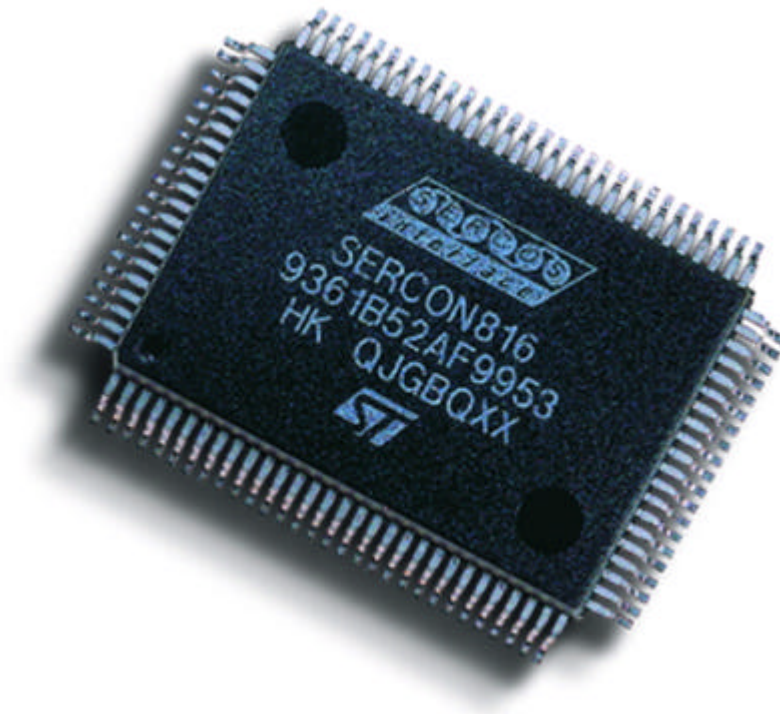




SERCON816

Spezifikation



Project: sertrans
Document: /proj/sertrans/doc/spec/sp81613doc
Version: 1.3

Author(s): Rolf Kassa, Rigobert Kynast (TWG-IGS)

Responsible:

Department: DIC

Revision history:

Version	Date	Remarks
1.0	4.12.1998	Vorschlag
1.1	17/12/1998	1. Überarbeitung
1.2	15/03/1999	
1.3	11.5.1999	Leistungsumfang, Bitrate

INHALT

1 Überblick	4
1.1 Kurzbeschreibung.....	4
1.2 Querverweise.....	4
2 Spezifikation SERCON816.....	5
2.1 Technologie.....	5
2.2 Pinning des Bausteins	5
2.3 Serielles Interface.....	5
2.3.1 Serieller Takt (SCLK).....	5
2.3.2 Kompatibilitätsmodus zum SERCON410B	6
2.3.3 Repeater	7
2.4 Telegrammverarbeitung.....	7
2.4.1 Takt der Telegrammverarbeitung (MCLK).....	7
2.5 Dual Port RAM.....	7
2.6 Watchdog.....	8
2.6.1 Watchdog Ausgang WDOGN	8
2.6.2 Einfluß des Watchdogs auf die Telegrammverarbeitung.....	9
2.6.3 Trigger Intervall	10
2.6.4 Triggern per Software.....	10
2.6.5 Triggern durch CYC_CLK	10
2.7 Ausgang DIV_CLK	11
2.8 Bugfix „Spezifikation BUSY-Signal“.....	12
2.9 Bugfix „Phase 1,2 Fehler“	12
2.9.1 Fehlerbeschreibung.....	12
2.9.2 Ursache und Fehlerbehebung	12
2.10 Bugfix „Asynchrone Zugriffe auf Dual-Port-RAM“	13
2.10.1 Steuerung des Doppelbuffers (VAL-Bit)	13
2.10.2 Kontrollwort für Service-Container.....	15
2.11 Steuerregister.....	16
2.11.1 Geänderte Steuerregister	16
2.11.2 Neue Steuerregister für SERCON816.....	18

BILDER

Abbildung 1: Watchdog Interrupt.....	8
Abbildung 2: Reset Prozessor über WDOGN Ausgang.....	9
Abbildung 3: Watchdog Trigger durch MST.....	10
Abbildung 4: Neuer Mode DIV_CLK	12
Abbildung 5: FLMDTADR.....	13
Abbildung 6: Lesen/Schreiben VAL-Bit bei Sendetelegrammen	14
Abbildung 7: Interrupt RSCEND.....	16

1 Überblick

1.1 Kurzbeschreibung

Der Baustein SERCON816 ist eine Weiterentwicklung des SERCOS interface Controllers SERCON410B. Ziel der Weiterentwicklung ist ein Baustein, der weitgehend abwärts kompatibel zum SERCON410B ist, jedoch in wesentlichen Leistungsmerkmalen verbessert wird. Der Baustein SERCON410B wird in der 0,7 μm Technologie des Herstellers ST Microelectronics gefertigt. Der SERCON816 soll in einer 0,5 μm , 5 V Technologie des gleichen Herstellers gefertigt werden.

Im einzelnen erfolgt neben der Technologieumsetzung eine Steigerung der Leistungsmerkmale bzw. Erweiterung des Funktionsumfangs um folgende Punkte:

- Steigerung der seriellen Übertragungsraten von 2 und 4 MBit/s auf 2, 4, 8 und 16 MBit/s. Die externe Taktregenerierung entfällt. Die Geschwindigkeit der parallelen Datenverarbeitung (Telegrammverarbeitung) wird entsprechend gesteigert.
- Verdoppelung des internen Dual Port RAM von 1k Worte x 16 Bit auf 2k Worte x 16 Bit.
- Einbau eines Watchdog zur Überwachung der ansteuernden Software und der externen Synchronisationssignale.
- Der Resetwert für den Repeater-Modus des seriellen Interface kann durch Pins konfiguriert werden.
- Neuer Modus für den Ausgang DIV_CLK

Der Baustein SERCON410B zeigt in einigen Betriebszuständen ein fehlerhaftes Verhalten, die den Einsatz des Bausteins einschränken. Im SERCON816 werden folgende Fehler korrigiert:

- *Spezifikation des BUSY-Signals*: das Timing dieses Signals ist in der aktuellen Spezifikation falsch beschrieben. Für das Datenblatt des SERCON816 wird die Beschreibung angepaßt.
- *Dauerhaftes Senden von Antriebstelegrammen in Phase 1 und Phase 2*: wird ein Antrieb zum Senden eines AT's aufgefordert, sendet er dieses nicht nur einmal, sondern fortlaufend.
- *Asynchrone Zugriffe auf das Daten-RAM*: werden über das Businterface zu einem beliebigen Zeitpunkt während eines Übertragungszykluses Daten in das Dual-Port-RAM geschrieben, kann es vorkommen, daß diese Daten von der internen Telegrammverarbeitung wieder überschrieben werden. Dieses Problem wird nicht generell gelöst. Statt dessen wird für zwei Datentypen, die sich in der Praxis als besonders kritisch herausgestellt haben, ein Workaround implementiert.

1.2 Querverweise

Diese Spezifikation baut auf folgende Dokumente auf:

- SERCON410B Reference Manual (08/93)
- Protokoll des Meetings bei ST am 17.11.98
- Zusatzangebot vom 5.3.99

2 Spezifikation SERCON816

Nachfolgend wird die Funktion des SERCON816 beschrieben. Es werden nur die Funktionen und Leistungsmerkmale beschrieben, die sich im Vergleich zum SERCON410B verändert haben.

2.1 Technologie

Der Baustein wird in der 0,5 μm / 5V Technologie des Herstellers ST Microelectronics entwickelt (HCMOS5).

2.2 Pinning des Bausteins

Der Baustein wird im QFP100 und TQFP100 Gehäuse gefertigt. Das Pinning des SERCON816 entspricht bis auf wenige Ausnahmen dem des SERCON410B. Durch die erweiterte Funktionalität wird es notwendig einige Pins des Bausteins neu zu definieren. Alle anderen Pins entsprechen denen des SERCON410B

Nr	SERCON410B	SERCON816	Funktion SERCON816
28	SREGEN	SBAUD16	Bitrate serielles Interface (zusammen mit SBAUD) und Kompatibilitätsmodus SERCON410B: 0: 8, 16 Mbit/s 1: 2, 4 Mbits/s
24	TxDNRZ	WDOGN	Watchdog Ausgang 0: Watchdog-Ausgang aktiv 1: Watchdog-Ausgang nicht aktiv
13	TxC (I/O)	TxC (O)	Ausgang für Transmitter Takt
12	RxC (I/O)	RxC (O)	Ausgang für den regenerierten Receive Takt

2.3 Serielles Interface

Das serielle Interface des SERCON816 empfängt und sendet die seriellen Daten. Aus dem eingehenden seriellen Datenstrom werden Daten und Takt des Bitstroms regeneriert. Der SERCON410 bietet die Möglichkeit den Bittakt mittels einer externen Schaltung zu rekonstruieren und Takt und Daten getrennt in den Baustein zu führen. Diese Möglichkeit entfällt beim SERCON816, da die interne Taktregenerierung ausreichend hohe Bitraten verarbeiten kann, wodurch die Notwendigkeit einer externen Taktregenerierung entfällt.

2.3.1 Serieller Takt (SCLK)

Die Taktregenerierung besteht aus einer Zustandsmaschine (DPLL_FSM), die mit der 16fachen Frequenz des eingehenden Bitstroms getaktet wird. Bei einer Bitrate von 16 Mbit/s ergibt sich damit eine Taktfrequenz von 256 MHz für die DPLL_FSM. Dieser Takt wird durch eine Vervierfachung des Eingangstaktes SCLK erreicht. Die

externe Frequenz des Taktes SCLK beträgt damit weiterhin 64 MHz, wie beim SERCON410B. Die Taktvervielfachung erfolgt mittels einer analogen PLL (Makro PLL51).

Die DPPLL_FSM generiert als Ergebnis einen Takt, welcher der gewünschten Bitrate entspricht (2, 4, 8 oder 16 MHz). Mit diesem Takt werden der seriellen Empfänger und Sender getaktet.

Die Einstellung der Bitrate kann über Pins oder durch Software erfolgen. Durch die Erweiterung der Einstellmöglichkeiten ist ein neuer Pin (SBAUD16) und ein neues Steuerbit (SWBAUD16) notwendig:

Bitrate für SCLK = 64 MHz					
ENTSBAUD (Adr. 1H.10)	SBAUD (Pin 29)	SBAUD16 (Pin 28)	SWBAUD (Adr. 1H.6)	SWBAUD16 (Adr. 1H.15)	Bitrate
0	1	1	x	x	2 Mbit/s
0	0	1	x	x	4 Mbit/s
1	x	1	1	x	2 Mbit/s
1	x	1	0	x	4 Mbit/s
0	1	0	x	x	8 Mbit/s
0	0	0	x	x	16 Mbit/s
1	x	0	1	1	2 Mbit/s
1	x	0	0	1	4 Mbit/s
1	x	0	1	0	8 Mbit/s
1	x	0	0	0	16 Mbit/s

Der Pin SBAUD16 steuert zusätzlich die Ausgangsfrequenz der Taktausgänge SCLKO2 und SCLKO4:

SCLKO2, SCLKO4 für SCLK = 64 MHz		
SBAUD16	SCLKO2	SCLKO4
1	32 MHz	16 MHz
0	64 MHz	32 MHz

2.3.2 Kompatibilitätsmodus zum SERCON410B

In SERCON410B Applikationen mit interner Taktrückgewinnung liegt der Pin SREGEN (SBAUD16) auf dem Pegel '1'. Dieser Pegel erzwingt im SERCON816 einen zum SERCON410B kompatiblen Modus.

In diesem Modus sind nur Bitraten von 2 und 4 Mbit/s einstellbar und die Ausgänge SCLKO2 und SCLKO4 geben die gleichen Frequenzen aus wie beim SERCON410B.

Sollen die neuen Bitraten des SERCON816 genutzt werden, muß SBAUD16 auf den Pegel '0' liegen.

2.3.3 Repeater

Der Repeater der seriellen Schnittstelle wird über das Steuerbit REPON an- (1) oder ausgeschaltet (0). Der Resetwert des Steuerbits REPON kann über die Pins TM0/1 eingestellt werden:

TM0	TM1	REPON nach Reset (Adr 1H.3)	Testmode
1	1	1: Repeater on	keiner
0	0	0: Repeater off	keiner
0	1	1	TxD1-6 continues signal light
1	0	1	Zero bit stream

Die Konfiguration der Pins TM0/1 bestimmt REPON nur während des Resets des Bausteins. Nach dem Reset haben TM0/1 keinen Einfluß auf den Inhalt von REPON.

Mit den Steuerbits TMODE2-0 kann der aktuelle Testmode überschrieben werden. Die Kodierung für diese Steuerbits ändert sich nicht. Auf den Wert REPON hat TMODE2-0 keinen Einfluß.

2.4 Telegrammverarbeitung

2.4.1 Takt der Telegrammverarbeitung (MCLK)

Die Telegrammverarbeitung und alle anderen Schaltungsteile außer dem seriellen Interface werden mit dem Takt MCLK angesteuert. Die erhöhte serielle Übertragungsrate macht auch eine Steigerung der Telegrammverarbeitung und damit des Taktes MCLK notwendig. Eine maximale Taktfrequenz von 32 MHz für MCLK kann auf Grund der schnelleren Technologie zugesagt werden. Zielvorgabe ist eine Frequenz von 64 MHz.

2.5 Dual Port RAM

Das interne Dual Port RAM wird von 1k Worte x 16 Bit (SERCON410B) auf 2k Worte x 16 Bit (SERCON816) erweitert. Hierdurch ändert sich die Adressierung des RAM's und der Steuerregister über das Businterface.

Das interne RAM wird über die Adreßpins A11-A1 und die Chip-Select Eingänge MCSN0/1 adressiert. Die internen Steuerregister werden über die Chip-Select Eingänge PCSN0 und PCS1 angesprochen. In einem bestehenden SERCON410B-System kann das Adreßbit A11 mit den Chip-Select-Eingängen MCSN1 und PCS1 verbunden sein, um zwischen RAM und Steuerregistern zu unterscheiden. Beim SERCON816 muß hierfür das Adreßbit A12 verwendet werden. Wird in einem bestehenden System der SERCON410B durch den SERCON816 ersetzt, können darum die oberen 1024 Worte des internen RAM's nicht genutzt werden.

Können die oberen 1k Worte des RAM's vom Businterface nicht angesprochen werden, muß über die Programmierung der Adreßpointer in den Telegramm-Hea-

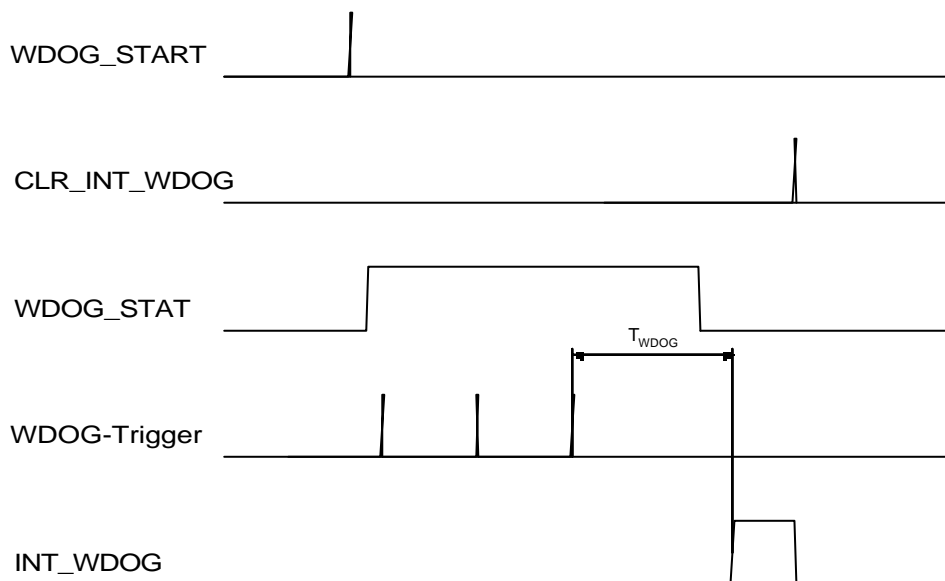
dem festgelegt werden, daß dieser Speicherbereich auch von der Telegrammverarbeitung nicht genutzt wird.

2.6 Watchdog

Der Watchdog des SERCON816 dient zur Überwachung der ansteuernden Software und des Eingangssignals CYC_CLK.

Für die Überwachung muß der Watchdog innerhalb eines programmierbaren Intervalls regelmäßig getriggert werden. Fällt das Triggersignal aus, spricht der Watchdog an. Mit dem Ansprechen des Watchdogs wird ein Interruptflag INT_WDOG gesetzt. Diese Interruptquelle besitzt wie alle anderen Interruptquellen des SERCON816 zwei Freigabebits für die Interruptausgänge INT0 und INT1 (Enx_INT_WDOG) und ein Steuerbit zum Löschen des Interruptflags (CLR_INT_WDOG).

Der Watchdog ist nach dem Reset des Bausteins abgeschaltet und kann durch Schreiben des Steuerbits WDOG_START aktiviert werden. Ist er einmal aktiviert worden, kann der Watchdog durch die Software nicht mehr abgeschaltet werden (Ausnahme Software-Reset). Spricht der Watchdog an, schaltet er sich automatisch wieder ab. Der Watchdog Interrupt bleibt jedoch gesetzt, bis er durch einen Reset oder Schreiben auf CLR_INT_WDOG gelöscht wird. Der aktuelle Zustand des Watchdogs wird im Statusbit WDOG_STAT angezeigt.



wdog1.vsd

Abbildung 1: Watchdog Interrupt

2.6.1 Watchdog Ausgang WDOGN

Das Interruptflag INT_WDOG kann zusätzlich den Watchdogausgang WDOGN aktivieren. Dies muß durch das Steuerbit WDOEN freigegeben werden. Der Ausgang WDOGN ist ein 0-aktives Signal, das mit dem Reseteingang des ansteuernden Prozessors verbunden werden kann. Damit der Prozessor den Resetzustand

wieder verläßt, wird WDOGN nach der Zeit T_{WDOGN} wieder abgeschaltet. Der Prozessor kann nach dem Booten durch Lesen des Watchdog-Interruptflags INT_WDOG feststellen, ob er durch den Watchdog oder den Power-Up zurückgesetzt wurde.

Die Zeitdauer für die der Ausgang WDOGN aktiv wird beträgt 2-3 μ s. Der Zeitraum 1 μ s wird durch den Vorteiler MCLKDIV bestimmt. Der Watchdog-Ausgang bleibt für mindestens 2, maximal 3 Periodendauern des Vorteilers MCLKDIV aktiv.

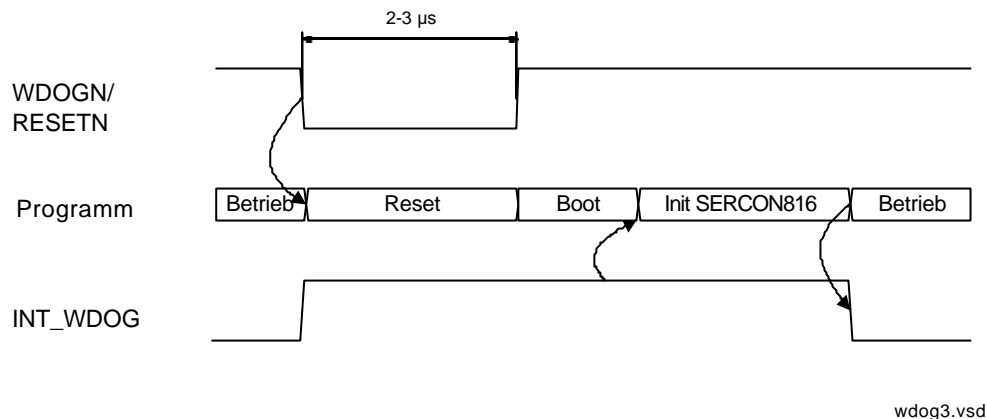


Abbildung 2: Reset Prozessor über WDOGN Ausgang

2.6.2 Einfluß des Watchdogs auf die Telegrammverarbeitung

Das Ansprechen des Watchdogs (INT_WDOG=1) setzt interne Register des SERCON816:

Register	Resetwert durch INT_WDOG=1
ENTMT (Adr.3H.8)	0
PHAS0 (Adr.AH.0-7)	0H

Das Registerbit ENTMT gibt das Senden von Telegrammen frei (Master- und Slave-Mode). Durch das zurücksetzen von ENTMT wird das Senden von AT's (Slaves) und MDT's (Master) nach dem Ansprechen des Watchdogs unterdrückt.

In dem Register PHAS0 wird die Phaseninformation festgelegt, die der SERCON816 beim Senden eines MST (Master-Mode) überträgt. Durch das Zurücksetzen dieses Registers überträgt der SERCON816 nach dem Ansprechen des Watchdogs MST's der Phase 0.

Diese Registerbits können vom Businterface erst wieder neu programmiert werden, wenn das Statusbit INT_WDOG gelöscht wurde.

Das Rücksetzen dieser Steuerbits über den Hardware- bzw. Softwarereset hat eine höhere Priorität, wie das Rücksetzen über INT_WDOG!

2.6.3 Trigger Intervall

Das Trigger Intervall wird über den Verteiler MCLKDIV und den Wert TWDOG15-0 bestimmt.

$$T_{\text{WDOG}} = \frac{(MCLKDIV + 1) \cdot TWDOG}{f_{\text{MCLK}}}$$

Nach der Freigabe des Watchdogs wird ein interner Rückwärtszähler mit dem Wert TWDOG geladen. Der Zähler zählt mit der durch MCLKDIV vorgegebenen Frequenz (üblicherweise 1µs). Erreicht der Zähler den Stand 0000H spricht der Watchdog an. Mit jedem Triggerereignis wird der Zähler erneut mit dem Wert TWDOG geladen.

2.6.4 Triggern per Software

Der Watchdog kann durch Software getriggert werden, dies erfolgt durch Schreiben des Wertes 42H auf die Steuerbits WDOG_TRIG (29H.7-0).

Das Triggern über Schreiben der Steuerbits WDOG_TRIG ist zu jedem Zeitpunkt möglich.

2.6.5 Triggern durch CYC_CLK

Im Master-Modus synchronisiert der Eingang CYC_CLK den Start eines oder mehrerer Sercos-Zyklen. Der SERCON816 kann überprüfen, ob das Eingangssignal CYC_CLK sich innerhalb eines programmierbaren Zeitfensters wiederholt (JTSCYC1, JTSCYC2). Liegt das Signal CYC_CLK außerhalb dieses Zeitfensters, wird der Interrupt MSTEARLY aktiv. Fällt der Eingang CYC_CLK jedoch ganz aus, spricht dieser Interrupt nicht an. Der Watchdog ergänzt die Überwachung des Signals CYC_CLK.

Der Watchdog wird nicht direkt mit CYC_CLK getriggert, sondern mit dem Startsignal für die von CYC_CLK ausgelösten MST's. Fällt der Eingang CYC_CLK aus, spricht der Watchdog nach Ablauf der Triggerzeit nach dem letzten Start eines MST an.

Dieser Modus ist nur im Master-Betrieb möglich. Das Triggern über das interne MST Startsignal muß mit dem Steuerbit WDOG_MSTEN freigegeben werden.

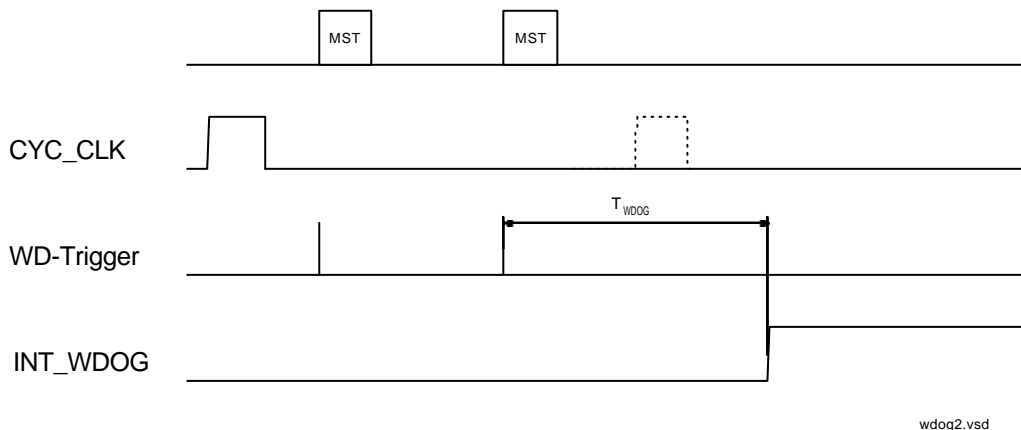


Abbildung 3: Watchdog Trigger durch MST

2.7 Ausgang DIV_CLK

Im SERCON410B wird der Ausgang DIV_CLK mehrfach während eines Sercos-Zyklus aktiv. Der erste Zeitpunkt für das Aktivieren des Ausgangs wird in dem Steuerregister TDIVCLK (1EH) programmiert. Der Abstand zwischen zwei DIV_CLK Impulsen in dem Steuerregister DTDIVCLK (1FH). Der aktive Pegel des Ausgangs wird in dem Registerbit POLDIVCLK (20H.8) und die Anzahl der Impulse innerhalb eines Sercos-Zyklus in den Steuerbits NDIVCLK (20H.7-0) programmiert.

Der SERCON816 kennt einen neuen Modus für den Ausgang DIV_CLK. Dieser wird in dem neuen Steuerbit DIVCLKMODE ausgewählt. In diesem Modus wird der Ausgang DIV_CLK nicht mehrfach während eines Sercos-Zyklus sondern einmal nach einer programmierbaren Anzahl von Sercos-Zyklen aktiv.

DIVCLKMODE (20H.9)	Funktion
0	mehrfach während eines Zyklus (wie SERCON410B)
1	einmal in mehreren Sercos-Zyklen

Der Resetwert für DIVCLKMODE beträgt '0'. Damit verhält sich der SERCON816 nach dem Reset wie der SERCON410B.

Die Anzahl der Sercos-Zyklen, nach denen der Ausgang DIV_CLK aktiv wird, wird in den Steuerbits NDIVCLK programmiert:

NDIVCLK (10H.7-0)	Funktion DIV_CLK
0	wird nie aktiv
1	einmal in jedem Zyklus
2	jeden zweiten Zyklus
3	jeden dritten Zyklus
N	jeden N-ten Zyklus

Der Ausgang DIV_CLK wird zum ersten Mal im ersten Sercos-Zyklus aktiv (bei NDIVCLK != 0) oder im ersten Zyklus nachdem NDIVCLK auf einen Wert ungleich 0 programmiert wurde.

Der Zeitpunkt innerhalb eines Sercos-Zyklus, zu dem DIV_CLK aktiv wird, bestimmt das Steuerregister TDIVCLK. Dieser entspricht dem ersten Zeitpunkt im alten DIV_CLK Modus. Die Polarität und Länge des DIV_CLK Impulses bleibt gegenüber dem alten Modus unverändert.

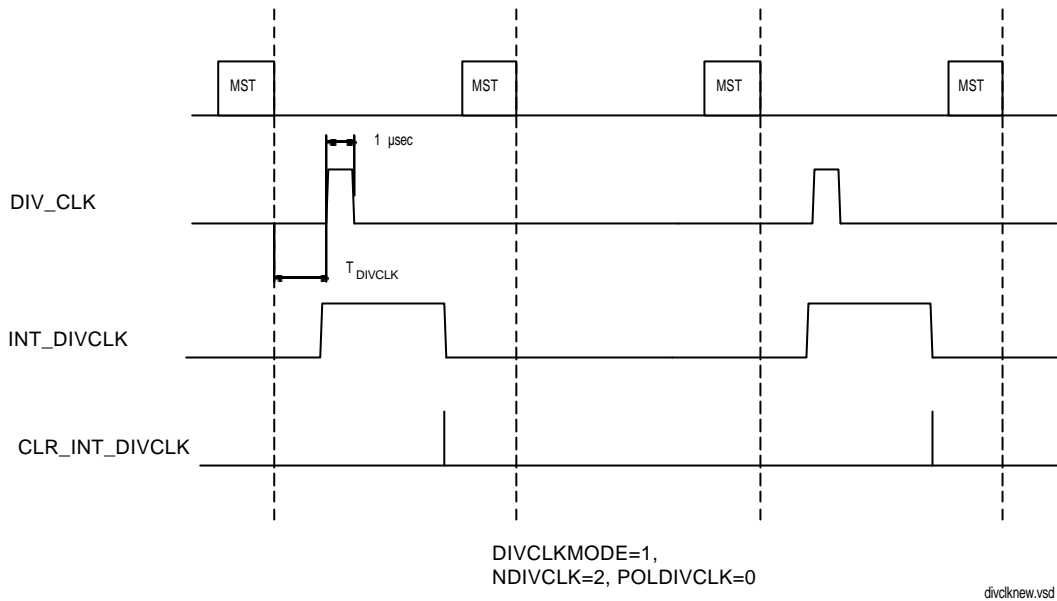


Abbildung 4: Neuer Mode DIV_CLK

2.8 Bugfix „Spezifikation BUSY-Signal“

Der Ausgang BUSYN des Mikrocontroller Businterfaces hat laut Spezifikation eine maximale Länge von 1 Zyklus von MCLK. Messungen haben ergeben, daß die maximale Länge des BUSYN Signals deutlich länger ausfallen kann. Die Logik des BUSY-Signals wird überprüft um die Ursache für diese langen BUSY Zeiten zu finden. Die maximale Länge des BUSY-Signals wird neu spezifiziert.

2.9 Bugfix „Phase 1,2 Fehler“

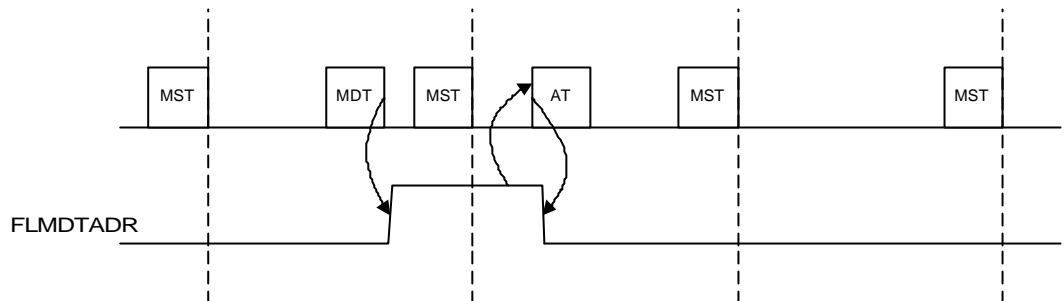
2.9.1 Fehlerbeschreibung

In den Kommunikationsphasen 1 und 2 soll ein Antrieb (SERCON im Slave-Modus) nur dann ein Telegramm senden, wenn er vom Master dazu aufgefordert wurde. Nach einer Anforderung vom Master soll das Antriebstelegramm nur einmal gesendet werden. Der SERCON410B sendet das Telegramm fortlaufend, bis der Master einen anderen Antrieb zum Senden des Telegramms auffordert.

2.9.2 Ursache und Fehlerbehebung

Der oben beschriebene Vorgang wird durch ein internes Flag im SERCON410B gesteuert (FLMDTADR, 25H.12). Das Flag wird gesetzt, wenn beim Empfang eines Telegramms die erwartete Adresse des Antriebs empfangen wurde. Ist dieses Flag gesetzt, wird im nächsten Übertragungszyklus das Antriebstelegramm gesendet. Im SERCON410B wird das Flag FLMDTADR erst dann wieder gelöscht, wenn ein neues MDT empfangen wird, mit einer Adresse ungleich der erwarteten Adresse. Ein fehlendes MDT setzt das Flag nicht zurück, hierdurch kann es zum mehrfachen Senden des Antriebstelegramms kommen, wenn in einem Zyklus kein MDT gesendet wird.

Im SERCON816 wird das Flag FLMDTADR gelöscht, wenn das Senden eines Antriebstelegramms durch ein gesetztes FLMDTADR gestartet wurde. Hierdurch kann das Senden eines Antriebstelegramms durch FLMDTADR nur einmal angestoßen werden. Soll ein weiteres Telegramm gesendet werden, muß zuvor ein neues MDT mit gültiger Adresse empfangen werden.



flmdtadr.vsd

Abbildung 5: FLMDTADR

2.10 Bugfix „Asynchrone Zugriffe auf Dual-Port-RAM“

Über das Dual-Port-RAM greifen die interne Telegrammverarbeitung (TV) und der externe Prozessor (μ P) auf gemeinsame Datenstrukturen zu. Diese Datenstrukturen enthalten Status- und Steuerinformationen zum Senden und Empfangen von Telegrammen.

Beim asynchronen Zugriff liest und schreibt der μ P diese Datenstrukturen ohne Rücksicht auf den aktuellen Zeitpunkt innerhalb eines Übertragungszyklus. Die TV liest Daten aus dem Dual-Port-RAM, verändert diese und schreibt sie zu einem späteren Zeitpunkt wieder ins RAM zurück. Schreibt der μ P in der Zwischenzeit neue Daten ins RAM, kann es vorkommen, daß einzelne Steuerinformationen von der TV mit falschen Daten überschrieben werden.

Dieses Problem kann ebenfalls auftreten, wenn der μ P Bitmanipulationen (lesen, ändern und zurückschreiben) auf Daten im Dual-Port-RAMs ausführt.

Das Problem des asynchronen Zugriffs wird nicht generell gelöst. Statt dessen werden für zwei Probleme, die sich im praktischen Einsatz als kritisch erwiesen haben, spezielle Lösungen implementiert die helfen Zugriffskonflikte zu vermeiden.

2.10.1 Steuerung des Doppelbuffers (VAL-Bit)

2.10.1.1 Fehlerbeschreibung

Das VAL-Bit im Telegrammheader steuert die Verwendung des Doppelbuffers für Telegrammdaten. Bei Sendetelegrammen wird dieses Bit vom μ P programmiert. Der SERCON wählt, abhängig von diesem Bit, die zu sendenden Telegrammdaten aus. Der Telegrammheader - und damit das VAL-Bit - werden für das nächste Telegramm am Ende des aktuellen Telegramms gelesen. Zum Startzeitpunkt des Sendetelegramms wird der Telegrammheader (Kontrollwort 0) in bestimmten Modi wieder ins RAM zurückgeschrieben (Sendetelegramm, EN=1), da sich im Telegramm-

header Statusinformationen geändert haben. Hat in der Zeit zwischen den beiden Telegrammen der μP einen neuen Wert für VAL programmiert, wird dieser neue Inhalt von der TV wieder mit dem alten Wert überschrieben. Dieses Problem tritt nur bei Sendetelegrammen auf, da bei Empfangstelegrammen das VAL-Bit ausschließlich von der TV verändert wird.

2.10.1.2 Fehlerbehebung

Der Zeitraum, in dem Schreibzugriffe auf den Telegrammheader verboten sind, soll auf wenige Takte zum Start des Sendetelegrammes begrenzt werden. Hierzu wird das VAL-Bit zum Start eines Telegramms aus dem Telegrammheader erneut gelesen (alle anderen Daten des Telegrammheaders: ADR, DMA, DBUF und EN werden nicht neu gelesen!). Muß der Telegrammheader in das Dual-Port-RAM zurückgeschrieben werden (Mode EN=1), wird das zuletzt gelesenen VAL-Bit mit dem Telegrammheader geschrieben. Bei diesem Ablauf kann der μP in der Zeit zwischen zwei Telegrammen das Steuerbit VAL neu programmieren. Das neu programmierte VAL-Bit wird dann bereits für das nächste Sendetelegramm verwendet.

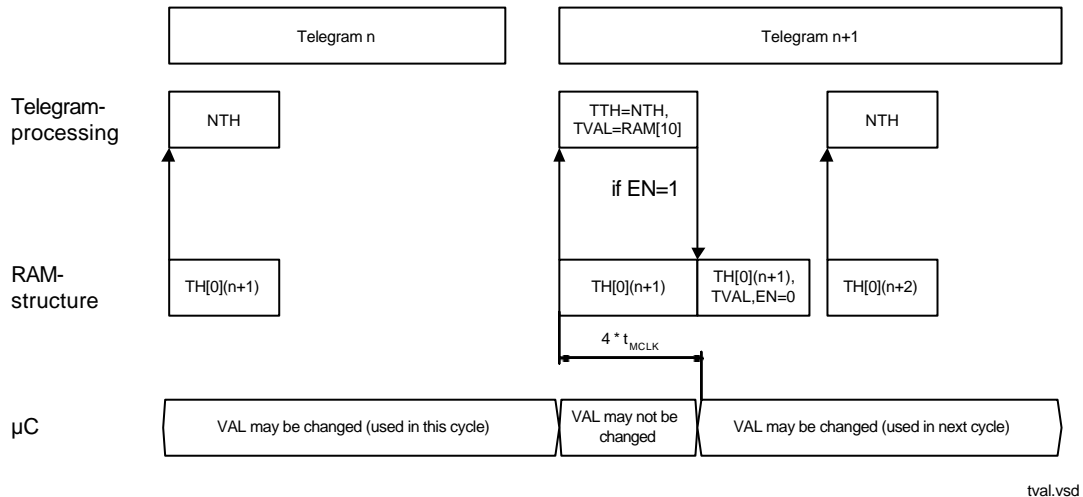


Abbildung 6: Lesen/Schreiben VAL-Bit bei Sendetelegrammen

Abbildung 6 zeigt schematisch diesen Ablauf. Nach dem Start eines Telegramms wird der Telegrammheader des nächsten erwarteten Telegramms ins Register NTH der TV gelesen. Beim Start des nächsten Telegramms wird NTH in das Register TTH kopiert und des Wert VAL wird erneut aus dem RAM gelesen (nur Sendetelegramme). Ist der Wert EN = 1, muß der Wert EN=0 ins RAM geschrieben werden. Da nur ganze Worte geschrieben werden können, werden auch alle anderen Steuerbits im Telegrammheader beschrieben. Für den Wert VAL wird der zuvor gelesene Wert TVAL geschrieben. Während des Lesens des Telegrammheaders und dem Zurückschreiben darf der μP den Telegrammheader nicht beschreiben!

Um das VAL-bit zur Steuerung des Doppelbuffers immer zu einem erlaubten Zeitpunkt zu Lesen, ist folgender Ablauf denkbar:

1. der μC liest den aktuellen Zeitpunkt im Sercos-Zyklus
2. der μC prüft ob noch ausreichend Zeit bis zum Start des Sendetelegramms vorhanden ist ($TCNTLT < t_{start} - dT$). Die Zeitdifferenz dT hängt von der Geschwindigkeit des Businterfaces und dem Takt MCLK ab.

3. wenn ja: VAL kann neu programmiert werden (VAL wird noch in diesem Zyklus übernommen)
4. wenn nein: warten bis Zeit größer als tstart + 4 (VAL wird im nächsten Zyklus übernommen)

Hinweis: Im SERCON410B ist dieser Ablauf, wie oben beschrieben, bereits implementiert. Für den SERCON816 wird der Ablauf überprüft und gegebenenfalls korrigiert. Es muß damit gerechnet werden, daß das Problem mit dem Doppelbuffer auf eine andere Ursache zurückzuführen ist, die nicht im Umfang der Entwicklung enthalten ist!

2.10.2 Kontrollwort für Service-Container

2.10.2.1 Fehlerbeschreibung

In einigen Anwendungen treten Probleme beim Beschreiben der Kontrollwörter für den Service-Container auf. In diesen Anwendungen wird das Kontrollwort am Ende eines Empfangstelegramms vom μ P neu beschrieben. Hierzu fragt der μ P die Interrupt-Statusbit für das Ende eines Empfangstelegramms (INT_REND) laufend ab. Zeigen die Statusbits das Ende eines Telegramms an, wird das Kontrollwort des Service-Containers vom μ P neu beschrieben. Hierbei kommt es zu Zugriffskonflikten zwischen dem μ P und der TV, da die TV ebenfalls nach dem Ende eines Empfangstelegramms mit der Bearbeitung der Service-Container beginnt.

2.10.2.2 Lösung

Es wird ein neuer Interrupt eingeführt (INT_RSCEND), der das Ende der Bearbeitung der Service-Container durch die TV anzeigt. Der neue Interrupt INT_RSCEND wird am Ende eines Empfangstelegramms aktiv, wenn keine Service-Container zu bearbeiten sind oder wenn die Bearbeitung der Service-Container abgeschlossen ist. Die Status- und Steuerbits des neuen Interrupts werden im Watchdog Steuerregister (28H) mit implementiert.

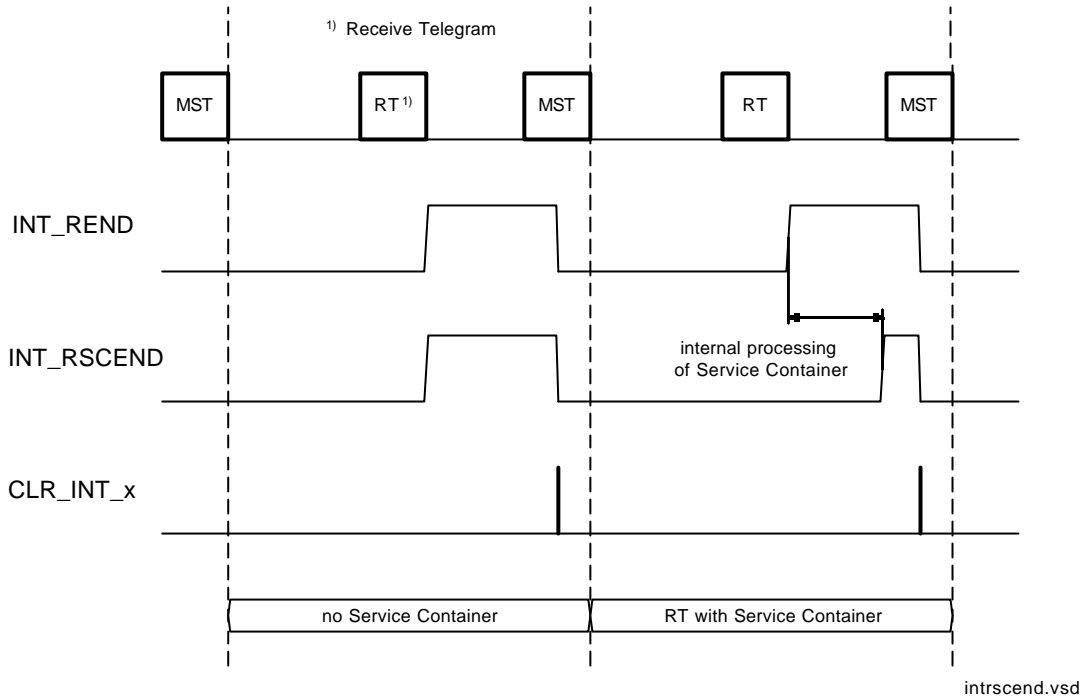


Abbildung 7: Interrupt RSCEND

2.11 Steuerregister

2.11.1 Geänderte Steuerregister

2.11.1.1 Register VERSION (00H)

In Adresse 0 der Steuerregister kann die aktuelle Version der verschiedenen SERCON Bausteine gelesen werden:

Baustein	VERSION (Adr. 0H)
SERCON410B	0002H
SERCON816	0010H

2.11.1.2 Register MCLKDIV, MCLKST (0EH)

Der Teiler für die Zeitbasis MCLKDIV und sein Startwert MCLKST wird um 1 Bit erweitert, um die höhere Taktfrequenz des SERCON816 herunter teilen zu können:

Baustein	MCLKDIV	MCLKST
SERCON410B	0EH.4-0	0EH.12-8
SERCON816	0EH.5-0	0EH.13-8

2.11.1.3 Register NDIVCLK (20H)

Adr	Bit	Name	R/W	Wert	Funktion
20H	7-0	NDIVCLK	R/W		Anzahl der DIV_CLK-Impuls für einen Sercos-Zyklus (DIVCLKMODE=0) oder Anzahl der Sercos-Zyklen, nach denen ein DIV_CLK-Impuls erzeugt wird (DIVCLKMODE=1)
	8	POLDIVCLK	R/W	0	DIV_CLK Pulse 1-aktiv
				1	DIV_CLK Pulse 0-aktiv
	9	DIVCLKMODE	R/W	0	wie SERCON410B (reset): DIV_CLK wird Normal in jedem Sercos-Zyklus aktiv
1				neuer Mode: DIV_CLK wird einmal nach N Sercos-Zyklen aktiv	
15-10			R	0	(nicht verwendet)

2.11.2 Neue Steuerregister für SERCON816

Adr	Bit	Name	R/W	Wert	Funktion
28H	0	WDOG_START	W	0	keine Aktion
				1	Watchdog einschalten
		WDOG_STAT	R	0	Watchdog aus (Reset)
				1	Watchdog ein
	1	INT_WDOG	R	0	Watchdog hat nicht angesprochen
				1	Watchdog hat angesprochen
		CLR_INT_WDOG	W	0	keine Aktion
				1	Lösche INT_WDOG
	2	WDOOUT_SET	W	0	keine Aktion
				1	Setze Freigabe für Ausgang WDOGN
		WDOOUT_STAT	R	0	Ausgang WDOGN gesperrt (Reset)
				1	Ausgang WDOGN freigegeben
	3	EN0_INT_WDOG	R/W	0	INT_WDOG löst INT0 nicht aus
				1	INT_WDOG löst INT0 aus
	4	EN1_INT_WDOG	R/W		Freigabe für INT1
	5	WDOG_MSTEN	R/W	0	Triggern über internen MST Start gesperrt
1				Master: Triggern über MST freigegeben Slave: keine Funktion	
6	INT_RSCEND	R	0	Interrupt RSCEND (Ende Service-Container Bearbeitung nach Telegramm Empfang) hat nicht angesprochen	
			1	Interrupt RSCEND hat angesprochen.	
	CLR_RSCEND	W	0	keine Aktion	
			1	Lösche INT_RSCEND	
7	EN0_INT_RSCEND	R/W	0	INT_RSCEND löst INT0 nicht aus	
			1	INT_RSCEND löst INT0 aus	
8	EN1_INT_RSCEND	R/W	0	INT_RSCEND löst INT1 nicht aus	
			1	INT_RSCEND löst INT1 aus	
29H	7-0	WDOG_TRIG	W	42H	Software Triggerung des Watchdogs
2AH	15-0	TWDOG	R/W		Watchdog Zeitkonstante