

PZS - SIS/ESR-Pulszentrale

P. Kainberger

Dieses Papier enthält die Beschreibung des Gerätemodells »PZS – SIS/ESR-Pulszentrale« und den Entwurf der Gerätesoftware für dieses Gerät.

Änderungsprotokoll

Datum	GM-Version	Name	Kommentar
03. Nov. 97	–	P. Kainberger	Beginn der Erstellung
14. Jan. 98	–	P. Kainberger	erste Vorabversion (unverbindlich und ohne Gewähr auf Vollständigkeit)
10. Feb. 98	–	P. Kainberger	ACCVIA wieder rein EXECREQU geändert
27. Mai 98	–	P. Kainberger	INFO, EVTARRAY und EVTSEQU geändert
08. Juni 98	–	P. Kainberger	THINFO ergänzt
05. Oct. 99	–	P. Kainberger	Erweiterung EXECBLCK (Änderung RIEGTAB-Belegung)
Feb.2000	–	M. Kühn	Überarbeitete und erweiterte TeX-Source, die in PostScript als auch in HTML konvertiert werden kann.
20. Juni 2000	–	P. Kainberger	Änderung RIEGTAB-Belegung
07. Jan. 2002	–	P. Kainberger	Änderung und Erweiterung RIEGTAB-Belegung
25. Sep. 2002	–	P. Kainberger	Erweiterung GAPNOPOS
27. Mar. 2003	–	U. Krause	Belegung der Therapie-Anfordereinheit
15. Dez. 2003	–	P. Kainberger	Änderung RIEGTAB-Belegung

Inhaltsverzeichnis

1	Die Schnittstelle zum Gerät	5
1.1	Definition der Bits des Hardwarestatus	5
I	Das Gerätemodell	7
2	Die Repräsentation des Gerätes	7
2.1	Kennzeichnung des Gerätemodells	7
2.2	Die Master-Properties	7
2.2.1	POWER	7
2.2.2	STATUS	8
2.2.3	INIT	8
2.2.4	RESET	8
2.2.5	VERSION	8
2.2.6	INFOSTAT	8
2.2.7	CONSTANT	10
2.2.8	ASYNACC	10
2.2.9	ASYNCDAT	10
2.2.10	CMSEQU	10
2.2.11	SUPCYCLE	11
2.2.12	PUSK	11
2.2.13	WAITNEXT	11
2.2.14	INFO	12
2.2.15	THINFO	13
2.2.16	RIEGTAB	15
2.2.17	SWRGTAB	17
2.2.18	PZMODE	17
2.3	Die Slave-Properties	17
2.3.1	ACTIV	18
2.3.2	COPYSET	18
2.3.3	EQMERROR	18
2.3.4	ACCVIA	19
2.3.5	STRALWEG	20
2.3.6	EVTARRAY	20
2.3.7	EVTSEQU	21
2.3.8	CHANSEQU	21
2.3.9	EXECCNT	22
2.3.10	EXECMODE	23
2.3.11	EXECREQU	23
2.3.12	ACCDESCR	23
2.3.13	ACCREQU	23
2.3.14	RIEGMASK	25
2.3.15	GAPNOPOS	25
2.3.16	EXECBLCK	26
II	Die Realisierung in Hard- und Software	27

3	Die Hardware der SIS/ESR-Pulszentrale	27
3.1	Die Schnittstellen der SIS-Pulszentrale	27
3.2	Belegung der Schnittstellen der SIS-Pulszentrale	28
3.3	Belegung der Therapie-Anfordereinheit	29
3.4	Die Schnittstellen der ESR-Pulszentrale	30
3.5	Belegung der Schnittstellen der ESR-Pulszentrale	31
3.6	Die Schnittstelle des Event-Piggy	32
3.7	Die externen Triggereingänge der Pulszentralen	32

Abbildungsverzeichnis

1	Die Schnittstellen der SIS-Pulszentrale	27
2	Modulbus-Einschub UNILAC/SIS	28
3	Modulbus-Einschub ESR/SIS	31
4	Verschaltung der Trigger-Eingänge ESR/SIS	34

1 Die Schnittstelle zum Gerät

???. Dieses Kapitel beschreibt den Aufbau der digitalen Schnittstelle zum Rechnersystem. Dabei geht es um die Beschreibung der *statischen* Schnittstelle und nicht um dynamische Anforderungen des Gerätes. Diese werden im Kapitel »Bedienung des Gerätes« behandelt.

1.1 Definition der Bits des Hardwarestatus

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	Eventbus transmit fuse	ok	error
9 ... 19	unused		
20 ... 22	Main modulbus components		
20	Event-Sequencer	ok	error
21	32-Bit I/O-card 1	ok	error
22	32-Bit I/O-card 2	ok	error
23 ... 25	Unilac modulbus components		
23	32-Bit I/O-card 1	ok	error
24	32-Bit I/O-card 2	ok	error
25	32-Bit I/O-card 3	ok	error
26 ... 31	ZKS, VAC and MNG modulbus components		
26	32-Bit I/O-card 1 (ZKS Bits 32 ... 63)	ok	error
27	32-Bit I/O-card 2 (ZKS Bits 0 ... 31)	ok	error
28	32-Bit I/O-card 3 (Vacuum Bits 32 ... 63)	ok	error
29	32-Bit I/O-card 4 (Vacuum Bits 0 ... 31)	ok	error
30	32-Bit I/O-card 5 (MNG Bits 32 ... 63)	ok	error
31	32-Bit I/O-card 6 (MNG Bits 0 ... 31)	ok	error

Teil I

Das Gerätemodell

2 Die Repräsentation des Gerätes

Dieses Kapitel definiert das Gerätemodell, also wie das Gerät nach höheren Ebenen hin abgebildet wird. Es beschreibt die Schnittstelle zwischen Benutzerebene (Operatingprogrammen) und Geräteebene (Gerätehard- und -software).

Ein Gerät erscheint zur Benutzerebene im Umfang des in Abschnitt definierten logischen Gerätes.

2.1 Kennzeichnung des Gerätemodells

Das Gerätemodell hat die Bezeichnung **PZS**.

Die Gerätemodellnummer ist 65_{dez} .

2.2 Die Master-Properties

Master-Properties							
Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
POWER	R/W	0	–	1	BitSet16	1	0
STATUS	R	0	–	1	BitSet32	1	0
INIT	N	0	–	0	–	–	–
RESET	N	0	–	0	–	–	–
VERSION	RA	0	–	48	BitSet8	1	0
INFOSTAT	RA	0	–	25	BitSet32	1	0
CONSTANT	RA	0	–	???	????????	1	0
ASYNCACC	W	0	–	1	BitSet16	1	0
ASYNCDAT	WA	0	–	8	BitSet16	1	0
CMSEQU	RA/WA	1	BitSet16	1025	BitSet16	1	0
SUPCYCLE	RA/WA	0	–	257	BitSet16	1	0
PUSK	RA/WA	0	–	2	BitSet16	1	0
WAITNEXT	RA/WA	0	–	2	BitSet16	1	0
INFO	RA	0	–	467	BitSet16	1	0
THINFO	RA	0	–	15	BitSet16	1	0
RIEGTAB	RA	0	–	12	BitSet32	1	0
SWRGTAB	RA/WA	0	–	3	BitSet32	1	0
PZMODE	R/W	0	–	1	BitSet16	1	0

2.2.1 POWER

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

Das Gerät PZS hat kein Leistungsteil. Ein Versuch **POWER** zu schalten, liefert einen Fehler.

Parameter: Keine.

Daten: Das Datum kann nur den Wert 1, d. h. eingeschaltet, annehmen.

2.2.2 STATUS

Bedeutung: Auslesen des 32 Bit Gerätestatus.

Parameter: Keine.

Daten: Das 32bit Statuswort. Die Bits entsprechen den Statusbits, wie sie in Abschnitt 1.1 auf Seite 5 und in der Tabelle 1.1 auf Seite 5 erklärt sind.

2.2.3 INIT

Bedeutung: Initialisierung des Gerätes (Kaltstart). Für die dabei durchzuführenden Aktionen siehe Abschnitt auf Seite 4.

Parameter: Keine.

Daten: Keine.

2.2.4 RESET

Bedeutung: Reset des Gerätes (Warmstart). Für die dabei durchzuführenden Aktionen siehe Abschnitt auf Seite 4.

Parameter: Keine.

Daten: Keine.

2.2.5 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 verwendeten MIL-Treibers
37 ... 48	Variante der EQMs

2.2.6 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:** EC-Mode. In den oberen 16 Bit des Langwortes steht der von der Gerätesoftware eingestellte Default-EC-Mode, in den unteren 16 Bit der aktuelle EC-Mode. Folgende Modi sind definiert:
 - 0:** *not set*
 - 1:** *Preset.Command* Der ECM hat das Umschalten in Command-Mode vorbereitet aber noch nicht beendet.
 - 2:** *Command* Der ECM läuft im Command-Mode.
 - 3:** *Preset.Event* Der ECM hat das Umschalten in Event-Mode vorbereitet aber noch nicht beendet.
 - 4:** *Event* Der ECM läuft im Event-Mode.
- 21:** EC-Performance-Mode. In den oberen 16 Bit des Langwortes steht der von der Gerätesoftware eingestellte Default-Performance-Mode, in den unteren 16 Bit der aktuelle Performance-Mode. Folgende Modi sind definiert:
 - 0:** *not set*
 - 1:** *Display* Der ECM läuft im Display-Mode.
 - 2:** *Preset.Turbo* Der ECM hat das Umschalten in den Turbo-Mode vorbereitet aber noch nicht beendet.
 - 3:** *Turbo* Der ECM läuft im Turbo-Mode.
- 22:** HW_Warning_Maske. Die 32 Bits geben an aus welchen Bits im Gerätestatus das HW-Warning-Bit im Status abgeleitet wird.
- 23** Pulszentralen-Identifikation:
 - 0:** TIF
 - 1:** SIS-PZ
 - 2:** ESR-PZ
 - 3 . . . 6:** undefiniert
 - 7:** Software-PZ
 - 8:** UNILAC, Master-PZ
 - 9:** UNILAC-PZ 1
 - 10:** UNILAC-PZ 2
 - 11:** UNILAC-PZ 3
 - 12:** UNILAC-PZ 4
 - 13:** UNILAC-PZ 5
 - 14:** UNILAC-PZ 6
 - 15:** UNILAC-PZ 7
- 24:** Reserviert für Erweiterungen.
- 25:** Reserviert für Erweiterungen.

2.2.7 CONSTANT

Bedeutung:

Parameter: Keine.

Daten:

2.2.8 ASYNCACC

Bedeutung: Ausführen eines Beschleunigers während die Pulszentrale im Superzyklusablauf angehalten ist. Da die Ausführung des Beschleunigers nicht über die Superzyklus-Steuerung synchronisiert erfolgt, wird diese Art der Beschleunigerausführung als *asynchron* bezeichnet.

Parameter: Keine

Daten: Das Datum gibt die Nummer des Beschleunigers an, der *asynchron* ausgeführt werden soll.

2.2.9 ASYNCDAT

Bedeutung: Verschicken asynchroner Datenevents zur Triggerung sogenannter ECCs (**E**vent **C**onnecte**d C**ommand). Dabei können mit einem Kommando bis zu 7 Datenbytes übertragen werden. Mit diesen Kommandos können auf der VME-Ebene (SE) EQMs *asynchron* (d. h. die Ausführung findet dann statt, wenn die SE nicht mit Real-Time-Aufgaben beschäftigt ist) gestartet werden.

Parameter: Keine

Daten: Die 1 bis 8 Worte, von denen jeweils nur das Low-Byte (Bit 0 ... 7) ausgewertet wird, haben folgende Bedeutung:

1: Kommandonummer (ECC-Nummer)

2 ... 8: 1 ... 7 Daten, die zum Kommando gehören (nur das Low-Byte wird ausgewertet).

2.2.10 CMSEQU

Bedeutung: Lesen bzw. Schreiben einer Folge von Submaschinen für den *Therapie-Test-Betrieb*.

Voraussetzung für die Ausführung dieser Submaschinenfolge ist, daß die Pulszentrale im *Therapie-Test-Betrieb* läuft (PZMODE = 1) und der Beschleuniger vom Cave-M aus angefordert wird.

Für jede Submaschine muß ein *execution count* (1 ... 255) angegeben werden.

Die Pulszentrale fährt diese Folge von Submaschinen selbständig ab und sorgt dabei auch für den richtigen *cycle count* (1 ... 255).

D. h. dann aber auch, daß die Gesamtlänge der Folge von Submaschinen (unter Berücksichtigung der *execution counts*) nicht länger als 255 sein darf? Oder lassen wir den *cycle count* überlaufen?

Soll diese Folge sofort ausgeführt werden, oder soll man sie explizit starten müssen?

Wie sollen wir denn die Wiederholung einer Submaschinenfolge implementieren?

Parameter: Das BitSet16 Datum gibt (nur beim Schreiben) die *MachineID* an, unter der die Submaschinenfolge ausgeführt werden soll.

Daten: Die 1 ... 509 BitSet16-Werte enthalten im Einzelnen:

- 1: Anzahl der Submaschinen in der Sequenz.
- 2 ... 5: Beschreibung der ersten Submaschine in der Sequenz:
 - Energiestufe** 1 ... 254
 - Fokussierungsstufe** 1 ... 7
 - Intensitätsstufe** 1 ... 15
 - Execution-Count** 1 ... 255
- 6 ... 9: Beschreibung der zweiten Submaschine in der Sequenz
- ⋮
- 1022 ... 1025: Beschreibung der 255-ten Submaschine in der Sequenz

2.2.11 SUPCYCLE

Bedeutung: Lesen bzw. Schreiben der Beschleunigerabfolge im Superzyklus.

Parameter: Keine

Daten: Die 1 bis 257 Worte haben folgende Bedeutung:

- 1: Anzahl Beschleuniger im Superzyklus.
- 2 ... 257: Nummern der Beschleuniger im Superzyklus.

2.2.12 PUSK

Bedeutung: Starten oder anhalten der Pulszentrale am Anfang des Superzyklus bzw. lesen, ob die Pulszentrale gestartet oder angehalten wurde.

Parameter: Keine

Daten: Die beiden Daten geben den Soll- und den Istwert an, wobei der Istwert beim Schreiben natürlich ignoriert wird.

- 1. Sollwert: 0 bedeutet **STOP**, 1 bedeutet **START**.
- 2. Istwert (nur beim Lesen): 0 bedeutet **ist gestoppt**, 1 bedeutet **ist gestartet**.

2.2.13 WAITNEXT

Bedeutung: Anhalten des Superzyklus im nächsten Gap oder Fortsetzen an der aktuellen Superzyklusposition, bzw. lesen, ob der Superzyklus angehalten oder fortgesetzt wurde.

Parameter: Keine

Daten: Die beiden Daten geben den Soll- und den Istwert an, wobei der Istwert beim Schreiben natürlich ignoriert wird.

- 1. Sollwert: 0 bedeutet **CONTINUE** an der aktuellen Position, 1 bedeutet **WAIT** im nächsten Gap,
- 2. Istwert (nur beim Lesen): 0 bedeutet **Superzyklus fortgesetzt**, 1 bedeutet **Superzyklus angehalten**.

2.2.14 INFO

Bedeutung: ??.

Parameter: Keine

Daten: Die 467 Werte haben folgende Bedeutung:

- 1: BufferNr** fortlaufende Nummer (0 ... 65535), die Änderungen im Superzyklus anzeigt.
- 2: Pusk.Soll** Siehe Property PUSK
- 3: Pusk.Ist** Siehe Property PUSK
- 4: Waitnext.Soll** Siehe Property WAITNEXT
- 5: Waitnext.Ist** Siehe Property WAITNEXT
- 6: PZMode:** Betriebsmodus der Pulszentrale (2: normal, 1: Test, 0: CaveM)
- 7: SupCyclePos** Aktuelle Position im Superzyklus.
- 8: ActVrtAcc** Nummer des aktuellen Beschleunigers im Superzyklus.
- 9: ActRequMode** Anforderungsmodus des aktuellen Beschleunigers (0: asynchron, 1: synchron).
- 10: ActState** Ausführungsstatus des Beschleunigers (0: läuft nicht, 1: läuft)
- 11: AccCount** Anzahl der Beschleuniger im Superzyklus.
- 12 ... 267: SupCycle** aktueller Superzyklus. Siehe Property SUPCYCLE
- 268: MediInfo** Zustandsinformationen zum Therapiebetrieb:

Bit	Bedeutung
0	Pulszentrale ist verriegelt (1) oder nicht (0)
1	Therapie-Test-Sequenz läuft (1) oder nicht (0)
2 ... 15	unbenutzt

269: Reserve

267 ... 394: AccInfo Beschreibung der Beschleuniger 0 ... 15, bestehend aus folgenden Daten (pro Beschleuniger):

- ExecMode** Ausführungsmodus (1: aktiv, 0: inaktiv)
- ExecCnt.Soll** Anzahl der Ausführungen in Folge (1 ... 255).
- ExecCnt.Ist** Anzahl der Ausführungen in Folge (1 ... 255).
- Request.Soll** Anforderungsmodus (0: frei laufend, 1: synchron, 2: asynchron)
- Request.Ist** Aktueller Zustand der Anfordereinheit (1: Beschleuniger ist angefordert, 0: Beschleuniger ist *nicht* angefordert). *Frei laufend* deklarierte Beschleuniger werden immer als *angefordert* dargestellt.
- InSupCycle** Beschleuniger ist im Superzyklus (1: Ja, 0: Nein)
- Executable** Ausführbarkeit des Beschleunigers (1: ausführbar, 0: Beschleuniger ist verriegelt)
- ExecBlock** siehe Property EXECBLCK
- ExecMask** Ausführbarkeitsmaske des Beschleunigers. Bit 0 hat eine spezielle Summenfunktion. Es ist nur dann *Eins*, wenn alle anderen Bits auch *Eins* sind. Zur einfachen Überprüfung der Ausführbarkeit eines Beschleunigers ist es demnach ausreichend Bit 0 zu überprüfen (ODD: ausführbar, EVEN: nicht ausführbar). Will man den Grund für die *Nichtausführbarkeit* wissen muß, man die restlichen Bits auswerten. Die Bits haben folgende Bedeutung:

Bit	Bezeichnung	Bedeutung
0	enabled	Beschleuniger ist ausführbar (Summenmeldung)
1	exec_blk_ok	Beschleuniger nicht blockiert
2	exec_mod_ok	Beschleuniger ist active
3	exec_cnt_ok	Execution-Count \neq 0
4	control_ok	Beschleuniger nicht verriegelt
5	InSupCyc_ok	Beschleuniger ist im Superzyklus
6	ESRReady_ok	Anforderung zur Reinjektion liegt vor
7	VM0Block_ok	Beschl. nicht ausführbar wegen VM0Block (Therapie)
8	MedLock_ok	Beschl. nicht ausführbar wegen Lock (Therapie)
9	MedTest_ok	HKR-Therapietest-Sequenz beendet oder leer
10	MedAcc_ok	Therapiebeschleuniger ist ausführbar
11	UniMed_ok	Unilacbestätigung für Therapieanforderung liegt vor
12	MedReq_ok	Therapieanforderung von Cave-M liegt vor
13	EvtData_ok	Kanalfolge und Eventsequenzen sind ok
14	MedMode_ok	Unilac nimmt am Therapiebetrieb teil
15	trans_blk_ok	Transmissionsüberwachung blockiert Beschleuniger nicht

ExecStat Ausführungsstatistik des Beschleunigers (Anzahl Ausführungen in der laufenden Schicht)

TransBlock siehe Property EXECBLCK

Reserve1

462 ... 467: LastEFICD aktuelle Therapieanforderung:

Energy Energiestufe (0 ... 254).

Focus Fokussierungsstufe (0 ... 7).

Intensity Intensitätsstufe (0 ... 15).

CycleCount Zykluscounter der aktuellen Bestrahlungssession (0 ... 255).

MachineID Maschinen-Datenkennung (0 ... 65535).

Device/Set-ID Geräte- (High-Byte) und Set-Datenkennung (Low-Byte).

2.2.15 THINFO

Bedeutung: Lesen aller für die Therapie relevanten Daten der Pulszentrale.

Parameter: Keine.

Daten: die 15 Daten haben folgende Bedeutung:

1: Pusk.Ist Siehe Property PUSK

2: Waitnext.Ist Siehe Property WAITNEXT

3: ExecMode Ausführungsmodus (1: aktiv, 0:inaktiv)

4: Request.Ist Aktueller Zustand der Anfordereinheit (1: Beschleuniger ist angefordert, 0: Beschleuniger ist *nicht* angefordert). *Frei laufend* deklarierte Beschleuniger werden immer als *angefordert* dargestellt.

5: InSupCycle Beschleuniger ist im Superzyklus (1: Ja, 0: Nein)

6: Security Verriegelungszustand der Pulszentrale
(1: verriegelt, 0: nicht verriegelt)

7: PZMode Betriebsmodus der Pulszentrale (2: normal, 1: Test, 0: CaveM)

- 8: ExecMask** Ausführbarkeitsmaske des Beschleunigers. Bit 0 hat eine spezielle Summenfunktion. Es ist nur dann *Eins*, wenn alle anderen Bits auch *Eins* sind. Zur einfachen Überprüfung der Ausführbarkeit eines Beschleunigers ist es demnach ausreichend Bit 0 zu überprüfen (ODD: ausführbar, EVEN: nicht ausführbar). Will man den Grund für die *Nichtausführbarkeit* wissen muß, man die restlichen Bits auswerten. Die Bits haben folgende Bedeutung:

Bit	Bezeichnung	Bedeutung
0	enabled	Beschleuniger ist ausführbar (Summenmeldung)
1	exec_blk_ok	Beschleuniger nicht blockiert
2	exec_mod_ok	Beschleuniger ist active
3	exec_cnt_ok	Execution-Count > 0
4	control_ok	Beschleuniger nicht verriegelt
5	InSupCyc_ok	Beschleuniger ist im Superzyklus
6	ESRReady_ok	Anforderung zur Reinjektion liegt vor
7	VM0Block_ok	Beschl. nicht ausführbar wegen VM0Block (Therapie)
8	MedLock_ok	Beschl. nicht ausführbar wegen Lock (Therapie)
9	MedTest_ok	HKR-Therapietest-Sequenz beendet oder leer
10	MedAcc_ok	Therapiebeschleuniger ist ausführbar
11	UniMed_ok	Unilacbestätigung für Therapieanforderung liegt vor
12	MedReq_ok	Therapieanforderung von Cave-M liegt vor
13	EvtData_ok	Kanalfolge und Eventsequenzen sind ok
14	MedMode_ok	Unilac nimmt am Therapiebetrieb teil
15	trans_blk_ok	Transmissionüberwachung blockiert Beschleuniger nicht

- 9 ... 14: LastEFICD** aktuelle Therapieanforderung:

Energy Energiestufe (0 ... 254).

Focus Fokussierungsstufe (0 ... 7).

Intensity Intensitätsstufe (0 ... 15).

CycleCount Zykluscounter der aktuellen Bestrahlungssession (0 ... 255).

MachineID Maschinen-Datenkennung (0 ... 65535).

Device/Set-ID Geräte- (High-Byte) und Set-Datenkennung (Low-Byte).

- 15: CMCntrl** Control-Bytes (Input und Output) der Therapieanfordereinheit. Die Bits haben folgende Bedeutung:

Bit	In/Out	Bezeichnung	Bedeutung
0	In	SessEnd	Anzeige: Bestrahlungsplanwechsel
1	In	—	keine (nur Output)
2	In	—	keine (nur Output)
3	In	MngEIN	Anforderung: Netzgeräte einschalten
4	In	VM0Block	Anforderung: nur Beschl. 0 darf laufen
5	In	DTAVal	Daten geprüft, sind ok
6	In	DTAval	Daten liegen an
7	In	Lock	Anforderung: Beschleuniger verriegeln
8	Out	SessEnd_Ack	Bestätigung des Bestrahlungsplanwechsel
9	Out	ReqEnabl	Therapiebetrieb zulässig (PZMode = 0)
10	Out	AccReady	Therapiebschl. ist bereit
11	Out	MngEIN_Ack	Bestätigung: Netzgeräte wurden eingeschaltet
12	Out	VM0Block_Ack	Bestätigung: nur Beschl. 0 darf laufen
13	Out	DTAVal_Ack	Bestätigung: Daten geprüft, sind ok
14	Out	DTAval_Ack	Bestätigung: Daten übernommen und gespiegelt
15	Out	Lock_Ack	Bestätigung: PZ ist verriegelt

2.2.16 RIEGTAB

Bedeutung: Lesen der Verriegelungseingänge in der Gruppierung ZKS, Vakuum, Magnetnetzgeräte und Software. Dabei sind für jede Gruppe 96 Bit vorgesehen, wovon jedes einem Strahlwegabschnitt entspricht.

Bit	ZKS	Vakuum	Magnete
0	NE1-0	S06VV1T	Heag Spitzenlast-Anzeige
1	NE1-1	TE2VV1T	–
2	NE2-0	TE5VV1S	–
3	NE2-1	TR1VV1T	–
4	NE3-0	TS1VV1T	–
5	NE3-1	HHDVV1T	–
6	NE2-6	TE5VV2T	–
7	NE4-0	TS3VV1T	–
8	NE4-1	TS5VV1T	–
9	NE2-7	TR1VV2T	–
10	NE5-1	TH2VV1T	–
11	NE5-2	TE3VV1T	–
12	NE5-3	TE4VV2T	–
13	NE5-4	TS7VV1T	–
14	NE5-5	TH3VV1T	–
15	NE5-6	TH4VV1S	–
16	NE6-0	HHTVV1T	–
17	NE7-0	HFSVV1T	–
18	NE8-0	TH4VV2T	–
19	NE8-1	TT1VV1T	–
20	NE8-17	TT1VV1S	–
21	NE2-8	TR1VV3T	–
22	NE8-4	HTTVV1T	–
23	NE8-5	TV2VV1T	–
24	NE2-9	S11VV2T	–
25	NE2-2	E02VV1T	–
26	NE9-0	HTCVV1T	–
27	NE10-0	HTAVV1T	–
28	NE11-0	HTBVV1T	–
29	NE2-3	E01VV2T	–
30	NE2-4	E02VV2T	–
31	NE3-5	E01VV1T	–
32	NE5-10	HADV1T	–
33	NE1-8	TE1VV1S	–
34	NE12-0	–	–
35	NE5-8	HADV1S	–
36	NE13-0	HADV2T	–
37	NE5-9	TP1VV1T	–
38	NE8-12	–	–
39	NE14-0	TK9VV1T	–
40	NE14-1	TK8VV1T	–
41	NE1-2	S05VV1T	–
42	NE1-3	S01VV1T	–

Forts. auf nächster Seite

Forts. von letzter Seite

Bit	ZKS	Vakuum	Magnete
43	NE1-4	S11VV1T	–
44	NE1-5	S06VV2T	–
45	NE3-3	TS1VV2T	–
46	NE3-4	TS2VV2T	–
47	NE4-2	TS4VV1T	–
48	NE7-1	TS4VV2T	–
49	NE6-1	HHTVV2T	–
50	NE8-13	TT1VV2T	–
51	NE8-14	TT1VV3T	–
52	NE9-1	HTCVV2T	–
53	NE11-1	HTBVV2T	–
54	NE12-1	HTMVV2T	–
55	NE13-1	HADV3T	–
56	NE5-11	TP1VV2T	–
57	NE1-9	S10VV1T	–
58	NE8-18	HTMVV1T	–
59	NE3-5	–	–
60	NE5-12	–	–
61	–	–	–
62	–	–	–
63	–	–	–
64	–	–	–
⋮	⋮	⋮	⋮
95	–	–	–

Nicht belegte Eingänge sind auf *ok* verdrahtet.

Parameter: Keine

Daten: Die 12 BitSet32 Werte haben folgende Bedeutung:

1. ZKS-Eingänge (Bit 64 ... 95)
2. ZKS-Eingänge (Bit 32 ... 63)
3. ZKS-Eingänge (Bit 0 ... 31)
4. VAC-Eingänge (Bit 64 ... 95)
5. VAC-Eingänge (Bit 32 ... 63)
6. VAC-Eingänge (Bit 0 ... 31)
7. MNG-Eingänge (Bit 64 ... 95)
8. MNG-Eingänge (Bit 32 ... 63)
9. MNG-Eingänge (Bit 0 ... 31)
10. SW-Verriegelung (Bit 64 ... 95)
11. SW-Verriegelung (Bit 32 ... 63)
12. SW-Verriegelung (Bit 0 ... 31)

Der Wert (0 oder 1) der jeweils 96 Bit gibt an, ob im entsprechenden Strahlwegabschnitt eine Verriegelung ansteht (Bit = 0) oder nicht (Bit = 1).

2.2.17 SWRGTAB

Bedeutung: Lesen bzw. Setzen der Softwareverriegelung. Die Belegung der 96 Bit ist identisch mit den Verriegelungseingängen.

Parameter: Keine

Daten: Die 3 BitSet32 Werte haben folgende Bedeutung:

1. SW-Verriegelung (Bit 64 ... 95)
2. SW-Verriegelung (Bit 32 ... 63)
3. SW-Verriegelung (Bit 0 ... 31)

Der Wert (0 oder 1) der 96 Bit gibt an, ob die Verriegelungsinformationen des entsprechenden Strahlwegabschnitts ausgewertet werden (Bit = 1) oder nicht (Bit = 0).

2.2.18 PZMODE

Bedeutung: Lesen bzw. Setzen des Betriebsmodus der Pulszentrale. Die derzeit möglichen Betriebsmoden sind:

- Therapiebetrieb (die Submaschinen im Therapiebeschleuniger werden von Cave-M aus angefordert).
- Therapie-Test-Betrieb (die Folge der Submaschinen im Therapiebeschleuniger wird vom Operating vorgegeben). Hat das irgendwelche Konsequenzen für den Superzyklus?
- Normalbetrieb (alle Beschleuniger stehen frei zur Verfügung und werden entsprechend dem Superzyklus ausgeführt).

?

Die ESR-Pulszentrale verfügt nur über den Betriebsmodus *Normalbetrieb* und kann deshalb auch nicht umgeschaltet werden.

Parameter: Keine

Daten: Das Datum kann folgende 3 Werte annehmen:

- 0:** *Therapiebetrieb* (nur SIS)
- 1:** *Therapie-Test-Betrieb* (nur SIS)
- 2:** *Normalbetrieb*

2.3 Die Slave-Properties

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	0	–	348	Integer32	1	0

Forts. auf nächster Seite

Forts. von letzter Seite

Property	Klasse	Parameter		Daten		Größe	
		Anz.	Typ	Anz.	Typ	Einh.	Exp.
ACCVIA	R/W	0	–	1	Integer16	1	0
STRALWEG	RA/WA	0	–	2	BitSet32	1	0
EVTARRAY	RA/WA	0	–	784	Integer32	1	0
EVTSEQU	RA/WA	1	BitSet16	49	Integer32	1	0
CHANSEQU	RA/WA	0	–	1281	Integer32	1	0
EXEC CNT	R/W	0	–	1	BitSet16	1	0
EXEC MODE	R/W	0	–	1	BitSet16	1	0
EXEC REQ	R/W	0	–	1	Integer16	1	0
ACCDESCR	RA/WA	0	–	6	Integer16	1	0
ACCREQ	RA/WA	0	–	7	Integer32	1	0
RIEGMASK	RA/WA	0	–	4	BitSet16	1	0
GAPNOPOS	R/W	1	Integer16	1	BitSet16	1	0
EXECBLCK	RA/WA	0	–	2	Integer16	1	0

2.3.1 ACTIV

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

Das Gerät SIS/ESR-Pulszentrale nimmt immer für alle Beschleuniger an der PPM teil. Ein Versuch ACTIV zu Schreiben führt zu einer entsprechenden Fehlermeldung.

Parameter: Keine.

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

2.3.2 COPYSET

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

Parameter: Keine.

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

2.3.3 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: Keine.

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

2.3.4 ACCVIA

Bedeutung: Setzen bzw. Lesen des *Strahlwegkey*. Der *Strahlwegkey* beschreibt den Weg, über den ein Beschleuniger von der Quelle zum Ziel kommt. Die Pulszentrale identifiziert anhand des *Strahlwegkey* das Strahlziel (i.d.R. ein Experimentierplatz) und ermittelt daraus die Nummer (und Adresse) der entsprechenden Anfordereinheit und belegt an der Anfordereinheit den nächsten freien Eingang mit diesem virtuellen Beschleuniger.

Der *Strahlwegkey* unterliegt folgenden Konventionen:

- jede Herkunft-Via Kombination ist eindeutig in einer 2-stelligen Zahl kodiert.
- jedes mögliche Strahlziel ist eindeutig in einer 3-stelligen Zahl kodiert.
- der gesamte Strahlweg ergibt sich aus der Kombination der beiden Zahlen und ist damit 5-stellig. Wobei Herkunft-Via in den beiden höherwertigen Dezimalstellen und das Strahlziel in den 3 niederwertigen Dezimalstellen untergebracht ist.

Der *Strahlwegkey* 0 ist zur Initialisierung eines virtuellen Beschleunigers explizit zugelassen. Dabei wird der bisher belegte Anforderkanal des Beschleunigers freigegeben und der Beschleuniger als *nicht ausführbar* gekennzeichnet.

Parameter: Keine.

Daten: Das Datum kann folgende Werte annehmen:

Herkunft-Via: Die an SIS und ESR möglichen Herkunft-Via-Kombinationen sind folgendermaßen kodiert:

Herkunft-Via	Wert
SIS-TS	20
SIS-TE	21
SIS-TH	22
SIS-TP	23
SIS-SIS	24
ESR-TT	30
ESR-TR	31
ESR-ESR	32

Ziel: Die möglichen Strahlziele für SIS und ESR sind folgendermaßen kodiert:

Ziel	Wert
HAD	600
HHT	610
HTA	620
HTB	630
HTC	640
HTM	650
HTP	660
HHD	670
HFS	680
ESR	690
SIS	700

Beispiele für gültige Strahlwegkeys:

Strahlweg	Herkunft-Via	Ziel	Strahlwegkey
SIS-TH-HTM	22	650	22650
ESR-TR-SIS	31	700	31700

2.3.5 STRALWEG

Bedeutung: Lesen bzw. Setzen der Strahlweginformationen eines Beschleunigers. Die Belegung der 64 Bit ist identisch mit den Verriegelungseingängen.

Parameter: Keine

Daten: Die 2 BitSet32 Werte haben folgende Bedeutung:

1. Strahlweg-Info (Bit 32 ... 63)
2. Strahlweg-Info (Bit 0 ... 31)

2.3.6 EVTARRAY

Bedeutung: Setzen bzw. Lesen der Eventfolgen aller Kanäle. Dabei gelten folgende Vereinbarungen:

- Die maximale zeitliche Länge einer Eventfolge beträgt 16.777215 s (Zeitauflösung $1\mu s$, Datenbreite 24 Bit).
- Die Events einer Folge müssen nicht in ihrer zeitlichen Reihenfolge aufsteigend sortiert angegeben werden. Sie werden auf der VME-Ebene entsprechend umsortiert.
- Wird der Mindesteventabstand zwischen 2 Events (derzeit $30\mu s$) nicht eingehalten, werden alle Eventsequenzen mit einer entsprechenden Fehlermeldung abgewiesen.

- Jede Eventfolge ist mit einer Endekennung (Eventcode 0 mit entsprechender Ausführungszeit) abzuschließen. Wird die Endekennung nicht explizit angegeben, wird sie auf der VME-Ebene selbständig ergänzt ($30\mu s$ nach dem letzten Event der Folge, was der Übertragungszeit des letzten Events über den Eventbus entspricht).

Parameter: Keine.

Daten: Die 1 ... 784 Langworte enthalten im Einzelnen:

1 ... 49: Eventfolge des ersten Kanals:

1: Anzahl der Events im Kanal.

2: Eventcode des ersten Events im Kanal.

3: Ausführungszeit des ersten Events im Kanal (in μs ab Kanalstart).

4: Eventcode des zweiten Events im Kanal.

5: Ausführungszeit des zweiten Events im Kanal (in μs ab Kanalstart).

⋮

48: Eventcode des 24-ten Events im Kanal.

49: Ausführungszeit des 24-ten Events im Kanal (in μs ab Kanalstart).

50 ... 98: Eventfolge des zweiten Kanals

⋮

736 ... 784: Eventfolge des 16-ten Kanals

2.3.7 EVTSEQU

Bedeutung: Setzen bzw. Lesen der Eventfolge eines einzelnen Kanals. Hierbei gelten die gleichen Randbedingungen wie bei EVTARRAY.

Parameter: Das Datum gibt die Kanalnummer (1 ... 16) an.

Daten: Die 1 ... 49 Langworte enthalten im Einzelnen:

1: Anzahl der Events im Kanal.

2: Eventcode des ersten Events im Kanal.

3: Ausführungszeit des ersten Events im Kanal (in μs ab Kanalstart).

4: Eventcode des zweiten Events im Kanal.

5: Ausführungszeit des zweiten Events im Kanal (in μs ab Kanalstart).

⋮

48: Eventcode des 24-ten Events im Kanal.

49: Ausführungszeit des 24-ten Events im Kanal (in μs ab Kanalstart).

2.3.8 CHANSEQU

Bedeutung: Setzen bzw. Lesen der Kanalfolge eines Beschleunigers. Die Kanalfolge legt fest in welcher Reihenfolge die Kanäle (1 ... 16) ausgeführt werden sollen. Dabei sind für jedes Element in der Folge jeweils die Kanalnummer und die Startbedingungen anzugeben.

Die möglichen Startbedingungen sind:

sequence: Die Ausführung des Kanals startet ohne besondere Randbedingungen mit dem Ende des vorhergehenden Kanals.

trigger-ext-timeout: Der Kanal wird durch einen externen Trigger gestartet. Im Falle eines *Timeout* wird die Kanalfolge mit einem *Timeoutkanal* terminiert.

trigger-ext-forced: Der Kanal wird durch einen externen Trigger gestartet. Im Falle eines *Timeout* wird die Kanalfolge trotzdem fortgesetzt.

trigger-ext-maxtime: Der Kanal wird durch einen externen Trigger oder nach Ablauf einer Maximalzeit gestartet.

Parameter: Keine

Daten: Die 1 ... 1281 BitSet16-Werte enthalten im Einzelnen:

1: Anzahl der Kanäle in der Sequenz (0 ... 256).

2 ... 6: Beschreibung des ersten Kanals in der Sequenz

Kanalnummer: Nummer des Kanals (1 ... 16).

Startbedingung: Startbedingung für den Kanal. Es sind die folgenden Startbedingungen definiert:

1: *sequence* (Der Kanal wird unmittelbar nach dem vorhergehenden Kanal gestartet)

2: *trigger-ext-timeout* (Der Kanal wird durch einen externen Trigger gestartet, im Falle eines *Timeout* wird zu einem *Timeoutkanal* verzweigt)

3: *trigger-ext-forced* (Der Kanal wird durch einen externen Trigger gestartet, im Falle eines *Timeout* wird der Kanal trotzdem gestartet)

4: *trigger-ext-maxtime* (Der Kanal wird durch einen externen Trigger oder nach Ablauf einer Maximalzeit gestartet)

Primärevent: Event, mit dem der Kanal getriggert wird (die Hardwareeingänge der Pulszentrale werden auf Eventcodes abgebildet). Dieser Wert wird nur bei den Startbedingungen *trigger-timeout*, *trigger-forced* und *trigger-ext-maxtime* ausgewertet.

Timeout: Maximale Zeit (1 ... 65535 ms), in der das Triggerereignis stattfinden muß. Dieser Wert wird nur bei den Startbedingungen *trigger-timeout* und *trigger-forced* ausgewertet. In der Startbedingung *trigger-ext-maxtime* bedeutet er die maximale Wartezeit bis zum Start des Kanals.

Timeoutkanal: Nummer des *Timeoutkanals*, zu dem im Falle eines *Timeouts* verzweigt wird. Dieser Wert wird nur bei Startbedingung *trigger-timeout* ausgewertet.

7 ... 11: Beschreibung des zweiten Kanals in der Sequenz.

⋮

1277 ... 1281: Beschreibung des 256-ten Kanals in der Sequenz.

2.3.9 EXECCNT

Bedeutung: Setzen bzw. Lesen der Anzahl von Ausführungen n eines virtuellen Beschleunigers. D. h. immer wenn ein virtueller Beschleuniger im Superzyklus zur Ausführung kommt, wird er n mal ausgeführt. Vorausgesetzt der Ausführung stehen keine Hinderungsgründe (z. B. Verriegelung, ...) im Wege.

Parameter: Keine.

Daten: Das Datum gibt an, wie oft ein Beschleuniger in Folge ausgeführt wird, wenn er im Superzyklus an der Reihe ist. Dabei können Werte von 0 bis 255 angegeben werden.

2.3.10 EXECMODE

Bedeutung: Gibt an, ob die Pulszentrale den zugehörigen virtuellen Beschleuniger zur Ausführung bringt oder nicht.

Parameter: Keine.

Daten: Das Datum kann nur zwei Werte annehmen. Null heißt, der Beschleuniger wird *nicht* ausgeführt bzw. soll *nicht* ausgeführt werden. Eins heißt, der Beschleuniger wird ausgeführt bzw. soll ausgeführt werden.

2.3.11 EXECREQU

Bedeutung: Beschreibung der Strahlanforderung durch einen Abnehmer (Experimentierplatz). Die Strahlanforderung durch den Experimentierplatz kann *synchron* oder *asynchron* erfolgen.

synchron: Der virtuelle Beschleuniger kann nur dann angefordert werden, wenn er im Superzyklus an der Reihe ist.

asynchron: Der virtuelle Beschleuniger wird nach einer Anforderung so bald wie möglich ausgeführt. Und zwar unabhängig von seiner Position im Superzyklus.

Parameter: Keine.

Daten: Das Datum kann nur zwei Werte annehmen. Null heißt, der Beschleuniger wird *asynchron* angefordert bzw. soll *asynchron* angefordert werden. Eins heißt, der Beschleuniger wird *synchron* angefordert bzw. soll *synchron* angefordert werden.

2.3.12 ACCDESCR

Bedeutung: Lesen der wichtigsten quasistatischen Einstellungen eines virtuellen Beschleunigers.

Parameter: Keine.

Daten: Die Bedeutung der einzelnen Elemente ist wie folgt:

1. Strahlwegkey (siehe Property ACCVIA).
2. Nummer des Strahllieferanten (siehe Property ACCREQU).
3. Nummer des jeweils anderen virtuellen Beschleunigers der von dieser Pulszentrale angefordert wird (für SIS-Pulszentrale: Unilac oder ESR, für ESR-Pulszentrale: SIS).
4. Anfordermodus (siehe Property ACCREQU)
5. Nummer der dem virtuellen Beschleuniger zugeordneten Anfordereinheit. Keine Anfordereinheit zugeordnet: -1.
6. Nummer des dem virtuellen Beschleuniger zugeordneten Anfordereingangs (auf der im vorigen Datum angegebenen Anfordereinheit). Keine Anfordereinheit zugeordnet: -1.

2.3.13 ACCREQU

Bedeutung: Lesen bzw. Setzen des virtuellen Beschleunigers eines Lieferbeschleunigers (SIS-Pulszentrale: Unilac oder ESR, ESR-Pulszentrale: SIS), der aus dem aktuellen virtuellen Beschleuniger angefordert wird.

In der ESR Pulszentrale erfolgt die Anforderung und die Synchronisation auf Superzyklus-Ebene. Dabei auftretende Fehler (z. B. angeforderter virtueller Beschleuniger nicht im Superzyklus; Time Out, d. h. angeforderter virtueller Beschleuniger wurde nicht rechtzeitig ausgeführt) können behandelt werden. Im Fehlerfall kann entweder im Superzyklus fortgefahren

werden, oder es wird *ein* spezieller virtueller Beschleuniger ausgeführt (»Fehlerbeschleuniger«) und die Abarbeitung des Superzyklus wieder am Beginn desselben gestartet.

Wie die Fehlerbehandlung erfolgen soll, ist in der Property anzugeben. Die Informationen zur Fehlerbehandlung werden nur in der ESR-Pulszentrale ausgewertet.

Parameter: Keine.

Daten: Die einzelnen Werte haben folgende Bedeutung:

1. Nummer des Strahllieferanten. Also von welchem Beschleuniger muß der Strahl angefordert werden.
 - 0:** Kein Lieferant. Alle weiteren Daten dieser Property werden nicht ausgewertet und können weggelassen werden. Im SIS bedeutet das, der Beschleuniger läuft *ohne Strahl*. Im ESR kann er trotzdem *Strahl* haben. Generell gilt aber:
 - Ein Beschleuniger, der keinen Lieferanten hat, wird im Superzyklus auch ohne Anforderung ausgeführt.
 - Alle anderen Beschleuniger (also mit Lieferant) werden nur auf Anforderung ausgeführt.
 - 1:** UNILAC
 - 2:** SIS
 - 3:** ESR
2. Nummer des virtuellen Beschleunigers der jeweils vorgeschalteten Beschleunigeranlage, der angefordert werden soll (Lieferbeschleuniger). Ausgewertet bei SIS und ESR.
3. Timeout-Zeit für die Anforderung. Erfolgt innerhalb dieser Timeout-Zeit keine Anforderung, wird das Warten auf eine Anforderung abgebrochen und die (im folgenden) angegebene Fehlerbehandlung ausgeführt.
4. Ablauf der primären Fehlerbehandlung. Die primäre Fehlerbehandlung wird ausgeführt, wenn es bei der *ersten* Ausführung eines virtuellen Beschleunigers im Superzyklus zu einem Fehler bei der Anforderung kommt.
 - 0:** Bei einem Fehler bei der Anforderung weitermachen im Superzyklus.
 - 1:** Bei einem Fehler bei der Anforderung den angegebenen Fehlerbeschleuniger ausführen und danach zum Beginn des Superzyklus zurückspringen.
 - 2:** Bei einem Fehler bei der Anforderung sofort an den Beginn des Superzyklus zurückspringen.
5. Nummer des Fehlerbeschleunigers, der bei einem Fehler bei der Anforderung ausgeführt werden soll.

Nur ausgewertet, wenn für den Ablauf der primären Fehlerbehandlung »Abbruch« (1) angegeben wurde.
6. Ablauf der sekundären Fehlerbehandlung. Die sekundäre Fehlerbehandlung wird ausgeführt, wenn es bei der *zweiten* oder folgenden Ausführung eines virtuellen Beschleunigers im Superzyklus zu einem Fehler bei der Anforderung kommt. Ausgewertet nur bei der ESR-Pulszentrale.
 - 0:** Bei einem Fehler bei der Anforderung weitermachen im Superzyklus.
 - 1:** Bei einem Fehler bei der Anforderung den angegebenen Fehlerbeschleuniger ausführen und danach zum Beginn des Superzyklus zurückspringen.
 - 2:** Bei einem Fehler bei der Anforderung sofort an den Beginn des Superzyklus zurückspringen.

7. Nummer des Fehlerbeschleunigers, der bei einem Fehler bei der Anforderung ausgeführt werden soll.

Wird nur ausgewertet, wenn für den Ablauf der sekundären Fehlerbehandlung »Abbruch« (1) angegeben wurde.

Wenn keine sekundäre Fehlerbehandlung angegeben wurde, wird die gleiche Fehlerbehandlung (mit demselben Fehlerbeschleuniger) wie bei der primären Fehlerbehandlung ausgeführt.

2.3.14 RIEGMASK

Bedeutung: Lesen bzw. Setzen, welche Verriegelungseingänge für einen Beschleuniger ausgewertet werden.

Parameter: Keine.

Daten: Die 4 Werte geben an, ob die entsprechenden Verriegelungsinformationen ausgewertet werden sollen oder nicht.

1. ZKS (1: wird ausgewertet, 0: wird nicht ausgewertet)
2. Vakuum (1: wird ausgewertet, 0: wird nicht ausgewertet)
3. Interlock der Magnetnetzgeräte (1: wird ausgewertet, 0: wird nicht ausgewertet)
4. Softwareverriegelung (1: wird ausgewertet, 0: wird nicht ausgewertet)

2.3.15 GAPNOPOS

Bedeutung: Setzen bzw. Lesen eines der folgenden Gap-Events im nächsten Gap vor dem virtuellen Beschleuniger:

- EVT_Gap_Pos_Mess
- EVT_Gap_Tra_Mess
- EVT_Gap_Scr_Mess
- EVT_Gap_DTS_Mess

D.h. bevor der Beschleuniger zum nächsten mal ausgeführt wird, wird im davorliegenden Gap das angegebene Gap-Event verschickt.

Über den optionalen Parameter können die Bedingungen, die an den Beschleuniger gestellt werden damit das Gap-Event vorgemerkt wird, ausgewählt werden.

Beim Lesen muß das zu lesende Gap-Event angegeben werden.

Parameter: 1 Integer16-Wert, der beim Lesen und Schreiben unterschiedliche Bedeutung hat:

Schreiben: Der Wert gibt an, welche Bedingungen der Beschleuniger zu erfüllen hat, damit das Gap-Event vorgemerkt wird.

Wert	Bedeutung
1	Beschleuniger muß ausführbar sein
0	Beschleuniger muß ausführbar und aktuell angefordert sein
-1	Gap-Event löschen (falls vorhanden)

Lesen: Das Datum gibt den Eventcode des zu prüfenden Gap-Events an. Ist das Event eingetragen, wird dessen Eventcode ansonsten wird 0 und eine PZS_GAPEVENT_NOT_FOUND (Information) als /emphcompletion-status geliefert.

Daten: Das Datum gibt den Eventcode des gewünschten Gap-Events an. Folgende Eventcodes sind zulässig:

Gap-Event	Eventcode (dezimal)
EVT_Gap_Pos_Mess	96
EVT_Gap_Tra_Mess	97
EVT_Gap_Scr_Mess	98
EVT_Gap_DTS_Mess	99

2.3.16 EXECBLCK

Bedeutung: Setzen bzw. Lesen der Ausführbarkeit eines Beschleunigers. Folgende Parameter sind bisher durch die Operating-Ebene einflußbar:

- Datenversorgung eines Beschleunigers ist konsistent. Erst wenn nach Änderungen im INIT oder im (SISMODI) von den Operatingprogrammen (SISMODI) alle notwendigen Vorbereitungen für die Ausführung eines Beschleunigers (Versorgung der benötigten Geräte mit Sollwerten, ...) fehlerfrei durchgeführt werden konnten, wird die Ausführung des Beschleunigers freigegeben.
- Transmissionsüberwachung bei Hochstrom. Nur wenn die Transmissionsüberwachung eine ausreichende Transmission bei der Injektion ins SIS feststellt, bleibt der Beschleuniger ausführbar. Wird die Transmissionschwelle unterschritten wird vom Überwachungsprogramm der Beschleuniger als nicht ausführbar gesetzt.

Parameter: Keine.

Daten: Die beiden Daten bedeuten:

1. *ExecBlock*: Konsistenz der Datenversorgung.
2. *TransBlock*: Transmissionsüberwachung bei Hochstrom.

und können folgende Werte annehmen:

- 0 Beschleuniger ist freigegeben bzw. soll freigegeben werden.
- 1 Beschleuniger ist *nicht* freigegeben bzw. soll *nicht* freigegeben werden.
- 1 (nur beim Schreiben) Wert wird nicht geändert.

Teil II

Die Realisierung in Hard- und Software

3 Die Hardware der SIS/ESR-Pulszentrale

Die Hardware der SIS/ESR-Pulszentrale besteht soweit wie möglich aus Standard-Komponenten, die bei GSI auch an anderen Stellen zum Einsatz kommen.

Die folgende Abbildung soll einen Überblick darüber vermitteln, wie die Schnittstellen der SIS/ESR-Pulszentrale in etwa aussehen werden.

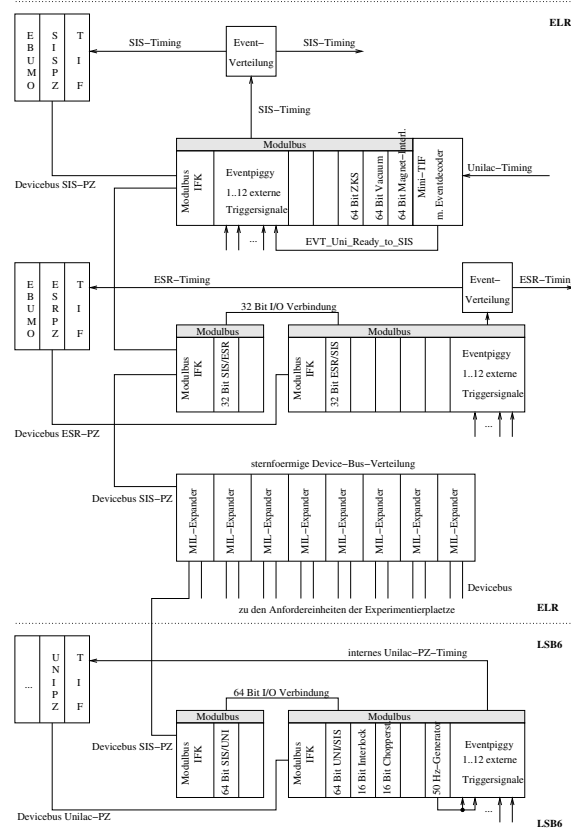


Abbildung 1: Die Schnittstellen der SIS-Pulszentrale

Bezüglich der Schnittstellen muß zwischen SIS- und ESR-Pulszentrale unterschieden werden.

3.1 Die Schnittstellen der SIS-Pulszentrale

Die SIS-Pulszentrale verfügt über folgende Schnittstellen:

- Datenaustausch mit der ESR-Pulszentrale (Daten zum Strahlaustausch, sowohl für Extraktion als auch Reinjektion).
- Bis zu 16 Anfordereinheiten an allen Experimentierplätzen.

Forts. von letzter Seite

I/O-Modul	Byte	I/O-Richtung	Bit	Bedeutung
S2/U2	1	SIS→Unilac	0 ... 7	MachineID Low-Byte
	2	SIS→Unilac	0 ... 7	MachineID High-Byte
	3	SIS→Unilac	0 ... 7	Zyklusnummer
S3/U3	1	SIS→Unilac	0 ... 3	Intensitätsstufe
			4 ... 7	Fokussierungsstufe
	2	SIS→Unilac	0 ... 7	Energiestufe
	3	—	0 ... 7	nicht belegt
U4	1	input	0 ... 7	Status PG-Abschnitte
			0	x: Anforderung X-Zweig
	2	input	1	y: Anforderung Y-Zweig
			2	z: Anforderung Z-Zweig
			3	m: M-Chopper: Kein Strahl
			4	n: N-Chopper: Kein Strahl
			5 ... 7	p1, p2, p3: nicht belegt
3	output	0 ... 7	Ansteuerung Interlocksystem	

Belegung I/O-Modul U4		
Byte	Bit	Bedeutung
1	0	PG in Abschnitt UN eingefahren
	1	PG in Abschnitt WS eingefahren
	2	PG in Abschnitt AT eingefahren
	3	PG in Abschnitt UT eingefahren
	4	PG in Abschnitt UX eingefahren
	5	PG in Abschnitt UY eingefahren
	6	PG in Abschnitt UZ eingefahren
2	7	PG in Abschnitt TK eingefahren
	0	Strahlanforderung X-Zweig
	1	Strahlanforderung Y-Zweig
	2	Strahlanforderung Z-Zweig
	3	M-Chopper: Strahl ausgetastet
3	4	N-Chopper: Strahl ausgetastet
	5 ... 7	nicht belegt
	0 ... 1	aktueller Stralweg: Kennung Quelle
2 ... 3	2 ... 3	aktueller Stralweg: Kennung Ziel
	4	aktueller Stralweg: Strahl zur Materialforschung
5	5	M-Chopper: Strahl austasten
	6	N-Chopper: Strahl austasten
	7	Interlockmeldung löschen

3.3 Belegung der Therapie-Anfordereinheit

Die Strahlanforderung für den Therapiebetrieb erfordert ein spezielles Protokoll. Die Belegung der Anfordereinheiten, über die der Therapiebetrieb abgewickelt werden kann, ist im Folgenden aufgelistet. Von der Pulszentrale werden pro Anfordereinheit 8 Byte Daten eingelesen und 8 Byte Daten an den Anforderer zurückübermittelt. Die Belegung ist für beide Datenrichtungen identisch.

Belegung der Anfordereinheit für Therapie-Anforderung	
Byte	Bedeutung
0	virt. Beschl. (immer 0)
1	Energienstufe E
2	Intensität I
3	Fokus F
4	Zyklusnummer C
5	Datenkennung D, Maschinenk. low Byte
6	Datenkennung D, Maschinenk. high Byte
7	Status, bitweise

Die Belegung des Statusbytes ist wie folgt, wobei sich die Datenrichtung (In/Out) auf die Pulszentrale bezieht. Die meisten Bits werden entweder vom Anforderer (In) oder von der Pulszentrale (Out) geschrieben und dann von der jeweiligen Gegenseite bestätigt. Lediglich Bit 5 und 6 werden nicht von der Pulszentrale eingelesen, sondern nur an den Anforderer gemeldet (Out).

Belegung des Status-Bytes für Therapie-Anforderung			
Bit	Name	Bedeutung	Datenrichtung
0	LOCK	Verriegelung des Gerätezugriffs	In/Out
1	DTAvail	Anforderungsdaten angelegt	In/Out
2	DTVal	Bestätigung Anforderungsdaten, virt. Beschl. starten	In/Out
3	VM0Block	Exklusive Bearbeitung von VrtAcc0	In/Out
4	MNGEIN	Einschaltbefehl für HTMMU1/2	In/Out
5	AccReady	Strahlweg zum Cave-M frei	Out
6	ReqEnabl	Anforderung freigegeben)	Out
7	SessEnd	Wechsel des Bestrahlungsplanes	In/Out

Die Handhabung von DTAvail und DTVal ist:

1. Die Bestrahlungssteuerung legt die EFICD-Daten an das Eingangsregister.
2. Die Bestrahlungssteuerung signalisiert das durch Setzen des DTAvail-Bits im Eingangsregister der Pulszentrale.
3. Die Pulszentrale spiegelt daraufhin die angelegten EFICD-Daten im Ausgangsregister.
4. Die Pulszentrale signalisiert das durch Setzen des DTAvail-Bits im Ausgangsregister.
5. Die Bestrahlungssteuerung prüft die von der Pulszentrale gespiegelten EFICD-Daten.
6. Sind die von der Pulszentrale gespiegelten Daten für gut befunden, und sind keine weiteren Störungen aufgetreten, setzt die Bestrahlungssteuerung das DTVal-Bit im Eingangsregister.
7. Die Pulszentrale quittiert das durch Setzen des DTVal-Bit im Ausgangsregister und startet daraufhin den Zyklus.

3.4 Die Schnittstellen der ESR-Pulszentrale

Die ESR-Pulszentrale verfügt über folgende Schnittstellen:

- Datenaustausch mit der SIS-Pulszentrale (Daten zum Strahlaustausch, sowohl für Extraktion als auch Reinjektion).

3.5 Belegung der Schnittstellen der ESR-Pulszentrale

Die Schnittstelle der ESR-Pulszentrale zur SIS-Pulszentrale besteht lediglich aus 2 I/O-Modulen (mit jeweils 24 Bit), die zum Informationsaustausch zwischen den beiden Pulszentralen verwendet werden. Die folgende Abbildung zeigt die Belegung dieses Modulbus-Einschubs.

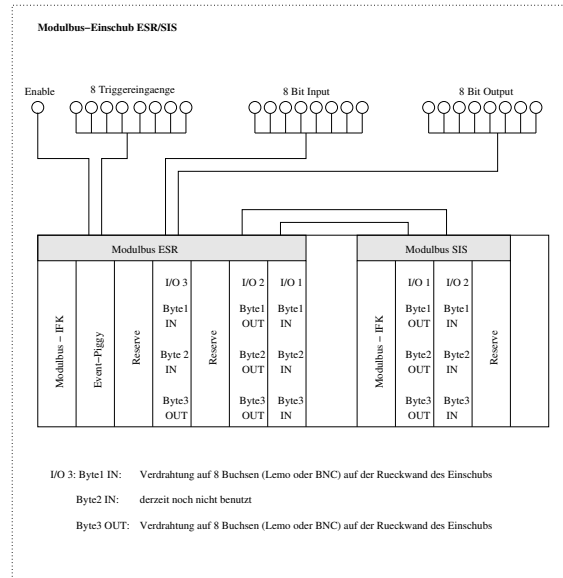


Abbildung 3: Modulbus-Einschub ESR/SIS

Die Belegung der I/O-Module des Modulbus-Einschub ESR/SIS im Detail:

Datenaustausch mit UNILAC-Pulszentrale				
I/O-Modul	Byte	I/O-Richtung	Bit	Bedeutung
1	1	output	0 ... 3	vom ESR angeforderte Beschl. Nr.
			4	Anforderung der SIS-Vorbereitung
			5	Anforderung des Strahls ???
			6 ... 7	nicht belegt
	2	output	0 ... 7	nicht belegt
			3	output
2	1	input	0 ... 7	???
	2	input	0 ... 7	???
	3	input	0	Bestätigung der SIS-Vorbereitung
			1	Bestätigung der Strahlanforderung ???

3.6 Die Schnittstelle des Event-Piggy

Zur Einspeisung zeitgenauer Informationen in die Pulszentrale wurde im Labor **Datentechnik** ein spezielles *Event-Piggy* (FG ??? ???) als Modulbus-Karte entwickelt.

Damit ist es möglich über einen normalen MIL-Bus, an dem auch andere Interfacekarten angeschlossen sein können, den Eventbus des Pulszentralen-VME-Rahmens mit Events zu versorgen (s. Abb. auf Seite 4).

Zur Synchronisierung der Pulszentrale mit externen Ereignissen (z. B. `EVT_Uni_Ready_to_SIS`) können bis zu 12 externe Trigger (TTL-Pegel) eingespeist werden. Für jeden Trigger kann in einem RAM ein Eventcode definiert werden, der bei Auftreten des Triggerpulses auf dem Eventbus erzeugt wird.

Die 12 Eingänge können einzeln und gemeinsam enabled oder disabled werden. Die Triggereingänge werden automatisch nach Eintreffen eines Triggersignals disabled, sodaß immer nur genau 1 Trigger zugelassen wird.

Zur Überwachung des Timings ist eine Timeout-Verwaltung implementiert, die vor dem *enable* der Triggereingänge versorgt werden muß. Kommt ein erwartetes Triggersignal nicht innerhalb der über die Timeout-Verwaltung eingestellten Zeit, werden die Eingänge automatisch disabled und ein frei programmierbares Timeout-Event erzeugt.

3.7 Die externen Triggereingänge der Pulszentralen

Zur Zeitsynchronisation mit externen Ereignissen können die Kanäle der SIS/ESR-Pulszentrale über Hardware-Triggersignale gestartet werden. Jede Pulszentrale besitzt dazu acht Triggereingänge. Diese Triggereingänge können bei der Festlegung der Kanalsequenz (Property CHANSEQU) als Startbedingung für die Kanäle festgelegt werden.

Die Triggereingänge werden von 0 bis 7 durchnummeriert. Um die Pulszentrale möglichst ähnlich zu der ersten in SIS und ESR verwendeten Generation erscheinen zu lassen, werden den Triggereingängen in der VME-Software sogenannte »Primäreventcodes« zugeordnet. Das sind im Kontrollsystem definierte Eventcodes, über die die Triggereingänge angesprochen werden. Die Primärevents sind zunächst nur Bezeichnungen für die Triggereingänge. Um die Kompatibilität zur ersten Generation von Pulszentralen möglichst weit zu treiben, werden die Primärevents von der Pulszentrale beim Start des Kanals automatisch verschickt. Einem Triggereingang können mehrere Primärevents zugeordnet werden.

Folgende Primärevents sind definiert:

SIS:

Primärevent	Event-Nr	Eingang
EVT_Uni_Ready	38	2
EVT_Trans_Start_1	61	0
EVT_Kick_Start_1	49	1
EVT_Trans_Start_2	62	0
EVT_Kick_Start_2	69	1
EVT_Extr_Stop_Slow	78	3
EVT_Async_Trans	79	0
EVT_Kick_Ready	105	4

ESR:

Primärevent	Event-Nr	Eingang
EVT_Trans_Start_1	61	0
EVT_Kick_Start_1	49	1
EVT_Trans_Start_2	62	0
EVT_Kick_Start_2	69	1
EVT_Async_Trans	79	0

Triggereingänge werden erst aktiviert, wenn ein daran geknüpfter Kanal zur Ausführung ansteht. über ein Enable-Signal (jeweils eines pro Triggereingang) wird angezeigt, dass der an das Triggersignal geknüpfte Kanal zur Ausführung ansteht. Das Enable-Signal wird mit dem Start des Kanals bzw. mit dem Ablauf der Timeout-Zeit wieder zurückgenommen.

Der sicherlich wichtigste Anwendungsfall ist die Synchronisation der Pulszentralen bei der Übergabe von Strahl von einem Timingbereich in den folgenden (Unilac nach SIS und SIS nach ESR bzw. ESR zurück zum SIS). über die Triggereingänge können die Kanäle μ s-genau gestartet werden.

Bei der Kopplung von SIS und ESR-Pulszentrale ist es erforderlich, dass in beiden Pulszentralen Kanäle zeitgleich gestartet werden. Dazu müssen die zu synchronisierenden Kanäle jeweils über Triggereingänge gestartet werden. Der Start darf aber erst erfolgen, wenn in *beiden* Pulszentralen die Sequenz bis zu dem Startpunkt abgearbeitet worden ist. Diese Synchronisation kann über die Enable-Signale erfolgen: Wenn die Enable-Signale sowohl der SIS als auch der ESR-Pulszentrale anstehen, ist dieser Zeitpunkt erreicht und der Trigger soll erfolgen.

Das kann leicht über eine zwischengeschaltete HW Logig-Einheit erfolgen: Die Enable-Signale der beiden Pulszentralen werden auf ein UND-Gatter gegeben und mit dem Ausgangssignal dieses Gatters werden beide Kanäle gestartet.

Auf entsprechende Weise kann der Kanalstart mit externen Ereignissen synchronisiert werden.

Die derzeitige Verschaltung der Trigger-Eingänge (mit Enable-Signalen) über die zwischengeschaltete Logig-Einheit ist in der folgenden Abbildung dargestellt.

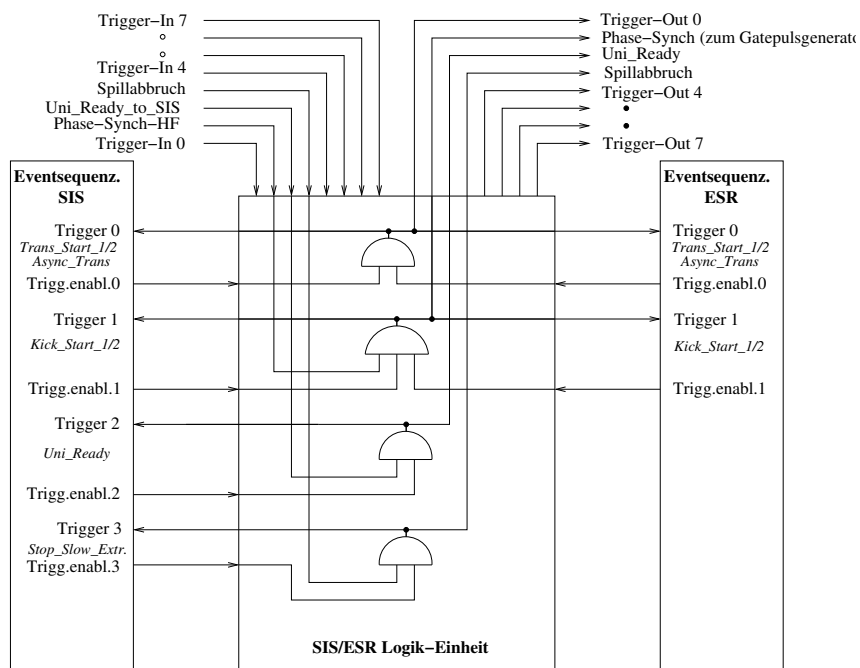


Abbildung 4: Verschaltung der Trigger-Eingänge ESR/SIS