Spinnaker to Neuelab [Spin2Neu]

Author: Francesco Diotalevi

Company: Istituto Italiano di Tecnologia - EDL

History:			
Revision	**Date**	**Author**	**Description**
1.0	May 16th, 2018	Francesco Diotalevi	First Release (DRAFT) after G. De Robertis design in 2015

Contents

Introduction

The Spin2Neu design aims to deliver and receive data to a spinnaker through 2 different unidirectional links: the input SpiNNaker Link and the output SpiNNaker link.

The user sends data to spinnaker by using the *Axi4 Stream Output Bus lines* and receives data from spinnaker by using the Axi4 Stream Input Bus lines.

The user can control the whole data processing thanks to the AXI 4 Lite Bus interface that is able to communicate with the Spin2Neu internal registers that will be detailed in Section 4.

1.1 Input SpiNNaker Link

The input SpiNNaker Link consists in 7 data wires named *data_2of7_from_spinnaker* and an acknowledge wire *ack_to_spinnaker*.

1.2 Output SpiNNaker Link

The output SpiNNaker Link consists in 7 data wires named *data_2of7_to_spinnaker* and an acknowledge wire *ack_from_spinnaker*.

In the Table 1 the input and output ports of the Spin2Neu module are listed.

Table 1 Spin2Neu interface signals description

Comment	Port name	Width	Dir	Description
AXI4 Lite Bus lines	S_AXI_ACLK	1	I	AXI Clock, System clock line
	S_AXI_ARESETN	1	I	AXI Reset active low line
	S_AXI_AWADDR	32	I	AXI Write address
	S_AXI_AWVALID	1	I	Write address valid
	S_AXI_WDATA	32	I	Write data
	S_AXI_WSTRB	4	I	Write strobes
	S_AXI_WVALID	1	I	Write valid
	S_AXI_BREADY	1	I	Response ready
	S_AXI_ARADDR	32	I	Read address
	S_AXI_ARVALID	1	I	Read address valid
	S_AXI_RREADY	1	I	Read ready
	S_AXI_ARREADY	1	O	Read address ready
	S_AXI_RDATA	32	O	Read data
	S_AXI_RRESP	2	O	Read response
	S_AXI_RVALID	1	O	Read valid
	S_AXI_WREADY	1	O	Write ready
	S_AXI_BRESP	2	O	Write response
	S_AXI_BVALID	1	O	Write response valid
	S_AXI_AWREADY	1	O	Write address ready
AXI4	S_AXIS_TREADY	1	O	Ready to receive data

Spinnaker to Neuelab [Spin2Neu]

Comment	Port name	Width	Dir	Description
	S_AXIS_TVALID	1	I	Read valid
	S_AXIS_TDATA	32	I	Read data
	S_AXIS_TLAST	1	O	Signal indicating the last data of the burst
Axi4 Stream Output Bus lines	M_AXIS_TREADY	1	I	Ready to receive data
	M_AXIS_TVALID	1	O	Write valid
	M_AXIS_TDATA	32	O	Write data
	M_AXIS_TLAST	1	O	Signal indicating the last data of the burst
Input Spinnaker Link	Data_2of7_from_spinnaker	7	I	Self-timed 2-of-7 protocol data input
	Ack_to_spinnaker	1	O	Acknowledge signal to spinnaker chip
Input Spinnaker Link	Data_2of7_to_spinnaker	7	O	Self-timed 2-of-7 protocol data output
	Ack_from_spinnaker	1	I	Acknowledge signal from spinnaker chip
	nRst	1	I	Active Low reset signal
	Clk_Spinn	1	I	Spinnaker clock
	Clk_Core	1	I	Core clock
	Interrupt_o	1	O	Interrupt active high signal for micro
	LpbkDefault	3	I	Loopback configure [1]

2 Spin2Neu Diagram

The Figure 1 shows all the modules used in the Spin2Neu design. The Spin2Neu design integrates the following modules that will be explained into the respective subsections into the next chapter:

- Spinn_Neu I/f,
- CoreMonSeq,
    - Monitor,
    - TimeStamper,
    - AEXSequencer,
    - InFIfo,
    - OutFifo
    - Timestamp wrap detector,
- AXI Stream I/f,
- AXI Lite REGs I/f.

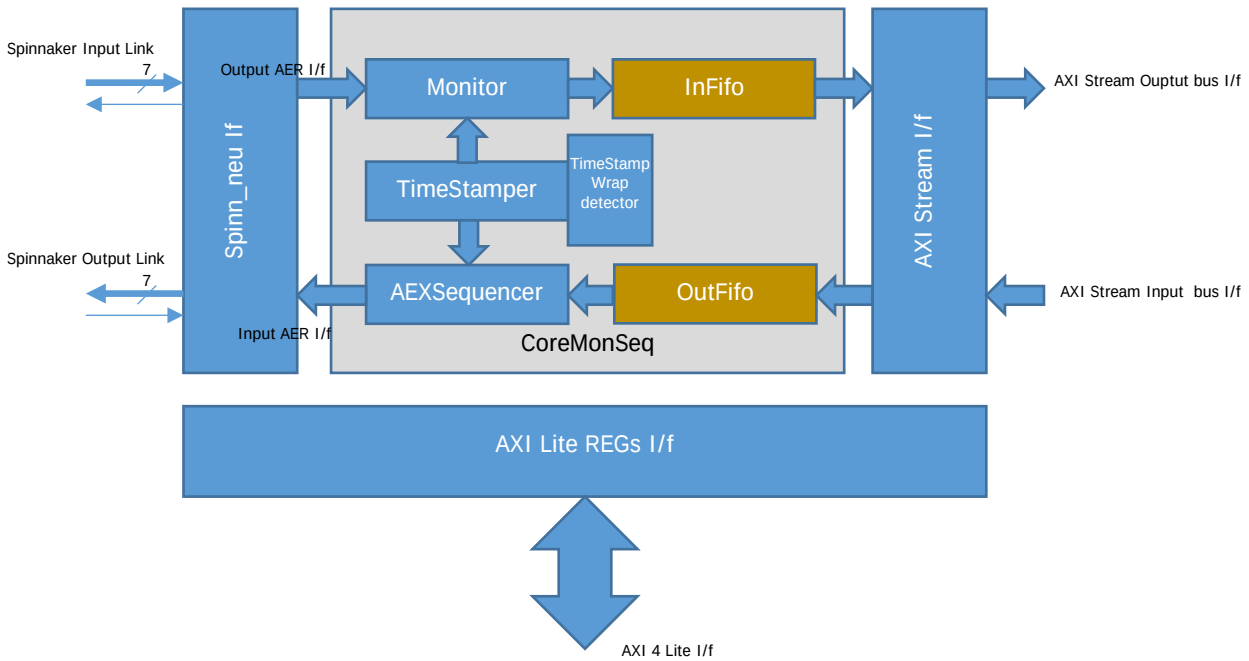


Figure 1 Spin2Neu architecture

3 Main components of the Spin2Neu architecture

In this Section we will describe the main components of the Spin2Neu architecture.

3.1 Spinn_neu I/f

This module (coming from the University of Manchester) is used to interface the *Spinnaker input link* and *Spinnaker Output link* with, respectively, the *output AER I/f* and the *input AER I/f*.

3.2 CoreMonSeq

The *CoreMonSeq* module by using one Input Fifo and one Output Fifo can manage the two events data flows that are the two spinnaker links. Using an internal 32 bit counter, it can identify any data with the specific time stamp.

3.2.1 Monitor

This module extract the data from the output AER I/f and feeds the InFifo with the data and relative time stamp.

3.2.2 Timestamper

This module is a 32 bit counter used to identify the events coming from/to the spinnaker links.

3.2.3 AEXSequencer

This module takes the data coming from the OutFifo and generates (depending on the timing information inside the couple Data/Timestamp of the OutFifo) the AER Input I/f of the *Spinn_neu I/f* module.

3.2.4 InFIfo

This Fifo is 2048 data of 64bit length. Both Fifo ports are 64 bits sized.

3.2.5 OutFIfo

This Fifo is 2048 data of 64bit length. The input port is 32 bits sized and the output port is 64 bits sized.

3.2.6 TimeStampWrap detector

This module is used to detect the wrapping of the *Timestamper*.

3.3 AXIStream I/f

The *AXIStream I/f* interfaces the AXI stream output and input I/f with the FIFOs interfaces.

3.4 AXI4Lite I/f

The AXI4Lite I/f module is used to interface the CPU with the internal Spin2Neu registers.

4 Spin2Neu internal registers

In this Section a detailed view of the registers internal to the Spin2Neu module.

	Offset	Mnemonic	Description	Type	Reset Value
	0x00	Control register (CTRL_REG)	Control register	rw	0x0?000000
	0x08	RX DATA buffer register (RXDATA_REG)	RX Data Buffer	ro	0x00000000
	0x0C	RX Time buffer register (RXTIME_REG)	RX Time Buffer	ro	0x00000000
	0x10	TX Data buffer register (TXDATA_REG)	TX Data Buffer	rw	0x00000000
	0x14	DMA register (DMA_REG)	DMA Burst Register	rw	0x00000100
	0x18	RAW Status Register (STAT_RAW_REG)	Status RAW register	ro	0x00000000
	0x1C	IRQ Register (IRQ_REG)	IRQ register	rc	0x00000000
	0x20	Mask Register (MSK_REG)	Mask register for the IRQ_REG register	rw	0x00000000
	0x28	Wrapping TimeStamp Register (WRAPTimeStamp_REG)	Wrapping TimeStamp Register	rc	0x00000000
	0x5C	Identification register (ID_REG)	ID Register	ro	0x53324E10

4.1 Control register (CTRL_REG)

This register is used to control the Spin2Neu IP.

CTRL_REG (Spin2Neu Base + 0x00) Reset Value: **0x0?000000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					Local Far LB	Local LB	Remote LB	Reserved							
					ro	ro	ro								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			Rearm DMA	Reserved							Flush Fifos	Reserved	IE enable	EN DMA	Reserved
			wc								wc		r/w	r/w	

- En DMA
  - If '1' the DMA I/f is enabled
- IE Enable
  - If '1' the Spin2Neu generates Interrupt depending on the unmasked bit of
- FlushFifo
  - If '1' the internal Spin2Neu Fifos are flushed. It's automatically cleared by the hardware.
- RearmDMA
  - If '1' the DMA is re-armed. It's automatically cleared by the hardware.
- Remote Loopback,
- Local Loopback and
- LocalFar Loopback
  - See **Error! Reference source not found.** for details.

4.2 RX DATA buffer register (RXDATA_REG)

This register contains the data (read from the InFifo) coming from the output AER I/f of the Spinn_neu I/f module.

RXDATA_REG (Spin2Neu Base + 0x08) Reset Value: **0x00000000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Data															
ro															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data															
ro															

- Data
  - 32 bit length data

4.3 RX Time buffer register (RXTIME_REG)

This register contains the timestamp associated to the received data (see RX DATA buffer register (RXDATA_REG)) from the output AER I/f of the Spinn_neu I/f module.

The time stamp value read from this register is the time stamp that the Spinn2Neu module sticks to the received Data available into the RX DATA buffer register (RXDATA_REG).

RXTIME_REG (Spin2Neu Base + 0x0C) Reset Value: **0x00000000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TimeStamp															
ro															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TimeStamp															
ro															

- TimeStamp
  - 32 bit length data

4.4 TX Data buffer register (TXDATA_REG)

This register is used to fill the OutFifo.

TXDATA_REG (Spin2Neu Base + 0x10) Reset Value: **0x00000000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXData															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXData															
rw															

When writing to this register, keep in mind that it is used by the internal hardware as follows: The register needs to be written twice to enable the correct behaviour. The first data written into the register represents the time, elapsed which, the second written data into this register is delivered to the Spinn_neu I/f module.

4.5 DMA register (DMA_REG)

This register is used to set the behaviour of the AXIStream interface. It represents the number of data (32 bit size length) sent to the DMA interface.

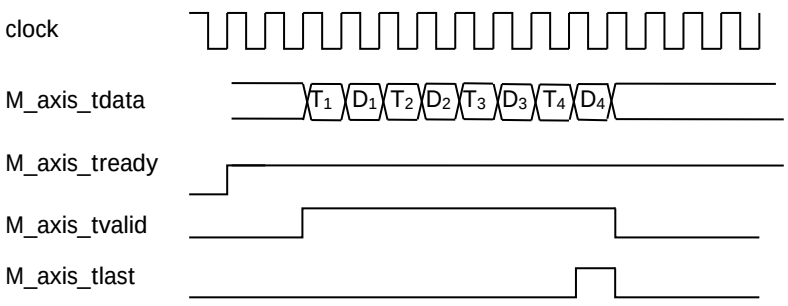
NOTE: This register can be written only if CTRL_REG.ENDMA='0'.

DMA_REG (Spin2Neu Base + 0x14) Reset Value: **0x000001n00**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					DMA Lenght										
					rw										

For example, if it is set to 8, the burst from/to the AXI Stream I/f will be in terms of 8 data length.



4.6 RAW Status Register (STAT_RAW_REG)

When read, this register gives a snapshot of the status of warning or errors signals. It is a Read Only register.

STAT_RAW_REG (Spinn2Neu Base + 0x18) Reset Value: **0x00000000**

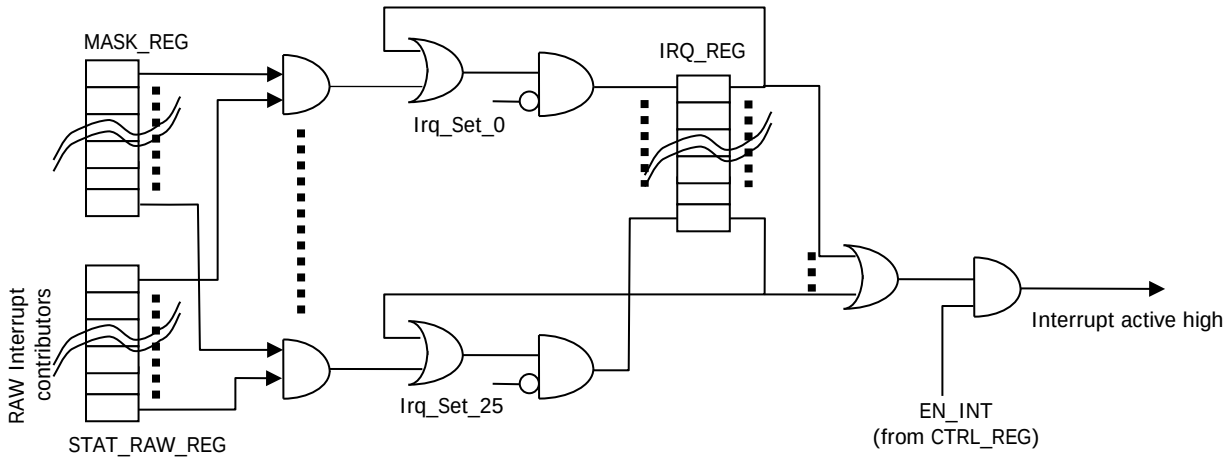
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RxModemErr	RXParityErr	TxDumpMode	Reserved		RX FIFO Not Empty	RxBufferReady	Time Stamp Wrapped	Reserved	TX Data Full	TX Data Almost Full	TX Data Empty	RX Data Full	RX Data Almost Empty	RX Data Empty
	ro	ro	ro			ro	ro	ro		ro	ro	ro	ro	ro	ro

- RxDataEmpty
  - When ‘0’, the INFIFO is not empty
  - When ‘1’ the INFIFO is empty
- RxDataAlmostEmpty
  - When ‘1’ the INFIFO has 1 or 0 data to be read.
  - When ‘0’ the INFIFO has more or equal two data to be read.
- RxDataFull
  - When ‘1’ the INFIFO is full.
  - When ‘0’ the INFIFO is not full.
- TxDataEmpty
  - When ‘0’, the OUTFIFO is not empty
  - When ‘1’ the OUTFIFO is empty
- TxDataAlmostFull
  - When ‘1’ the OUTFIFO has 2047 or 2048 data within himself.
  - When ‘0’ the OUTFIFO has less than 2047 data within himself.
- TxDataFull
  - When ‘1’ the OUTFIFO is full.
  - When ‘0’ the OUTFIFO is not full.
- Time stamp wrapped (this bit is high for one clock period only, when the counter wraps its value)
  - When ‘1’ the counter inside the TimeStamp module has wrapped its value.
  - When ‘0’ the counter inside the TimeStamp module has not yet wrapped its value
- RXBufferReady
  - When ‘1’ the Rx Fifo has at least DMA_REG value of data available
  - When ‘0’ the Rx Fifo has less than DMA_REG value of data available
- RXFifoNotEmpty
  - When ‘1’ the RX Fifo is not empty.
  - When ‘0’ the RX Fifo is empty
- TxDumpMode
  - When ‘1’ the spinn_neu I/f module through its in_mapper module has highlighted a DumpMode event.
- RXParityErr
  - When ‘1’ the spinn_neu I/f module through its out_ module has highlighted a Parity error event.
- RxModeErr
  - When ‘1’ the spinn_neu I/f module through its spinn_receiver module has highlighted a Rx Mode error event.

4.7 IRQ Register (IRQ_REG)

When read, this register gives the status of the collected warning or errors signals. It is a Read/Set register, i.e., to clear the warning/error bit the user has to write '1' on the corresponding bit position.



IRQ_REG (Spinn2Neu Base + 0x1C) Reset Value: **0x00000000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RxModemErr	RXParityErr	TxDumpMode	Reserved		RX FIFO Not Empty	RxBufferReady	Time Stamp Wrapped	Reserved	TX Data Full	TX Data Almost Full	TX Data Empty	RX Data Full	RX Data Almost Empty	RX Data Empty
	r/c	r/c	r/c			r/c	r/c	r/c		r/c	r/c	r/c	r/c	r/c	r/c

The meaning of the masked contributors of this register is the same of the RAW Status Register (STAT_RAW_REG).

4.8 Mask Register (MSK_REG)

This is the Mask register used to mask the contributors for the interrupt signal.

MSK_REG (Spinn2Neu Base + 0x20) Reset Value: **0x00000000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	RxModemErr	RXParityErr	TxDumpMode	Reserved		RX FIFO Not Empty	RxBufferReady	Time Stamp Wrapped	Reserved	TX Data Full	TX Data Almost Full	TX Data Empty	RX Data Full	RX Data Almost Empty	RX Data Empty
	r/w	r/w	r/w			r/w	r/w	r/w		r/w	r/w	r/w	r/w	r/w	r/w

The meaning of the masked contributors of this register is the same of the RAW Status Register (STAT_RAW_REG).

4.9 Wrapping TimeStamp Register (WRAPTimeStamp_REG)

This register is used to read how many times the internal 32bit counter of the TimeStamp module has wrapped its value.

In case the user writes any value in this register, it will be cleared and also the internal 32bit counter of the TimeStamp module will be cleared.

WRAPTIMESTAMP_ REG (Spinn2Neu Base + 0x28) Reset Value: **0x00000000**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Wrapping times															
r/c															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Wrapping times															
r/c															

4.10 Identification register (ID_REG)

This register contains the ID of the Spinn2Neu module.

ID_REG (Spin2Neu Base + 0x5C) Reset Value: **0x53324E10**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
S								2							
r/o								r/o							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N								Major				Minor			
r/o								r/o				r/o			

Minor = 1;

Major = 0;

5 References

[1] "Application Notes for Spin2Neu", by F. Diotalevi, May 2016.

6 Appendixes