



HFS - HF-Kavitätensynchronisation

Gerätemodell und Softwareentwurf

U. Krause, G.Riehl

Änderungsprotokoll

Datum	GM-Version	Name	Kommentar
4. Jun. 07	HFS_01	UK	1. Diskussionsgrundlage
8. Aug. 09	HFS_01	GuRi	Aktuelle Softw.Vers. 9.00
18. Apr. 10	HFS_01	GuRi	Erweitertes GM, Bunch-Merging
29. Jul. 14	HFS_02	LH	Property MHBPAR2 added

Inhaltsverzeichnis

I	Das Gerätemodell	5
1	Die Aufgabe des Gerätes	5
2	Die Hardware des Gerätes	5
2.1	Gerätevarianten	6
3	Use case(s)	6
3.1	Use case Bunch Merging	6
3.1.1	Geräteansteuerung	7
4	Die Schnittstelle zum Gerät	8
4.1	Funktionscodes der Interfacekarte	8
4.2	Interlock Interrupt	9
4.3	Data Request (DRQ) Interrupts	9
4.4	Data Ready (DRD) Interrupts	9
4.5	Definition der Bits des Hardwarestatus	9
5	Die Steuerung des Gerätes	10
5.1	Geräte-Parameter	10
5.2	Experiment-Geräte-Parameter	10
5.3	Device- und Register-Liste	11
5.4	Übergabe der Geräte-Parameter	11
5.5	Genauigkeitsanforderungen	12
5.6	Zeitkritische Anforderungen	12
5.7	Einordnung in das Timing	12
5.8	Festlegung von Startwerten	13
5.8.1	Kaltstarts	13
5.8.2	Warmstarts	13
5.9	Handbetrieb	13
5.10	Ableitung des Hardwarefehler-Bits aus dem Gerätestatus	13
5.11	Verhalten bei Störungen	13
5.11.1	Geräteinterlock	13
5.11.2	Event-Sequenzfehler	13
5.11.3	Event-Overrun	13
5.11.4	Emergency-Event	13
5.11.5	Ausfall der Kommunikation EC – Gerät	14
5.11.6	Sonstiges	14
5.12	Bedienungsfehler vom Operating	14
5.13	Sonstige Anforderungen	14
6	Die Repräsentation des Gerätes	14
6.1	Kennzeichnung des Gerätemodells	14
6.2	Die Master-Properties	14
6.2.1	INFOSTAT	14
6.2.2	INIT	15
6.2.3	POWER	15
6.2.4	RESET	16
6.2.5	STATUS	16
6.2.6	VERSION	16

6.3	Die Slave-Properties	16
6.3.1	ACTIV	17
6.3.2	COPYSET	17
6.3.3	EQMERROR	17
6.3.4	SETDATA	18
6.3.5	WORKMODE	19
6.3.6	SYNCMODE	19
6.3.7	MHBPAR	20
6.3.8	MHBPAR2	20
6.3.9	EXPPAR	21
6.3.10	MILDATA	21
6.3.11	DPRDATA	22
II Der Entwurf der Software		23
7	Softwareentwurf	23
8	Lokale Datenbasis	23
8.1	Tabelle der Konstanten	23
9	Dualport RAM	23
9.1	M_data_type	23
9.2	S_data_type	24
10	Zyklus-Ablauf	24
Literatur		26
Index		27

Teil I

Das Gerätemodell

1 Die Aufgabe des Gerätes

Aufgabe des Gerätes ist die Synchronisation der beiden HF-Anlagen im SIS18.

Zur Zeit sind im wesentlichen drei Funktionsgruppen im Gerät bzw. Gerätemodell implementiert: a) Per Kommando und mittels Blob-Format beliebige Werte in beliebige Register des Gerätes zu beschreiben. Diese Funktion wird hauptsächlich in der experimentellen Phase von den Geräte-Entwicklern (HF-Gruppe) benutzt bzw. benötigt. b) HF-Phasen-Synchronisation. Die zwei HF-Kavitäten werden in bestimmter Phase (0-360-Grad zueinander betrieben, Vacc spezifisch und Event-gesteuert c) Mehr-harmonischen-Betrieb, Register-Zugriffe am Zyklus-Anfang (Prep) und zu zwei definierbaren Zeitpunkten im Zyklus, Vacc spezifisch und Event-gesteuert d) Experiment-Mode, Zugriffe auf beliebige Register mit beliebigen Daten (neues Blob-Format), Register-Zugriffe am Zyklus-Anfang (Prep) und zu zwei definierbaren Zeitpunkten im Zyklus, Vacc spezifisch und Event-gesteuert.

2 Die Hardware des Gerätes

Blockschaltbild Kavitäten-Ansteuerung...

In Abbildung 1 ist ein vereinfachtes Blockschaltbild der Kavitätenansteuerung und der Kavitäten-synchronisation der zwei SIS-Kavitäten S02BE1 und S08BE2 dargestellt.

Für die Kavität S02BE1 erzeugt das Gerät S02BE1FS die Frequenzrampe. Das Gerät S02BE1A erzeugt die Soll-Amplituden-Rampe. Der Block Amplitude Control stellt sicher, dass die Amplitude des Gap-Signales der Soll-Amplitude entspricht, indem die HF-Treiberleistung (der Treiber ist als gelber Verstärkerblock dargestellt) nachgeregelt wird. Das Gerät S02BE1P erzeugt die Soll-Phasen-Rampe. Diese wird an das DSP-System 1 geschickt, damit die in diesem System implementierte Phasenregelung (weiter unten erklärt) den Sollwert berücksichtigen kann

Die Signalgenerierung und Amplitudenregelung der Anlage S08BE2 ist gleichartig wie die soeben beschriebene der Anlage S02BE1 aufgebaut.

Man beachte, dass die Kavitäten-Signale ohne eine Regelung nicht synchron zueinander laufen, da sich beide in verschiedenen Überrahmen befinden und von unterschiedlich getakteten Interfackarten angesteuert werden.

Um die Kavitäten miteinander synchronisieren zu können, wurde die Anlage durch ein neues Gerät Frequency Generation und Distribution Amplifier erweitert, dem man in derselben Weise wie bei den Geräten S02BE1FS und S08BE2FS Frequenzrampen schickt. Dieses Gerät enthält zwei sogenannte Gruppen-DDS-Module und zwei Offset-LO-DDS-Module. Das Gerät wird nicht mit der Betriebsfrequenz-Rampe einer Kavität, sondern mit der Umlauffrequenz-Rampe versorgt.

Man kann nun durch Konfiguration des Verteilerverstärkers (Distribution Amplifier) eines der Gruppen-DDS-Signale und das dazu passende LO-Signal als Referenz für das DSP-System einer Kavität auswählen. Das DSP-System stellt dann sicher, dass die Phasendifferenz zwischen Gapsignal und Gruppen-DDS-Signal der Sollphase entspricht, indem die Frequenz der Kavitäten-DDS entsprechend korrigiert wird (Phasenregelung).

Die Kavitäten-DDS-Signale laufen ohne diese Regelung nicht synchron zu den Gruppen-DDS-Signalen, da sich beide in verschiedenen Überrahmen befinden und von unterschiedlich getakteten

Interfacekarten angesteuert werden. Demgegenüber ist es möglich und hier vorgesehen, das Signal der Group DDS RF2 durch exakte Frequenzverdopplung aus der Versorgung der *Group DDS RF1* abzuleiten (oder umgekehrt). Da beide Gruppen-DDS-Module in demselben Überrahmen platziert sind und von derselben Umlauffrequenz-Rampe angesteuert werden, ist sichergestellt, dass sie zu jedem Zeitpunkt dieselbe Phasenlage zueinander haben, obwohl die Frequenz variabel ist.

Das Gerät zur HF-Synchronisation S00BE_PS besteht aus einer Interface-Karte (IFK) zur Anbindung an die VME-Kontrollsystemrechner und dem in einem Modulbusrahmen installierten CCS_Fib. Das CCS_Fib steuert über einen Datenbus (Token-Ring, orange Linie im Blockschaltbild) die zur HF-Synchronisation erforderlichen HW-Komponenten (Sub-Devices) der Kavitätenansteuerung (Frequenz-Generation, Distribution-Amplifier, DSP-Systeme 1,2).

2.1 Gerätevarianten

Es sind keine Varianten vorgesehen, allerdings sind hardware-spezifische Details (z.B. register-Adressen) auf der SE d.h. in den EQMs definiert, sodass die zukünftige Einrichtung von varianten erleichtert ist.

3 Use case(s)

Zur Erläuterung der Funktionsweise des Gerätes werden hier angeforderte Funktionen des Gerätes beschrieben.

3.1 Use case Bunch Merging

Ziel des Bunch-Merging ist es, einen einzigen Bunch auf Flattop zu erzeugen. Dieser kann dann beispielsweise mit dem SIS18-Bunchkompressor komprimiert und schnell extrahiert werden. Die Erzeugung des Einzel-Bunches läuft wie folgt ab (frev=Umlauffrequenz):

- Konfiguration der Gruppen-DDS-Signale:
 - Group DDS RF1: 2frev
 - Offset LO DDS 1: fIF+2frev
 - Group DDS RF2: 4frev
 - Offset LO DDS 2: fIF+4frev
- Die Kavität S02BE1 wird mit dem Signal der Group DDS RF1 versorgt, die S08BE2 mit dem der Group DDS RF2.
- Normaler adiabatischer Einfang mit der S08BE2 bei $h=4$ und 11.4 MeV/u
- Normale Beschleunigung mit der S08BE2 bei $h=4$ auf 210 MeV/u (Umlauffrequenz 800 kHz, HF-Frequenz 3.2 MHz)
- Die Amplitude der S02BE1 ($h=2$) wird linear von 0 auf einen Maximalwert hochgefahren, während gleichzeitig die Amplitude der S08BE2 ($h=4$) von ihrem aktuellen Wert auf 0 heruntergefahren wird. Dabei verschmelzen die 4 Bunche zu 2 Bunchen.
- Nun sind die Gruppen-DDS-Module umzukonfigurieren:
 - Group DDS RF2: 1frev

- Offset LO DDS 2: $f_{IF} + 1f_{rev}$
- Die Amplitude der S08BE2 (h=1) wird linear von 0 auf einen Maximalwert hochgefahren, während gleichzeitig die Amplitude der S02BE1 (h=2) von ihrem aktuellen Wert auf 0 heruntergefahren wird. Dabei verschmelzen die 2 Bunche zu 1 Bunch.
- Am Ende wird der Verteilerverstärker so umgeschaltet, dass auch die S02BE1 mit dem Signal der Group DDS RF2 bei h=1 versorgt wird. Dies ermöglicht es, eine Bunchkompression mit beiden Ferrit-Kavitäten zu unterstützen.

3.1.1 Geräteansteuerung

Aus den oben genannten Schritten, folgt unmittelbar, welche Befehle zu welchen Zeitpunkten an das Gerät S00BE_PSGeschickt werden müssen:

- Beginn des Zyklus:
 - Festlegung der Harmonischenzahlen
 - * Group DDS RF1: h=2 (GROUP_DDS1.H=2)
 - * Group DDS RF2: h=4 (GROUP_DDS2.H=4)
 - Konfiguration Verteilerverstärker:
 - * DSP-System BE2: Group DDS RF2 (BE2.GROUP_DDS=2)
 - * DSP-System BE1: Group DDS RF1 (BE1.GROUP_DDS=1)
 - Konfiguration DSP-System 1
 - * Regelung ein: BE1_DSP_CONTROL=1
 - * Sollphasenrampe berücksichtigen: BE1_DSP_EXT_PHASE=1
 - Konfiguration DSP-System 2
 - * Regelung ein: BE2_DSP_CONTROL=1
 - * Sollphasenrampe berücksichtigen: BE2_DSP_EXT_PHASE=1
- Im Abschnitt 2 Bunches:
 - Festlegung der Harmonischenzahlen
 - * Group DDS RF2: h=1 (GROUP_DDS2.H=1)
 - Konfiguration Verteilerverstärker:
 - * bleibt
- Zu Beginn des Abschnitt 1 Bunch:
 - Konfiguration Verteilerverstärker:
 - * DSP-System BE1: Group DDS RF2 (BE1.GROUP_DDS=2)

Die Datenversorgung entspricht exakt einer Untermenge der in [1], Tabelle 6 definierten physikalischen Größen:

- Harmonischenzahl für jede Kavitäatengruppe (hier sind nur zwei Kavitäten und damit nur 2 Kavitäatengruppen beteiligt): GROUP_DDS1.H, GROUP_DDS2.H
- Referenzsignal für jede Kavität: Hiermit ist das jeweilige Group DDS-Signal gemeint: BE1_GROUP_DDS, BE2_GROUP_DDS

Wie man dem skizzierten Ablauf entnimmt, sind zu drei Zeitpunkten im HF-Zyklus ein bis vier Werte zu modifizieren (für jeden virtuellen Beschleuniger unterschiedlich). Das Datenaufkommen ist also gering (daraus resultieren jedoch wegen der Verteilung über den Optical Token Ring an die Subsysteme deutlich mehr Registerzugriffe). Die geforderte Zeitgenauigkeit liegt im Bereich von 100 usec...1 ms.

4 Die Schnittstelle zum Gerät

Dieses Kapitel beschreibt den Aufbau der digitalen Schnittstelle zum Rechnersystem.

Die Anbindung des Gerätes an das Kontrollsystem ist über ein sog. Ccs.Fib in einem Modulbus-Rahmen realisiert. Das Kontroll-System schreibt Daten über den MIL-Bus in eine IFK und den angeschlossenen Modulbus-Rahmen und das dortige Ccs.Fib. Dieses wiederum gibt die Daten an das/die eigentlichen HW-Komponenten (Sub-Devices) der Kavitätenansteuerung weiter. Auf dem Ccs.Fib wird ein Sub-Device über eine 16-Bit Device-Adresse adressiert. Das Ziel der Daten auf dem Device wird wiederum über eine 16-Bit Parameter-Adresse festgelegt. Auf dem HW-Device entsprechen die Daten bestimmten Einstellungen und Funktionen. Zur Kommunikation mit der VME-Ebene belegt das Ccs.Fib im Modulbus vier 16-Bit Register mit folgenden Bedeutung:

Reg.-Adr.Offset	Bedeutung
00	Kontroll-Register
02	HW-Device-Adresse
04	16-Bit Daten
06	Register-Adresse

Die Bits des Kontrollregisters sind wie folgt definiert:

Bit	Name	Richtung	Bedeutung
0	Send-Data	In	übertrage Daten (Auto-Reset)
1	RnW	In	0: Schreiben, 1: Lesen
2	Reset Parameter	In	Reset der Parameter-Tabelle
3	Block-transfer	In	Optional: Block-Transfer
4	Data Available	Out	Daten verfügbar bei Lesezugriff
5	Error	Out	Übertragungsfehler (Auto-Reset)
6-15	reserved	-	Reserviert

Für die beiden mit Autoreset gekennzeichneten Bits gilt:

Bit 0: Rückname nach Abarbeitung der Übertragung

Bit5: Rückname nach Auslesen des Kontroll-Registers

Die Verwendung dieser Register zur Datenübertragung zum/vom gerät wird in Kapitel - Übergabe der Geräteparameter - beschrieben

4.1 Funktionscodes der Interfacekarte

Die für die Kommunikation mit der Modulbus-Interfacekarte definierten Funktionscodes sind in der folgenden Tabelle aufgelistet. Als Modus ist angegeben, ob Daten von der Interfacekarte gelesen werden, ob Daten zu der Interfacekarte geschrieben werden, oder ob nur eine Funktion ausgeführt wird.

Da die Modulbus-Interfacekarte nur Treiberfunktion für den Modulbus hat, beschränken sich die verwendbaren Funktions-Codes auf die wenigen, die man zum Handling von Adreß- und Daten-Bus benötigt und einige Standard-Codes.

Funktionscode		Modus	Bedeutung
Name	hex		
ifb_reset	01	Funktion	Reset
ifb_data_bus_w	10	Schreiben	16 Bit Daten auf den Datenbus schreiben
ifb_adr_bus_w	11	Schreiben	8 Bit Daten auf den Adreßbus schreiben
ifb_data_bus_r	90	Lesen	16 Bit Daten vom Datenbus lesen
ifb_rdstat_int	C9	Lesen	Status der Interfacekarte lesen

ifb_reset

Das Gerät wird in einen definierten Ausgangszustand gebracht.

4.2 Interlock Interrupt

Es werden keine Geräte-Interlocks verwendet.

4.3 Data Request (DRQ) Interrupts

DRQ-Interrupts werden nicht erwartet bzw. verarbeitet.

4.4 Data Ready (DRD) Interrupts

DRD-Interrupts werden nicht erwartet bzw. verarbeitet.

4.5 Definition der Bits des Hardwarestatus

Es ist kein Gerätestatus definiert.

Statusbits	Funktionscode
8 ... 15	ifb_rdstat
16 ... 23	ifb_rdstat_1
24 ... 31	ifb_rdstat_2

Die Bits 0 ... 7 sind die systemweiten sogenannten generierten Softwarestatusbits (in engl. derived status bits). Die Statusbits im Einzelnen sind in der folgenden Tabelle zusammengefasst.

Bit	Name	Bedeutung	
		High (1)	Low (0)
0	Power	on	off
1	Remote/Local	Remote	Local
2		reserved	
3		reserved	
4	Emergency	no	yes
5	Interlock	no	yes
6	HW Error	no	yes
7	SW Error	no	yes
8..15	undef	–	–
16..23	undef	–	–
24..31	undef	–	–

5 Die Steuerung des Gerätes

Hier wird beschrieben, wie das Gerät (die Hardware) bedient werden muss. Das beinhaltet die Anforderungen *vom* Gerät als auch die Anforderungen *an das* Gerät.

Von der Gerätesoftware sind im Gerät (bestehend aus einem Verbund von Sub-Geräten (hier als Sub-Devices oder einfach Devices bezeichnet) eine Reihe von Werten (Harmonischen-Zahl, Gain, Phase,...) einzustellen. Diese einzustellenden Werte werden als Geräte-Parameter bezeichnet.

5.1 Geräte-Parameter

Für jeden der einzustellenden Parameter ist ein oder sind mehrere interne Register in einem oder mehreren Sub-Devices vorgesehen. Die Register werden über zwei interne Adressen angesprochen, die *Device-Adresse* und *Register-Adresse*.

Die einzelnen Geräte-Parameter können unterschiedliche Größe haben. Register-Zugriffe erfolgen immer 16-Bit weise, größere Parameter werden auf mehrere aufeinanderfolgende 16-Bit Register aufgeteilt.

Und natürlich hat jeder Geräte-Parameter einen Wert, das eigentliche *Datum*.

Somit ist jeder einzelne Geräte-Parameter vollständig beschrieben durch

Device Adresse	16-Bit
Register Adresse	16-Bit (0..4294967295)
Register Datum	16-Bit

Innerhalb der Controls-Software sind die bereits bekannten Register in einer Definitionsliste mit Register-Namen (Nummer), Device-Adresse und Register-Adresse definiert. Dadurch kann ein parameter durch 2 Werte vollständig definiert werden:

Register Nummer	16-Bit
Register Datum	16-Bit

5.2 Experiment-Geräte-Parameter

Im Test- oder Experiment-Mode werden die Register direkt angesprochen, Device- und Register-Adressen sowie zugehörige Daten werden durch Listen, sog. BLOBS beschrieben. Hier haben sich im Laufe der Geräte-Entwicklung 2 BLOB-Formate entwickelt, zur Zeit werden noch beide unterstützt. Die VME-Software überprüft die BLOB-Inhalte NICHT, sodass hier auch in die Hardware neu implementierte Devices und Register angesprochen werden können.

- BLOB-Format 1 (s.PROP SETDATA) beinhaltet eine variable Datenlänge (1 bis 4 16-Bit Worte) pro Register. Dabei wird das 1 Wort der Register-Adresse zugeordnet, das zweite, dritte und vierte Wort wird im Blocktransfer-Mode dem CCS.Fib übergeben, das die Adressierung an die anschliessenden Register übernimmt.
 - count, Anzahl nachfolgender Parameter
 - Par-1, Device-Adresse 16-Bit
 - Par-1, Register-Adresse 16-Bit
 - Par-1, BytCount Datum 1-8
 - Par-1, Datum Low 32-Bit (Wort 1 und 2)
 - Par-1, Datum High 32-Bit (Wort 3 und 4)
 - Par-2,

- BLOB-Format 2 (s.PROP EXPPAR) arbeitet mit einer festen Datums Größe von 16-Bit, größere Daten werden auf 2,3,.. Register aufgeteilt.
 - count, Anzahl nachfolgender Parameter
 - Par-1, Device-Adresse 16-Bit
 - Par-1, Register-Adresse 16-Bit
 - Par-1, Datum 16-Bit
 - Par-2,

5.3 Device- und Register-Liste

Im folgenden werden die (Sub-) Devices und deren Register in einer Tabelle gelistet. Die angegebenen Adressen der Geräte und Register entsprechen den in das CCS-Fib zu ladenden Device-Adressen und Parameter-Adressen. (s. subsection Übergabe der Parameter).

Sub-Geraet Name	Subadr.	Registername	Reg.Adr.	Reg.-Breite (Bytes)	Werte
DSP-System BE1	60	Schatten o. Control	0xfffe	2	0,1
..	..	Phase	0x8002	2	0-3600
..	..	Gain-Low	0x8008	2	0-0xffff
..	..	Gain-High	0x800a	2	0-0xffff
DSP-System BE2	61	Schatten o. Regelung	0xfffe	2	0,1
..	..	Phase	0x8002	2	0-3600
..	..	Gain-Low	0x8008	2	0-0xffff
..	..	Gain-High	0x800a	2	0-0xffff
Group DDS-RF1		harmonische h			1,2,4
Group DDS-RF2		harmonische h			1,2,4
Offset-Lo DDS1		harmonische h			1,2,4
Offset-Lo DDS2		harmonische h			1,2,4
Distribution Amp		Output1			1,2,3,4
..		Output2			1,2,3,4
..		Output3			1,2,3,4
..		Output4			1,2,3,4

5.4 Übergabe der Geräte-Parameter

Die Geräte-Parameter werden der Controls-Software übergeben und von dieser auf die entsprechenden Register-Werte transformiert. Dazu legt die Controls-Software für jeden Vacc eine Liste der bekannten Devices und Register und deren Daten an.

Um nun einen Geräte-Parameter von der VME-Ebene über das Ccs.Fib in ein bestimmtes Device-Register zu laden, ist folgender Ablauf vorgeschrieben:

- Device-Adresse in Modulbus-Register 2 laden
- Register-Adresse in Modulbus-Register 6 schreiben
- 16-Bit Datum in Modulbus-Register 4 schreiben
- Bit 0 im Control-Register setzen zur Übername und Senden des Datums

Möglich ist das Schreiben von Daten größerer Bitbreite in aufeinanderfolgende Adressen mittels des Blockmodes:

- Device-Adresse in Modulbus-Register 2 laden
- Register-Adresse in Modulbus-Register 6 schreiben
- 16-Bit Datum in Modulbus-Register 4 schreiben
- Bit 0 im Control-Register setzen (command=1) zur Übernahme und Senden des Datums
- Nächstes 16-Bit Datum in Modulbus-Register 4 schreiben
- Bit 0 und Bit 4 im Control-Register setzen (command=9), dadurch wird die Registeradresse inkrementiert und das Datum übernommen und gesendet
- Nächstes 16-Bit Datum

Um ein 16-Bit Modulbus Register zu beschreiben sind immer zwei MIL-Schreibzugriffe erforderlich, den ersten zur Anwahl der Modulbus Adresse, den zweiten zum Beschreiben dieser Adresse mit dem 16-Bit Datum, siehe Tabelle oben, - Funktionscodes Interfacekarte - Daraus ergibt sich, dass zur Übertragung eines Datenwortes von der VME-Ebene zum HW-Device immer 8-MIL-Zugriffe erforderlich sind.

Über eine Programmierung der Interfacekarte (neu programmierter Funktionscode = Daten schreiben auf Modulbus-Adresse x) kann die Anzahl der MIL-Zugriffe auf 4 pro Datenwort halbiert werden.

5.5 Genauigkeitsanforderungen

Entfällt

5.6 Zeitkritische Anforderungen

Alle in der Liste definierten Geräte-Parameter werden zu Beginn eines Zyklus in das Gerät geschrieben.

Die Geräte-Parameter sind zyklusspezifisch, das heisst, für jeden virtuellen Beschleuniger sind eigene Parametersätze zu verwalten.

5.7 Einordnung in das Timing

Das Gerät wird zum frühestmöglichen Zeitpunkt mit den jeweiligen Daten versorgt.

Die Datenversorgung (Prep) ist daher an das Event `Evt_Start_Cycle` geknüpft, zu diesem Zeitpunkt werden alle bekannten H-Register mit den Werten entsprechend der vacc-Einstellung versorgt.

Unterstützt werden bis zu 4 zusätzliche Datenversorgungen im Zyklus. Der Zeitpunkt der Datenversorgung(en) wird bestimmt durch ein einstellbares Delay sowie das Event `—Evt_Flattop_Start—`. Dabei addieren sich die Delays der einzelnen Datenversorgungen:

- Zeitpunkt Datenversorgung 1: Delay-1 + Event-Flattop-Start
- Zeitpunkt Datenversorgung 2: Delay-2 + Delay-1 + Event-Flattop-Start
-

5.8 Festlegung von Startwerten

5.8.1 Kaltstarts

Bei einem Kaltstart werden folgende Aktionen durchgeführt:

- Es wird ein Gerätereset durchgeführt.
- Ein Default-Sollwertsatz wird für alle virtuellen Beschleuniger am Gerät eingestellt.
- Das Gerät wird für alle virtuellen Beschleuniger inaktiv geschaltet.
- Die SE wird in den Eventmode-Betrieb geschaltet.
- Die Standard-Eventkonnektierungen müssen gesetzt werden (siehe Tabelle ?? auf Seite ??).

5.8.2 Warmstarts

Bei einem Warmstart werden folgende Aktionen durchgeführt:

- Es wird ein Gerätereset durchgeführt.

5.9 Handbetrieb

Ein Handbetrieb ist nicht vorgesehen.

5.10 Ableitung des Hardwarefehler-Bits aus dem Gerätestatus

Entfällt, da kein Geräte-Status definiert ist.

5.11 Verhalten bei Störungen

5.11.1 Geräteinterlock

Entfällt.

5.11.2 Event-Sequenzfehler

Entfällt.

5.11.3 Event-Overrun

Bei einem Overrun-Fehler ist der Zyklus normal abzuarbeiten.

5.11.4 Emergency-Event

Entfällt.

5.11.5 Ausfall der Kommunikation EC – Gerät

Der Ausfall der Kommunikation zwischen EC und Gerät führt zu Timeouts. Tritt ein Timeout auf so ist der Zyklus abzuberechnen.

5.11.6 Sonstiges

Keine weiteren Störungen vorhanden.

5.12 Bedienungsfehler vom Operating

Es erfolgt keine Überprüfung der Eingabewerte des Operatings.

5.13 Sonstige Anforderungen

Keine.

6 Die Repräsentation des Gerätes

Dieses Kapitel beschreibt, wie das Gerät nach höheren Ebenen hin abgebildet wird.

6.1 Kennzeichnung des Gerätemodells

Das Gerätemodell hat die Bezeichnung **HFS**.

Die Gerätemodellnummer ist 78_{dez} .

6.2 Die Master-Properties

Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
INFOSTAT	RA	0	–	25	BitSet32	1	0
INIT	N	0	–	0	–	–	–
POWER	R/W	0	–	1	BitSet16	1	0
RESET	N	0	–	0	–	–	–
STATUS	R	0	–	1	BitSet32	1	0
VERSION	RA	0	–	36	BitSet8	1	0

6.2.1 INFOSTAT

Bedeutung: Diese Property liefert einige wichtige Geräteinformationen in einem Zugriff. Die Informationen werden direkt aus dem Dualport-RAM gelesen, also ohne den expliziten Aufruf eines EQMs, und sind daher in der Abarbeitung nicht abhängig von Kommandoevents.

Parameter: Keine.

Daten: Die 25 Langworte enthalten im Einzelnen:

- 1: Gerätestatus (wie in der Property STATUS)

- 2:** Gibt in den oberen 16 Bits an, welcher virtuelle Beschleuniger aktiv gesetzt ist (ein Bit pro Beschleuniger). Das niederwertigste Bit (Bit 16) gibt den Beschleuniger 15 an, das Bit 31 den Beschleuniger 0. Die unteren 16 Bit sind nicht verwendet. Dabei bedeutet Null, dass der Beschleuniger inaktiv ist und Eins, dass der Beschleuniger aktiv ist.
- 3:** Master-Fehler. Hier ist derjenige Master-Gerätefehlercode mit dem schwersten Fehlergrad eingetragen. Bei mehreren Fehlern mit dem gleichen Fehlergrad wird der erste eingetragen, der gefunden wurde.
- 4:** Slave Fehler für virtuellen Beschleuniger 0. Entsprechend dem Master-Fehler wird hier der nach dem Fehlergrad schwerste Slave-Gerätefehlercode für den Beschleuniger 0 eingetragen.
- 5:** Entsprechend Punkt 4, aber für virtuellen Beschleuniger 1.
- ⋮
- 19:** Entsprechend Punkt 4, aber für virtuellen Beschleuniger 15.
- 20:** Reserviert für Erweiterungen.
- ⋮
- 25:** Reserviert für Erweiterungen.

6.2.2 INIT

Bedeutung: Initialisierung des Gerätes (Kaltstart). Durchzuführenden Aktionen: Init-Kommando zur IFB, alle Vaccs inaktiv setzen, den definierten (und im Softwarecode festgeschriebenen) Init-Parametersatz zum Gerät schicken. Dieser Parametersatz wird zugleich als Default-Datensatz und Sollwert-Datensatz für alle Vaccs gesetzt.

Parameter: Keine.

Daten: Keine.

R-INFOs:

6.2.3 POWER

Bedeutung: Gibt an, ob der Leistungsteil des Gerätes ein- oder ausgeschaltet ist bzw. werden soll.

Parameter: Keine.

Daten: Das Datum kann nur zwei Werte annehmen. Null heißt, das Gerät ist eingeschaltet bzw. soll eingeschaltet werden. Eins heißt, das Gerät ist ausgeschaltet bzw. soll ausgeschaltet werden.

Hat keine Funktion, als Rückgabewert wird immer 1 (Power On) geliefert.

R-INFOs: SIND WIRKLICH KEINE INFORMATIONEN ÜBER EIN-/AUSSCHALTUNG DER GERÄTE VERFÜGBAR ?

6.2.4 RESET

Bedeutung: Reset des Gerätes (Warmstart). Durchzuführende Aktionen: Reset-Kommando zum Gerät, Default-Datensatz zum Gerät schicken. 5.8.2 auf Seite 13.

Parameter: Keine.

Daten: Keine.

R-INFOs:

6.2.5 STATUS

Bedeutung: Auslesen des 32bit Gerätestatus.

Parameter: Keine.

Daten: Das 32bit Statuswort. Die Bits entsprechen den Statusbits, wie sie in Abschnitt 4.5 auf Seite 9 und in der Tabelle ?? auf Seite ?? erklärt sind.

R-INFOs:

6.2.6 VERSION

Bedeutung: Lesen der Versionskennung der Gerätesoftware.

Parameter: Keine.

Daten: Versionskennung als ASCII-String, pro Datum ein ASCII-Zeichen.

Bytes	Inhalt
1...12	Version der USRs
13...24	Version der EQMs
25...36	Version des Standard-MIL-Treibers
37...48	Variante der EQMs

6.3 Die Slave-Properties

Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
ACTIV	R/W	0	-	1	BitSet16	1	0
COPYSET	W	0	-	1	BitSet16	1	0
EQMERROR	RA	217	Integer32	348	Integer32	1	0
SETDATA	R/W	0	-	1501	BitSet32	-	0
MILDATA	R/W	0	-	601	BitSet32	-	0
SYNCMODE	R/W	0	-	1	SWord	-	0
WORKMODE	R/W	0	-	1	SWord	-	0
MHBPAR	R/W	1	Integer16	5	Integer32	-	0
MHBPAR2	R/W	0	-	15	Integer32	-	0
EXPPAR	R/W	1	Integer16	150	Integer16	-	0

6.3.1 ACTIV

Bedeutung: Gibt an, ob das Gerät für den zugehörigen virtuellen Beschleuniger an der Puls-zu-Puls-Modulation teilnehmen soll bzw. teilnimmt.

Parameter: Keine.

Daten: Das Datum kann nur zwei Werte annehmen. Null heißt, das Gerät nimmt für den zugeordneten Beschleuniger *nicht* an der PPM teil bzw. soll *nicht* an der PPM teilnehmen. Eins heißt, das Gerät nimmt für den zugeordneten Beschleuniger an der PPM teil bzw. soll an der PPM teilnehmen.

R-INFOs: Wann reagiert das Gerät auf einen am Geraet eintreffenden Event ? Muss es von der VME-Ebene scharf gemacht werden, oder arbeitet es einen eintreffenden Event immer ab ? D.h. hat das Activ setzen von Seiten der VME Ebene überhaupt irgendeinen Einfluss auf die Arbeit des Gerätes ? Um eine definierte Steuerung von Seiten der VME-Ebene und einen sicheren Ablauf der Geraete-Funktion zu gewährleisten, sollte das Geraet von Seiten der VME-Ebene scharf gemacht werden können (aktiv-Signal) und einen entsprechenden Zyklus erst dann abarbeiten. (In anderen Worten: Zyklus nur bearbeiten, wenn aktiv-Signal von VME UND Vacc-Event eingetroffen sind)

6.3.2 COPYSET

Bedeutung: Kopiert alle Geräteeinstellungen (Sollwerte) eines virtuellen ('fremden') Beschleunigers in den zugehörigen ('eigenen') Beschleuniger.

Parameter: Keine.

Daten: Nummer des virtuellen ('fremden') Beschleunigers, von dem die Einstellungen (Sollwerte) kopiert werden sollen.

6.3.3 EQMERROR

Bedeutung: Fehlermeldungen der auf der SE installierten Gerätesoftware. Es werden die aktuellen Fehlermeldungen sowohl für die Masterfehler als auch für die Slavefehler der Geräteebene geliefert. Dazu wird auch der Inhalt des Fehlerpuffers zurückgegeben, in dem die letzten aufgetretenen Fehler abgespeichert wurden.

Parameter: Hier hat nur der erste der 217 Parameter eine Bedeutung.

- 1:** Wird bei konnektierten Aufträgen ausgewertet. 0: Es wird bei jeder Ausführung des Auftrages eine Antwort verschickt. 1: Es wird bei jeder Ausführung des Auftrages nur dann eine Antwort verschickt, wenn sich seit dem letzten Aufruf der Inhalt der Daten geändert hat.
- 2...217:** Dummy, sie werden vom MOPS intern verwendet und können vom Benutzer beliebig gesetzt werden.

Daten: Die Anzahl der Fehlermeldungen sei bezeichnet durch:

- m* Zahl der Master-Fehlermeldungen
- s* Zahl der Slave-Fehlermeldungen
- b* Größe des Fehlerpuffers

Weiterhin soll gelten:

$$l = m + s$$
$$t = m + s + b$$

Die Daten im Einzelnen:

1: In den unteren beiden Bytes sind die Anzahl der Master-Fehlermeldungen m und die Anzahl der Slave-Fehlermeldungen s angegeben:

0	0	s	m
---	---	-----	-----

2: erste Master-Fehlermeldung

⋮

$m + 1$: letzte Master-Fehlermeldung

$m + 2$: erste Slave-Fehlermeldung

⋮

$l + 1$: letzte Slave-Fehlermeldung

$l + 2$: Länge b des Fehlerpuffers

$l + 3$: Zahl der Einträge im Fehlerpuffer

$l + 4$: Index des ersten freien Platzes im Fehlerpuffer (der Fehlerpuffer ist ein Ringpuffer)

$l + 5$: Erster Speicherplatz im Fehlerpuffer

⋮

$t + 4$: Letzter Speicherplatz im Fehlerpuffer

6.3.4 SETDATA

Bedeutung: Setze oder lese Sollwert-Parameter des Gerätes für einen Vacc. Die Werte werden auf der VME Ebene zwischengespeichert und in der nächsten Zykluspause per Kommando (!) an das Geraet geschrieben oder (soweit möglich) vom Gerät gelesen.

Parameter: Keine.

Daten: Insgesamt 1501 Elemente vom SIS-Datentyp BitSet32, mit folgender Bedeutung:

count: Anzahl der Parameter-Sätze, 1 .. 300

devAddr 0: Device Adresse, 0. Paramter-Satz

parAddr 0: Paramter Adresse, 0. Paramter-Satz

size 0: Größe der Parameter-Daten, in Byte (1..8), 0. Paramter Satz

dataLow 0: Parameter Daten, untere Bytes (Byte 0 .. 3), 0. Paramter-Satz

dataHigh 0: Parameter Daten, obere Bytes (Byte 4 .. 7), 0. Paramter-Satz

devAddr 1: Device Adresse, 1. Paramter-Satz

parAddr 1: Paramter Adresse, 1. Paramter-Satz

size 1: Größe der Parameter-Daten, in Byte (1..8), 1. Paramter Satz

dataLow 1: Parameter Daten, untere Bytes (Byte 0 .. 3), 1. Paramter-Satz

dataHigh 1: Parameter Daten, obere Bytes (Byte 4 .. 7), 1. Parameter-Satz
 ...
devAddr 299: Device Adresse, 299. Parameter-Satz
parAddr 299: Parameter Adresse, 299. Parameter-Satz
size 299: Größe der Parameter-Daten, in Byte (1..8), 299. Parameter Satz
dataLow 299: Parameter Daten, untere Bytes (Byte 0 .. 3), 299. Parameter-Satz
dataHigh 299: Parameter Daten, obere Bytes (Byte 4 .. 7), 299. Parameter-Satz

R-INFOs: Es müssen nicht alle 300 max. möglichen Parametersätze gesendet werden sondern nur die durch count angegebene Anzahl.

6.3.5 WORKMODE

Bedeutung: Liest oder Schreibt den Workmode eines Vaccs zur SE.

Parameter: Keine.

Daten: Workmode: Das Datum kann folgende Werte annehmen:

- 0:** Standard/Default: Im aktivierten Vacc wird das Gerät mit den Standard/Default Daten geladen
- 1:** HF-Sync: Im aktivierten Vacc wird das Gerät mit den Daten zur HF-Synchronisation geladen, entsprechend Prop. SYNCMODE.
- 2:** MHB-Mode: Im aktivierten Vacc wird das Gerät mit den Daten zum MHB-Mode geladen, entsprechend Prop. MHBPAR.
- 3:** EXP-Mode: Im aktivierten Vacc wird das Gerät mit den Daten des EXPERIMENT-Mode geladen, entsprechend Prop. EXPPAR.

R-INFOs: ..

6.3.6 SYNCMODE

Bedeutung: Liest oder Schreibt einen Synchronisations-Modus zum Gerät. Welche der Geräte-Register wie gesetzt werden müssen, ist auf der VME-Ebene vor-definiert, diese Property arbeitet nur mit den Nummern der Modii. Einfach für die Operateure aber wenig flexibel durch festverdrahtete Werte. Die eingestellten Register und deren Werte können aber danach mit der Property DPRDATA und MILDATA gelesen werden.

Parameter: Keine.

Daten: SyncModus: Das Datum kann folgende Werte annehmen:

- 1:** Synchronisation aus,
- 2:** Synchronisation EIN mit 0-grad Phase,
- 3:** Synchronisation EIN mit 1800-grad Phase,

R-INFOs: ..

6.3.7 MHBPAR

Bedeutung: Liest oder Schreibt die nötigen Parameter für den Multi-Harmonischen-Betrieb vom/zum Gerät. Die eingestellten Register und deren Werte können danach mit der Property DPRDATA und MILDATA kontrolliert werden. Hier sind nur die Registeradressen festgelegt, die Parameter aber variabel einstellbar.

Parameter: 1 Integer 16, EvtNum.

Bedeutung: Event-/Zeitpunkt-Index, dem die Daten zugeordnet werden. Im MHB-Betrieb werden zu drei Events-/Zeitpunkten Daten zum Gerät geschickt. EvtNum=0=Prep=Zyklusstart, EvtNum=1=Evt1,zur Zeit Flattop+Delay1, EvtNum=2=Evt2 = Flattop + Delay-1 + Delay2

Daten: Die Daten liegen in folgender Struktur vor:

Delay: Delay-Zeit in usec, immer 0 bei Evt=0=Prep

Harm-1: Harmonischnzahl für Group-RF1 und Offset Low 1

Harm-1: Harmonischnzahl für Group-RF1 und Offset Low 1

DAmp1: Distribution Amplifier 1

DAmp2: Distribution Amplifier 2

R-INFOs: ..

6.3.8 MHBPAR2

Bedeutung: Exakt wie die Property MHBPAR. Die Property wurde designed, um mit LSA/JAPC zugreifen zu können. LSA kann keine Property-Parameter versorgen.

Parameter: Keine.

Bedeutung: Exakt wie die Property MHBPAR, nur dass *alle* Daten in einem Zugriff gelesen oder geschrieben werden.

Daten: Die Daten liegen in folgender Struktur vor:

Delay_1: Delay-Zeit in usec, immer 0 bei EvtStart_Cycle.

Harm1_1: Harmonischnzahl für Group-RF1 und Offset Low 1

Harm2_1: Harmonischnzahl für Group-RF1 und Offset Low 1

DAmp1_1: Distribution Amplifier 1

DAmp2_1: Distribution Amplifier 2

Delay_2: Delay-Zeit in usec ab Evt_Flattop.

Harm1_2: Harmonischnzahl für Group-RF1 und Offset Low 1

Harm2_2: Harmonischnzahl für Group-RF1 und Offset Low 1

DAmp1_2: Distribution Amplifier 1

DAmp2_2: Distribution Amplifier 2

Delay_3: Delay-Zeit in usec ab Evt_Flattop plus Delay_2.

Harm1_3: Harmonischnzahl für Group-RF1 und Offset Low 1

Harm2_3: Harmonischnzahl für Group-RF1 und Offset Low 1

DAmp1_3: Distribution Amplifier 1

DAmp2_3: Distribution Amplifier 2

R-INFOs: ..

6.3.9 EXPPAR

Bedeutung: Setze oder lese Registerdaten des Gerätes für einen Vacc. Die Werte werden auf der VME Ebene zwischengespeichert und im nächsten Zyklus eines aktivierten und im workmode= EXP stehenden Vaccs an das Geraet geschrieben. Das Datenformat entspricht einem optimierten BLOB-Format. Zu beachten ist, dass die Experiment-Parameter nicht für jeden Vacc getrennt eingestellt werden können, es gibt nur einen entsprechenden Speicher. Jeder Vacc, der im Experiment-Mode aktiviert wird, nimmt sich die Parameter aus diesem gemeinsamen Speicher !.

Parameter: 1 Integer16, EvtNum

Bedeutung: Event-Index, zu dem die Daten zugeordnet werden. EvtNum=0, d.h. Event=Prep, Zyklusstart, EvtNum=1, d.h. Event=Evt1, Flattop (z.B.)+Delay1, ... EvtNum=4, d.h. Event=Evt4, Evt3+Delay4, Max. EvtNum=4

Daten: Insgesamt 152 Elemente vom SIS-Datentyp Bitset32, mit folgender Bedeutung:

Delay: Delay in usec vom vorhergehenden Event (Evt1,Evt2..)

count: Anzahl der Register-Sätze, 1..50

DevAdr 1: Device Adresse, 0. Register-Satz

RegAdr 2: Register Adresse, 0. Register-Satz

RegDat 3: Register Daten, 16-Bit, 0. Register-Satz

...

DevAdr 50: Device Adresse, 50. Register-Satz

RegAdr 50: Register Adresse, 50. Parameter-Satz

RegDat 50: Register Daten, 16-Bit, 50. Register-Satz

R-INFOs: Es müssen nicht alle 50 max. möglichen Registersätze gesendet werden sondern nur die durch count angegebene Anzahl. Und: Zur vollständigen Einstellung der Experiment-Parameter eines Vaccs muss diese Property fünfmal aufgerufen werden ! (Für EvtNum = 0, 1,...,4). Achtung: Wird für Evt1,2,...,4 das Delay=0 gesetzt bedeutet dies den Abbruch des Zyklus, d.h. diese und andere Evt1 werden nicht bearbeitet und keine Daten gesendet !

6.3.10 MILDATA

Bedeutung: Lese die MIL-Transfers des letzten/aktuellen Vacc-Zyklus. Von einem MIL-Transfer werden MIL-Command (IFB-Funktionscode) sowie zugehöriges Datum gelesen. Die Controls-Software speichert alle MIL-Transfers eines Zyklus, sodass hier sowohl die am Zyklus-Anfang gesetzten Werte als auch die zu späteren Zeitpunkten geschriebenen Registerwerte kontrolliert werden können.

Parameter: Keine.

Daten: Insgesamt max. 601 Elemente vom SIS-Datentyp BitSet32, mit folgender Bedeutung:

count: Anzahl vorhandener/gelesener MIL-transfers, 0 .. 600

MilDat 1: 1. Mil-Command (Ifb-Funktionscode)

MilDat 2: 1. Mil-Datum

MilDat 3: 2. Mil-Command (Ifb-Funktionscode)
MilDat 4: 2. Mil-Datum
...: —

R-INFOs: Es müssen nicht alle 600 max. möglichen MilDaten gesendet werden sondern nur die durch count angegebene Anzahl.

6.3.11 DPRDATA

Bedeutung: Lese zur Diagnose wichtige Daten des VME-DPR's.

Parameter: Keine.

Daten: Insgesamt 200 Elemente vom SIS-Datentyp BitSet32, mit folgender Bedeutung:

Dprdat 0: workmode
DprDat 1: Syncmode
DprDat 2: ohne Bedeutung
DprDat 3: EvtNum, Anzahl Events, z.B. 0 im Workmode Default, 2 im workmode MHB
DprDat 4: EvtLoop, Index des letzten, bereits abgearbeiteten Events, d.h. 0=Prep, 1=Evt-1,..
DprDat 5: Register 1, Name
DprDat 6: Register 1, Datum
DprDat 7: Register 2, Name
DprDat 8: Register 2, Datum
...: —
DprDat 103: Register 50, Name
DprDat 104: Register 50, Datum
DprDat 105-199:] reserviert

R-INFOs: Die Registerliste enthält den Namen eines bekannten HFS-Hardware-Registers sowie die daten, die zuletzt im Zyklus auf dieses Register geschrieben wurden. An EvtLoop kann man ablesen, zu welchem Zeitpunkt im Zyklus dies geschehen ist.

Teil II

Der Entwurf der Software

7 Softwareentwurf

Dies ist noch ein sehr allgemeiner Punkt. Hier sollte unter anderem hingehören:

- Datenstrukturen,
- Datenflussdiagramme,
- Kontrollflussdiagramme
- ...

8 Lokale Datenbasis

8.1 Tabelle der Konstanten

Keine Konstanten vorgesehen.

Siehe auch die Beschreibung der Property CONSTANT.

9 Dualport RAM

Hierher kommt die Beschreibung des gerätespezifischen Teils des Dualport RAM (`m_data_type`, `s_data_type`, `Dev_Common_Buf_Type`, ...).

9.1 M_data_type

In den Masterdaten stehen alle Daten, die NICHT vacc abhängig sind. Von den HFS-spezifischen Master-Daten sind hier folgende zu erläutern:

MregList Ist die Definitionsliste der Hardware-Register. Sie beinhaltet Name/Nummer, Device-Adresse, Register-Adresse, Registerbreite und das Default-Datum aller bekannten Register.

Synclist Ist die Reg.-Liste der Default HFS-Sync Mode Einstellung, beinhaltet Name und Datum aller bekannten Register.

Mhblist Ist die Reg.-Liste der Default MHB Einstellung, beinhaltet Name und Datum aller bekannten Register.

ExpPar Registerdaten im BLOB-2 Format (Device-Adresse, Register-Adresse, 16-Bit Datum. Beliebige Adressen und Daten können gesetzt werden, sodass dem Experimentator wenig Grenzen gesetzt sind. Einer Liste von max. 50 Registersätzen stehen eine EvtNum und ein Delay voran, durch die Registersätze zu max. 5 Zeitpunkten im Zyklus geschrieben werden können.

EvtNum 0: Zeitpunkt Startcycle, Prepare. Delay=0

EvtNum 1: Zeitpunkt: Flattop_Start + Delay-1

EvtNum 2: Zeitpunkt: Flattop_Start + Delay-1 + Delay-2

–

Zu beachten ist, dass die ExpParameter im Zyklus gesendet werden, wenn der entsprechende vacc im workmode EXP aktiviert ist. Man kann mehrere Vaccs gleichzeitig in diesem workmode aktivieren, alle greifen aber auf diesen Datensatz zu. Dies wurde so realisiert, da ein Betrieb mehrerer Vaccs im Experimentier-Betrieb nicht sinnvoll erscheint. Es sollte jedoch der Experimentierbetrieb parallel zum normalen Betriebsmode ermöglicht werden.

9.2 S_data_type

In den Slavedaten stehen alle vacc abhängigen Daten. Von den HFS-spezifischen Slave-Daten sind hier folgende zu erläutern:

hfsSync Gibt den Synchronisationsmodus an, 0=keine Sync., 1=Sync.ein -Phase 180-grad, 2=Sync.ein Phase 0-grad. Die Software muss dazu die Register für gain und Phase beider Kavitäten ändern.

MhbPar Array 0-2 (0=Daten für Prep, 1=Daten für Evt-1,...) einer Parameter-Struktur. Die Struktur enthält 5 Werte (Delay, Harm-1, Harm-2, DisAmp1,DisAmp-2) von denen die beiden Harmonischenzahlen in jeweils 2 der definierten Register geschrieben wird. Delay ist bei den Daten für Prepare nicht relevant. Für Evt1,Evt2 gilt: Delay gibt den Abstand zum vorangehenden Event, s. Beschreibung PROP MhbPar.

SdataSet Daten im älteren BLOB-1 Format. steht nur noch aus Kompatibilitätsgründen mit alten Nodal-Programmen in der Controls-Software, neuere Programme sollen das neue BLOB-2 Format und ExpPar verwenden.

EvtNum,EvtLoop Zwei variablen zur Kontrolle der Datensendungen innerhalb des Zyklus zu den Zeitpunkten Evt1,2,... EvtNum zeigt die Gesamtzahl der durchzuführenden Datensendungen (Evts) exkl. Prep. (Start-Cyclus). EvtLoop zeigt die bereits abgelaufenen Evts, 0 bei Prep, 1=Evt1...

WriteRegs Eine Register-datenliste aller definierten Register in dem die zuletzt zur HW geschriebenen daten in diesem Zyklus stehen. Damit wird unter anderem geprüft, ob die aktuellen Daten überhaupt neu sind und wirklich gesendet werden müssen.

MilBuf Diagnosebuffer, in dem alle in einem Zyklus ausgeführten MIL-Transfers gespeichert sind. Kann mit PROP MILDATA gelesen werden.

PROP DPRDATA Mittels dieser Propertie werden hfsSync,EvtNum,Evtloop und WriteRegs eines Beschleunigers gelesen.

10 Zyklus-Ablauf

Der Ablauf bzw. die ausgeführten Funktionen im Beschleuniger-Zyklus sind abhängig vom eingestellten workmode und dessen Parameter.

Prepare In den Workmodes Default,Sync und MHB werden alle definierten Register zum Zyklus Beginn (Event Cycle.start) mit Daten versorgt. Die Daten werden wie folgt ermittelt: Laden einer Registerliste (WriteRegs) mit den konstanten Daten des Workmodes + den variablen Daten des Workmodes. Dies ist in der Funktion Set_Prep_regs zusammengefasst. Im workmode EXP werden zum Zyklus Beginn (Event Cycle.start) die gesetzten Registerdaten (BLOB-2 Format) aus ExpPar[0] gesetzt. Hier erfolgt keine Kontrolle, welche oder wieviel Register gesetzt werden.

Mhb Zyklus Im MHB workmode werden an zwei Zeitpunkten (Evts) im Zyklus Daten zur Hardware geschickt. Hierzu werden die Datenarrays Mhbpar[1] und Mhbpar[2] verwendet. In jedem Array stehen Registerwerte und ein Delay. Delay-1 beschreibt die Zeitspanne von Evt_Flattop_start bis zur ersten Datensendung an Evt-1. Delay-2 beschreibt die Zeitspanne von Evt-1 bis zur zweiten Datensendung an Evt-2. In der SE Software wird dazu an das Beschleuniger-Event Evt_Flattop_start das Eqm_EvtX angehängt, das sich mit dem Delays 1,2 immer wieder selbst aufruft und die Registerdaten in Mhbpar 1,2 zur Hardware schreibt. Dabei wird überprüft, ob sich der Registerinhalt wirklich ändert und nur neue Daten werden geschickt.

Exp Zyklus Im EXP workmode werden an bis zu 4 Zeitpunkten (Evts) im Zyklus Daten zur Hardware geschickt. Hierzu werden die Datenarrays ExpPar[1] bis ExpPar[4] verwendet. In jedem Array stehen Registerwerte und ein Delay. Für die Delays und die Funktion des Eqm_EvtX gilt das gleiche wie im Mhb-Zyklus beschrieben. Dabei werden im EXP workmode die Registerinhalt nicht überprüft und die Daten werden immer geschickt. Das Senden der Daten im Zyklus kann verhindert bzw. unterbrochen werden, wenn das Delay auf 0 gesetzt wird. Beispiel: Delay-1 = 10000usec, Delay-2 = 0: Die Daten in ExpPar[1] werden 10000 usec nach Evt_Flattop_start gesendet, die Daten von ExpPar[2,3,4] werden nicht gesendet.

Literatur

Index

— Symbole —

Übergabe der Geräte-Daten.....11

— A —

Aufgabe des Gerätes.....5

— B —

Bedienungsfehler.....14

— D —

Datenbasis.....23

DRD Interrupt.....9

DRQ Interrupt.....9

Dualport RAM.....23

— E —

Emergency-Event.....13

Event-Overrun.....13

Event-Sequenzfehler.....13

Eventkonnektierungen.....12

— F —

Funktionscodes.....8

- ifb_reset.....9

— G —

Genauigkeitsanforderungen.....12

Gerät

- Aufgabe.....5
- Hardware.....5
- Repräsentation.....14
- Steuerung.....10

Geräte

- Schnittstelle.....8

Gerätemodell.....5

- Kennzeichnung.....14
- Master-Properties.....14
- Slave-Properties.....16

Gerätevarianten.....6

— H —

Handbetrieb.....13

Hardware des Gerätes.....5

Hardwarefehler-Bit.....13

Hardwarestatus.....9

— I —

ifb_reset.....9

Init.....13

Interfacekarte.....8

Interlock.....9, 13

Interrupt

- DRD Interrupt.....9
- DRQ Interrupt.....9
- Interlock.....9

— K —

Kaltstarts.....13

— L —

Lokale Datenbasis.....23

Lokalen Datenbasis

- Tabelle der Konstanten.....23

— M —

M_data_type.....23

Master-Properties.....14

— O —

Overrun.....13

— P —

Properties

- ACTIV.....17
- COPYSET.....17
- DPRDATA.....22
- EQMERROR.....17
- EXPPAR.....21
- INFOSTAT.....14
- INIT.....15
- Master-.....14

- MHBPAR 20
- MHBPAR2 20
- MILDATA 21
- POWER 15
- RESET 16
- SETDATA 18
- Slave- 16
- STATUS 16
- SYNCMODE 19
- VERSION 16
- WORKMODE 19

— **Z** —

- Zeitkritische Anforderungen 12
- Zyklus Ablauf 24

— **R** —

- Repräsentation des Gerätes 14
- Reset 13

— **S** —

- S_data_type 24
- Schnittstelle zum Gerät 8
- Sequenzfehler 13
- Slave-Properties 16
- Softwareentwurf 23
- Softwarestatus 9
- Störungen 13
 - Emergency-Event 13
 - Event-Overrun 13
 - Event-Sequenzfehler 13
 - Interlock 13
 - Kommunikation EC – Gerät 14
- Startwerte 13
- Statusbits 9
- Steuerung des Gerätes 10

— **T** —

- Timing 12

— **U** —

- Use Case(s) 6

— **V** —

- Varianten
 - Betriebs- 12
 - Geräte- 6

— **W** —

- Warmstarts 13