IPM429-E-0T-16R
IPM429-EI-0T-16R
IPM429-ELB-0T-16R
IPM429-ELBI-0T-16R

Document N°: MANH705-53

IPM429-E-0T-16R™

User Manual

Version 2.01
October 15, 2007

MAX technologies

Disclaimer

MAX Technologies, IPM429-E-0T-16R, and the MAX Technologies logo are all trademarks of MAX Technologies Inc. All other trademarks and registered trademarks are the property of their respective owners.

Changes are periodically made to the information in this document. These changes will be incorporated into new editions of this document. MAX Technologies may make improvements and/or changes in the products and/or programs described in this document at any time. MAX Technologies Inc. makes no warranties as to the contents of this manual or the accompanying software. Although every effort has been made to ensure that the manual is accurate and that the software is reliable, MAX Technologies Inc. cannot be held responsible for any damages suffered from the use of this product.

How to reach MAX Technologies for Product Support

7005 Taschereau Blvd., Suite 350
Brossard, Quebec
Canada, J4Z 1A7
Tel.: (450) 443-3332
Fax: (450) 443-1618
Toll free: (800) 361-1629
www.maxt.com

IPM429-E-0T-16R REVISION CROSS REFERENCE

Document version	**Modification revision***	**Mod. Effective date**	**Note**
Version 1.0	Rev. 1.0	Jul 12, 2000	Initial release
Version 1.22	Rev. 1.01 Rev. 1.02 Rev. 1.03	Feb 8, 2001 Mar 15, 2002 Sep 15, 2003	
Version 2.0	Rev. 1.1 Rev. 1.11 Rev. 1.12 Rev. 1.2 Rev. 1.3 Rev. 1.31	Dec 14, 2005 Apr 27, 2006 May 30, 2006 Aug 15, 2006 Sep 5, 2006 Feb 1, 2007	
Version 2.01	Rev. 1.31	Feb 1, 2007	User manual revision history update

* Refer to MAX Technologies Product History documents for a full hardware history of all MAX Technologies products. More details on software compatibility, product revision, global modification revision, PCB revision and firmware revision are available in the Product History documents.

Table of content

List of tables

1. INTRODUCTION

This document describes the operation of the IPM429-E-0T-16R Module. The IP module has the following characteristics:

- 16 ARINC 429 receivers.
- Single size, 16-bit Industry Pack interface at 32 MHz.
- Meets or exceeds ARINC 429 electrical and timing specifications.
- Data transfer controlled by interruptions or pooling.
- 32 ARINC words buffer per port.
- Exact time tagging of every ARINC word received with a 32 bit, 1us precision timer.
- Label filtering
- Buffer Overflow detection and error generation.
- Programmable parity (odd, even or none).
- Word length selection (25 or 32 bits).
- Two programmable bit rate generators (from 8 to 256 us / bit with .5us increments).
- 8 general usage TTL compatible IO lines.
- 2 Event capture systems on IO lines with a resolution of 1 us..
- Word length error detection on RX ports..
- Industrial temperature range available (I option).
- On-board IRIG-B decoder (rev.B1 and higher).

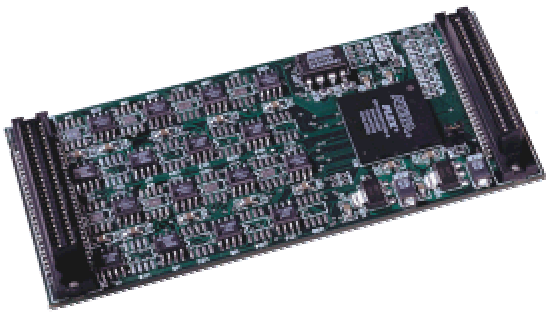


Figure 1-1 : IPM429-E-8T-8R

2. HARDWARE SPECIFICATION

2.1 The Main Controller

The main controller on the IP includes 7 sections:

- IP Bus interface controller.
- IP ID ROM emulation block.
- A set of control and status registers.
- The SRAM data buffers.
- The DATA buffer controller (16 FIFOs within the SRAM).
- Two bit rate generators.
- 16 ARINC 429 receiver controller blocks

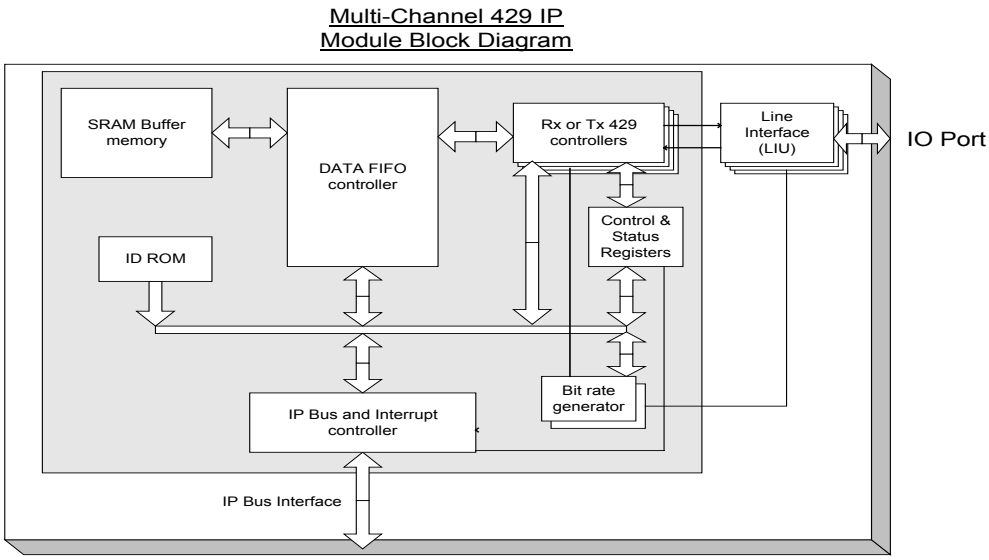


Figure 2-1: IPM429 Block Diagram

2.1.1 The IP Bus Interface Controller

The host uses this interface to configure the module, read statuses and to transfer the data transmitted and received.

2.1.2 The Data Buffers

The FIFO buffers are implemented with an internal SRAM, shared between the IP interface and the ports. A 128 words deep FIFO 16 bits wide is implemented for each port (32 ARINC words per port). Each ARINC word received from the line interface and the associated time tag are saved in the receiver port FIFO. See chapter 4 for details on data transfer structure.

2.1.3 The Main Timer

A 32 bit, 1 us resolution timer is implemented in the main controller. It is primary used for data time stamping for RX ports and IO event capture.

2.1.4 Data Reception

The receiver uses the main timer to tag the ARINC word received. When a word is received, the value of the 32 bits main timer is written to the RX buffers with the ARINC word.

2.1.5 The Bit Rate Generators

Two bit-rate generator registers are provided to program the low and high speed rate of every port. Each port is programmed to operate at the high or the low rate.

2.1.6 The Time Captures

A falling or rising edge on IOs 0 and 1 may generate an interrupt and be time stamped. There is a separate time capture register and interrupt control register for both IOs. The time capture may be used whenever the IO is configured as an input or an output.

2.1.7 The ARINC 429 Label Selection

Before a received ARINC 429 word is written to the RX buffer, the FIFO controller checks if the word's label is in the selection list. If it is, the word is written to the RX buffer, otherwise, the word is trashed. The selection list is mapped in the IP memory region. For each port there is a 256 bits selection array in which each bit is associated with an ARINC label.

2.1.8 The IRIG-B decoder

An IRIG-B decoder on the module can optionally provide IRIG-B time code to the user application and synchronize the IPM-429E to them for true real-time critical applications. Since the module has a 32bit 1us Time tag precision and the IRIG-B time code has a one second resolution, we've implemented an algorithm to precisely correlate the 1us timer to the Pr point of the IRIG-B (start of the time code) or the 1PPS signal if available. (see 3.4.23 IRIGB_SEL Register)

The decoder logic accepts both TTL and standard amplitude-modulated (AM) IRIG-B signal. When using the TTL input, the 1us synchronicity is exact since the rising edge of the TTL output corresponds exactly to the Pr point. When using an AM IRIG-B signal, the decoder circuitry and logic will synchronize on the zero crossing of the beginning of the Pr point of the AM modulated signal. This zero crossing of the low frequency AM modulated signal is not as accurate as the TTL or 1PPS signal. The user may use a 1PPS signal on the TTL input to improve synchronicity when using the IRIG-B AM signal or compensate. (See application Note: <Calibration process for MAX Technologies product with IRIG.pdf>)

2.2 Electrical Interfaces

2.2.1 Input Interfaces

The input line interfaces are the integrated circuit HOLT HI-8588 that are compliant with the ARINC-429 input specification:

- Differential input Threshold 5 +/-0.5V.
- Bipolar (RZ) coding.
- Differential Input Impedance > 100 Kohm

3. MEMORY MAP

3.1 The IP ID ROM region

The IP identification ROM is implemented in one of the ALTERA memory cells. It is preloaded with the proper device definition during the programming phase of the ALTERA.

IP Region	Address	Description	Value (hex)
ID	0	ASCII "VI"	0x5649
	2	ASCII "TA"	0x5441
	4	ASCII "4 "	0x3420
	6	Manufacturer ID high 8 bit	0x0000
	8	Manufacturer ID low 16 bit	0x0000
	A	Model #	0x0009
	C	Revision #	0x00B1
	E	Reserved	0x0000
	10	Driver ID LSB	0x10E8
	12	Driver ID MSB	0x0008 => OPTION*
	14	16-bit flag	0x0004
	16	Number of byte used in the ROM	0x0030
	18	CRC of the byte used in the ROM	0xXXXX
	1A	Active TX PORT	0x0000 (no TX)
	1C	Active RX PORT	0xFFFF(port 0 to 15 are RX)
	1E	Not Used	0x0000
	20	Not Used	0x0000
	22	Last Compatible Model	0x0001
	24	ASCII "IP"	0x4950
	26	ASCII "M4"	0x4D34
	28	ASCII "29"	0x3239
	2A	ASCII "E "	0x4520
	2C	ASCII " "	0x2020
	2E	ASCII " " or ASCII "I " (I option)	0x2020 or 0x4920

Table 3-1: IP ID ROM map

* : Included option bit definition :
  bit 0 : Future use
  bit 1 : Future use
  bit 2 : Future use
  bit 3 : IRIG-B decoder included (Option D)

3.2 The IP MEMORY region

The IP memory region is used to store the label selection table of the RX channels. The following table gives the map of the memory region:

IP Region	Address	Access	Description	RESET Value
MEMORY	00 to 1E	R/W	Label selection bits for RX 0	All 1
	20 to 3E	R/W	Label selection bits for RX 1	All 1
	40 to 5E	R/W	Label selection bits for RX 2	All 1
	60 to 7E	R/W	Label selection bits for RX 3	All 1
	80 to 9E	R/W	Label selection bits for RX 4	All 1
	A0 to BE	R/W	Label selection bits for RX 5	All 1
	C0 to DE	R/W	Label selection bits for RX 6	All 1
	E0 to FE	R/W	Label selection bits for RX 7	All 1
	100 to 11E	R/W	Label selection bits for RX 8	All 1
	120 to 13E	R/W	Label selection bits for RX 9	All 1
	140 to 15E	R/W	Label selection bits for RX 10	All 1
	160 to 17E	R/W	Label selection bits for RX 11	All 1
	180 to 19E	R/W	Label selection bits for RX 12	All 1
	1A0 to 1BE	R/W	Label selection bits for RX 13	All 1
	1C0 to 1DE	R/W	Label selection bits for RX 14	All 1
	1E0 to 1FE	R/W	Label selection bits for RX 15	All 1

Table 3-2: IP Memory map

3.2.1 The Label Selection Tables

The label selection tables are place in the memory region. Each port has a 256 bits representing the ARINC 429 labels. Table 3-3 shows the memory organization for RX 0. The other ports have the same organization.

Address	Word	From label (oct)	To label (oct)
00	LABEL_SELECTION_WORD00	000	017
02	LABEL_SELECTION_WORD01	020	037
04	LABEL_SELECTION_WORD02	040	057
06	LABEL_SELECTION_WORD03	060	077
08	LABEL_SELECTION_WORD04	100	117
0A	LABEL_SELECTION_WORD05	120	137
0C	LABEL_SELECTION_WORD06	140	157
0E	LABEL_SELECTION_WORD07	160	177
10	LABEL_SELECTION_WORD08	200	217
12	LABEL_SELECTION_WORD09	220	237
14	LABEL_SELECTION_WORD10	240	257
16	LABEL_SELECTION_WORD11	260	277
18	LABEL_SELECTION_WORD12	300	317
1A	LABEL_SELECTION_WORD13	320	337
1C	LABEL_SELECTION_WORD14	340	357
1E	LABEL_SELECTION_WORD15	360	377

Table 3-3: Memory region vs ARINC 429 label selection for RX 0

A 1 in the bit corresponding to a label means that an incoming ARINC 429 word with this label will be kept and written to the receiver BUFFER. The next table shows two examples of the LABEL_SELECTION_WORD.

Examples:

LABEL_SELECTION_WORD00

Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARINC 429 label (oct)	017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000

LABEL_SELECTION_WORD15

Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARINC 429 label (oct)	377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360

Table 3-4: LABEL_SELECTION_WORD usage

3.3 The IP IO region

IP Region	Address	IP Bus Bits	Access	Name	RESET Value (hex)
IO	00	[15:0]	R/W	PORT_ENABLE	0x0000
	02	[15:0]	R	BUFFER_STATUS	0x0000
	04	[15:0]	R/W	BUFFER_OVERFLOW	0x0000
	06	[15:0]	R/W	INTERRUPT_ENABLE	0x0000
	08	[15:0]	R/W	PARITY_ENABLE	0x0000
	0A	[15:0]	R/W	WORD_LENGTH	0x0000
	0C	[15:0]	R/W	RATE SELECT	0x0000
	0E	[8:0]	R/W	HIGH_RATE	0x0013
	10	[8:0]	R/W	LOW_RATE	0x009F
	12	[7:0]	R/W	IO_DATA	0x00FF
	14	[7:0]	R/W	IO_DIR	0x0000
	16			N.U.	0x0000
	18			N.U.	0x0000
	1A	[15:0]	R/W	RX_INT_CONFIG	0x0001
	1C	[15:0]	R	TIMER[31:16]	0x0000
	1E	[15:0]	R	TIMER[15:0]	0x0000
	20	[15:0]	R	RX #0 BUFFER	Empty
	22	[15:0]	R	RX #1 BUFFER	Empty
	24	[15:0]	R	RX #2 BUFFER	Empty
	26	[15:0]	R	RX #3 BUFFER	Empty
	28	[15:0]	R	RX #4 BUFFER	Empty
	2A	[15:0]	R	RX #5 BUFFER	Empty
	2C	[15:0]	R	RX #6 BUFFER	Empty
	2E	[15:0]	R	RX #7 BUFFER	Empty
	30	[15:0]	R	RX #8 BUFFER	Empty
	32	[15:0]	R	RX #9 BUFFER	Empty
	34	[15:0]	R	RX #10 BUFFER	Empty
	36	[15:0]	R	RX #11 BUFFER	Empty
	38	[15:0]	R	RX #12 BUFFER	Empty
	3A	[15:0]	R	RX #13 BUFFER	Empty
	3C	[15:0]	R	RX #14 BUFFER	Empty
	3E	[15:0]	R	RX #15 BUFFER	Empty
	40	[15:0]	R/W	TX_RX_ERROR	0x0000
	42	[15:0]	R/W	TX_RX_ERROR_MASK	0x0000
	44			N.U.	0x0000
	46			N.U.	0x0000
	48	[5:0]	R/W	EDGE_CAPTURE_CTRL	0x0000
	4A	[15:0]	R	CAPTURE_TIME0	0x0000
	4C	[15:0]	R	CAPTURE_TIME1	0x0000
	4E-72			N.U.	
	74	[15:0]	R	IRIG_CTT_LOW	0000
	76	[15:0]	R	IRIG_CTT_HIGH	0000
	78	[15:0]	R	IRIG_DATA_LOW	0000
	7A	[13:0]	R	IRIG_DATA_HIGH	0000
	7C	[15:11]	R/W	IRIGB_SEL	0000
	7E		R	Reserved for internal use	0004

Table 3-5: IP IO map.

3.4 Registers Definition and usage

3.4.1 The PORT_ENABLE Register (IO Addr: 0x00)

This register is used to enable or disable each PORT.

Bit	Function	Reset value
0	RX #0 enable	0 (port disabled)
1	RX #1 enable	0 (port disabled)
2	RX #2 enable	0 (port disabled)
3	RX #3 enable	0 (port disabled)
4	RX #4 enable	0 (port disabled)
5	RX #5 enable	0 (port disabled)
6	RX #6 enable	0 (port disabled)
7	RX #7 enable	0 (port disabled)
8	RX #8 enable	0 (port disabled)
9	RX #9 enable	0 (port disabled)
10	RX #10 enable	0 (port disabled)
11	RX #11 enable	0 (port disabled)
12	RX #12 enable	0 (port disabled)
13	RX #13 enable	0 (port disabled)
14	RX #14 enable	0 (port disabled)
15	RX #15 enable	0 (port disabled)

Table 3-6: PORT_ENABLE Register

3.4.2 The BUFFER_STATUS Register (IO Addr: 0x02)

This register contains the status of the data buffers of every port on the module. When set, the status bit indicates that there is some data available in the receive buffer.

For receivers:

IF (ReceiverBuffer[PORT] contains RX_FIFO_AF[1] ARINC words or ARINC words older than AGGING_TIMEOUT) then
    BUFF_STAT[PORT] goes to '1'
ELSE IF (ReceiverBuffer[PORT] is empty) then
    BUFF_STAT[PORT] return to '0'
END IF

Bit	Function		Reset
0	RX #0 Buffer status	RX: 0 => Buffer empty 1 => data available in buffer	0
1	RX #1 Buffer status	"	0
2	RX #2 Buffer status	"	0
3	RX #3 Buffer status	"	0
4	RX #4 Buffer status	"	0
5	RX #5 Buffer status	"	0
6	RX #6 Buffer status	"	0
7	RX #7 Buffer status	"	0
8	RX #8 Buffer status	"	0
9	RX #9 Buffer status	"	0
10	RX #10 Buffer status	"	0
11	RX #11 Buffer status	"	0
12	RX #12 Buffer status	"	0
13	RX #13 Buffer status	"	0
14	RX #14 Buffer status	"	0
15	RX #15 Buffer status	"	0

Table 3-7: BUFFER_STATUS Register

[1] See RX_INT_CONFIG section 3.4.12

3.4.3 The BUFFER_OVERFLOW Register (IO Addr: 0x04)

This register indicates if any of the data buffers has overflowed. The IP_Error* is asserted when there is an overflow condition on at least one port. To reset an overflow condition, there must be a one written to the associated bit (sticky bit).

Bit	Function	Reset
0	RX #0 Buffer OV	0 (no overflow)
1	RX #1 Buffer OV	0
2	RX #2 Buffer OV	0
3	RX #3 Buffer OV	0
4	RX #4 Buffer OV	0
5	RX #5 Buffer OV	0
6	RX #6 Buffer OV	0
7	RX #7 Buffer OV	0
8	RX #8 Buffer OV	0
9	RX #9 Buffer OV	0
10	RX #10 Buffer OV	0
11	RX #11 Buffer OV	0
12	RX #12 Buffer OV	0
13	RX #13 Buffer OV	0
14	RX #14 Buffer OV	0
15	RX #15 Buffer OV	0

Table 3-8: BUFFER_OVERFLOW Register

3.4.4 The INT_ENABLE Register (IO Addr: 0x06)

This register enables or disables the interrupt at each port as a possible interrupt source. When the buffer status bit of a port and the corresponding bit in the interrupt mask register are set, the IntReq0* signal is asserted.

Bit	Function	Reset value
0	RX #0 interrupt enable	0 (INT disabled)
1	RX #1 interrupt enable	"
2	RX #2 interrupt enable	"
3	RX #3 interrupt enable	"
4	RX #4 interrupt enable	"
5	RX #5 interrupt enable	"
6	RX #6 interrupt enable	"
7	RX #7 interrupt enable	"
8	RX #8 interrupt enable	"
9	RX #9 interrupt enable	"
10	RX #10 interrupt enable	"
11	RX #11 interrupt enable	"
12	RX #12 interrupt enable	"
13	RX #13 interrupt enable	"
14	RX #14 interrupt enable	"
15	RX #15 interrupt enable	"

Table 3-9: INT_ENABLE Register

3.4.5 The PARITY_ENABLE Register (IO Addr: 0x08)

This register enables or disables the automatic parity generation. The receivers with the automatic parity check enabled have their parity verified on the module. The parity bit returned to the host is cleared if the ARINC word received has an even parity and set if parity was odd.

Bit #	Function	Description	Reset value
0	RX #0 parity enable	0 => parity check disabled 1 => parity check enabled	0 (no parity check)
1	RX #1 parity enable	"	"
2	RX #2 parity enable	"	"
3	RX #3 parity enable	"	"
4	RX #4 parity enable	"	"
5	RX #5 parity enable	"	"
6	RX #6 parity enable	"	"
7	RX #7 parity enable	"	"
8	RX #8 parity enable	"	"
9	RX #9 parity enable	"	"
10	RX #10 parity enable	"	"
11	RX #11 parity enable	"	"
12	RX #12 parity enable	"	"
13	RX #13 parity enable	"	"
14	RX #14 parity enable	"	"
15	RX #15 parity enable	"	"

Table 3-10: PARITY_ENABLE Register

3.4.6 The WORD_LENGTH Register (IO Addr: 0x0A)

Each port can be programmed to work on 25 or 32 bit ARINC words.

Bit	Function	Reset value
0	RX #0 word length select 0 => 25 bit word 1 => 32 bit word	0 (25 bit word)
1	"	"
2	"	"
3	"	"
4	"	"
5	"	"
6	"	"
7	"	"
8	"	"
9	"	"
10	"	"
11	"	"
12	"	"
13	"	"
14	"	"
15	RX #15 …	"

Table 3-11: WORD_LENGTH Register

3.4.7 The RATE_SELECT Register (IO Addr: 0x0C)

This register selects which of the two rate generators is assigned to every port.

Bit	Function	Reset value
0	RX #0 Rate select	0 Low Rate
1	RX #1 Rate select	“
2	RX #2 Rate select	“
3	RX #3 Rate select	“
4	RX #4 Rate select	“
5	RX #5 Rate select	“
6	RX #6 Rate select	“
7	RX #7 Rate select	“
8	RX #8 Rate select	“
9	RX #9 Rate select	“
10	RX #10 Rate select	“
11	RX #11 Rate select	“
12	RX #12 Rate select	“
13	RX #13 Rate select	“
14	RX #14 Rate select	“
15	RX #15 Rate select	“

Table 3-12: RATE_SELECT Register

3.4.8 The HIGH_RATE Register (IO Addr: 0x0E)

This 9 bits register contains the bit rate used by the ports programmed to operate at the high rate (see the RATE_SELECT register). The rate should never be less than 8 us per bit.

RATE = (HIGH_RATE + 1) * 0.5us

Example:

HIGH_RATE Value (Hex)	Corresponding bit rate
1FF	3906.25 bits / s
09F (low rate reset value)	12.5 Kbit / sec
014	95.238Kbit / sec
013 (high rate reset value)	100 Kbit / sec
00F	125 Kbit / sec

Table 3-13: HIGH_RATE Register

3.4.9 The LOW_RATE Register (IO Addr: 0x10)

This register is identical to the HIGH_RATE register, but is used by ports configured to operate at the low rate.

3.4.10 The IO_DATA Register. (IO Addr: 0x12)

Depending on the data direction chosen for each IO signal, this register returns the value of signals configured as inputs or controls the level on signals configured as outputs.

Bit	Function	Reset value
0	IO #0	1
1	IO #1	1
2	IO #2	1
3	IO #3	1
4	IO #4	1
5	IO #5	1
6	IO #6	1
7	IO #7	1

Table 3-14: IO_DATA Register

3.4.11 The IO_DIR Register. (IO Addr: 0x14)

Controls whether the IO[7:0] signals are input or output signals. When an IO signal is configured as an input, the level present on the line is reflected in the corresponding bit of the IO_DATA register. When configured as an output, the IO_DATA register controls the level present on the corresponding IO signal.

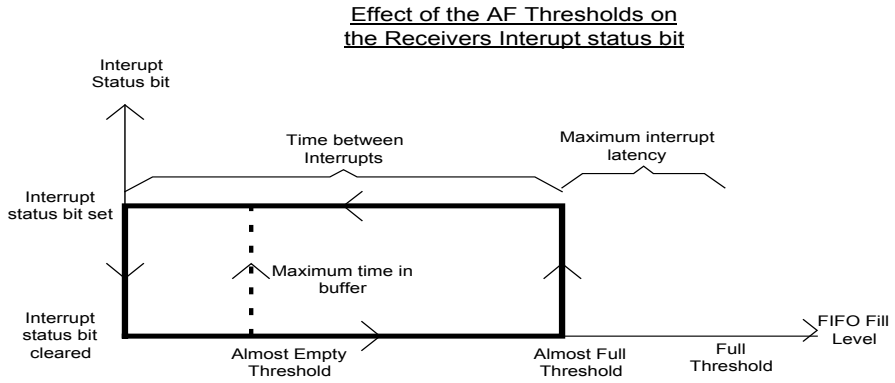
Bit	Function	Reset value
0	0 => IO0 is an input 1 => IO0 is an output	0
1	0 => IO1 is an input 1 => IO1 is an output	0
2	0 => IO2 is an input 1 => IO2 is an output	0
3	0 => IO3 is an input 1 => IO3 is an output	0
4	0 => IO4 is an input 1 => IO4 is an output	0
5	0 => IO5 is an input 1 => IO5 is an output	0
6	0 => IO6 is an input 1 => IO6 is an output	0
7	0 => IO7 is an input 1 => IO7 is an output	0

Table 3-15: IO_DIR Register

3.4.12 The RX_INT_CONFIG Register (IO Addr: 0x1A)

Contains the almost FULL (AF) and the data aging time-out value. The AGING_TIMEOUT parameter is used to prevent the stagnation of received data when the AF threshold is not reached. To obtain the actual time-out period multiply the AGING_TIMEOUT value by 512us. Programming a value of "0" disables the maximum aging surveillance process.

The following figure graphically presents the effect of the AF threshold and the AGING_TIMEOUT time-out on the receivers interrupt status bit.



Bit	NAME	Description	Reset value
[0:5]	RX_FIFO_AF	Fifo Almost Full threshold	04h
[9:6]	AGGING_TIMEOUT	Maximum time that data can reside in a receiver FIFO before setting the receiver interrupt status bit	0 (Disabled)

Table 3-16: RX_INT_CONFIG register

Note that 32 is the maximum quantity of ARINC words that the RX buffer may contain. Values greater than 32 will result in an unreachable threshold.

3.4.13 The TIMER Register (IO Addr: 0x1C & 0x1E)

This register is a latch of the 32 bit main timer. When reading the MSB (address 1C), the latch will freeze until the LSB (address 1E) is read. The IP_Strobe* signal from the carrier board is used to reset this timer. As an example, the IP_Strobe* signal could be asserted simultaneously to all IP modules in order to reset and thus synchronize the timers present on every MAX Technologies IP modules that support this feature.

3.4.14 The TX_RX_ERROR Register (IO Addr: 0x40)

This register sets a flag when a receptor detects a word length error. If the associated mask in the TX_RX_ERROR_MASK_REG register is set to one for receiver, the IP_ERROR* line will be asserted. To reset an error condition, there must be a 1 written to the associated bit (sticky bit).

The RX error detection state machine calculates the word length after detection of a gap of at least 3 bit time on the line interface. We could also say that if no falling/rising edge is detected for a period of at least 3 bit time the word is considered completely transmitted. So this error detection system does not detect silence within a word unless the silence is longer than 3 bit time, but the word length error in terms of bit quantity.

When a short or a long word error occurs, this word, even if it is wrong, is written to the RX buffer. The resulting written word will be, of course, wrong.

Bit #	Function	Description	Reset value
0	RX #0 word error flag	Set to one when a word length received is wrong	0
1	RX #1 word error flag	Set to one when a word length received is wrong	0
2	RX #2 word error flag	Set to one when a word length received is wrong	0
3	RX #3 word error flag	Set to one when a word length received is wrong	0
4	RX #4 word error flag	Set to one when a word length received is wrong	0
5	RX #5 word error flag	Set to one when a word length received is wrong	0
6	RX #6 word error flag	Set to one when a word length received is wrong	0
7	RX #7 word error flag	Set to one when a word length received is wrong	0
8	RX #8 word error flag	Set to one when a word length received is wrong	0
9	RX #9 word error flag	Set to one when a word length received is wrong	0
10	RX #10 word error flag	Set to one when a word length received is wrong	0
11	RX #11 word error flag	Set to one when a word length received is wrong	0
12	RX #12 word error flag	Set to one when a word length received is wrong	0
13	RX #13 word error flag	Set to one when a word length received is wrong	0
14	RX #14 word error flag	Set to one when a word length received is wrong	0
15	RX #15 word error flag	Set to one when a word length received is wrong	0

Table 3-17: TX_RX_ERROR Register

3.4.15 The TX_RX_ERROR_MASK Register (IO Addr: 0x42)

This register enables/disables the error notification on line IP_Error* from the reception of a word with a wrong word length.

Bit #	Function	Description	Reset value
0	RX #0 word error mask	Enable/disable the error on IP_ERROR*	0
1	RX #1 word error mask	Enable/disable the error on IP_ERROR*	0
2	RX #2 word error mask	Enable/disable the error on IP_ERROR*	0
3	RX #3 word error mask	Enable/disable the error on IP_ERROR*	0
4	RX #4 word error mask	Enable/disable the error on IP_ERROR*	0
5	RX #5 word error mask	Enable/disable the error on IP_ERROR*	0
6	RX #6 word error mask	Enable/disable the error on IP_ERROR*	0
7	RX #7 word error mask	Enable/disable the error on IP_ERROR*	0
8	RX #8 word error mask	Enable/disable the error on IP_ERROR*	0
9	RX #9 word error mask	Enable/disable the error on IP_ERROR*	0
10	RX #10 word error mask	Enable/disable the error on IP_ERROR*	0
11	RX #11 word error mask	Enable/disable the error on IP_ERROR*	0
12	RX #12 word error mask	Enable/disable the error on IP_ERROR*	0
13	RX #13 word error mask	Enable/disable the error on IP_ERROR*	0
14	RX #14 word error mask	Enable/disable the error on IP_ERROR*	0
15	RX #15 word error mask	Enable/disable the error on IP_ERROR*	0

Table 3-18: TX_RX_ERROR_MASK Register

3.4.16 The EDGE_CAPTURE_CTRL Register (IO Addr: 0x48)

This register controls the edge type and holds the interrupt masks and flags.

Bit	NAME	Description	Reset value	Access
0	EDGE0	1 : Rising edge on IO0 will set EFLAG0 0 : Falling edge on IO0 will set EFLAG0	0	R/W
1	EDGE1	1 : Rising edge on IO1 will set EFLAG1 0 : Falling edge on IO1 will set EFLAG1	0	R/W
2	EFLAG0	Set when EDGE0 occurred	0	R/WC[1]
3	EFLAG1	Set when EDGE1 occurred	0	R/WC[1]
4	EMASK0	When set and EFLAG0 is set an interrupt is issued on IntReq0*	0	R/W
5	EMASK1	When set and EFLAG1 is set an interrupt is issued on IntReq0*	0	R/W

[1] R/WC is write 1 clear type. Writing 1 to this bit clears this bit.

Table 3-19: EDGE_CAPTURE Register

3.4.17 The TIME_CAPTUREx register. (IO Addr: 0x4Ah and 0x4C)

These 16 bit registers hold the TIMER LSBs copied when the first EDGEx that triggered the EFLAGx occurred. Here are the characteristics of the captured time:

- Resolution : 1 us (LSB weight)
- Precision : + 0.075us (time capture delay)

3.4.18 The IRIGB_CTT register. (IO Addr: 0x74 and 0x76)

When the IRIG-B is enabled, IRIG_CTT is used as a correlation time-tag with the IRIG-B time. When the IRIG-B second is reached, the internal 32-bit microsecond timer is latched, which gives the correlation between the IRIG-B time and microsecond time. In that case, when reading the IRIG_CTT LSB (0x18), the IRIG_CTT MSB and the IRIG_DATA are latched, and will only be latched again after the IRIG_DATA is accessed.

Bit	Description	Reset	Access
[31..0]	IRIG_CTT	0x00000000	R

3.4.19 The IRIGB_DATA register. (IO Addr: 0x78 and 0x7A)

This register shows the last set of information the IRIG-B decoder has captured. If no IRIG-B signal is fed into the decoder circuitry, these registers will remain at their reset value. The following table describes the content of the registers (bits 16 to 31 represent bits 0 to 15 of the second register). These registers are read-only. Note: IRIG-B signals often give Greenwich Mean Time (GMT+00:00).

Bit	Description	Reset value
0-3	Units of seconds	0000
6-4	Tens of seconds	000
10-7	Units of minutes	0000
13-11	Tens of minutes	000
17-14	Units of hours	0000
19-18	Tens of hours	00
23-20	Units of days	0000
27-24	Tens of days	0000
29-28	Hundreds of days	00

3.4.20 The IRIGB_SEL register. (IO Addr: 0x7C)

Bit	Name	Description	Reset value
0..13	Reserved		0
14	IRIGB_TTL_EN	When set to one, the DIO6 is used as a TTL IRIG-B signal. 1 => IRIG-B TTL enable	0
15	IRIGB_TTL_CONFIG	Gives the option between IRIG-B TTL entry or 1PPS signal used in correlation with IRIG-B AM signal. 0 => IRIG-B TTL digital 1 => IRIG-B 1PPS signal	0

The IRIGB_TTL_EN gives the possibility to use the TTL entry. When TTL enabled, the IRIGB_TTL_CONFIG is used to choose between an IRIG-B digital input and a 1PPS digital input, which is a 1 Hz TTL signal, with the rising edge precisely on the IRIG second (used in conjunction with the AM IRIG-B signal). When the IRIG-B digital input is used, the AM IRIG-B input is ignored. (See application note: <Q&A IRIG-B correlation.pdf> for more details on how to correlate Time Tag with IRIG-B Time code).

Note that to use the IRIG-B feature, jumpers of JP1 must be set on pins 4-6 and pins 3-5. When these are set, the IOSYNC6 and the IOSYNC7 cannot be used.

4. DATA TRANSFER

4.1 Data transfer from the ARINC 429 Receivers

Each ARINC word received is time tagged with a 32 bit, 1us timer value. This time tag is read with the data word as shown in the table below.

Data transfer sequence from a receiver:

Read order	Word read from a RX_FIFO
1 (first)	TIME_TAG[31:16]
2	TIME_TAG[15:0]
3	ARINC_DATA[32:17]
4 (last)	ARINC_DATA[16:1]

Table 4-1: Data Read Order

Performances of time tagging :

Resolution : 1us

Precision : Limited by rise/fall time as per ARINC specification and HI-8588 receiver. A deterministic delay range is specified for all RX port from the edge detection on the ARINC line.

Delay range : 0.28 us to 1.28 us

5. Hardware Configuration

5.1 IP IO Connector Definition

The IO connector pin-out is given in the following table:

PIN #	Function
1	GND
2	RX_0A
3	RX_0B
4	RX_1A
5	RX_1B
6	GND
7	RX_2A
8	RX_2B
9	RX_3A
10	RX_3B
11	GND
12	RX_4A
13	RX_4B
14	RX_5A
15	RX_5B
16	GND
17	RX_6A
18	RX_6B
19	RX_7A
20	RX_7B
21	GND
22	RX_8A
23	RX_8B
24	RX_9A
25	RX_9B
26	GND
27	RX_10A
28	RX_10B
29	RX_11A
30	RX_11B
31	GND
32	RX_12A
33	RX_12B
34	RX_13A
35	RX_13B
36	GND
37	RX_14A
38	RX_14B
39	RX_15A
40	RX_15B
41	GND
42	Not Used
43	IO0
44	IO1
45	IO2
46	IO3
47	IO4
48	IO5
49	IO6 / IRIG-B TTL
50	IO7 / IRIG-B AM

Table 5-1: IO Connector Definition

5.2 IP Signal Connector Definition

The IP signal connector pin-out is given in the following table:

PIN #	Signal
1	GND
2	IP_CLK
3	IP_RESET*
4	IP_D0
5	IP_D1
6	IP_D2
7	IP_D3
8	IP_D4
9	IP_D5
10	IP_D6
11	IP_D7
12	IP_D8
13	IP_D9
14	IP_D10
15	IP_D11
16	IP_D12
17	IP_D13
18	IP_D14
19	IP_D15
20	IP_BS0*
21	IP_BS1*
22	- 12 V
23	+ 12 V
24	+ 5 V
25	GND
26	GND
27	+ 5 V
28	IP_R/W*
29	IP_IDSEL*
30	N.U.
31	IP_MEMSEL*
32	N.U.
33	N.U.
34	N.U.
35	IP_IOSEL*
36	IP_RESERVED
37	IP_A1
38	N.U.
39	IP_A2
40	IP_ERROR*
41	IP_A3
42	IP_INTREQ0*
43	IP_A4
44	N.U.
45	IP_A5
46	IP_STROBE*
47	IP_A6
48	IP_ACK*
49	IP_RESERVED
50	GND

N.U. = Not Used

Table 5-2: IP Connector Definition

5.3 Jumpers configuration

J3 is a shunt may be used to disconnect the IP_STROBE* signal to the IP module main controller. This permits to use the IPM429-E-0T-16R on a host board that does not support the IP_STROBE* feature or that has a definition for this signal that is not compatible with the IPM429-E-0T-16R.

J4, J5 and J6 are reserved for future use and shall not have a shunt installed on them.

6. SPECIFICATIONS

6.1 IP specifications

IPM429-E-0T-16R is fully compliant to the Industry Pack module single size type II mechanical specifications

6.2 Environmental Specifications

Operation Temperature (natural convection)	0-55 º C -40-85 º C (I option)
Relative Humidity (non-condensing)	0-95%
Storage Temperature	-55 to 125 º C

Table 6-1: Environmental Specifications

6.3 Electrical Specifications

ARINC 429 input :

- Fully compliant, see Holt HI-8588 for detailed specification

IO Input characteristics :

- TTL compatible
- On board 10K pull-up to +5V
- Under/over shoot diode protection.

IO Output characteristics :

- TTL compatible
  - I_{OH} = +25 mA
  - I_{OL} = -25 mA
- Not current limited *

***WARNING:** Be careful not to switch to an output mode (see next section 3.4.11) if the IO pin is physically connected to another source of voltage (possibly intended for the IO in input mode).

Consumption :

+5V	75mA Typical
+12V	0mA Typical
-12V	0mA Typical

Table 6-2: Electrical Specifications