

<b>Einleitung .....</b>	<b>2</b>
<b>Speicherbereiche.....</b>	<b>2</b>
<b>PLX Register (Speicherbereich 0, Read + Write) .....</b>	<b>2</b>
<b>RAM (Speicherbereich 1, Read + Write).....</b>	<b>2</b>
<b>FPGA WR Register (Speicherbereich 2, Write Only) .....</b>	<b>3</b>
<b>Control (Base + 0x00).....</b>	<b>3</b>
1) Play/Capture Enable .....	3
2) Counter Reset.....	4
3) Run.....	4
4) Clock Select .....	4
5) Analog Output Routing.....	4
6) Digital Output Routing .....	5
7) ADDA Freq High .....	5
8) Clear Interrupt .....	5
<b>SyncBus (Base + 0x04) .....</b>	<b>5</b>
<b>Internal Clock Generator (Base + 0x08).....</b>	<b>6</b>
<b>Extended Control (Base + 0x18).....</b>	<b>6</b>
1) Serial Coder .....	6
2) Clock Generator Range .....	7
3) Wordclock Sync .....	7
4) Dat-Marker .....	7
5) SyncBus Clock/Mode .....	8
6) Rücksetzen der Clockerkennung (nur Marc 2 PRO) .....	8
<b>RAM Control (Base + 0x1C).....</b>	<b>8</b>
<b>FPGA RD Register (Speicherbereich 2, Read Only) .....</b>	<b>8</b>
<b>Counter (Base + 0x00...0x0C).....</b>	<b>9</b>
<b>Status (Base + 0x10).....</b>	<b>9</b>
1) Der Interrupt Status.....	9
2) Der Interrupt Status des Digital Codec.....	9
3) Clockerkennung (nur Marc 2 PRO).....	9
4) Erkennung des Extenders (nur Marc 2 PRO) .....	9
<b>FPGA Load (Speicherbereich 3, nach Power Up oder lokalem Reset) .....</b>	<b>10</b>
<b>HW WR-Register (Speicherbereich 3, Write Only) .....</b>	<b>10</b>
<b>„Marc 2“: Digital Codec Control (Base + 0x00).....</b>	<b>10</b>
<b>„Marc 2 PRO“: Digital Codec Control (Base + 0x00).....</b>	<b>10</b>
<b>HW RD-Register (Speicherbereich 3, Read Only) .....</b>	<b>10</b>
<b>Digital Codec Status (Base + 0x00).....</b>	<b>10</b>
<b>Hardware Initialisierung .....</b>	<b>11</b>
<b>Kontakt zu den Entwicklern .....</b>	<b>12</b>

## Einleitung

Diese Dokumentation enthält alle technischen Informationen die benötigt werden, um Treibersoftware für die digitale Audioverarbeitung im Rahmen der technischen Eigenschaften des Produkts „MARC 2“ und „MARC 2 PRO“ zu erstellen. Die „MARC 2 PRO“ stellt eine Erweiterung der „MARC 2“ dar. Technische Eigenschaften, die nur durch diese „PRO“ Version gegeben sind, sind entsprechend gekennzeichnet.

## Speicherbereiche

Die MARC 2 / PRO stellt 4 Speicherbereiche zur Verfügung, die für die Audiofunktionalität der Karte relevant sind.

**Der Zugriff auf diese Speicherbereiche hat ausschließlich DWORD orientiert zu erfolgen.**

Nr.	Länge (Bytes)	Bereich	Verwendung	Kommentar
0	128	PLX-Register	<a href="#">Konfiguration der PCI-Bridge</a>	
1	131072	RAM	<a href="#">Play- und Capture Daten</a>	
2	1024	FPGA WR-Register	<a href="#">Steuerungsregister</a>	alle Write Only
		FPGA RD-Register	<a href="#">Statusregister</a>	alle Read Only
3	512	FPGA Load	<a href="#">Laden des FPGA</a>	Nur verfügbar nach: Power Up oder lokalem Reset
		HW WR-Register	<a href="#">Digital Input Source Select</a>	Verfügbar nach <a href="#">FPGA laden</a> Write Only
		HW RD-Register	<a href="#">Digital Input Status</a>	Verfügbar nach <a href="#">FPGA laden</a> Read Only

### PLX Register (Speicherbereich 0, Read + Write)

Über diesen Speicherbereich haben Sie Zugriff auf die Register des PLX PCI-Bridge-Controllers. Die Speicheradresse eines Registers errechnet sich aus der Basisadresse des Speicherbereiches + Registernummer. Von Interesse sind vor allem die Register 0x4C und 0x50 zur Interrupt- und Resetsteuerung. Näheres zu der Funktionsweise dieser Register erfahren Sie in der Dokumentation des Herstellers '[plx.pdf](#)'.

### RAM (Speicherbereich 1, Read + Write)

Der RAM belegt einen Gesamtspeicherbereich von 0x20000 (128k) Byte und dient dem Austausch der Audiodaten zwischen Karte und Treiber.

Adresse	Bereich
0x00000000	Analog Play Buffer Linker Kanal
0x00004000	Analog Play Buffer Rechter Kanal
0x00008000	Analog Capture Buffer Linker Kanal
0x0000C000	Analog Capture Buffer Rechter Kanal
0x00010000	Digital Play Buffer Linker Kanal
0x00014000	Digital Play Buffer Rechter Kanal
0x00018000	Digital Capture Buffer Linker Kanal
0x0001C000	Digital Capture Buffer Rechter Kanal

Der Hardwarebuffer für die einzelnen Kanäle ist 16 KByte lang und 32 Bit Mono-Sample orientiert. Für jedes Mono-Sample sind nur die Bits 0..23 benutzbar. Samples sind immer MSB-bündig (höchstwertiges Bit = D23) zu schreiben und zu lesen.

Beispielcode zum Schreiben und Lesen von Samples finden Sie in der Datei „[transfer.pas](#)“.

## **FPGA WR Register (Speicherbereich 2, Write Only)**

Alle Signale sind grundsätzlich H-aktiv. Nach dem Initialisieren des FPGA sind sämtliche Register mit 0 initialisiert.

Adresse (Base + n)	Register
0x00	<a href="#">Control</a>
0x04	<a href="#">SyncBus</a>
0x08	<a href="#">Internal Clock Generator</a>
0x0C	-
0x10	-
0x14	-
0x18	<a href="#">Extended Control</a>
0x1C	<a href="#">RAM Control</a>

### **Control (Base + 0x00)**

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

#### 1) Play/Capture Enable

Das Setzen des jeweiligen Bits bewirkt das Aktivieren des entsprechenden Devices. Nach Einführung des *SyncBus*-Features bedeutet dies NICHT mehr den Start von Aufnahme oder Wiedergabe des entsprechenden Gerätes. Vielmehr geht das Gerät in „Lauerstellung“ und startet erst, wenn im [SyncBus-Master-Mode](#) das [Run-Bit](#) gesetzt wird oder im [SyncBus-Slave-Mode](#) über den SyncBus ein Start signalisiert wird.

Das Rücksetzen der jeweiligen Bits bewirkt das Deaktivieren der entsprechenden Devices. Die Aufnahme oder Wiedergabe hält an, auch wenn über „Run“ oder SyncBus kein Stop signalisiert wurde.

„Enable“ und „Run“/„Start von SyncBus“ sind also immer UND-verknüpft.

Beispielcode finden Sie in der Datei „[control.pas](#)“ unter „\_EnablePlay“, „\_EnableRec“.

## 2) Counter Reset

Um die **Counter** der jeweiligen Devices zurückzusetzen, wird das entsprechende Bit gesetzt.

Damit erhält der Counter den Wert 0. Der Counter bleibt solange gesperrt, bis das entsprechende Bit zurückgesetzt wird.

Beispielcode finden Sie in der Datei „[control.pas](#)“ unter „\_Reset“.

## 3) Run

Diese Bit startet im Zusammenhang mit dem **Enable-Bit** die Aufnahme oder Wiedergabe eines Gerätes. Im **SyncBus-Master-Mode** wird zusätzlich das Startsignal auf dem SyncBus erzeugt, so dass anhängige Slave-Geräte mit aktiviertem Enable-Bit auch starten. Im **SyncBus-Slave-Mode** hat das Setzen dieses Bits keine Auswirkungen.

Beispielcode finden Sie in der Datei „[control.pas](#)“ unter „\_Start“.

## 4) Clock Select

Diese Einstellungen bestimmen die Herkunft des Master-Clocks für das jeweilige Device.

<u>Bit</u>	<u>1 0</u>	
	0 0	No Clock
	0 1	Internal Clock Generator
	1 0	SyncBus Clock
	1 1	Digital Input Clock

Ein Device muss zunächst mit einer Clock-Quelle verbunden werden, so dass es mit der Samplerate arbeitet, für die der interne Clockgenerator konfiguriert wurde bzw. mit der eine externe Clock-Quelle arbeitet.

**Bevor Sie die Clock für den Analog Output verändern, muss der ADDA Chip im Reset gehalten und in den Power Down Mode geschickt werden. Nach Umschaltung der Clock, Power Up und Freigabe von Reset ist der ADDA Chip wieder betriebsbereit. Eine Reinitialisierung ist nicht notwendig.**

Sehen Sie hierzu auch den Beispielcode in der Datei „[control.pas](#)“ unter „hws\_SetDevsClockSource“.

**Für Digital Play müssen zusätzlich auch interne Register des Digital Codec behandelt werden.**

Eine Registerbeschreibung ist in der Herstellerdokumentation „[cs8427.pdf](#)“ enthalten. Beispielcode finden Sie in der Datei „[control.pas](#)“ unter „dig\_SetPlayStandartMode“ und „dig\_SetPlaySlaveMode“.

## 5) Analog Output Routing

Diese Einstellungen bestimmen, was am analogen Ausgang zu hören ist.

<u>Bit</u>	<u>1 0</u>	
	0 0	Wiedergabe der Analog Play Audiodaten im RAM
	0 1	Digital Output
	1 0	Analog Input
	1 1	Digital Input

**Wenn Digital Output auf Analog Input geroutet ist, darf Analog Output nicht auf Digital Play oder Digital Input geroutet werden.**

**Ist Analog Record aktiv, so darf der Analog Output nur auf Analog Input oder Analog Wiedergabe geroutet werden.**

**Bevor Sie das Routing für den Analog Output verändern, muss der ADDA Chip im Reset gehalten und in den Power Down Mode geschickt werden. Nach Umschaltung des Routing, Power Up und Freigabe von Reset ist der ADDA Chip wieder betriebsbereit. Eine Reinitialisierung ist nicht notwendig.**

Die Beschreibung der Register des ADDA Chips finden Sie in der Herstellerdokumentation „[4524.pdf](#)“

Beispielcode finden Sie in der Datei „[control.pas](#)“ unter „anlg\_outSetRouting“.

### 6) Digital Output Routing

Diese Einstellungen bestimmen, was am digitalen Stereo-Ausgang zu hören ist.

Bit	1 0	
	0 0	Wiedergabe der Digital Play Audiodaten im RAM
	0 1	Analog Output
	1 0	Analog Input
	1 1	Digital Input

**Wenn Analog Output auf Digital Input geroutet ist, darf Digital Output nicht auf Analog Play oder Analog Input geroutet werden.**

**Ist Digital Record aktiv, so darf der Digital Output nur auf Digital Input oder Digital Wiedergabe geroutet werden.**

Beispielcode finden Sie in der Datei `„control.pas“` unter `„dig_outSetRouting“`.

### 7) ADDA Freq High

Über dieses Bit wird eine Filterfrequenzanpassung im ADDA Chip geregelt. Dieses Bit muss gesetzt werden, wenn analog mit einer Samplerate  $\geq 48\text{kHz}$  aufgenommen bzw. wiedergegeben werden soll. Liegt die Samplerate unter  $48\text{kHz}$  so muss dieses Bit 0 sein.

Beispielcode ist in der Datei `„control.pas“` unter `„anlg_OnRateChanged“` enthalten.

### 8) Clear Interrupt

Nachdem ein Interrupt ausgelöst worden ist, muss dieser auf der Karte zurückgesetzt werden. Dies geschieht durch das Setzen und anschließendes Rücksetzen des Bits. Beispielcode finden Sie in der Datei `„isr.asm“`

### SyncBus (Base + 0x04)

Der SyncBus ermöglicht den synchronen Betrieb von mehreren *MARIAN SyncBus* kompatiblen Audiokarten. Über dieses Register wird das Start/Stop Verhalten der Geräte gegenüber dem SyncBus konfiguriert. Zur Konfiguration des Kartenverhaltens gegenüber dem SyncBus lesen Sie bitte mehr unter [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

Ist ein Gerät als Slave konfiguriert, so startet es nach gesetztem [Enable-Bit](#) erst, wenn auf dem SyncBus ein Startsignal vorliegt.

Ist ein Gerät als Master konfiguriert, so bewirkt ein Setzen des entsprechenden [Run-Bits](#) das Erzeugen eines Startsignals auf dem SyncBus.

Ein Gerät muss als Master und Slave konfiguriert sein, wenn einerseits ein SyncBus Startsignal erzeugt werden soll und das Gerät gleichzeitig gestartet werden soll.

Ist ein Gerät weder als Master noch als Slave konfiguriert, so verhält es sich autonom. Zum Start und Stop werden also nur die Bits [Enable](#) und [Run](#) herangezogen.

**Internal Clock Generator (Base + 0x08)**

Diese Einstellungen bestimmen die Frequenz des Internal Clock Generators, die über die entsprechenden [Clock Select Einstellungen](#) verschiedenen Devices verfügbar gemacht werden. Der Generator erzeugt den Master Clock für die angegebene Samplefrequenz.

Für die verbundenen Geräte sind ggf. noch zusätzliche Einstellungen vorzunehmen. Diese entnehmen Sie bitte dem Beispielcode in der Datei „[control.pas](#)“ unter „\_OnRateChanged“.

Bit 0..15 Frequenz

Den Wert zur Erzeugung einer bestimmten Frequenz berechnet sich wie folgt:

$$\text{Value} = \text{SampleRate} * 0.0524288$$

Zusätzlich muss der Generator in Abhängigkeit der erforderlichen Samplerate in den passenden Arbeitsbereich gefahren werden. (siehe [Clock Generator Range](#)).

Beispielcode finden Sie in der Datei „[control.pas](#)“ unter „SetDcoRate“.

**Extended Control (Base + 0x18)**

Bit	0	Serial Coder	Data	1)
	1	"	Clock	
	2	"	Chip Select Digital	
	3	"	Chip Select Analog	
	4	<a href="#">Clock Generator Range</a>	0	2)
	5	"	1	
	6	<a href="#">Wordclock Sync</a>		3)
	7	<a href="#">DatMarker</a>		4)
	8	SyncBus	<a href="#">ClockSource</a>	5)
	9	"	<a href="#">ClockMode</a>	
	10	<a href="#">Flip/Flop SyncBus Clock Present Reset</a>		6)
	11	<a href="#">Flip/Flop Word Clock Present Reset</a>		

1) Serial Coder

Die Serial Coder Bits dienen dem Erzeugen eines seriellen Datenstroms für den ADDA Chip und den Digital Transmitter/Receiver Chip. Im seriellen Datenstrom werden durch die Software Registernummern und Registerwerte für die Chips codiert. Durch den Zugriff auf die Register der Chips können Parameter für die Analog Aufnahme- und Wiedergabe sowie die Digital Aufnahme und Wiedergabe beeinflusst werden.

Eine Registerbeschreibung für den ADDA Chip ist in der Datei „[4524.pdf](#)“ enthalten. Diese Register werden vorwiegend zur Pegel- und Lautstärkeinstellung benötigt. Beispielcode finden Sie in der Datei „[volume.pas](#)“.

Eine Registerbeschreibung für den Digital Transmitter/Receiver ist in der Datei „[cs8427.pdf](#)“ enthalten. Diese Register werden vorwiegend zur Clock-Master/Slave Einstellung für den Digital Output benötigt. Beispielcode finden Sie in der Datei „[control.pas](#)“ unter „SetCodecReg“.

Um ein Registerwert zu schreiben, muss zunächst das Chip Select für den entsprechenden Chip erzeugt werden. Dies geschieht über die Bits „Chip Select Digital“ oder „Chip Select Analog“.

Die Register erreicht man durch das Serialisieren einer Byte-Sequenz, indem jedes einzelne Bit dieser Byte-Sequenz vom Höherwertigen zum Niederwertigen am Data-Bit ausgegeben wird. Die Ausgabe dieser Bits erfolgt synchron zu dem Takt, der am Clock-Bit durch die Software zu erzeugen ist. Das entsprechende Data-Bit wird mit einer L-H Flanke des Clock-Bits übernommen.

Die Steuersequenz für den Digital Transmitter/Receiver besteht aus drei Bytes:

Hi-Byte: 0x20  
 Middle-Byte: Registernummer  
 Low-Byte: Registerwert

Die Steuersequenz für den Analog Codec besteht aus zwei Bytes:

Hi-Byte	0..3	Registernummer
	4..7	1010b
Low-Byte		Registerwert

#### Schritte zum Serialisieren

1. Variable mit aktuellem Status des Extended Control Registers initialisieren
2. Gewünschtes Chip Select Bit in Variable setzen
3. Clock-Bit in Variable setzen
4. Data-Bit entsprechend des Bit-Status des aktuellen Steuersequenz-Bits in Variable setzen oder rücksetzen
5. Variable ausgeben
6. Dummy-Read auf Extended Control Register zwecks Einhalten des Clocktimings ausführen
7. Clock-Bit in Variable rücksetzen
8. Variable ausgeben
9. Dummy-Read auf Extended Control Register zwecks Einhalten des Clocktimings ausführen

Schritte ab 2. für weitere 23 Bit bzw. 15 Bit ausführen. Dann wie folgt weiter:

1. Data-Bit und Chip Select-Bit in Variable rücksetzen
2. Variable ausgeben
3. Dummy-Read auf Extended Control Register zwecks Einhalten des Clocktimings ausführen

Auch auf das Extended Control Register darf ausschließlich mittels DWORD-Zugriff geschrieben oder gelesen werden.

**Während des Serialisierens darf kein anderer Zugriff auf das Register stattfinden. Programmcode, der konkurrierend als Bestandteil einer Interrupt Service Routine oder eines anderen Threads ggf. auch andere Bits in diesem Register schreibt oder liest, muss gegeneinander verriegelt werden!**

Beispielcode finden Sie in der Datei ["control.pas"](#) unter „SetCodecReg“.

#### 2) Clock Generator Range

In Abhängigkeit der erforderlichen Samplerate für den Clockgenerator, muss dieser zunächst in den passenden Arbeitsbereich gefahren werden.

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

Beispielcode finden Sie in der Datei ["control.pas"](#) unter „SetDcoRate“.

#### 3) Wordclock Sync

**Nur MARC 2 PRO:** Durch Setzen dieses Bits wird der interne Clockgenerator durch den am Wordclock-Eingang (Extender) anliegenden Clock synchronisiert. Damit laufen alle mit dem internen Clockgenerator verbundenen Geräte synchron zur Samplerate der Wordclock.

#### 4) Dat-Marker

Das Setzen dieses Bits aktiviert die Dat-Marker Unterstützung für das Digital Input Device. Dabei werden die niederwertigsten 8 Bit eines aufgenommenen Samples durch User-Bits der Channelstatus Informationen des S/PDIF Datenstroms ersetzt. Diese Bits ermöglichen verschiedenen Anwenderprogrammen das Feststellen von Start-Markern eines DAT-Recorders. Informationen, die das Feststellen von CD Indizes ermöglichen, werden NICHT ausgegeben.

5) SyncBus Clock/Mode

Der SyncBus ermöglicht den synchronen Betrieb von mehreren *MARIAN SyncBus* kompatiblen Audiokarten. Diese Bits ermöglichen die Konfiguration des Clockverhaltens der Karte gegenüber dem SyncBus.

Setzen Sie das Mode-Bit wenn, die Karte eine Clock auf den SyncBus geben soll. Dieser kann dann durch Geräte anderer Karten als Clock-Quelle herangezogen. Sehen Sie dazu „[Clock Select](#)“ im Control Register.

Das Clock-Bit legt fest, welche Clock auf den SyncBus gegeben wird:

- 0 = Clock des Internen Clock Generator
- 1 = Clock des Digital Input

Durch die Software muss sichergestellt werden, dass im Multikartenbetrieb immer nur eine Karte eine Clock auf den SyncBus gibt. Bevor Sie den Clock-Mode oder die Clock-Quelle für den SyncBus ändern, sollten Sie [anhängige Geräte](#) vom SyncBus trennen und danach wieder zuschalten, so dass gerätespezifische Aktionen zur Ausführung kommen.

6) Rücksetzen der Clockerkennung (nur Marc 2 PRO)

Zu jeder Interruptzeit sollte der Clockstatus aus dem [Statusregister](#) gelesen werden und danach die Bits 10..11 gesetzt und sofort wieder rückgesetzt werden. Damit wird ermöglicht, daß die Ausgänge der Flip/Flops für die Clockerkennung bei Abfrage immer aktuelle Werte enthalten.

**RAM Control (Base + 0x1C)**

Durch dieses Register kann die Audiobuffergröße der Kanäle, die für die Aufnahme oder Wiedergabe genutzt wird, angepasst werden. Dies ist insbesondere von Bedeutung im Zusammenspiel mit Anwenderprogrammen, denen ein direkter Zugriff auf den Karten-RAM ermöglicht werden kann (ASIO). Durch eine Verkleinerung der Audiobuffergröße kann auch auf Applikationsebene eine variable Buffergröße für das Audio-Streaming realisiert werden, wodurch kürzere Latenzzeiten oder ein stabileres Systemverhalten erreicht werden.

Beachten Sie, dass sich die Anpassung der Buffergrößen immer auf alle Geräte einer Karte bezieht. Die gelesenen [Counter](#) müssen durch die Software entsprechend der eingestellten Buffergröße maskiert werden.

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 (Speicherbereich 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)**

Durch Lesen dieser Counter lassen sich die aktuellen Abspiel- bzw. Aufnahmepositionen relativ zur Basisspeicheradresse des jeweiligen Abspiel- bzw. Aufnahmeegerätes ermitteln.

Bit	8..0	Counter
	31..9	undefiniert

Die Counter repräsentieren die oberen 9 Bit des jeweiligen 12 Bit breiten, internen DWORD Sample Counters. Ein Schritt des internen Counters repräsentiert den Fortschritt von einem Mono Sample. Die Counter sind daher auf 8 Mono Samples genau.

**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-Aktiv)	4)

**1) Der Interrupt Status**

gibt darüber Aufschluss, ob die Karte einen Interrupt generiert hat. Teilt sich die Karte einen IRQ mit einer anderen Systemkomponente, so muss über diese Bit festgestellt werden, ob die Karte den Interrupt ausgelöst hat und demzufolge der Interrupt bearbeitet werden darf. Alternativ hierzu können Sie den Interrupt Status auch über das PCI Register 0x4C abfragen (Interruptquelle A). Sehen Sie hierzu die Herstellerdokumentation „[plx.pdf](#)“.

Interrupts werden asynchron zu jeglichen, Samplerate abhängigen Aufnahme- oder Wiedergabefortschritten ca. alle 1.5ms erzeugt.

**2) Der Interrupt Status des Digital Codec**

Dieses Bit spiegelt den Interrupt Status des Digital Codec wider. Dieser Codec kann für verschiedene Ereignisse einen Interrupt auslösen. Konsultieren Sie hierzu die Herstellerdokumentation „[cs8427.pdf](#)“.

**3) Clockerkennung (nur Marc 2 PRO)**

Über Bit 1 des Statusregisters kann abgefragt werden, ob am SyncBus-Clockeingang ein Clock anliegt.

Über Bit 2 des Statusregisters kann abgefragt werden, ob am Wordclockeingang des Extenders ein Clock anliegt.

Dabei werden durch die Hardware Frequenzen ab 2,5 kHz ausgewertet und das Bit gesetzt. Zu jeder Interruptzeit sollte der Clockstatus gelesen werden und anschließend das [entsprechende Flip/Flop](#) zurückgesetzt werden, damit die kontinuierliche Erkennung der Clocks gewährleistet ist.

**4) Erkennung des Extenders (nur Marc 2 PRO)**

Über Bit 3 des Statusregisters kann in Erfahrung gebracht werden, ob der Extender angeschlossen ist. Dieses Bit ist L-Aktiv und darf nur ausgewertet werden, wenn Bit 2 des Statusregisters nicht gesetzt ist. Ist Bit 2 des Statusregisters gesetzt, so ist dies ohnehin ein Merkmal dafür, daß der Extender angeschlossen ist, da sich der Wordclockeingang auf dem Extender befindet.

## **FPGA Load (Speicherbereich 3, nach Power Up oder lokalem Reset)**

Damit die MARC 2 / PRO außer PnP überhaupt eine Funktion ausübt, muss in den FPGA ein gültiges Schaltungsdesign geladen werden. Das Schaltungsdesign wird in den sogenannten BIT-Dateien gespeichert. Das Laden eines Schaltungsdesigns geschieht durch einen bit-seriellen Datenstrom an D0 der FPGA Ladeadresse. Das Schaltungsdesign für die „Marc 2“ ist in der Datei „marc2.bit“, das Design für die „Marc 2 PRO“ in der Datei „marc2p.bit“ enthalten.

Beispielcode zum FPGA-Laden finden Sie in der Datei [“init.pas”](#)

## **HW WR-Register (Speicherbereich 3, Write Only)**

### **„Marc 2“: Digital Codec Control (Base + 0x00)**

Über dieses Register wird u.a. der Eingang für das Digital Input Device festgelegt.

Bit	0	muss immer 0 sein
	1	muss immer 0 sein
	2	Digital Input Source 0 (1)
	3	Digital Input Source 1
	4	muss immer 1 sein

#### 1) Digital Input Source

Bit	1	0	
	0	0	Optischer Eingang
	0	1	Elektrischer Eingang
	1	0	OnBoard CD Eingang

Beispielcode finden Sie in der Datei [„control.pas“](#) unter „Routing“.

### **„Marc 2 PRO“: Digital Codec Control (Base + 0x00)**

Über dieses Register wird u.a. der Eingang für das Digital Input Device festgelegt. Jedes Bit repräsentiert einen möglichen Eingang. Die Bits sind daher alternativ zu setzen.

Bit	0..	muss immer 0 sein
	1	S/PDIF Cinch Eingang (Extender)
	2	AES/EBU XLR Eingang (Extender)
	3	Digital CD Eingang (OnBoard)
	4	Optischer Eingang (OnBoard)

Beispielcode finden Sie in der Datei [„control.pas“](#) unter „Routing“.

## **HW RD-Register (Speicherbereich 3, Read Only)**

### **Digital Codec Status (Base + 0x00)**

Dieses Register gibt Aufschluss über das Signal am Digital Input Device.

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

## **Hardware Initialisierung**

Nachfolgend werden die Schritte aufgeführt, um die Hardware erfolgreich zu initialisieren. Danach ist die Karte bereit für die Aufnahme- und Wiedergabe. Beispielcode finden Sie in der Datei „[init.pas](#)“.

1. PCI Controller initialisieren (siehe „InitPciController“)
  2. Reset der Hardware auslösen, FPGA initialisieren (siehe „WriteConfigData“)
  3. Überprüfen Sie den Erfolg der Initialisierung (siehe „FpgaInitialized“)
  4. Setzen Sie den [Internal Clock-Generator](#) auf eine Samplerate (z.B. 44100Hz)
  5. [Verbinden](#) Sie die Analog- und Digitalgeräte mit der Clock des Clock Generator
  6. Initialisieren Sie den Analog Codec. (siehe „InitADDA“). Hierfür MUSS für die Analoggeräte eine aktive Clock-Quelle eingestellt worden sein. Eine Registerbeschreibung für den Analog Codec ist in der Herstellerdokumentation „[4524.pdf](#)“ enthalten.
  7. Initialisieren Sie den Digital Codec (siehe „InitDigital“) Hierfür MUSS für die Digitalgeräte eine aktive Clock-Quelle eingestellt worden sein. Eine Registerbeschreibung für den Digital Codec ist in der Herstellerdokumentation „[cs8427.pdf](#)“ enthalten.
  8. Setzen Sie das [Output Routing](#) für die Analog- und Digital Geräte. Die Software sollte dem Nutzer die Möglichkeit geben, das Output Routing festzulegen. Nach Neustart des Systems sollte die Software diese Einstellungen wiederherstellen.
  9. Setzen Sie die Lautstärkeinstellungen für die Analogen Geräte. (siehe „[volume.pas](#)“) Die Software sollte dem Nutzer die Möglichkeit geben, das Lautstärkeinstellungen festzulegen. Nach Neustart des Systems sollte die Software diese Einstellungen wiederherstellen.
  10. Setzen Sie die [SyncBus-Clock](#) und [SyncBus-Start/Stop](#) Einstellungen. Die Software sollte dem Nutzer die Möglichkeit geben, diese Einstellungen festzulegen. Nach Neustart des Systems sollte die Software diese Einstellungen wiederherstellen.
-

## **Kontakt zu den Entwicklern**

Fragen zu dieser Dokumentation stellen Sie bitte schriftlich an:



Elektronik & Software Entwicklung

Eisenacher Straße 72

04155 Leipzig

Email: [developers@marian.de](mailto:developers@marian.de)