PAM-ISA Technical Manual

Ordering Number: 006.8037

Rev. December 6, 1999

This version replaces all previous versions of this document. Atlas Copco Controls has made every effort to insure this document is complete and accurate at the time of printing. In accordance with our policy of continuing product improvement, all data in this document is subject to change or correction without prior notice.

Atlas Copco Controls SA
Zone industrielle La Rippe
CH-1303 Penthaz
Switzerland

Ordering No. 006.8037
Rev. December 6, 1999

Table of Contents

Introduction

General information

The PAM – Programmable Axes Managers – is a module developed by Atlas Copco Controls to precisely control and synchronize SAM digital motion controllers.

PAM-ISA is a plug-in axis manager for PC based machine control application. The PAM-ISA has been designed to provide a general purpose low cost PAM platform for standard interfaces using the ISA bus. The PAM-ISA and all SAM controllers are interconnected on EasyBus, a proprietary optical fiber fieldbus (see figure 1)

Features

- On-board Intel 80960KB (25 Mhz).
- Up to 16 Mbytes EDO DRAM.
- Up to 2 Mbytes Flash EPROM.
- Real time clock with 8 Kbytes NVRAM.
- High speed fiber optical ring (EasyBus).
- Safety relay output for incorporation into machine level safety circuit.
- RS-232 serial communication port for tools and service.
- Communication between PC and PAM via 4 KByte dualport memory.
- ISA interface based on IEEE P996 standard.

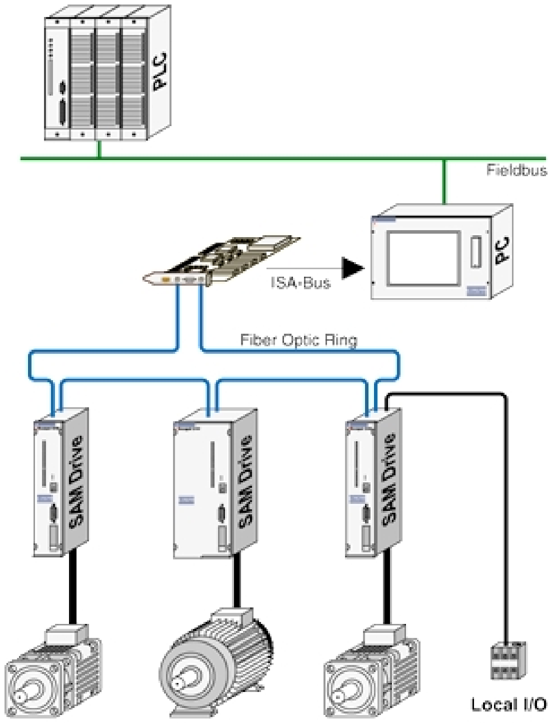


Fig. 1 PAM-ISA system architecture

PAM-ISA block diagram

The PAM-ISA is based on a high performances RISC processor. Supporting this processor are memories, serial interface, EasyBus interface and ISA interface. Figure 2 illustrates the PAM-ISA's block diagram.

The RS-232 interface is used to connect PAM-ISA with a terminal for application monitoring and debugging purposes. The service port is compatible with PAM/SAM test device.

PAM-ISA is equipped with a safety output (the Fatal Error relay) for incorporation into a machine level safety circuit.

The application parameters are stored in battery backed SRAM to ensure fast and safe restart following a power failure.

A 4 Kbyte dualport memory is used for communication between the host PC and the PAM-ISA.

PAM-ISA Architecture

- Intel i960KB
- 32 bit Risc Processor
- 25 Mhz CPU speed

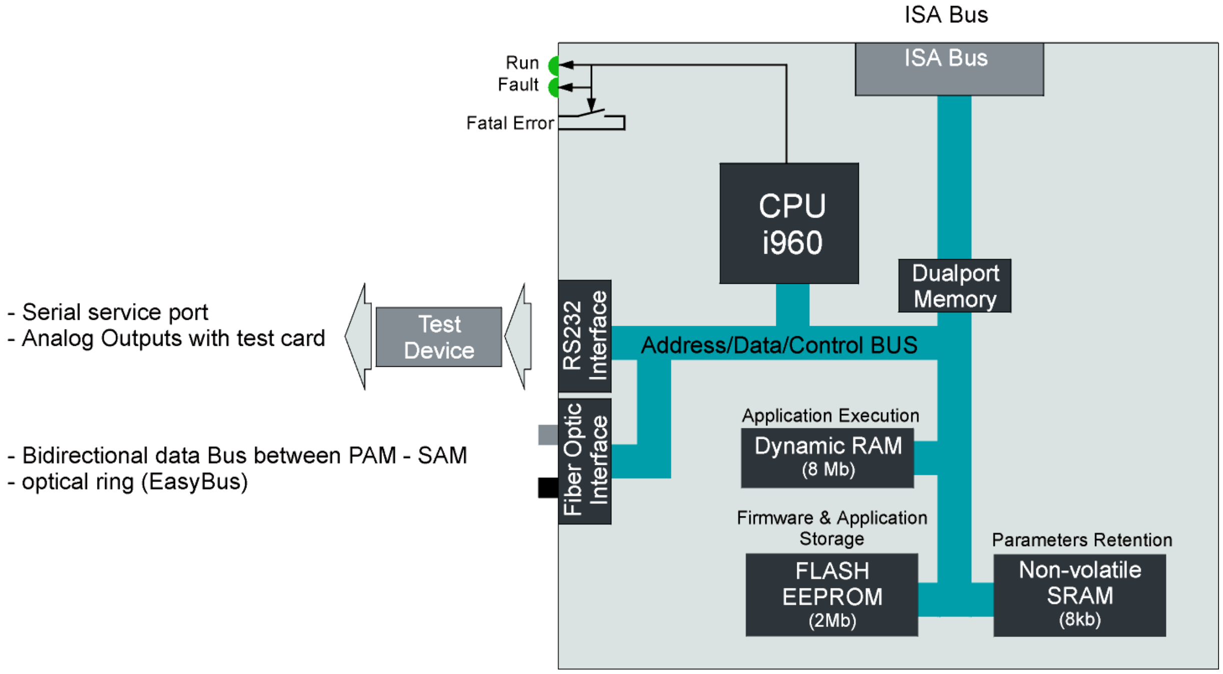


Fig. 2 PAM-ISA general block diagram

ISA interface architecture

Figure 3 illustrates the architecture of the ISA interface. ISA Bus pin assignments are listed in table 1.

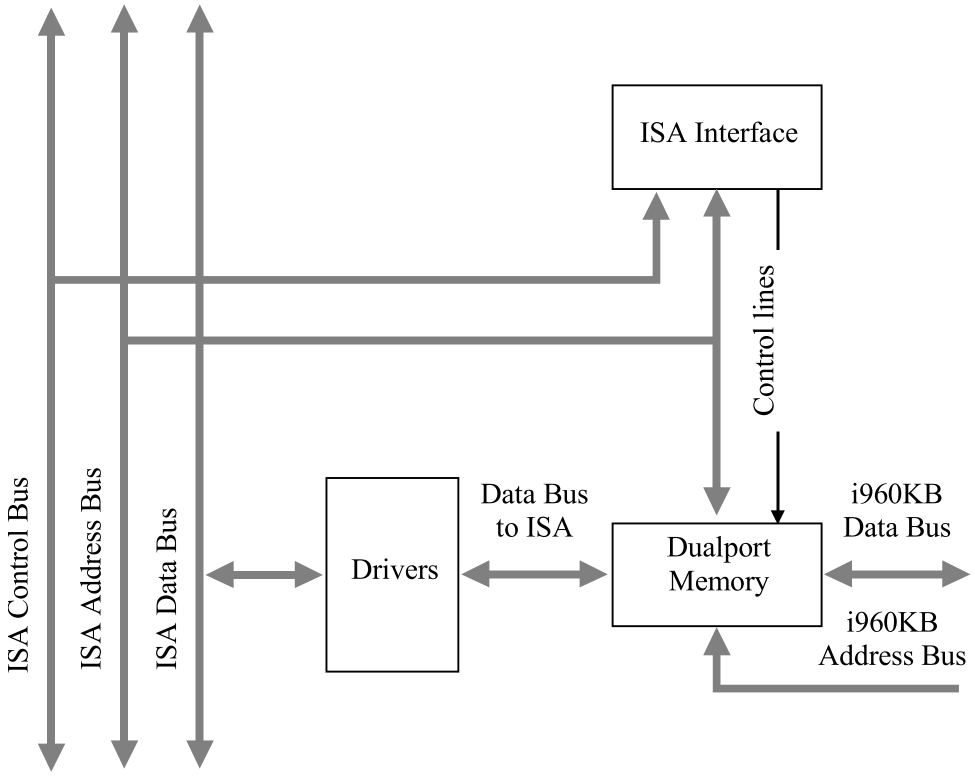


Fig. 3 ISA interface block diagram

Signal	Pin	Pin	Signal
0V	B01	A01	Not Used
RESETDRV	B02	A02	SD7
+5V	B03	A03	SD6
IRQ9	B04	A04	SD5
Not Used	B05	A05	SD4
Not Used	B06	A06	SD3
Not Used	B07	A07	SD2
Not Used	B08	A08	SD1
Not Used	B09	A09	SD0
Not Used	B10	A10	IOCHRDY
Not Used	B11	A11	AEN
Not Used	B12	A12	SA19
#IOW	B13	A13	SA18
#IOR	B14	A14	SA17
Not Used	B15	A15	SA16
Not Used	B16	A16	SA15
Not Used	B17	A17	SA14
Not Used	B18	A18	SA13
Not Used	B19	A19	SA12
Not Used	B20	A20	SA11
Not Used	B21	A21	SA10
Not Used	B22	A22	SA9
IRQ5	B23	A23	SA8
Not Used	B24	A24	SA7
Not Used	B25	A25	SA6
Not Used	B26	A26	SA5
Not Used	B27	A27	SA4
BALE	B28	A28	SA3
+5V	B29	A29	SA2
Not Used	B30	A30	SA1
0V	B31	A31	SA0

Signal	Pin	Pin	Signal
Not Used	D01	C01	Not Used
Not Used	D02	C02	LA23
IRQ10	D03	C03	LA22
IRQ11	D04	C04	LA21
IRQ12	D05	C05	LA20
IRQ15	D06	C06	LA19
Not Used	D07	C07	LA18
Not Used	D08	C08	LA17
Not Used	D09	C09	#MEMR
Not Used	D10	C10	#MEMW
Not Used	D11	C11	Not Used
Not Used	D12	C12	Not Used
Not Used	D13	C13	Not Used
Not Used	D14	C14	Not Used
Not Used	D15	C15	Not Used
+5V	D16	C16	Not Used
Not Used	D17	C17	Not Used
0V	D18	C18	Not Used

Table 1 ISA Bus pin assignments

Note:

The comment “not used” means the ISA signal is not used by the PAM-ISA.

Signals beginning with # are active in the low state.

PAM-ISA I/O port address configuration

PAM-ISA needs I/O access via IN or OUT to configure the dualport's mapping. The I/O address areas used by PAM-ISA can be selected by jumpers of X2 (see figure 4).

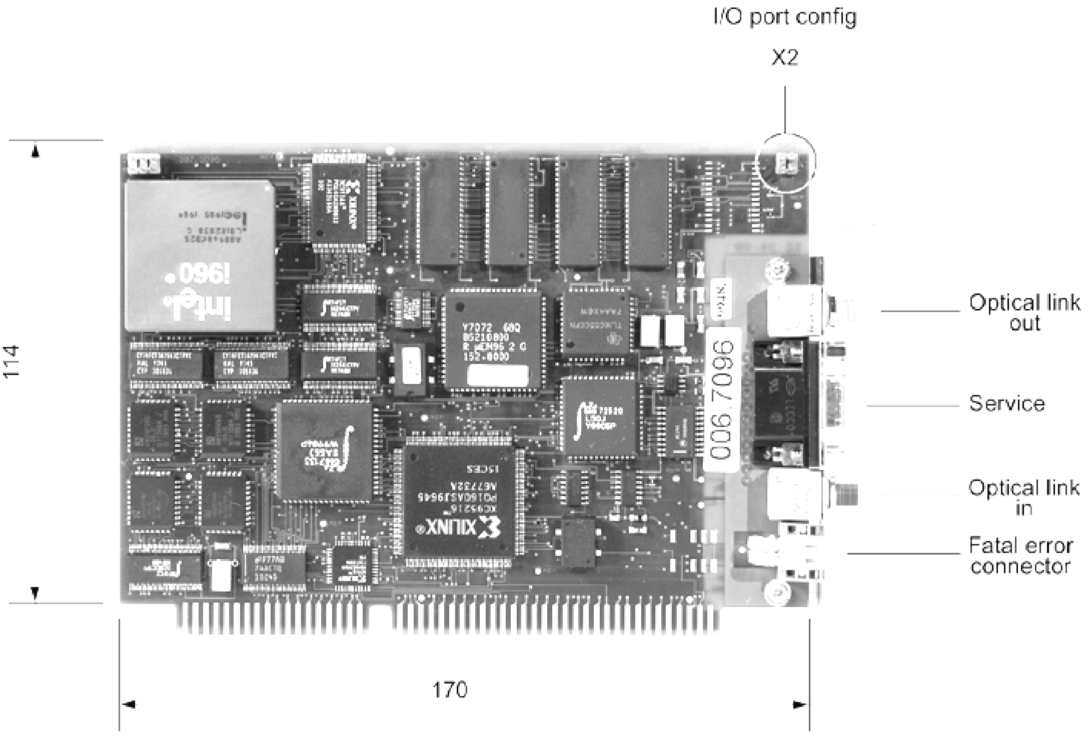


Fig. 4 PAM-ISA

Table 2 shows 4 available configurations of PAM-ISA's I/O port address. The factory (default setting) address configuration is 300 Hex to 307 Hex.

I/O port address	Jumper of X2
300 Hex – 307 Hex	1 – 2 : short 3 – 4 : short (factory default)
308 Hex – 30F Hex	1 – 2 : short 3 – 4 : open
310 Hex – 317 Hex	1 – 2 : open 3 – 4 : short
318 Hex – 31F Hex	1 – 2 : open 3 – 4 : open

Tab. 2 ISA I/O port address configuration

PAM-ISA Internal registers

There are 6 internal registers implemented on the ISA interface of PAM-ISA. These registers are used to update the PAM's hardware and to configure dualport's mapping.

All internal registers are shown in table 3. The " IO Base address " means the start address of the I/O port address configuration (300 Hex is the factory default IO base address).

IO Base address + 00 h	Interrupt control register, reserved
IO Base address + 01 h	Start address register (LSB)
IO Base address + 02 h	Start address register (MSB)
IO Base address + 03 h	Command register
IO Base address + 04 h	Interrupt Acknowledge, Reserved
IO Base address + 05 h	Reset PAM-ISA register
IO Base address + 06 h	Jtag control register, reserved
IO Base address + 07 h	LFSR

Tab. 3 Internal registers of PAM-ISA

The internal registers block diagram is illustrated in figure 5.

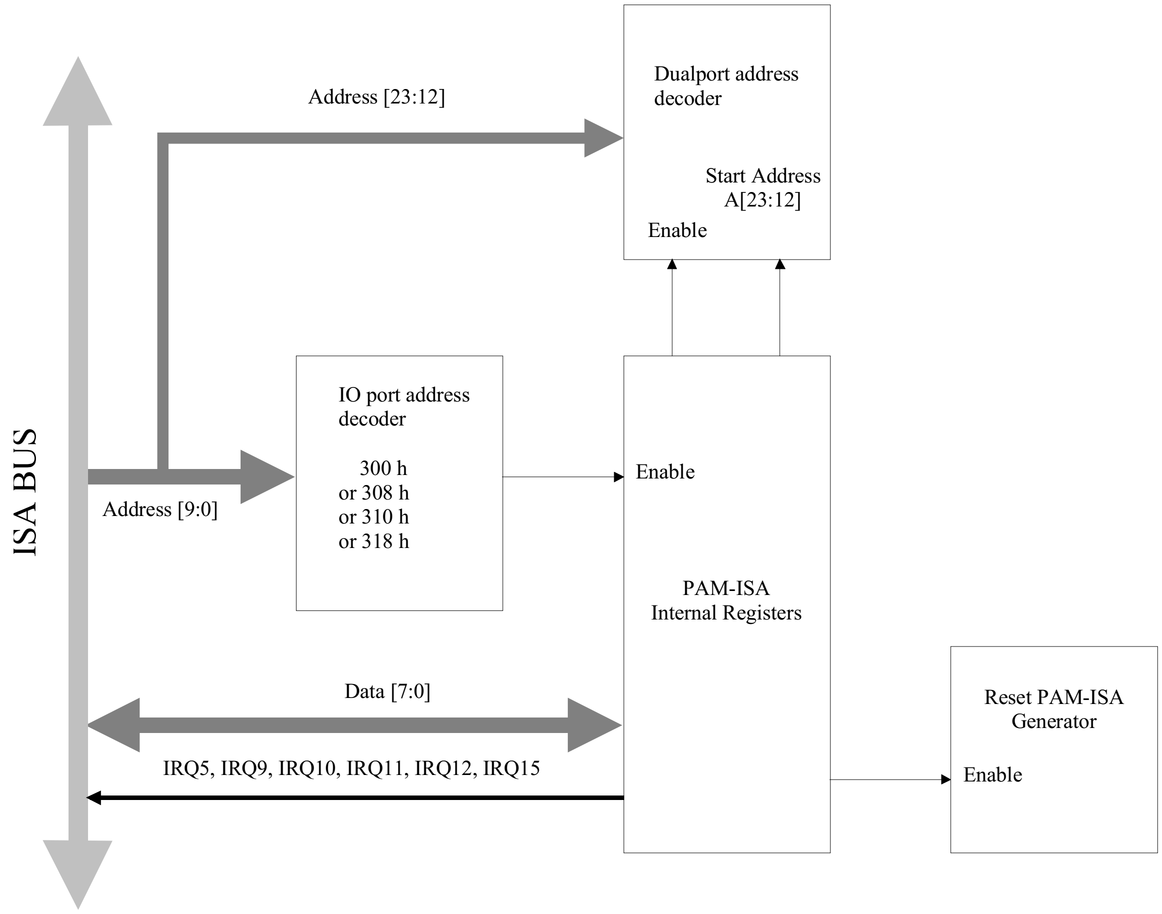


Fig. 5 PAM-ISA internal registers block diagram

Linear Feedback Shift Register (LFSR)

This register is used to identify the presence of PAM-ISA on the PC. The PAM-ISA card uses this LFSR register to generate the values that verify the initiation key. The LFSR is a 4-bit shift register that resets to the value 0x0A. Figure 6 shows a diagram of the LFSR.

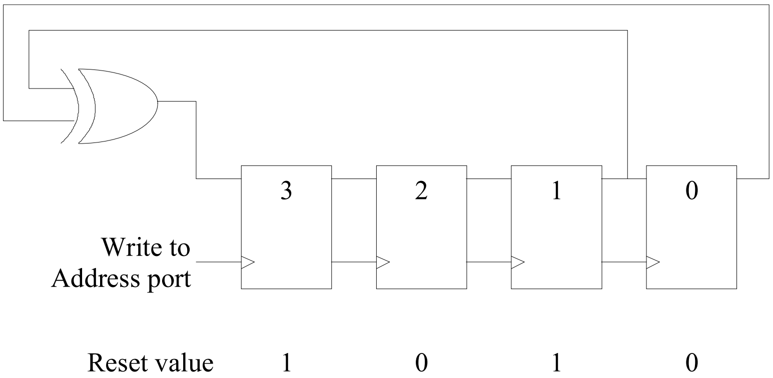


Fig. 6 Initiation key LFSR

The LFSR resets to its initial state (0x0A) any time the PAM-ISA card receives a write at this address that does not match the value currently in LFSR. To ensure that the LFSR is in the initial state, the system software performs two write operations of the value 0x00 to the LFSR address port before sending the initiation key.

The system software then begins writing the values of the initiation key to the LFSR address port. The sequence of value for the initiation key in hexadecimal notation are :

0x0A, 0x0A, 0x0D, 0x0E, 0x0F, 0x07, 0x03, 0x01, 0x08, 0x04, 0x02, 0x09

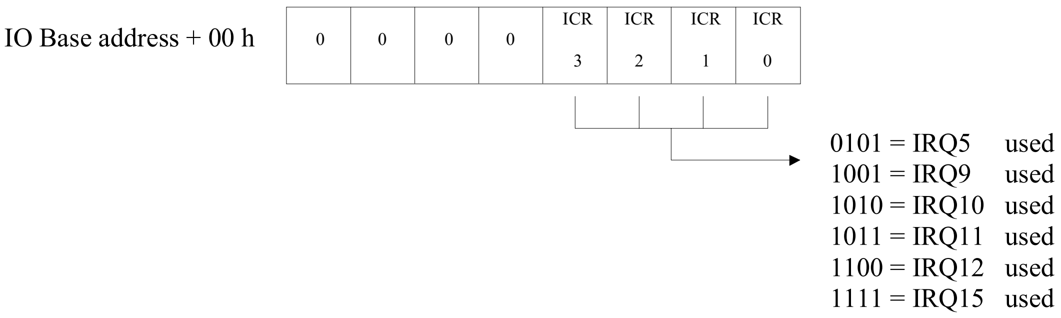
Interrupt control register (ICR)

One of six PC IRQ lines can be used by the PAM-ISA. The read/write register enables the corresponding IRQ line of the PC when the PAM-ISA generates an ISA interrupt request. This function is reserved and not implemented in the current Promotion.

Figure 7 illustrates the contents of this register.

The reset value (reset only by RESETDRV signal on ISA slot) of the ICR is 0x00. The system software reset or test device button reset do not affect the ICR's contents.

Interrupt control register (ICR), reserved



IO Base address + 00 h	0	0	0	0	ICR 3	ICR 2	ICR 1	ICR 0

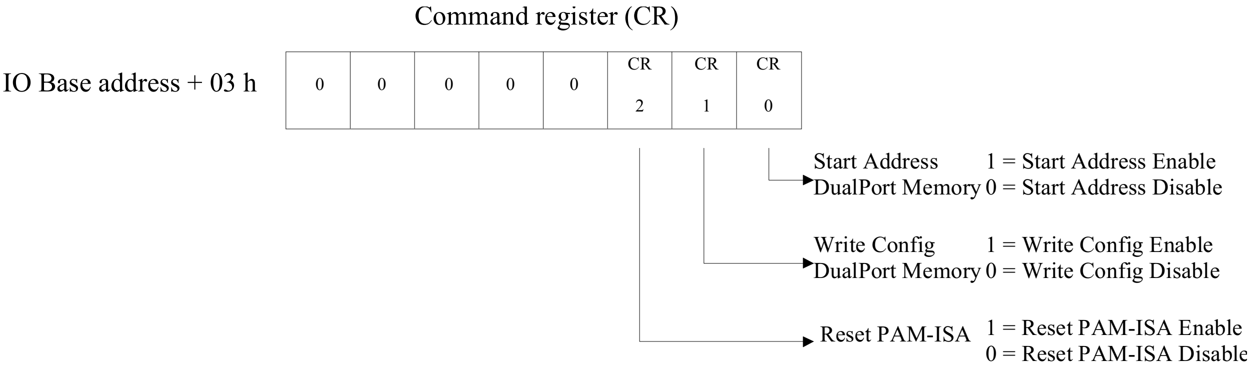
0101 = IRQ5	used
1001 = IRQ9	used
1010 = IRQ10	used
1011 = IRQ11	used
1100 = IRQ12	used
1111 = IRQ15	used

Fig. 7 Interrupt register

Command register (CR)

This read/write register is used to enable the writing registers and the to enable the address decoder of the PAM's dualport memory. The contents of this register is described in figure 8.

The reset value (reset only by RESETDRV signal on ISA slot) of the Command Register (CR) is 0x00. The system software reset or test device button reset do not affect the CR's bit0 (Start Address DualPort Enable/Disable).



Command register (CR)

IO Base address + 03 h

0	0	0	0	0	CR 2	CR 1	CR 0

Start Address DualPort Memory 1 = Start Address Enable
0 = Start Address Disable

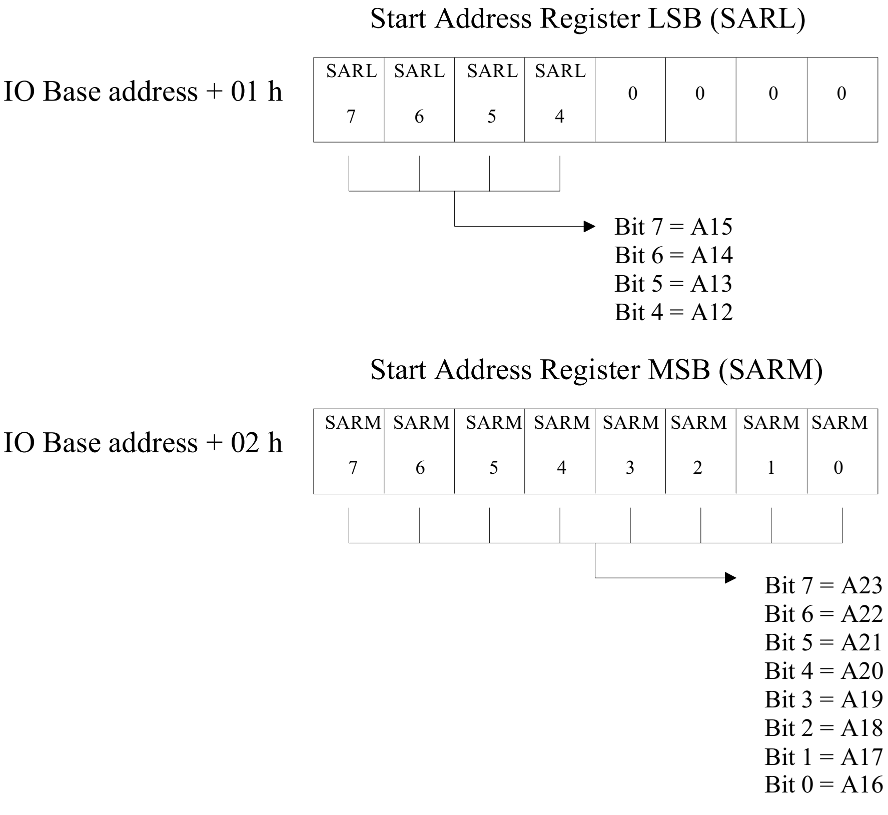
Write Config DualPort Memory 1 = Write Config Enable
0 = Write Config Disable

Reset PAM-ISA 1 = Reset PAM-ISA Enable
0 = Reset PAM-ISA Disable

Fig. 8 Command register

Start address registers (MSB & LSB)

These read/write registers are used to configure the start address of the dualport memory (mapping). The contents of these registers are shown in figure 9.



Start Address Register LSB (SARL)

IO Base address + 01 h

SARL 7	SARL 6	SARL 5	SARL 4	0	0	0	0

Bit 7 = A15
Bit 6 = A14
Bit 5 = A13
Bit 4 = A12

Start Address Register MSB (SARM)

IO Base address + 02 h

SARM 7	SARM 6	SARM 5	SARM 4	SARM 3	SARM 2	SARM 1	SARM 0

Bit 7 = A23
Bit 6 = A22
Bit 5 = A21
Bit 4 = A20
Bit 3 = A19
Bit 2 = A18
Bit 1 = A17
Bit 0 = A16

Fig. 9 Start address registers

Start address register MSB (SARM) and start address register LSB (SARL) form a complete base address of the dualport memory.

The reset value (reset only by RESETDRV signal on ISA slot) of the SARM and

SARL respectively is 0xFF and 0xF0. The system software reset or test device button reset do not affect these registers.

The dualport configuration procedure is as follows:

- Enable the writing SARL and SARM by setting bit1 of Command Register (CR)
- Write the LSB start address into SARL (A15:A12)
- Write the MSB start address into SARM (A23:A16)
- Enable the decoder address by setting bit0 of Command Register (CR)

Reset PAM-ISA register (RPR)

PAM-ISA can be reset by the ISA signal (RESETDRV) or by the system software. Resetting by the system software generates a hardware reset of PAM-ISA (see figure 5). The PAM-ISA software reset effect is identical to the test device button reset.

The reset procedure is as follows:

- Enable the PAM's reset generator by setting bit2 of the Command Register (CR)
- Generate a hardware reset by writing 0x01 value to Reset Pam-ISA Register (RPR)

The reset value of Reset PAM-ISA register (RPR) is 0x00.

JTAG control register (JCR)

This register is reserved to update PAM's hardware. Accidentally writing to this register when the value in the LFSR is 0x09 may cause the malfunction of the PAM-ISA.

PAM-ISA DualPort Memory Configuration

The total capacity of the DualPort Memory (DPM) is 4K bytes. It can be mapped into free space memory of ISA Bus by the system software.

On power up of the PC, the DPM is disabled and can not be used. Access to the DPM is possible after completed the following configuration procedure:

- Enable the writing SARL and SARM by setting bit1 of Command Register (CR)
- Write the LSB start address into SARL (A15:A12)
- Write the MSB start address into SARM (A23:A16)
- Enable the decoder address by setting bit0 of Command Register (CR)

See the programming examples on the following pages for more details.

EXAMPLES 4

The following example in C code gives an ideas for configuration programming. The users can modify these codes to adapt to their drivers.

```
/*************************************************************/
/* File: main                                                */
/* Abstract: This file contains the main function            */
/*        call for configure the mapping's DPM of PAM-ISA    */
/*************************************************************/


#include <io.h>
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>


void main (void)
{
        printf ("\nPAM-ISA DualPort Configuration - version 1.00 - \n");
        printf ("Copyrigh (C) ACC 1999 All rights reserved.\n\n");

        /* Identify the PAM-ISA card */
        if (FindCard() == 1) {
                printf("ERROR: PAM-ISA Identification\n");
                exit (1);
        }

        /* Configure the DualPort Memory */
        if (Config_DPM (0x300, 0xD4000) == 1) {
                printf("ERROR: PAM-ISA DualPort Configuration\n");
                exit (1);
        }

        /* Reset PAM-ISA */
        if (Reset_PAM (0x300) == 1) {
                printf("ERROR: PAM-ISA Reset\n");
                exit (1);
        }

        printf ("Configuration done.\n\n");
        exit (0) ;
}
```

Identification Example

```
/************************************************************/
/* This routine looks for the PAM-ISA card                  */
/* and returns the following values:                        */
/*    VALUE         CAUSE                                   */
/*     0            PAM-ISA found                           */
/*     1            PAM-ISA not found                       */
/************************************************************/
int findcard (short PortAddr)
{
        short POUT_lfsr;            /* addr of LFSR write Port */
        short PSTS_lfsr;/* addr of LFSR read  port */

        POUT_lfsr  = PortAddr + 7;
        PSTS_lfsr  = PortAddr + 7;


   /* To ensure that the LFSR is in the initial state, the system software performs two
   write operations of the value 0x00 to the LFSR address port */

        _outp(POUT_lfsr, 0x0);  /* dummy write */
        _outp(POUT_lfsr, 0x0);  /* dummy write */
        /* Send the initiation key */
        _outp(POUT_lfsr, 0x0a);
        if ((_inp(PSTS_lfsr) & 0x0f) != 0x0a) return (1);
        _outp(POUT_lfsr, 0x0a);
        if ((_inp(PSTS_lfsr) & 0x0f) != 0x0d) return (1);
        _outp(POUT_lfsr, 0x0d);
        if ((_inp(PSTS_lfsr) & 0x0f) != 0x0e) return (1);
        _outp(POUT_lfsr, 0x0e);
        if ((_inp(PSTS_lfsr) & 0x0f) != 0x0f) return (1);
        _outp(POUT_lfsr, 0x0f);
        if ((_inp(PSTS_lfsr) & 0x0f) != 0x07) return (1);
        _outp(POUT_lfsr, 0x07);
        if ((_inp(PSTS_lfsr) & 0x0f) != 0x03) return (1);
        _outp(POUT_lfsr, 0x03);
        if ((_inp(PSTS_lfsr) & 0x0f) != 0x01) return (1);
        _outp(POUT_lfsr, 0x01);
        if ((_inp(PSTS_lfsr) & 0x0f) != 0x08) return (1);
        _outp(POUT_lfsr, 0x08);
        if ((_inp(PSTS_lfsr) & 0x0f) != 0x04) return (1);
        _outp(POUT_lfsr, 0x04);
        if ((_inp(PSTS_lfsr) & 0x0f) != 0x02) return (1);
        _outp(POUT_lfsr, 0x02);
        if ((_inp(PSTS_lfsr) & 0x0f) != 0x09) return (1);
        /* To disable the writing to the Jtag Control Register (JCR) accidentally ,
        LFSR must be in the initial state.

        The system software performs two write operations of the value 0x00 to the
        LFSR address port to ensure it */
        _outp(POUT_lfsr, 0x0);  /* dummy write */
        _outp(POUT_lfsr, 0x0);  /* dummy write */

        return (0);           /* PAM-ISA found */

}
```

DualPort Configuration Example

```
/**************************************************************/
/* This routine configures the DualPort memory                */
/* and returns the following values:                          */
/*    VALUE        CAUSE                                      */
/*     0           Config. OK                                 */
/*     1           Config. Problem                            */
/**************************************************************/

#define ADDR_EN             0x01    /* mapping bit enable */
#define CONFIG_EN           0x02    /* config bit enable  */

int Config_DPM (short PortAddr, long Start_Address)
{
        short PADDL_addr;       /* addr of lsb address port */
        short PADDH_addr;       /* addr of msb address Port */
        short PCMD_addr;        /* addr of command port     */
        int Address_LSB, Address_MSB ;

        PADDL_addr = PortAddr + 1 ;
        PADDH_addr = PortAddr + 2 ;
        PCMD_addr  = PortAddr + 3 ;

        /* Enable the config bit in xilinx chip */
        _outp(PCMD_addr, CONFIG_EN);
        if (_inp(PCMD_addr != CONFIG_EN) {
                printf("ERROR write command\n");
                return (1); /* error write*/
        }
        /* Configure A15-A12 of DPM
         D7 -> A15, D6 -> A14, D5 -> A13, D4 -> A12
         D3-D0 -> don't care */
        Address_LSB = (int)((Start_Address >> 8) & 0xF0);
        _outp(PADDL_addr, Address_LSB);
        if (_inp(PADDL_addr != Address_LSB) {
                printf("ERROR write lsb address\n");
                return (1); /* error write*/
        }
        /* Config A23-A16 of DPM
         D7 -> A23, D6 -> A22, D5 -> A21, D4 -> A20
         D3 -> A19, D2 -> A18, D1 -> A17, D0 -> A16 */
        Address_MSB = (int)((Start_Address >> 16) & 0xFF);
        _outp(PADDH_addr, Address_MSB);
        if (_inp(PADDH_addr) != Address_MSB) {
                printf("ERROR write msb address\n");
                return (1); /* error write*/
        }
        /* Enable the DPM decoder in xilinx chip */
        _outp(PCMD_addr, ADDR_EN);
        if (_inp(PCMD_addr) != ADDR_EN) {
                printf("ERROR write command\n");
                return (1); /* error write*/
        }
        printf ("\nDPM Configuration completed successfully.\n");
        return (0)          /* Successfully */

}
```

Pam Reset Example

```
/*************************************************************/
/* This routine resets the PAM-ISA                          */
/* and returns the following values:                        */
/*    VALUE        CAUSE                                     */
/*     0           Reset OK                                  */
/*     1           Reset Problem                             */
/*************************************************************/

#define RESET_EN        0x04     /* Reset bit enable   */
#define RESET  0x01     /* Reset value        */

int Reset_PAM (short PortAddr)
{
        short PCMD_addr;        /* addr of command port    */
        short PRST_addr;        /* addr of Reset port      */
        int cmd;

        PCMD_addr  = PortAddr + 3 ;
        PRST _addr = PortAddr + 5 ;

        /* Enable the Reset Control bit in xilinx chip */
        cmd = _inp(PCMD_addr) | RESET_EN;
        _outp(PCMD_addr, cmd);
        if (_inp(PCMD_addr) != cmd) {
                printf("ERROR write command\n");
                return (1); /* error write*/
        }

        /* Reset PAM-ISA */
        _outp(PRST_addr, RESET);

        printf("Reset PAM-ISA done...\n");

        return (0);         /* Successfully */
}
```

PAM-ISA Specifications 5

	MIN	NOM	MAX
Supply voltage VCC	4.75 VDC	5 VDC	5.25 VDC
Supply current		1.4 A	
Operating temperature (without fan)	0 °C		55 °C
Safety relay output Nominal switching capacity (resistive) Max. switching current Max. switching voltage AC Max. switching voltage DC		2A 24 VDC	2 A 50 VAC 75 VDC

Tab. 4 PAM-ISA Specifications