

SpaceWire-Link-Interface Handbuch (PL)

1. Memory-Mapped IO

Das Link-Interface ist in drei Bereiche aufgeteilt: Transmitting (TX), Receiving (RX) und Register (REG). Jedem davon wird in einem AXI-basierten Memory-Mapped-Aufbau eine eigene Offset-Adresse zugewiesen. Diese ist vor der endgültigen Implementierung unbekannt. Innerhalb des zugewiesenen Adressbereichs, ist die Adressverteilung konstant und wie folgt aufgebaut:

1.1. Transmitting (TX)

Offset_TX + 0x0000_0000 : Adresse des Transmit-FIFOs (write-only).

1.2. Receiving (RX)

Offset_RX + 0x0000_0000 : Adresse des Receive-FIFOs (read-only).

1.3. Register (REG)

Offset_REG + 0x0000_0000 : Link-Konfiguration (read/write)

Offset_REG + 0x0000_0004 : Transmit-Rate (read/write)

Offset_REG + 0x0000_0008 : Time-Codes (ausgehend) (read/write)

Offset_REG + 0x0000_000C : Time-Codes (eingehend) (read only)

Offset_REG + 0x0000_0010 : Link-Status (read only)

2. Speicherzugriff

Die Verwaltung und der Zugriff auf die Transmit bzw. Receive-Funktionalität erfolgt über das AXI4-Full-Interface. Die Register werden dagegen über AXI4-Lite angesprochen. Der grundlegende Unterschied ist, dass das Full-Interface Burst-Transfers ermöglicht, während die Lite-Version nur einzelne Datentransfers managen kann.

2.1. Transmitting (TX)

Um Daten in den Transmit-FIFO zu schreiben, muss die unter 1.1. angegebene Adresse als Zieladresse eines Write-Requests angegeben werden. Bei Burst-Transfers ist darauf zu achten, dass die Adresse nicht inkrementiert wird, sondern konstant bleibt.

2.2. Receiving (RX)

Um Daten aus dem Receive-FIFO zu lesen, muss die unter 1.2. angegebene Adresse als Quelladresse eines Read-Requests angegeben werden. Bei Burst-Transfers ist darauf zu achten, dass die Adresse nicht inkrementiert wird, sondern konstant bleibt.

2.3. Register (REG)

Zugriff auf einzelne Register sind mit einem einzelnen Read-/Write-Request und der entsprechenden Angabe der Ziel-/Quelladresse des Registers möglich. Die Konfiguration erlaubt nicht, Read-only-Register zu überschreiben, es ist dahingehend also keine besondere Vorsicht walten zu lassen.

3. Technische Beschreibung

Im folgenden werden die Elemente hinter den unter 1. angegebenen Adressen näher beschrieben und erläutert.

3.1. Transmitting (TX)

Daten, die in diesen FIFO-Block geschrieben werden, werden nach und nach durch ein Inbetriebgenommenes Link-Interface verschickt (siehe 2.1.). Befindet sich das Link-Interface in einem nicht-verbundenen Zustand, werden die Daten innerhalb des FIFOs gesammelt und verschickt sowie eine Verbindungsaufnahme des Links erfolgreich war.

Die Breite des FIFOs beträgt dabei 9 Bits, die wie folgt aufgeteilt sind:

Bit 8 : Flag Bit (1 : EOP/EEP, 0 : Daten)

Bit 7 – Bit 0 : zu übertragene Byte (0x00 für EOP, 0x01 für EEP)

Die Größe des FIFOs ist 18 kbit, in der Tiefe passen 2048 Datensätze in den FIFO.

Daten die über die 9 Bits hinaus an das FIFO gesendet werden, werden ignoriert und nicht in den Speicher geschrieben. Ein letztendlich gültiger Datentransfer sieht wie folgt aus: 0x0000_xxvv,

wobei xx entweder 0x00 oder 0x01 ist und vv ein beliebiges Datenbyte sein kann.

Zurückgesetzt werden kann der FIFO-Speicher mittels eines Resets über die CPU. Dabei wird der FIFO pseudo-entleert und kann neu mit Daten befüllt werden. Der Link wird keine vor dem Zeitpunkt des Resets eingefügte Daten mehr versenden.

Um zu verhindern, dass in einen vollen FIFO geschrieben wird, gibt es ein Ausgangssignal, welches den Status des FIFO anzeigt. Das Schreiben in einen vollen FIFO versucht keinen Schaden an der Hardware oder am System.

3.2. Receiving (RX)

Über SpaceWire empfangene Daten werden durch den Link nach und nach in diesen FIFO geschrieben und können durch das Processing System abgerufen werden (siehe 2.2.).

Die Breite des FIFOs beträgt dabei 9 Bits, die wie folgt aufgeteilt sind:

Bit 8 : Flag Bit (1 : Prozess ID/EOP/EEP, 0 : Daten)

Bit 7 – Bit 0 : empfangenes Datenbyte (0x00 für EOP, 0x01 für EEP)

Die Größe des FIFOs ist 18 kbit, in der Tiefe passen 2048 Datensätze in den FIFO.

Aus dem FIFO gelesene Datensätze enthalten zwei gültige Bytes (0x0000_xxvv), wobei xx entweder 0x00 oder 0x01 und vv ein beliebiges Datenbyte sein kann. Ist xx = 0x01, muss vv unterschieden werden: Gilt für vv = 0x00 (EOP), 0x01 (EEP), größer als 0x01 (Prozess ID)

Zurückgesetzt werden kann der FIFO-Speicher mittels des Resets des Links (PL, unabhängig von CPU Reset!). Dabei wird der FIFO pseudo-entleert und kann neu mit

Daten befüllt werden. Das Processing System kann dabei keine Daten mehr lesen, die vor dem Zeitpunkt des Resets in den Speicher geschrieben wurden.

Um zu verhindern, dass aus einem leeren FIFO gelesen wird, gibt es ein entsprechendes Ausgangssignal, welches den Status des FIFO anzeigt. Das Lesen aus einem leeren FIFO versucht keinen Schaden an der Hardware oder am System.

3.3. Register (REG)

Die Breite des gesamten Registers beträgt 32 Bit. Je nach Adresse ist die Belegung der einzelnen Bits unterschiedlich.

1.3.1. - Link-Konfiguration: (read/write)

Bit **0** : linkdis
Bit **1** : linkstart
Bit **2** : autostart
Bit **31** – Bit **3**: *unbelegt*

linkdis	Deaktiviert den Link, der Empfang und das Versenden von Nachrichten via SpaceWire ist nicht möglich. Standardwert: 0. 1 : Link deaktiviert 0 : Link nicht-deaktiviert
linkstart	Aktiviert den Link, es wird aktiv versucht, eine Verbindung mit der gegenüberliegenden Stelle aufzubauen. Dabei wird in regelmäßigen Intervallen ein Verbindungsversuch unternommen. Standardwert: 0. 1 : Link starten 0 : Link nicht-starten
autostart	Aktiviert den Link, es wird jedoch nicht aktiv versucht, eine Verbindung mit der gegenüberliegenden Stelle aufzubauen. Auf Verbindungsversuche der gegenüberliegenden Stelle wird jedoch verbindungs Aufbauend reagiert. Standardwert: 0. 1 : Automatischer Verbindungsaufbau aktiviert 0 : Automatischer Verbindungsaufbau deaktiviert

1.3.2. - Transmit Rate (read/write)

Bit **7** – Bit **0** : txdivcnt
Bit **31** – Bit **8** : *unbelegt*

txdivcnt<7:0>	Skalierungsfaktor minus 1. Wird benutzt um die Sendebitrate aus dem Transmit-Takt abzuleiten. Standardwert: 0x01.
---------------	---

1.3.3. - Time-Codes (ausgehend) (read/write)

Bit **5** – Bit **0** : Counter Value
Bit **7** – Bit **6** : *unbelegt*
Bit **9** – Bit **8** : Control Flag
Bit **31** – Bit **10** : *unbelegt*

Counter Value <5:0>	Wert des Time-Codes, der zur Synchronisierung benutzt wird (Wertebereich: 0-63)
Control Flag <1:0>	Flag des Time-Codes. Standardwert: 0b00.

1.3.4. - Time-Codes (eingehend) (read only)

Bit **5** – Bit **0** : Counter Value
 Bit **7** – Bit **6** : *unbelegt*
 Bit **9** – Bit **8** : Control Flag
 Bit **31** – Bit **10** : *unbelegt*

Counter Value <5:0>	Wert des Time-Codes, der zur Synchronisierung benutzt wird (Wertebereich: 0-63)
Control Flag <1:0>	Flag des Time-Codes.

1.3.5. - Link-Status (read only)

Bit **0** : started
 Bit **1** : connecting
 Bit **2** : running
 Bit **7** – Bit **3** : *unbelegt*
 Bit **8** : errdisc
 Bit **9** : errpar
 Bit **10** : erresc
 Bit **11** : errcred
 Bit **15** – Bit **12** : *unbelegt*
 Bit **16** : rxhalff
 Bit **17** : txhalff
 Bit **31** – Bit **18** : *unbelegt*

started	Zeigt an, ob der SpaceWire Link im started-Zustand ist. (Nicht verbunden, aber bereit eine Verbindung aufzunehmen)
connecting	Zeigt an, ob der SpaceWire Link im connecting-Zustand ist. (Verbindungsaufnahme im Gange)
running	Zeigt an, ob der SpaceWire Link mit einem anderen Link erfolgreich verbunden und lauffähig ist.
errdisc	Zeigt an ob ein Disconnect entdeckt wurde. Der Link cleart diesen Fehler automatisch und versucht eine erneute Verbindungsaufnahme.
errpar	Zeigt an ob ein Paritätsfehler während einer Datenübertragung entdeckt wurde. Der Link cleart diesen Fehler automatisch und versucht eine erneute Verbindungsaufnahme.
erresc	Zeigt an ob ein Escape-Fehler während der Datenübertragung entdeckt wurde. Der Link cleart diesen Fehler automatisch und versucht eine erneute Verbindungsaufnahme.
errcred	Zeigt an ob ein Credit-Fehler entdeckt wurde. Der Link cleart

	diesen Fehler automatisch und versucht eine erneute Verbindungsaufnahme.
rxhalff	Zeigt an ob der interne Eingangs-FIFO des Link (nicht der RX FIFO unter 1.2.) zur Hälfte befüllt ist.
txhalff	Zeigt an ob der interne Ausgangs-FIFO des Link (nicht der TX FIFO unter 1.1.) zur Hälfte befüllt ist.

4. Steuerung

4.1. Inbetriebnahme

Um den Link in Betrieb zu nehmen sind folgende Schritte erforderlich:

1. Ggf. Durchführen eines globalen Resets um alle Elemente erst-zu-initialisieren.
2. Schreiben des Wertes 0x0000_0002 in das Link-Konfigurationsregister (1.3.1.)
3. Warten bis Verbindung aufgebaut ist (dies ist der Fall wenn im Status-Register (1.3.5.) das dritte Bit 1 ist. (Leicht abzufragen mit der Bitmaske 0x0000_0004)
4. Ab jetzt können Daten verschickt und empfangen werden.

4.2. Außerbetriebnahme

Um den Link zu deaktivieren oder abzuschalten genügt es, in das Link-Konfigurationsregister den Wert 0x0000_0001 zu schreiben. Damit wird linkdis aktiviert, welche alle anderen Einstellung innerhalb dieses Registers überschreibt.