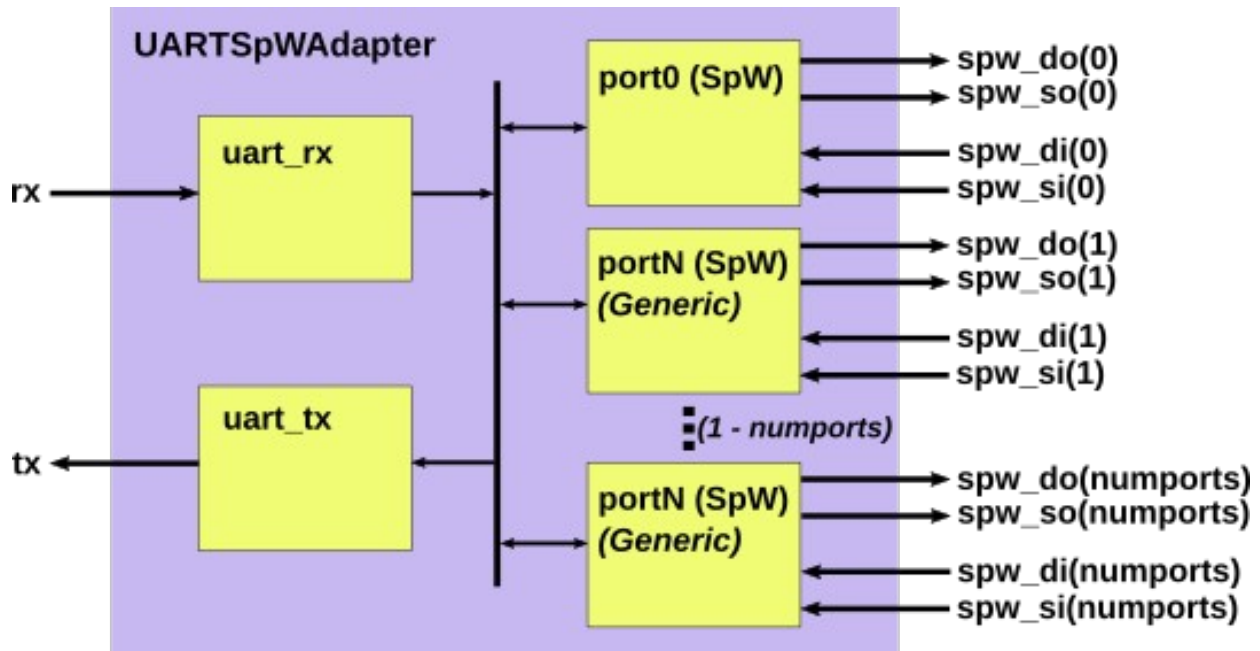


UART – SpaceWire Adapter



Beschreibung: Ermöglicht die Übersetzung von über UART gesendeten Daten zu SpaceWire und umgekehrt. Dabei existiert eine vorkonfigurierte Variante des Adapters, bei der die Konfiguration nach der Implementierung nicht mehr verändert werden kann und eine auf Kommandos basierte Version, bei der die Konfiguration dynamisch im Betrieb verändert werden kann. Dies geht allerdings auf Kosten der frei belegbaren Bits für Daten, da bestimmte Konstellationen für Kommandos reserviert sind. Über die generische Eingabe `activate_commands` wird dies bei der Implementierung gesteuert.

Interface:

Konfiguration durch VHDL Generics:

Name	Beschreibung	Typ	Standardwert
<code>clk_cycles_per_bit</code>	Quotient aus Systemtaktfrequenz und der UART Baudrate. (Bsp.: 100 MHz / 115200 1/s = 868)	Integer	-/-
<code>numports</code>	Anzahl der SpaceWire Ports, die der Adapter besitzen soll. (0-31)	Integer	-/-
<code>init_input_port</code>	Initialer SpaceWire Port über den eingehender UART Datenverkehr verschickt werden soll.	Integer	0
<code>init_output_port</code>	Initialer SpaceWire Port über den eingehender SpaceWire Datenverkehr zurück über UART verschickt werden soll.	Integer	0
<code>activate_commands</code>	Legt fest ob der unveränderbare Adapter (false) oder der Kommando-basierte Adapter (true) verwendet werden sollen.	Boolean	-/-

sysfreq	Frequenz des Systemtaktes in Hz. Wird benötigt um Counter innerhalb des SpaceWire Ports aufzusetzen.	Real	-/-
txclkfreq	Transmit Systemtakt in Hz. Wird nur benötigt wenn das impl_fast-Front-End verwendet wird. (Siehe Anleitung SpaceWire Light)	Real	0.0
rximpl	Auswahl des Receiver-Front-Ends der SpaceWire Ports des Adapters. (Siehe Anleitung SpaceWire Light)	spw_implementation_type_rec (s. spwpkg.vhd)	-/-
rxchunk	Maximale Anzahl an Bits die innerhalb eines Taktes empfangen werden können. (1-4)	Integer	1
WIDTH	Anzahl an Flipflops, die zum Einsynchronisieren von SpaceWire Signalen verwendet werden sollen. (Wird nur benötigt falls das Taktrückgewinnungs-Front-End verwendet wird. (impl_clkrec))	Integer	2
tximpl	Auswahl des Transmitter-Front-Ends der SpaceWire Ports des Adapters. (Siehe Anleitung SpaceWire Light)	spw_implementation_type_xmit (s. spwpkg.vhd)	-/-
rxfifo_size_bits	Größe des FIFO-Eingangsspeichers eines SpaceWire Ports des Adapters. (Bsp.: 6 entspricht 64 bytes (log2(64))). (6-14)	Integer	11
txfifo_size_bits	Größe des FIFO-Ausgangsspeichers eines SpaceWire Ports des Adapters.	Integer	11

Folgende Ein- und Ausgangssignale sind in der VHDL Entity definiert:

Name		Beschreibung
clk	in	Systemtakt
rxclk	in	Receiver-Takt (wird nur für impl_fast) benötigt.
txclk	in	Transmitter-Takt (wird nur für impl_fast) benötigt.
rst	in	Synchroner Reset.
autostart<numports:0>	in	Aktiviert automatischen Start eines SpaceWire Ports sobald eine NULL empfangen wird.
linkstart<numports:0>	in	SpaceWire Port startet und beginnt damit NULLEN zu verschicken. Ohne autostart oder linkstart verbleibt ein SpaceWire Port im ‚ready‘-Zustand.
linkdis<numports:0>	in	Deaktiviert einen SpaceWire Port.
txdivcnt<7:0>	in	Skalierungsfaktor minus 1. Wird genutzt um die Versenderate des Transmitters zu konfigurieren. (Siehe Anleitung SpaceWire Light)

started<numports:0>	out	Zeigt an, welcher SpaceWire Port sich im ,started‘-Zustand befindet.
connecting<numports:0>	out	Zeigt an, welcher SpaceWire Port sich im ,connecting‘-Zustand befindet.
running<numports:0>	out	Zeigt an, welcher SpaceWire Port sich im ,running‘-Zustand befindet.
errdisc<numports:0>	out	Disconnect im ,running‘-Zustand aufgetreten. Löst einen Reset des SpaceWire Ports und einen Reconnect aus.
errpar<numports:0>	out	Paritätsfehler im Datenverkehr aufgetreten. Löst einen Reset des SpaceWire Ports und einen Reconnect aus.
erresc<numports:0>	out	Ungültige Escape Sequenz entdeckt. Löst einen Reset des SpaceWire Ports und einen Reconnect aus.
errcred<numports:0>	out	Credit-Fehler entdeckt. Löst einen Reset des SpaceWire Ports und einen Reconnect aus.
txhalff<numports:0>	out	Gibt an, wenn mindestens die Hälfte des FIFO-Ausgangsspeichers eines SpaceWire Ports belegt ist.
rxhalff<numports:0>	out	Gibt an, wenn mindestens die Hälfte des FIFO-Eingangsspeichers eines SpaceWire Ports belegt ist.
spw_di<numports:0>	in	SpaceWire Data In
spw_si<numports:0>	in	SpaceWire Strobe In
spw_do<numports:0>	out	SpaceWire Data Out
spw_so<numports:0>	out	SpaceWire Strobe Out
rx	in	Eingehender UART Stream.
tx	out	Ausgehender UART Stream.

UART Kommandos:

Grundlegende Syntax:

0 X₆ X₅ X₄ X₃ X₂ X₁ X₀

1 X₆ X₅ X₄ X₃ X₂ X₁ X₀

Datenbyte oder Time Code
Kommando

Spezifische Darstellung der Kommandos:

0 0 D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	Datenbyte (D ₅ -D ₀ , Paketanfang/-inhalt)
0 1 T ₅ T ₄ T ₃ T ₂ T ₁ T ₀	Time Code (T ₅ -T ₀ entspricht Time Code Counter Value)
1 0 0 0 0 - - -	Reset (setzt alle Kommandos zurück)
1 0 0 0 1 - - -	Ausgabe Information1

1 0 0 1 0 - - -	Ausgabe Information2
1 0 0 1 1 - - -	Ausgabe Information3
1 0 1 P₄ P₃ P₂ P₁ P₀	Eingangsport festlegen (P ₄ -P ₀)
1 1 0 P₄ P₃ P₂ P₁ P₀	Ausgangsport festlegen (P ₄ -P ₀)
1 1 1 0 - - - -	EOP (End of Packet)
1 1 1 1 - - - -	EEP (Error End of Packet)