserved" and cannot be written using any SPI command. Further, these register addresses will not provide meaningful data in response to read commands.

Register Direct Addressing

Two 2-byte SPI instructions read and write data bytes to registers anywhere in the register address space from 0x8000 through 0x807F for writes or 0x8000 through 0x8078 for reads. The explicit register address is included in the op-code bit sequence. The opcodes take the form:

Write: 11011AAA AAAA0000
Read: 1110AAAA AAAA0000

Where AAA AAAA is the least significant seven or eight bits of the register address.

RAM and Register Indirect Addressing

To access the entire address range of the part, SPI commands are included that use an address pointer to indicate the address for read or write transactions. This sixteen-bit memory address pointer (MAP) must be initialized before any indirect read or write operation. Two dedicated SPI instructions are used to write and read the MAP. SPI Instruction 0x98 followed by two data bytes is used to write MAP. SPI instruction 0x90 reads two data bytes from MAP. The first byte is the most significant eight bytes of the address. For example, SPI sequence 0x98, 0x12, 0x34 write the value 0x1234 into the MAP.

Two SPI instructions read and write data bytes to memory or registers using the MAP as an address pointer. Single or multi-byte reads and writes may be performed. MAP is incremented after each byte access.

Two command bytes cannot be "chained"; $\overline{HCS}$ must be de-asserted after the command, then reasserted for the following Read or Write command.

Note: When the primary or fast-access address pointer is used for auto-incrementing multi-word read/write and reaches the top of the memory address range (0x7FFF), or the top of the register address range (0xFFFF) attempts to read further bytes will result the terminal address (0x7FFF or 0xFFFF) being output again. The host should avoid this situation.

Two single-byte SPI commands use the current address pointer value in MAP without first loading or otherwise modifying it:

Command	**Read Operation**
0x80	read location addressed by pointer value

Command	**Write Operation**
0x88	write location addressed by pointer value

Either of these commands can be used to read or write a single location, or may be used when starting a multi-byte read or write by using the pointer's auto-increment feature.

MAP is only used with opcodes 0x80 and 0x84.

Special Purpose Commands

Several other HI-3220 SPI commands load or otherwise modify the memory address pointer before initiating a read or write process. These commands are designed to allow speedy access to messages received on the ARINC 429 buses.

Using a 2-byte SPI command, the address pointer can be directly loaded with the memory address for the last received ARINC 429 message which triggered an interrupt.

Op Code 11001000 CCCC0000
The HI-3220 will retrieve the current ARINC Receive Interrupt Vector for a given receive channel (CCCC), calculate the memory address for the first word of the corresponding receive memory data block, and then read the location at that address.

This command can be used to read just the most recent ARINC 429 Receive Status Byte, or may be used to start a four-byte read because memory pointer auto-increment occurs after the Status Byte is read.

Op Code 11010000 CCCC0000
The HI-3220 will retrieve the current ARINC Receive Interrupt Vector for a given channel (CCCC), calculate the memory address for the first word of the corresponding receive memory data block, then output the value of the Receive Interrupt Vector (ARINC 429 label byte).

This command can be used to read just the most recent ARINC 429 label value received, or may be used to start a four-byte read to output the entire four-byte ARINC message, because memory pointer auto-increment occurs after the label byte is output.

Op Code 10100TTT
Writes an ARINC 429 message to ARINC 429 transmit scheduler TTT for immediate transmission, where TTT represents the transmit channel number.

Op Code 11111010 01011010
This op-code forces a reset of the HI-3220. The part will re-initialize in exactly the same manner as if the MRST pin had been asserted.

TABLE 2. DEFINED SPI COMMANDS

OP CODE byte 1	OP CODE byte 2	Auto Increment	Number of Data Bytes	DESCRIPTION
0RRRRR00	None	Yes	1++	Fast Register access at register RRRRR
10000000	None	Yes	1++	Read memory at address MAP
10001000	None	Yes	1++	Write memory at address MAP
10010000	None	No	2	Read MAP
10011000	None	No	2	Write MAP
10100TTT	None	No	4, 8, 12...	Transmit ARINC 429 message on transmit bus TTT
11000000	CCCC0000	Yes	4, 8, 12...	Read ARINC 429 FIFO # CCCC. Read multiples of four bytes
11001000	CCCC0000	No	4	Read ARINC block at receive channel CCCC, label <IARn> (Reads status word plus data). See Note 3 below.
11010000	CCCC0000	No	4	Read ARINC message at receive channel CCCC, label <IARn> (Reads label plus data). See Note 3 below.
11011AAA	AAAA0000	Yes	1++	Write register number AAA AAAA (0x8000 - 0x807F)
1110AAAA	AAAA0000	Yes	1++	Read register number AAAA AAAA (0x8000 - 0x8087)
11111010	01011mmm	No	0	Software Reset with mode 0 - mode 5, 000-101

FAST-ACCESS SPI COMMANDS FOR REGISTERS 0-31
Command Bits 6:2 Convey the 5-Bit Register Address

COMMAND BITS 7 6 5 4 3 2 1 0	HEX BYTE	FUNCTION
0 0 0 0 0 0 0 0	0x00	Write MCR
0 0 0 0 0 1 0 0	0x04	Read MCR
0 0 0 0 1 0 0 0	0x08	Read MSR
0 0 0 0 1 1 0 0	0x0C	Reserved
0 0 0 1 0 0 0 0	0x10	Read PIR
0 0 0 1 0 1 0 0	0x14	Reserved
0 0 0 1 1 0 0 0	0x18	Read RPIRL
0 0 0 1 1 1 0 0	0x1C	Reserved
0 0 1 0 0 0 0 0	0x20	Read RPIRH
0 0 1 0 0 1 0 0	0x24	Reserved
0 0 1 0 1 0 0 0	0x28	Read PIR+RPIRL/H
0 0 1 0 1 1 0 0	0x2C	Read RPIRL/H
0 0 1 1 0 0 0 0	0x30	Read RPIRH
0 0 1 1 0 1 0 0	0x34	Read FTFL
0 0 1 1 1 0 0 0	0x38	Read FTFH
0 0 1 1 1 1 0 0	0x3C	Read TFR

COMMAND BITS 7 6 5 4 3 2 1 0	HEX BYTE	FUNCTION
0 1 0 0 0 0 0 0	0x40	Read IAR0
0 1 0 0 0 1 0 0	0x44	Read IAR1
0 1 0 0 1 0 0 0	0x48	Read IAR2
0 1 0 0 1 1 0 0	0x4C	Read IAR3
0 1 0 1 0 0 0 0	0x50	Read IAR4
0 1 0 1 0 1 0 0	0x54	Read IAR5
0 1 0 1 1 0 0 0	0x58	Read IAR6
0 1 0 1 1 1 0 0	0x5C	Read IAR7
0 1 1 0 0 0 0 0	0x60	Read IAR8
0 1 1 0 0 1 0 0	0x64	Read IAR9
0 1 1 0 1 0 0 0	0x68	Read IAR10
0 1 1 0 1 1 0 0	0x6C	Read IAR11
0 1 1 1 0 0 0 0	0x70	Read IAR12
0 1 1 1 0 1 0 0	0x74	Read IAR13
0 1 1 1 1 0 0 0	0x78	Read IAR14
0 1 1 1 1 1 0 0	0x7C	Read IAR15

Notes:
1. Op-code 0x28 is used to read PIR, RPIRL and RPIRH as a three-byte data field.
2. Op-code 0x02C is used to read RPIRL and RPIRH as a two-byte data field.
3. To use OpCodes 11000000 CCCC 0000 and 11010000 CCCC 0000 the corresponding Received Data Interrupt Look-Up Table bits must be set for the desired channels and labels. Enabling interrupts in the interrupt enable registers are not required.

PROGRAMMING THE AUTO-INITIALIZATION EEPROM.

Following reset, the HI-3220 may be completely configured by automatically copying the contents of an external EEPROM into HI-3220 memory and registers. An SPI enabled 64KByte EEPROM is used for this purpose. The EEPROM memory space is mapped to the HI-3220.

All configuration memory blocks are copied. The ARINC 429 Received Data Memory contents and ARINC 429 Receive log FIFO contents are not copied to or from the EEPROM.

The HI-3220 can be used to program the Auto-Initialization EEPROM. When the HI-3220 is in its IDLE state (RUN input = “0”), a five step sequence must be performed to begin the EEPROM programming cycle:

1. Write data value 0x5A to HI-3220 memory address 0x8FFE.

2. Write data value 0x96 to HI-3220 memory address 0x8FFA. Step 2 must occur within 1ms of step 1.

3. Write data value 0xF0 to memory address 0x8FF8 within 10ms.

4. Wait > 1ms, but less than 10ms.

5. Write 0xC3 to 0x8FFC

If the five step sequence is interrupted by any intervening host activity or improper timing between the steps the programming cycle is aborted.

Step 5 initiates the cycle. The READY pin goes low, and the contents of the HI-3220 memory and registers are copied to the EEPROM. When copying is complete, the HI-3220 executes a byte-by-byte comparison of the EEPROM and its own register / memory contents. If the verification completes successfully, the READY pin goes high.

A 2’s complement of the checksum is also written to the EEPROM at location 0x8080. The total read back checksum should be zero. The following locations are excluded from the checksum because they are either read-only or unused locations:

0x8001 - 0x800F
0x8010 - 0x801F
0x8040 - 0x8047
0x8068 - 0x807C.

If the comparison of the EEPROM contents and HI-3220 memory / register contents results in a discrepancy, the HI-3220 enters the SAFE state, the PROGERR bit is set in the Pending Error Register and the INT output is asserted.

The user must clear the PROGERR issue before normal operation can resume.

If an error occurs during programming the EEABORTERR flag is set in the PIR register. The EE programming time on the Holt ADK-3220 development board using an Atmel EEPROM is approximately 278ms.

ABSOLUTE MAXIMUM RATINGS

Supply voltage (VDD)	-0.3 V to +5.0 V
Logic input voltage range	-0.3 V DC to +3.6 V
ARINC 429 input voltage	-120 V to +120 V
Power dissipation at 25°C	1.0 W
Reflow Solder Temperature	260°C
Junction Temperature	175°C
Storage Temperature	-65°C to +150°C

RECOMMENDED CONDITIONS

Operating Supply Voltage

VDD..... 3.3 VDC ±5%

Operating Temperature Range

Industrial ..... -40°C to +85°C
Extended ..... -55°C to +125°C

NOTE: Stresses above absolute maximum ratings or outside recommended operating conditions may cause permanent damage to the device. These are stress ratings only. Operation at the limits is not recommended.

DC ELECTRICAL CHARACTERISTICS

Digital Pins

VDD = 3.3 V, GND = 0V, T_A = Operating Temperature Range (unless otherwise specified).

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS
Operating Voltage	VDD		3.15	3.30	3.45	V
Supply Current	IDD				75	mA
Min. Input Voltage (HI)	V_{IH}	Digital inputs	70%			VDD
Max. Input Voltage (LO)	V_{IL}	Digital inputs			30%	VDD
Pull-Up / Pull-Down Current	I_{PUD}	See pin definitions table		30	100	µA
Output Current (HI)	I_{OH}	V_{OH} = 0.8 VDD VDD = 3.15 - 3.45 V	-6.0			mA
Output Current (LO)	I_{OL}	V_{OL} = 0.4 V VDD = 3.15 - 3.45 V	6.0			mA
Min. Output Voltage (HI)	V_{OH}	I_{OUT} = -1.0mA, Digital outputs	90%			VDD
Max. Output Voltage (LO)	V_{OL}	I_{OUT} = 1.0mA, Digital outputs			10%	VDD
Input Capacitance	C_I			5		pF
Output Capacitance	C_O			5		pF

ARINC 429 Receiver Inputs (ARXx-40)

VDD = 3.3 V, GND = 0V, T_A = Operating Temperature Range (unless otherwise specified). Measured at ARINC 429 bus with required 40KOhm isolation resistors installed.

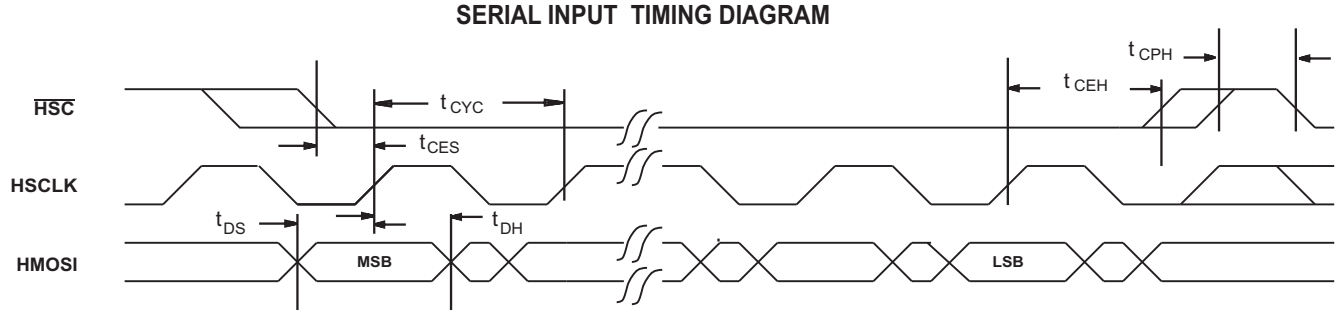
PARAMETER		SYMBOL	CONDITION	MIN	TYP	MAX	UNITS
Input Voltage	ONE or ZERO	V_{DIN}	Differential input voltage	6.5	10	13	V
	NULL	V_{NIN}	Differential input voltage			2.5	V
	Common mode	V_{COM}	With respect to GND			+/-5.0	V
Input Resistance	ARXnP-40 to ARXnN-40	R_{DIFF}	Supplies floating		90		kΩ
	Input to GND or VDD	R_{SUP}	Supplies floating		45		kΩ
Input Hysteresis		V_{HYS}		0.5	1.0		V
Input Capacitance	ARINC bus differential	C_{AD}			5	10	pF
	ARINC single ended to GND	C_{AS}				10	pF

AC ELECTRICAL CHARACTERISTICS

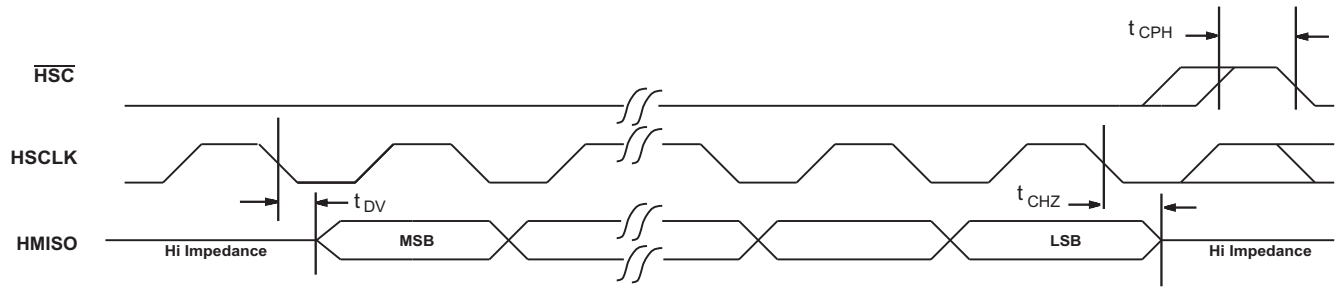
VDD = 3.3 V, GND = 0V, T_A = Operating Temperature Range (unless otherwise specified).

PARAMETER	SYMBOL	LIMITS			UNITS
		MIN	TYP	MAX	
SPI Host Bus Interface					
HSCK clock period	t_{CYC}	25			ns
$\overline{HSC}$ set-up time to first HSCK rising edge	t_{CES}	15			ns
$\overline{HSC}$ hold time after last HSCK falling edge	t_{CEH}	15			ns
$\overline{HSC}$ inactive between SPI instructions	t_{CPH}	100			ns
SPI SI Data set-up time to SCK rising edge	t_{DS}	10			ns
SPI HMOSI Data hold time after HSCK rising edge	t_{DH}	10			ns
HMISO valid after HSCK falling edge	t_{DV}			10	ns
HMISO high-impedance after $\overline{HSC}$ inactive	t_{CHZ}			90	ns

SERIAL INPUT TIMING DIAGRAM



SERIAL OUTPUT TIMING DIAGRAM



Power Supply Recommended bypass capacitors (near the device)

Provide a 0.1µF ceramic bypass capacitor on GND and VDD power pins (2-4 capacitors).
Provide a single 22µF ceramic or tantalum bypass capacitor on any GND and VDD power pins.

ORDERING INFORMATION

HI-322xxx x x

PART NUMBER	PACKAGE DESCRIPTION
Blank	Tin / Lead (Sn / Pb) Solder or NiPdAu
F	100% Matte Tin or NiPdAu (Pb-free RoHS compliant)

PART NUMBER	TEMPERATURE RANGE	FLOW	BURN IN
I	-40°C TO +85°C	I	No
T	-55°C TO +125°C	T	No
M	-55°C TO +125°C	M	Yes

PART NUMBER	PACKAGE DESCRIPTION	LINE DRIVER SLOPE CONTROL	INTERNAL LINE RECEIVERS	# RX/TX Channels
3220PQ	80-PIN PLASTIC QFP (80PQTS)	Y	Y	16 / 8
3221PC	72-PIN PLASTIC QFN (72PCS)	N	Y	16 / 8
3222PC	48-PIN PLASTIC QFN (48PCS7)	N	Y	8 / 4
3223PC	64-PIN PLASTIC QFN (64PCS) [1]	Y	Y	8 / 4
3223PQ	52-PIN PLASTIC QFP (52PQS)	Y	Y	8 / 4
3225PQ	80-PIN PLASTIC QFP (80PQTS)	Y	N	16 / 8
3226PC	72-PIN PLASTIC QFN (72PCS)	N	N	16 / 8

NOTE 1: Available Pb-free only, HI-3223PCxF.

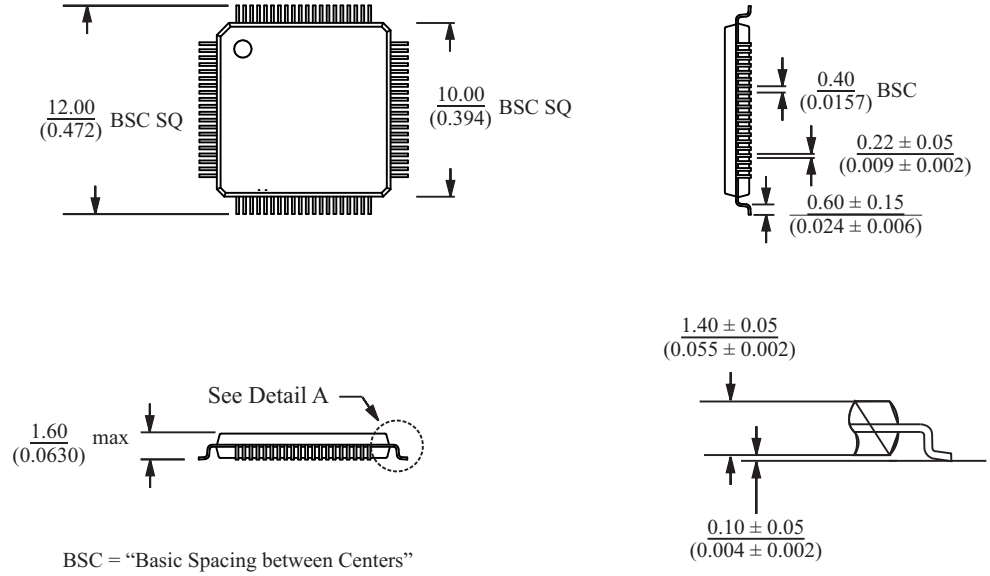
REVISION HISTORY

Document	Rev.	Date	Description of Change
DS3220	New	01-12-18	Initial Release.
	A	05-01-18	Add HI-3223 device and related package and ordering information. . Correct typo in 80PTQS drawing. Correct other typos and add additional clarification notes.
	B	10-15-18	Add “Fully compliant to ARINC 429 Specification” to Features. Correct typo in 3225PQ ordering information (slope control should be “Y”). Update package lead finish. Correct other numerous typos.
	C	12-14-18	Add ARINC input voltage to “Absolute Maximum Ratings Table”. Correct typos.

80 PIN PLASTIC QUAD FLAT PACK (PQFP)

millimeters (inches)

Package Type: 80PTQS



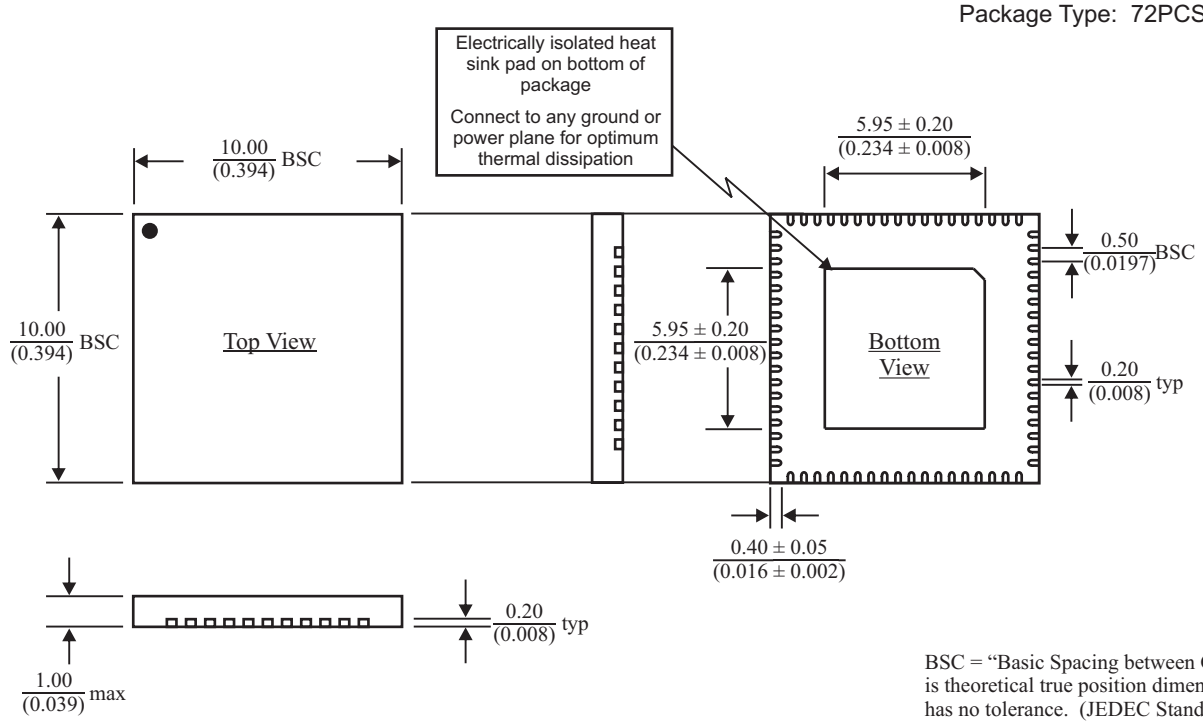
Detail A

BSC = "Basic Spacing between Centers" is theoretical true position dimension and has no tolerance. (JEDEC Standard 95)

72-PIN PLASTIC CHIP-SCALE PACKAGE (QFN)

millimeters (inches)

Package Type: 72PCS

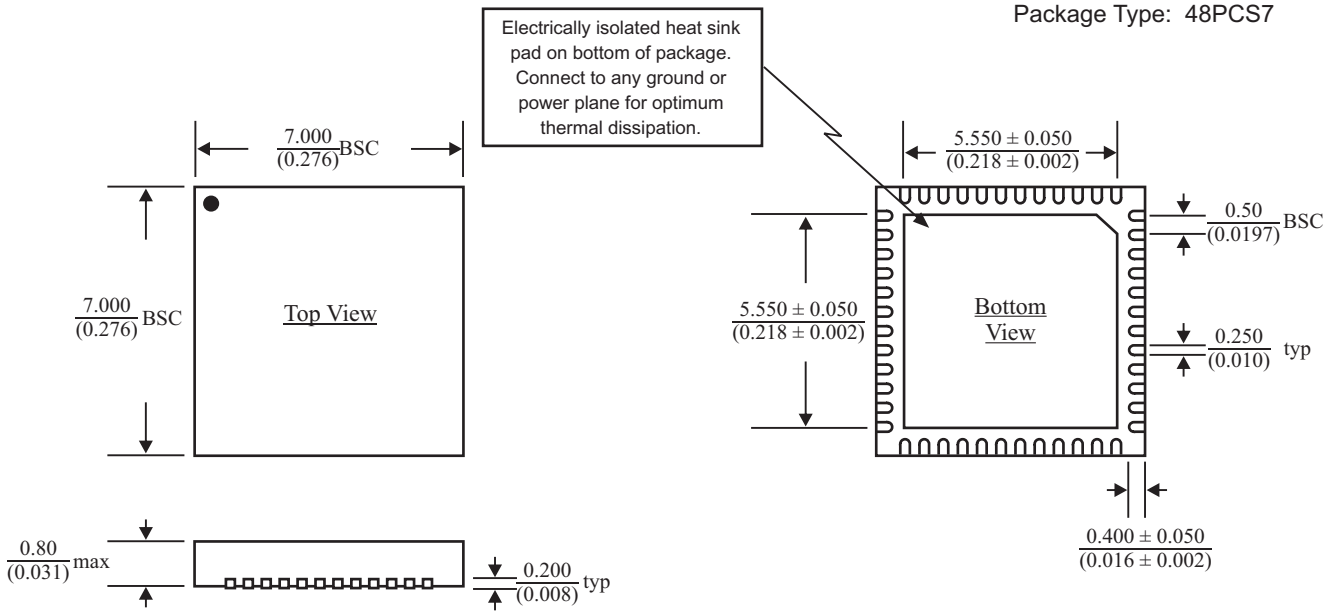


BSC = "Basic Spacing between Centers" is theoretical true position dimension and has no tolerance. (JEDEC Standard 95)

48-PIN PLASTIC CHIP-SCALE PACKAGE (QFN)

millimeters (inches)

Package Type: 48PCS7

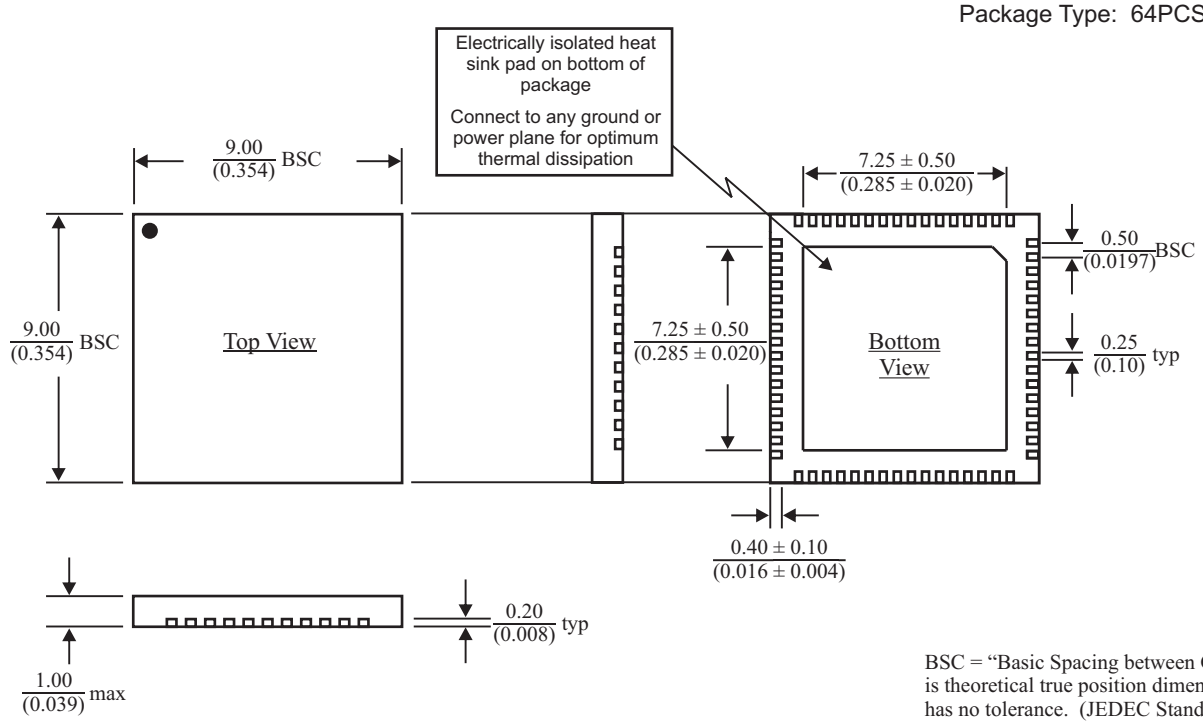


BSC = "Basic Spacing between Centers" is theoretical true position dimension and has no tolerance. (JEDEC Standard 95)

64-PIN PLASTIC CHIP-SCALE PACKAGE (QFN)

millimeters (inches)

Package Type: 64PCS

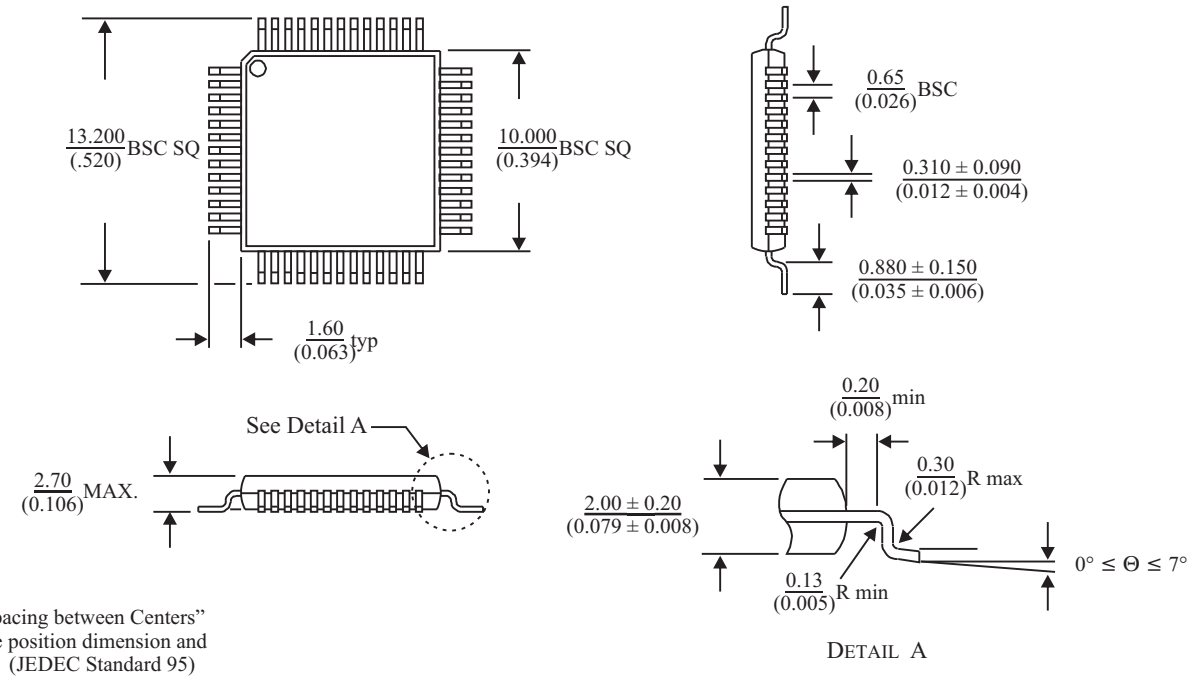


BSC = "Basic Spacing between Centers" is theoretical true position dimension and has no tolerance. (JEDEC Standard 95)

52-PIN PLASTIC QUAD FLAT PACK (PQFP)

millimeters (inches)

Package Type: 52PQS



DETAIL A

BSC = "Basic Spacing between Centers" is theoretical true position dimension and has no tolerance. (JEDEC Standard 95)