marian

MARC 2 / PRO Hardware Documentation

Version 1.1
01.11.2001

Introduction

This document contains all technical information which are required to develop driver software for digital audio processing within the framework of the technical properties of the “MARC 2 “ and the “MARC 2 PRO”. The “MARC 2 PRO” is the extended version of the “MARC 2”. Technical properties given by this “PRO”-version only are identified properly.

Memory Ranges

The MARC 2 / PRO provides 4 memory ranges, which are of interest for the audio functionality of the card.

The access to this memory ranges has do be DWORD oriented only.

Nr.	Length (Bytes)	Area	Use	Commentary
0	128	PLX-Register	Configuration of the PCI-Bridge	
1	131072	RAM	Play and Capture Data	
2	1024	FPGA WR-Register	Control Register	All Write Only
		FPGA RD-Register	Status Register	All Read Only
3	512	FPGA Load	Loading the FPGA	Only available after: Power Up or local Reset
		HW WR-Register	Digital Input Source Select	Available after FPGA loading Write Only
		HW RD-Register	Digital Input Status	Available after FPGA loading Read Only

PLX Register (Memory Range 0, Read + Write)

This memory range provides the access to the registers of the PLX PCI-Bridge-Controllers. The memory address of a register can be calculated: base address of the memory range + register number. Of main interest are the register 0x4C and 0x50 for interrupt and reset control. For further information about the functionality of these registers please see the manufacturers documentation ‘plx.pdf'.

RAM (Memory Range 1, Read + Write)

The RAM occupies a memory range of 0x20000 (128k) byte and serves to exchange the audio data between card and driver.

Adresse	Bereich
0x00000000	Analog Play Buffer Left Channel
0x00004000	Analog Play Buffer Right Channel
0x00008000	Analog Capture Buffer Left Channel
0x0000C000	Analog Capture Buffer Right Channel
0x00010000	Digital Play Buffer Left Channel
0x00014000	Digital Play Buffer Right Channel
0x00018000	Digital Capture Buffer Left Channel
0x0001C000	Digital Capture Buffer Right Channel

The hardware buffer of the several channels has a length of 16 Kbytes and is 32 bit mono sample oriented.
Only the Bits 0..23 can be used for every mono sample.
Samples are to write and read MSB justified (highest bit = D23) always.

You will find sample code for audio data transfer in file „transfer.pas“.

FPGA WR Register (Memory Range 2, Write Only)

All signals are general H-active. After the FPGA initialisation all registers are initialised to 0.

Address (Base + n)	Register
0x00	Control
0x04	SyncBus
0x08	Internal Clock Generator
0x0C	-
0x10	-
0x14	-
0x18	Extended Control
0x1C	RAM Control

Control (Base + 0x00)

Bit	0	Analog	Play	Enable		1)
	1	Analog	Capture	‘’		
	2	Digital	Play	‘’		
	3	Digital	Capture	‘’		
	4	Analog	Play	Counter Reset		2)
	5	Analog	Capture	‘’		
	6	Digital	Play	‘’		
	7	Digital	Capture	‘’		
	8	Analog	Play	Run		3)
	9	Analog	Capture	‘’		
	10	Digital	Play	‘’		
	11	Digital	Capture	‘’		
	12	Analog	Play+Capture	Clock Select	0	4)
	13			‘’	1	
	14	Digital	Play	Clock Select	0	4)
	15			‘’	1	
	16	-				
	17	-				
	18	Analog	Output	Routing	0	5)
	19			‘’	1	
	20	Digital	Output	Routing	0	6)
	21			‘’	1	
	22	ADDA Freq High				7)
	23	Clear Interrupt				8)

1) Play/Capture Enable

The setting of the respective bits activates the appropriate devices. Since the *SyncBus*–Feature is available Play/Capture Enable will NOT start recording or playing of the devices. The device will remain in a sort of „standby“ mode and get started only if the Run-Bit is set (while in SyncBus-Master-Mode) or a start is signalized via SyncBus (while in SyncBus-Slave-Mode.)

Setting back the respective bits disables the appropriate devices. A running recording or playback stops even if there is no stop signal via „Run“ or SyncBus.

„Enable“ and „Run“/“Start of the SyncBus“ are AND connected always.

For sample code see the file „control.pas“ under „_EnablePlay“, „_EnableRec“.

2) Counter Reset

To set the Sample Counter of the respective devices to zero, the appropriate bit has to be set.

The counter receives the value 0. The counter will be blocked until the appropriate bit is set back.

For sample code see the file „control.pas“ under „_Reset“.

3) Run

This bit connected with the Enable-Bit starts the recording or playback of a device. Within the SyncBus-Master-Mode the start signal will be generated on the SyncBus, in such a way as to enable the pending slave devices with active Enable-Bit to start. The setting of this bit is not of any importance to the SyncBus-Slave-Mode.

For sample code see the data „control.pas“ under „_Start“.

4) Clock Select

These settings determine the origin of the master clock for the appropriate device.

Bit	1 0	
	0 0	No Clock
	0 1	Internal Clock Generator
	1 0	SyncBus Clock
	1 1	Digital Input Clock

A device has to be connected with a clock source to work at that sample rate, for which the internal clock generator had been configured respectively, which an external clock source is working with.

Before you change the Clock for the Analog Output, you have to hold the ADDA Chip at Reset and send it in the Power Down Mode. After switching the Clock, Power Up and release of Reset, the ADDA Chip will be ready to operate without having to reinitialise it.

The registers description of the ADDA Chips you will find in the manufacturers documentation „4524.pdf“

For sample code see the file „control.pas“ under „hws_SetDevsClockSource“.

For Digital Play you have to handle internal registers of the Digital Codec as well.

You will find a register description in the manufacturers documentation “cs8427.pdf“. For sample code see „control.pas“ under „dig_SetPlayStandartMode“ and „dig_SetPlaySlaveMode“.

5) Analog Output Routing

These settings determine what you will hear at the analog output.

Bit	1 0	
	0 0	Playback of the Analog Play Audio data on RAM
	0 1	Digital Output
	1 0	Analog Input
	1 1	Digital Input

If the Digital Output is routed to Analog Input, it is not allowed to route the Analog Output to Digital Play or Digital Input.

If the Analog Record is enabled, the Analog Output has to be routed to Analog Input or Analog Playback only.

Before changing the routing of the Analog Output, you have to hold the ADDA Chip at Reset and send it in the Power Down Mode. After finishing the switch of Routing, Power Up and release of Reset, the ADDA Chip is ready to be operated without having to reinstall the chip.

The registers description of the ADDA Chips you will find in the manufacturers documentation „4524.pdf“

For sample code see the file “control.pas” under “anlg_outSetRouting“.

6) Digital Output Routing

These settings determine, what you hear at the digital stereo output.

Bit	1 0	
	0 0	Playback of the Digital Play Audio data in RAM
	0 1	Analog Output
	1 0	Analog Input
	1 1	Digital Input

If the Analog Output is routed to Digital Input, it is not allowed to route the Digital Output to Analog Play or Analog Input.

If the Digital Record is enabled, the Digital Output has to be routed to Digital Input or Digital playback only.

For sample code see the file "control.pas" under "dig_outSetRouting".

7) ADDA Freq High

Through this bit a filter frequency adjustment within the ADDA Chip will be regulated. You have to set this bit for analog recording and playing back with a sample rate >= 48kHz. Is the sample rate below 48kHz, the bit has to be 0.

For sample code see the file „control.pas" under „anlg_OnRateChanged".

8) Clear Interrupt

After an interrupt has been released, you have to clear it on the card. It will be cleared after setting and subsequently resetting the bit. For sample code see the file „isr.asm"

SyncBus (Base + 0x04)

The SyncBus ensures that several *MARIAN SyncBus* compatible audio cards run synchronously. Through this register, the devices Start/Stop behaviour towards the SyncBus can be configured. For further information about the configuration of the cards behaviour towards the SyncBus please see SyncBus Clock/Mode.

Bit		
	0	Analog Play Slave
	1	Analog Record Slave
	2	Digital Play Slave
	3	Digital Record Slave
	4	Analog Play Master
	5	Analog Record Master
	6	Digital Play Master
	7	Digital Record Master

If a device is configured as Slave and the Enable-Bit is set, it will start only if the SyncBus detects a start signal.

If a device is configured as Master, the setting of the appropriate Run-Bits will lead to a start signal at the SyncBus.

A device has to be configured as Master and Slave, if you want to generate a SyncBus start signal and start the device simultaneously.

If a device is configured neither as Master nor as Slave, it will be autonomous. To start and stop only the bits Enable and Run will be used.

Internal Clock Generator (Base + 0x08)

These settings determine the frequency of the Internal Clock Generator, which is available via the appropriate Clock Select Settings for the several devices. The generator generates the Master Clock for the declared sample frequency.

Eventually additional settings for the connected devices are necessary. For further information see then sample code in „control.pas“ under „_OnRateChanged“.

Bit 0..15 Frequency

The value to generate a determined frequency has to be calculated as follows:

Value = Sample Rate * 0.0524288

Depending on the required sample rate the generator has to be in a suitable work range. (see Clock Generator Range).

For sample code see the file „control.pas“ under „SetDcoRate“.

Extended Control (Base + 0x18)

Bit	0	Serial Coder	Data	1)
	1	‘’	Clock	
	2	‘’	Chip Select Digital	
	3	‘’	Chip Select Analog	
	4	Clock Generator Range	0	2)
	5	‘’	1	
	6	Wordclock Sync		3)
	7	DatMarker		4)
	8	SyncBus	Clock Source	5)
	9	‘’	Clock Mode	
	10	Flip/Flop SyncBus Clock Present Reset		6)
	11	Flip/Flop Word Clock Present Reset		

1) Serial Coder

The Serial Coder Bits serve to generate a serial data stream for the ADDA Chip and the digital Transmitter/Receiver Chip. The software will code register numbers and register values for the chips in the serial data stream. Through accessing the chip registers you can affect the parameters of the analog record and playback respectively the digital record and playback.

For a register description of the ADDA Chip see the file „4524.pdf“. These registers are mainly used to determine the level and the volume. For sample code see the file „volume.pas“.

For a register description of the digital Transmitter/Receiver see the file „cs8427.pdf“. These registers are used mainly, to set the clock master/slave for the digital output. For sample code see the file „control.pas“ under „SetCodecReg“.

To write a register value, you have to generate the Chip Select for the appropriate chip in the first place. This happens via the Bits „Chip Select Digital“ or „Chip Select Analog“.

You can achieve the registers via serializing of a Byte-Sequence. Each bit of this Byte-Sequence will be output from the higher to the lower at the Data-Bit. The output of these Bits is synchronously to the tact, which is generated by the software at the Clock-Bit. The appropriate Data-Bit will be taken over by L-H of the Clock-Bit.

The control sequence for the Digital Transmitter/Receiver has three Bytes:

Hi-Byte: 0x20
Middle-Byte: register number
Low-Byte: register value

The control sequence of the Analog Codec consists of two Bytes:

Hi-Byte	0..3	register number
	4..7	1010b
Low-Byte		register value

To serialize follow these steps

1. Initialise a variable with the current status of the Extended Control Register
2. Set the appropriate Chip-Select-Bit in the variable
3. Set the Clock-Bit in the variable
4. Set or reset the Data-Bit corresponding to the Bit-Status of the current control sequence in the variable
5. Output the variable
6. Do a Dummy-Read-Operation from the Extended Control Register. This leads to accurate clock timing.
7. Reset the Clock-Bit in the variable
8. Output the variable
9. Do the Dummy-Read-Operation from the Extended Control Register once again. This leads to accurate clock timing.

Do the Steps ex 2 for the remaining 23 bits or respectively 15 bits. Go on as follows:

1. Reset the Data-Bit and the Chip-Select-Bit in the variable
2. Output the variable
3. Do the Dummy-Read-Operation from the Extended Control Register. This leads to accurate clock timing.

At the Extended Control Register you have to write or read via DWORD-access only.

While serializing no other access to the register is allowed. Competing program code, who is part of an Interrupt Service Routine or other Threads and who writes or reads this register, have to be bared against each other!

For sample code see the file "control.pas" under „SetCodecReg".

2) Clock Generator Range

Depending on the required sample rate for the clock generator, the generator has to be in a suitable work range.

Bit 1	0	
1	1	6..14 kHz
1	0	12..29 kHz
0	1	24..58 kHz
0	0	47..115 kHz

For sample code see the file „control.pas" under „SetDcoRate".

3) Wordclock Sync

MARC 2 PRO only: Setting this bit causes the internal clock generator to be synchronized to the external workclock signal (extender input). That means that all devices, which are connected to the internal clock source, running with the samplerate given by the external wordclock.

4) Dat-Marker

Setting this bit enables the Dat-Marker support for the Digital Input Device. The 8 lowest Bits of a recorded Sample will be substituted by user bits of the S/PDIF channel status information. These Bits provide several user programs the possibility to get Start-Marker of a DAT-Recorder. Information, which provide the possibility to capture CD Indices, will NOT be generated.

5) SyncBus Clock/Mode

The SyncBus ensures that several *MARIAN SyncBus* compatible audio cards run synchronously. These Bits allow the configuration of the cards clock behaviour towards the SyncBus.

Set the Mode-Bit if you want the card to give a clock to the SyncBus. It can be used as clock source by the devices of other cards. Please see „Clock Select“ in the Control Register.

The Clock-Bit determines, which clock will be output to the SyncBus:

0 = Clock of the Internal Clock Generator
1 = Clock of the Digital Input

The software has to ensure, that while operating several cards parallel, only one of them outputs a Clock to the SyncBus. Before changing the clock mode or the clock source for the SyncBus, you should detach connected devices from the SyncBus and book them up again after the changing.

6) Reset Clock Detection (Marc 2 PRO only)

Every time a interrupt is handled by the driver software, the handler should read the state of the clock inputs and should set and reset the bits 10..11 afterwards. This ensures to get „fresh“ states of the clocks everytimes you read it.

RAM Control (Base + 0x1C)

Through this register you can adjust the audio buffer size of the channels, which are used for recording and playback. This is of main interest for the interaction with user programs, which have the possibility to access directly on the cards RAM (ASIO). You can realize a variable buffer size on the applications level for the audio streaming by reduction of the audio buffer size. This leads to lower latency or more stable system behaviour.

Please note, that the adjustment of the buffer sizes refers to all card devices always. The software has to mask the read Counter appropriate to the determined buffer size.

Bit	0..7	0x00	1/1 Buffer	4096 Samples
		0x01	1/2 Buffer	2048 Samples
		0x03	1/4 Buffer	1024 Samples
		0x07	1/8 Buffer	512 Samples
		0x0F	1/16 Buffer	256 Samples
		0x1F	1/32 Buffer	128 Samples
		0x3F	1/64 Buffer	64 Samples
		0x7F	1/128 Buffer	32 Samples
		0xFF	1/256 Buffer	16 Samples

FPGA RD Register (Memory Range 2, Read Only)

Adresse (Base + n)	**Register**
0	Analog Play Counter
4	Analog Record Counter
8	Digital Play Counter
12	Digital Record Counter
16	Status

Counter (Base + 0x00…0x0C)

Through reading these counters you can get the current playback respectively record positions in relation to the base memory address of the respective playback respectively record device.

Bit	8..0	Counter
	31..9	undefinied

The counter represent the highest 9 bit of the respective 12 bit wide, internal DWORD Sample Counter. One step of the internal counter represents the progress of one Mono Sample. Therefore the counters are exact to 8 Mono samples.

Status (Base + 0x10)

Marc 2:

Bit	0..5	-	
	6	Interrupt Status	1)
	7..14	-	
	15	Interrupt Status Digital Codec	2)

Marc 2 PRO:

Bit	0	-	
	1	Flip/Flop SyncBus Clock present	3)
	2	Flip/Flop Word Clock present	
	3	Extender present (L-Active)	4)

1) The Interrupt Status

gives the information, whether the card has generated an Interrupt. If the card shares an IRQ with another system component, you will have to detect via this bit, whether the card has caused an Interrupt and as a result the Interrupt can be handled. Alternative you can interrogate the Interrupt Status via PCI Register 0x4C (Interrupt source A). For further information see the manufacturers documentation „plx.pdf“.

Interrupts will be generated asynchronous to any record or playback progress and sample rate, approximately every 1.5ms.

2) The Interrupt Status of the Digital Codec

This bit reflects the Interrupt Status of the Digital Codec. This Codec can produce an Interrupt for several incidents. Please see manufacturers documentation „cs8427.pdf“.

3) Clock Detection (Marc 2 PRO only)

Bit 1 of the states register signals the presence of a clock at the SyncBus-Input.

Bit 2 of the states register signals the presence of a clock at the Wordclock Input of the extender.

The hardware detects frequencies beginning from 2.5 kHz as a valid clock. When ever the driver software handles a interrupt it should read the state of these clocks and should reset the appropriate flip/flops to ensure actual clock states.

4) Extender Detection (Marc 2 PRO only)

Bit 3 of the state register signals the presence of the extender board. This bit is L-active. It must only be evaluated if Bit 2 of the states register is 0. Else the state of Bit 2 must be used because a valid clock at the word clock input is show the presence of the extender too.

FPGA Load (Memory Range 3, after Power Up or Local Reset)

To give the MARC 2 / PRO functionality besides PnP, you have to load a control design to the FPGA. The control design is stored in so-called BIT-Data files. The loading of a control design happens through a bit-serial data stream at D0 of the FPGA Load address. You will find the control design for the “Marc 2” in the file „marc2.bit“ and for the “Marc 2 PRO” in the file “marc2p.bit”.

For sample code to FPGA-Loading see the file “init.pas”

HW WR-Register (Memory Range 3, Write Only)

“Marc 2”: Digital Codec Control (Base + 0x00)

Through this register you determine among other things the input of the Digital Input Device.

Bit		
	0	has to be 0 always
	1	has to be 0 always
	2	Digital Input Source 0 1)
	3	Digital Input Source 1
	4	has to be 1 always

1) Digital Input Source

Bit	1	0	
	0	0	Optical Input
	0	1	Electrical Input
	1	0	OnBoard CD Input

For sample code see the file „control.pas“ under „Routing“.

“Marc 2 PRO”: Digital Codec Control (Base + 0x00)

Through this register you determine among other things the input of the Digital Input Device. Each bit represents one possible digital input. Therefore the bit must be set alternatively.

Bit		
	0	has to be 0 always
	1	S/PDIF RCA Input (Extender)
	2	AES/EBU XLR Input (Extender)
	3	Digital CD Input (OnBoard)
	4	Optical Input (OnBoard)

For sample code see the file „control.pas“ under „Routing“.

HW RD-Register (Memory Range 3, Read Only)

Digital Codec Status (Base + 0x00)

This register gives you information about the signal at the Digital Input Device.

Bit		
	0..3	-
	4	Error (No Lock)
	5	Emphasis

Hardware Initialisation

Subsequently you will find the steps to initialise the hardware successfully. Afterwards the card will be ready to record and playback. For sample code see the file „init.pas“.

1. Initialise the PCI Controller (see „InitPciController“)
2. Release the Reset of the hardware, initialise the FPGA (see „WriteConfigData“)
3. Verify the success of the initialisation (see „FpgaInitialised“)
4. Set the Internal Clock Generator to a sample rate (i.e. 44100Hz)
5. Connect the analog and digital devices with the Clock of the Clock Generator
6. Initialise the Analog Codec. (see „InitADDA“). You have to determine an active clock source for the analog devices before.
   You will find a register description for the Analog Codec in the manufacturers documentation „4524.pdf.
7. Initialise the Digital Codec (see „InitDigital“). You have to determine an active clock source for the digital devices before.
   You will find a register description for the Digital Codec in the manufacturers documentation „cs8427.pdf“.
8. Set the Output Routing for the analog und digital devices. The software should provide the possibility to determine the Output Routing. After restarting the system the software should restore these settings.
9. Define the volume settings for the Analog devices (see „volume.pas“). The software should provide the possibility to define the volume settings. After restarting the system the software should restore these settings.
10. Define the SyncBus-Clock and SyncBus-Start/Stop settings. The software should provide the possibility to define these settings. After restarting the system the software should restore these settings.

Contact the Developers

Should you have any problems or questions please do not hesitate to contact:

marian

Elektronik & Software Entwicklung

Eisenacher Straße 72
04155 Leipzig

Email: developers@marian.de