FPGA Firmware for new Kintex UltraScale board (AXIStream)

Matthäus Krein

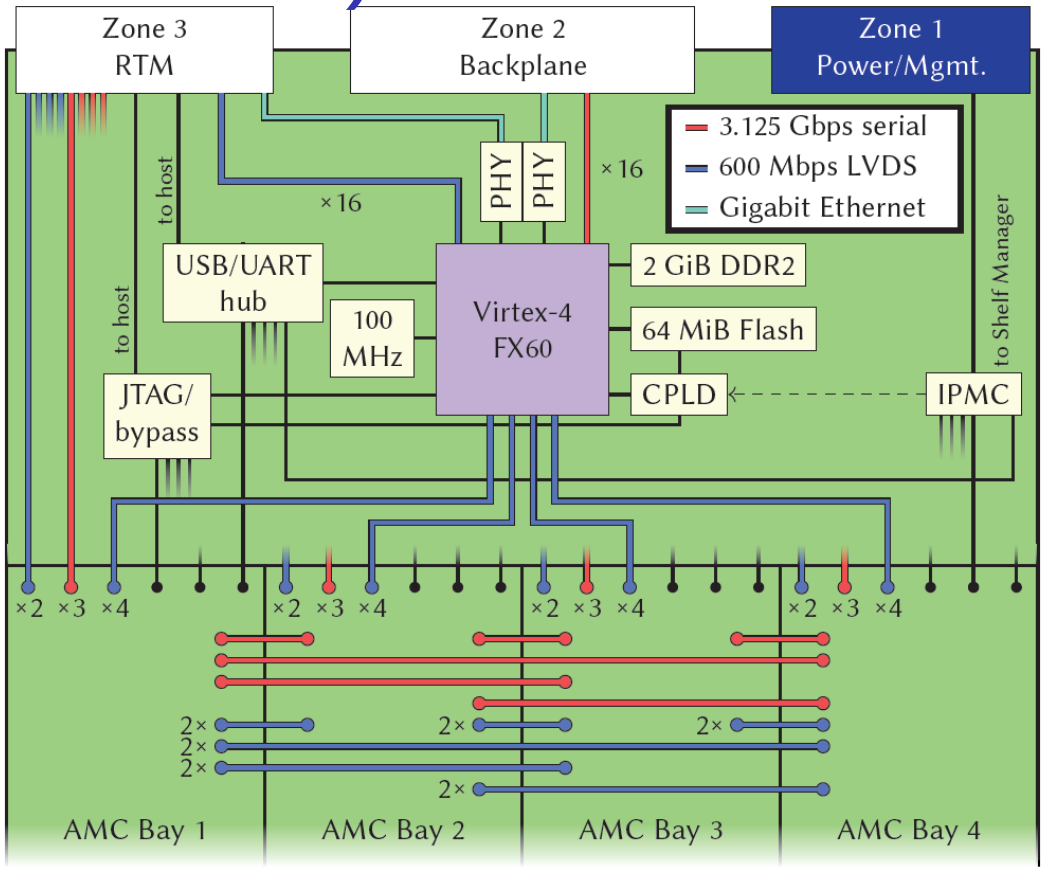
JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN

Fast Realtime Systems Workshop

April 09, 2024

Current Hardware (CNCB v3.3)

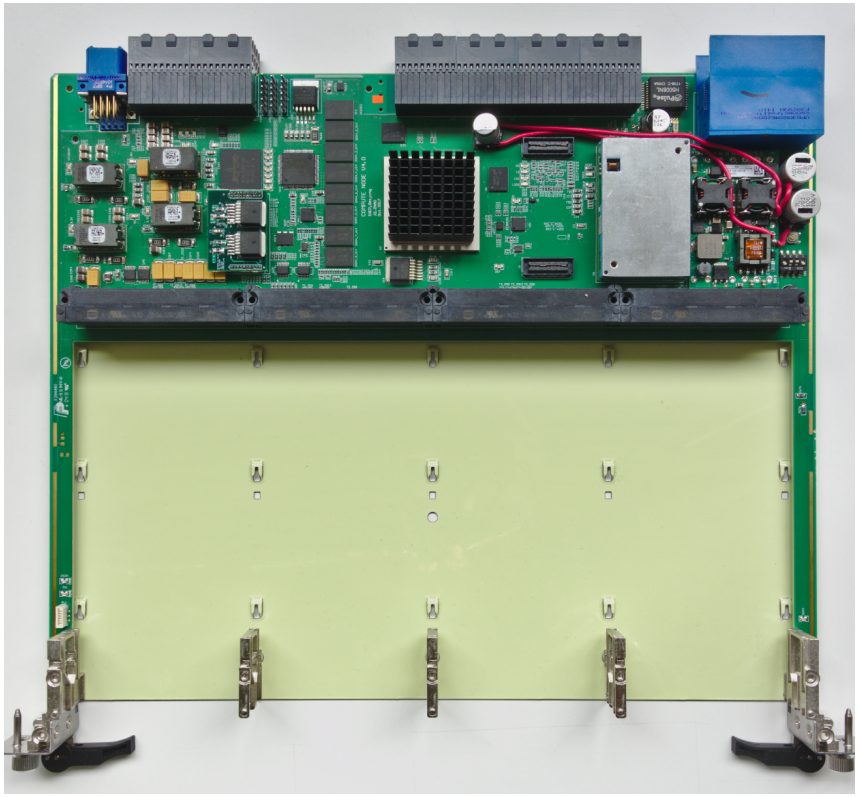
- Hardware components of the Carrier are outdated and not produced any more
- Development of a new Carrier with updated/alternative components
- For the existing firmware adjustments are necessary



Compute Node Carrier Board (CNCB) v3.3

Hardware Upgrade (CNCB v4.0)

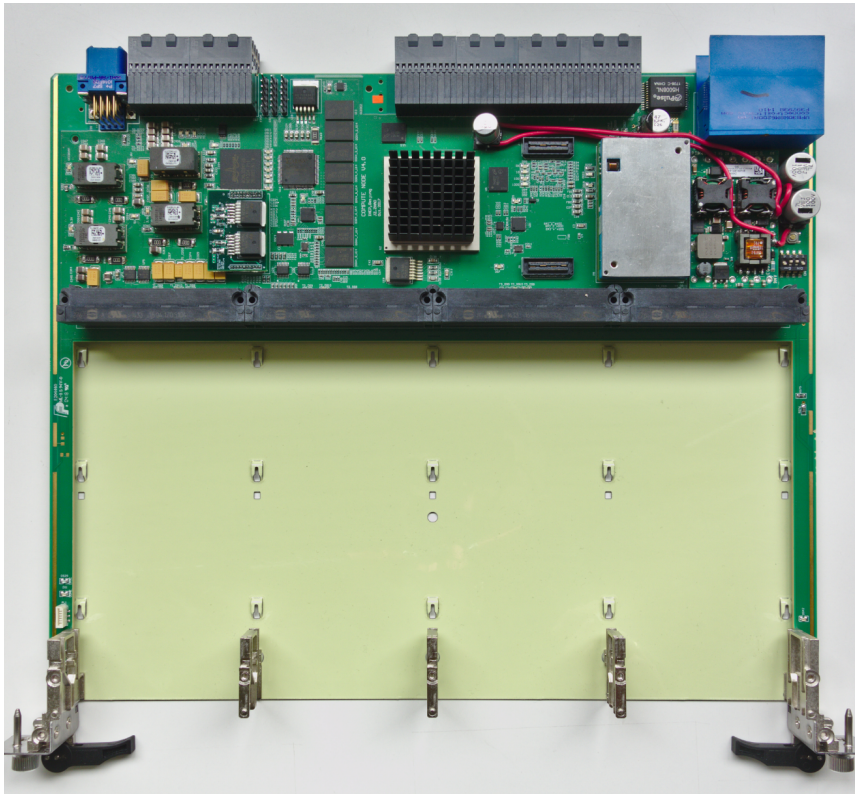
	CNCB v3.3	CNCB v4.0
FGPA	Virtex 4	Kintex UltraScale



Compute Node Carrier Board (CNCB) v4.0

Hardware Upgrade (CNCB v4.0)

	CNCB v3.3	CNCB v4.0
FGPA	Virtex 4	Kintex UltraScale
Registers	50k	663k
LUTs	50k x 4-input	332k x 6-input
BRAM	4 Mb	38 Mb
MGT	16 x 6.5 Gbps	32 x 16.3 Gbps

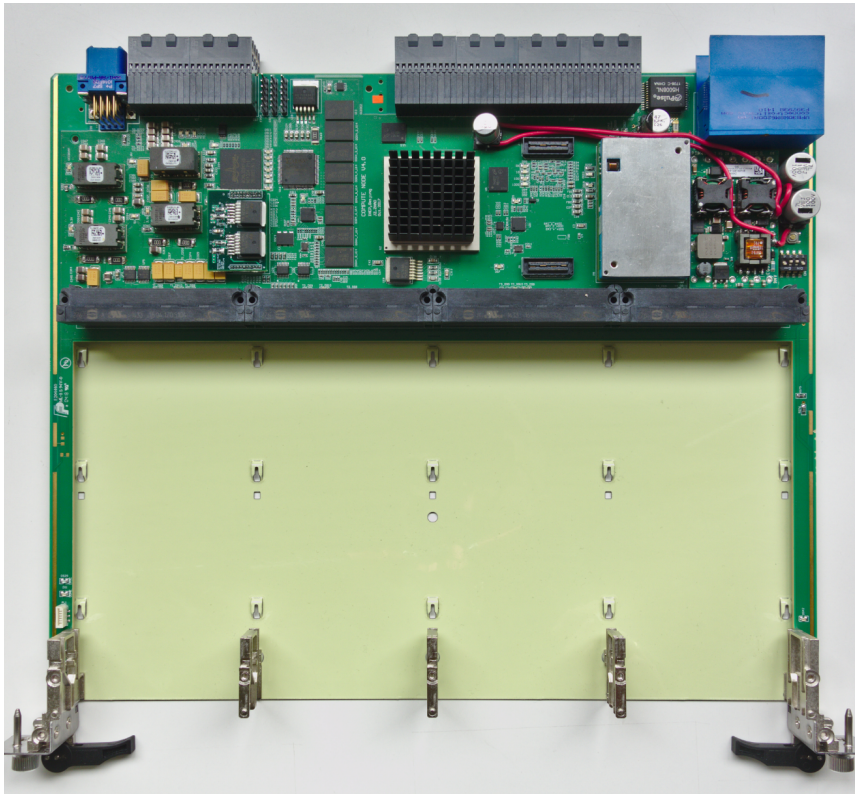


Compute Node Carrier Board (CNCB) v4.0

Hardware Upgrade (CNCB v4.0)

	CNCB v3.3	CNCB v4.0
FGPA	Virtex 4	Kintex UltraScale
Registers	50k	663k
LUTs	50k x 4-input	332k x 6-input
BRAM	4 Mb	38 Mb
MGT	16 x 6.5 Gbps	32 x 16.3 Gbps
CPU	PPC405	

- The CPU is replaced by a soft core **MicroBlaze** for slow control

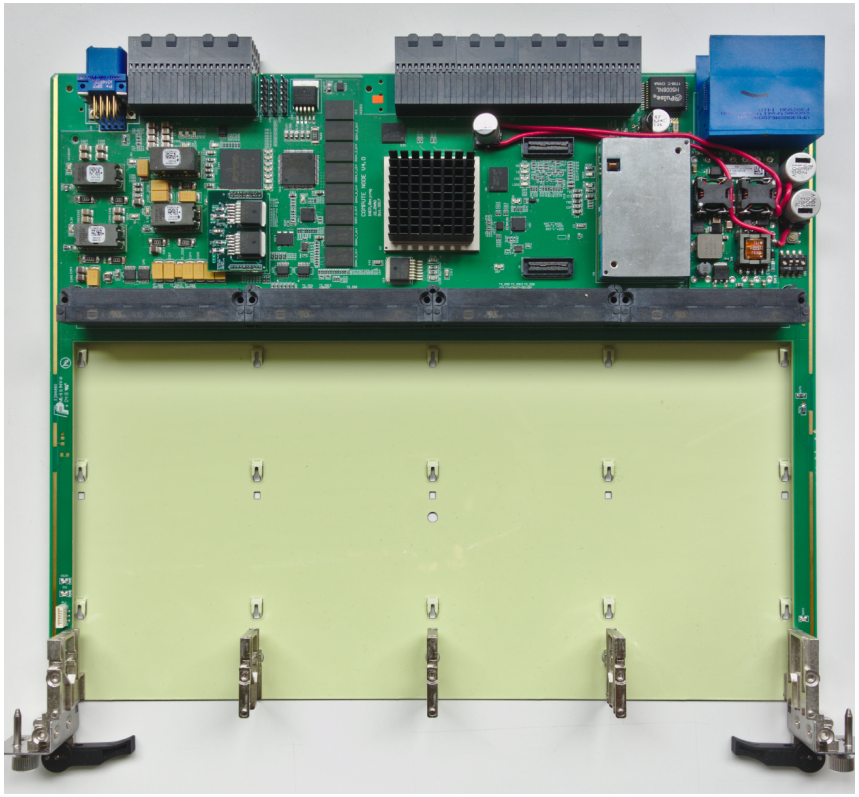


Compute Node Carrier Board (CNCB) v4.0

Hardware Upgrade (CNCB v4.0)

	CNCB v3.3	CNCB v4.0
FGPA	Virtex 4	Kintex UltraScale
Registers	50k	663k
LUTs	50k x 4-input	332k x 6-input
BRAM	4 Mb	38 Mb
MGT	16 x 6.5 Gbps	32 x 16.3 Gbps
CPU	PPC405	
RAM	2 GiB DDR2	16 GiB DDR4

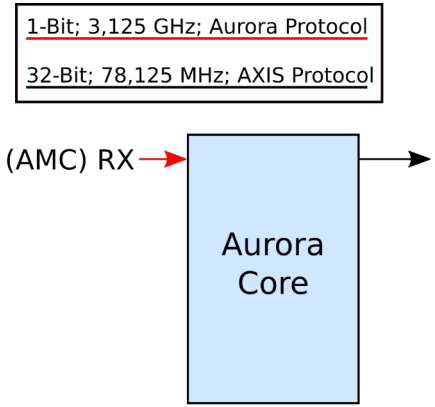
- The CPU is replaced by a soft core **MicroBlaze** for slow control



Compute Node Carrier Board (CNCB) v4.0

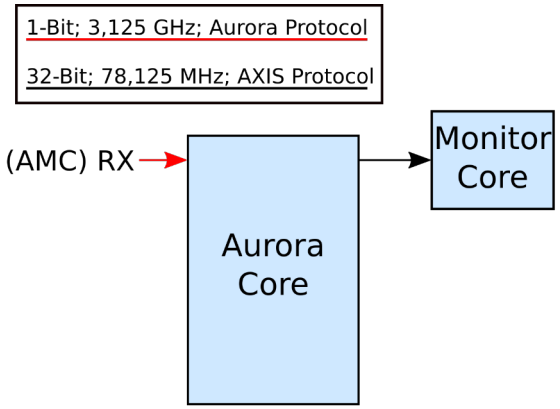
Merger Firmware Block Design

- **Aurora Core**
  - Aurora Wrapper
  - FIFOs
  - Slave register and interrupts



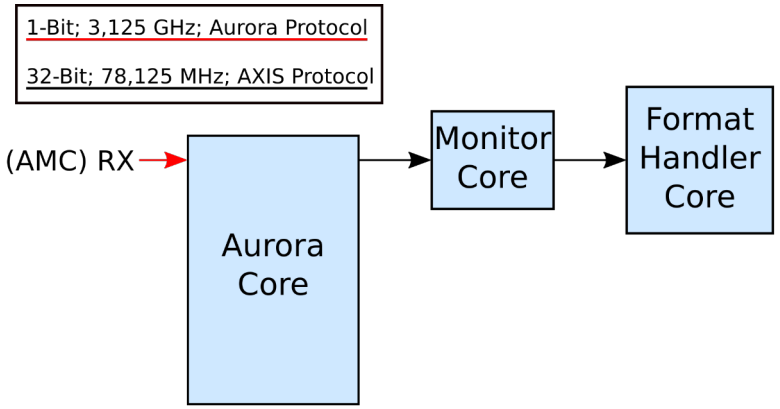
Merger Firmware Block Design

- **Aurora Core**
  - Aurora Wrapper
  - FIFOs
  - Slave register and interrupts
- **Monitor Core**
  - Slave register



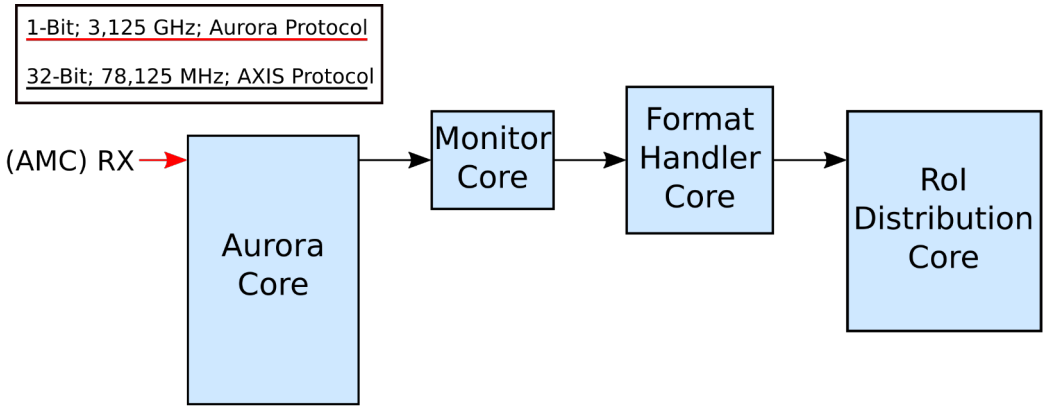
Merger Firmware Block Design

- **Aurora Core**
  - Aurora Wrapper
  - FIFOs
  - Slave register and interrupts
- **Monitor Core**
  - Slave register
- **Belle II Format Handler Core**
  - RoI Parser
  - Slave register and interrupts



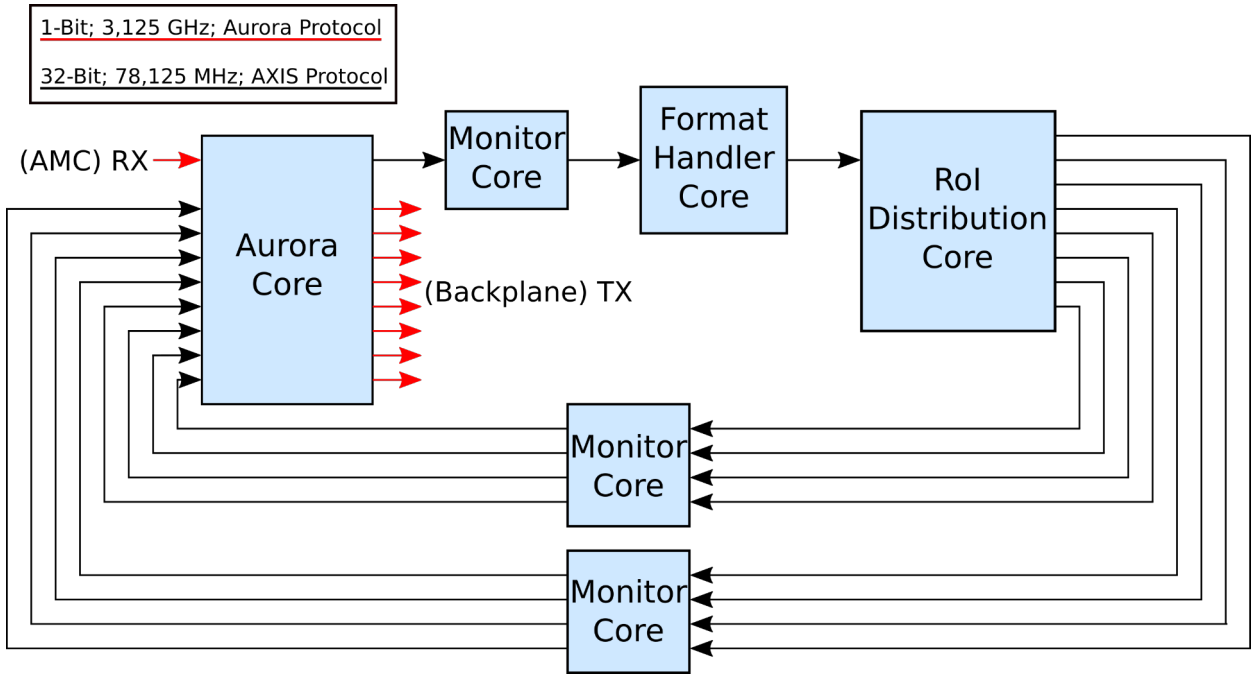
Merger Firmware Block Design

- **Aurora Core**
  - Aurora Wrapper
  - FIFOs
  - Slave register and interrupts
- **Monitor Core**
  - Slave register
- **Belle II Format Handler Core**
  - RoI Parser
  - Slave register and interrupts
- **RoI Distribution Core**
  - Slave register

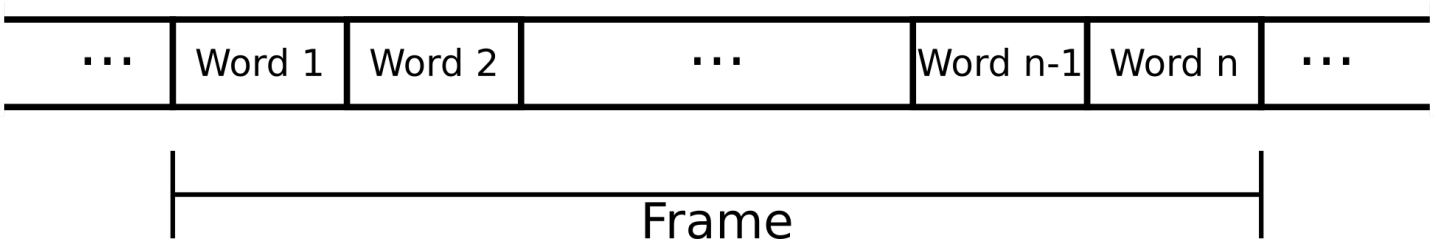


Merger Firmware Block Design

- **Aurora Core**
  - Aurora Wrapper
  - FIFOs
  - Slave register and interrupts
- **Monitor Core**
  - Slave register
- **Belle II Format Handler Core**
  - RoI Parser
  - Slave register and interrupts
- **RoI Distribution Core**
  - Slave register

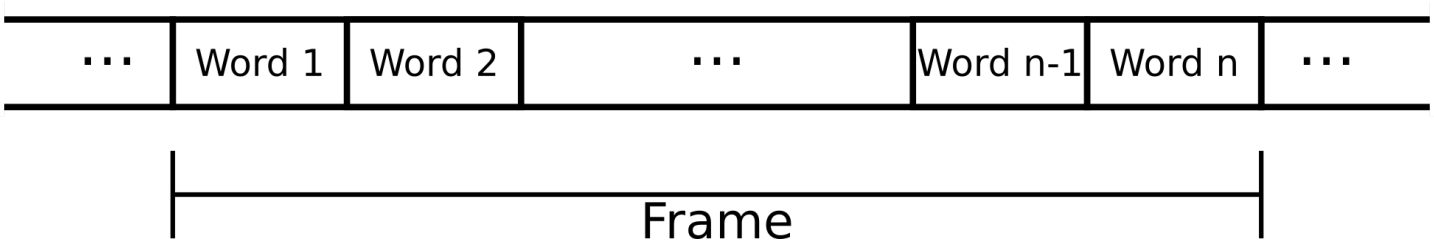


LocalLink



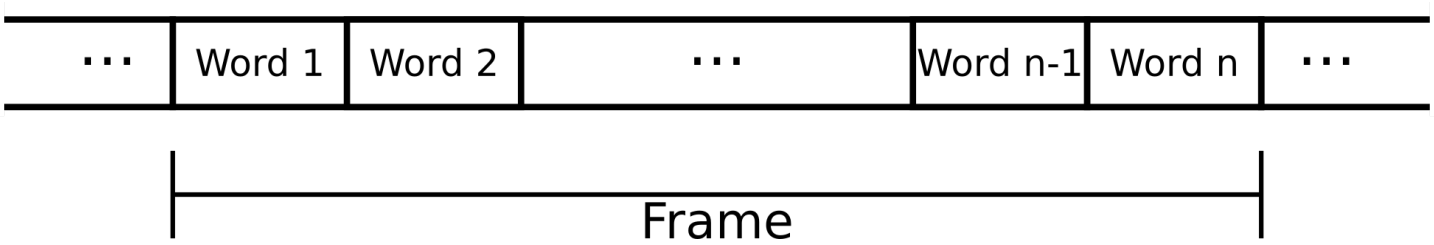
- The transmission consists of **frames**, which consists of any number of **words**
- **`DATA`** contains payload (usually 32 bits)

LocalLink



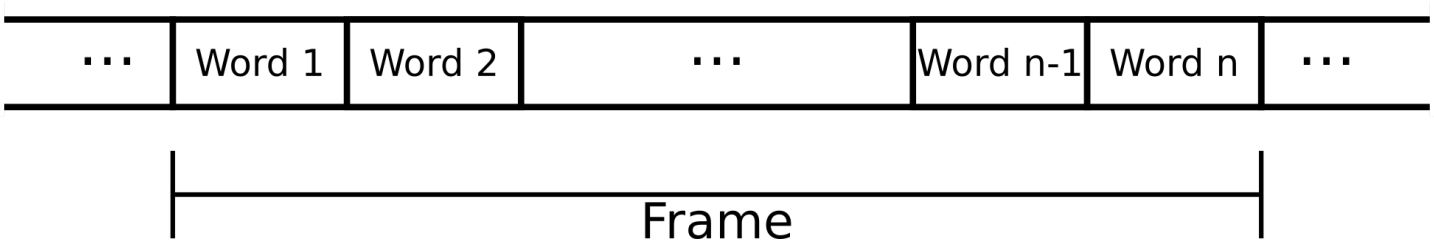
- The transmission consists of **frames**, which consists of any number of **words**
- **`DATA`** contains payload (usually 32 bits)
- **`REM`** sets number of remaining bytes in last word

LocalLink



- The transmission consists of **frames**, which consists of any number of **words**
- **`DATA`** contains payload (usually 32 bits)
- **`REM`** sets number of remaining bytes in last word
- **`SOF`** indicates start of frame
- **`EOF`** indicates end of frame

LocalLink



- The transmission consists of **frames**, which consists of any number of **words**
- **`DATA`** contains payload (usually 32 bits)
- **`REM`** sets number of remaining bytes in last word
- **`SOF`** indicates start of frame
- **`EOF`** indicates end of frame
- **`SRC_RDY`** indicates valid data
- **`DST_RDY`** indicates readiness of the source

LocalLink to AXI-Stream

LocalLink port	Master	Slave	Bandwidth	Mode
LL_DATA	Output	Input	32	-

AXI-Stram port	Master	Slave	Bandwidth	Mode
AXIS_TDATA	Output	Input	32	-

LocalLink to AXI-Stream

LocalLink port	Master	Slave	Bandwidth	Mode
LL_DATA	Output	Input	32	-
LL_REM	Output	Input	2	Active high

AXI-Stram port	Master	Slave	Bandwidth	Mode
AXIS_TDATA	Output	Input	32	-
AXIS_TKEEP	Output	Input	4	Active high

LocalLink to AXI-Stream

LocalLink port	Master	Slave	Bandwidth	Mode
LL_DATA	Output	Input	32	-
LL_REM	Output	Input	2	Active high
LL_SRC_RDY_N	Output	Input	1	Active low
LL_DST_RDY_N	Input	Output	1	Active low

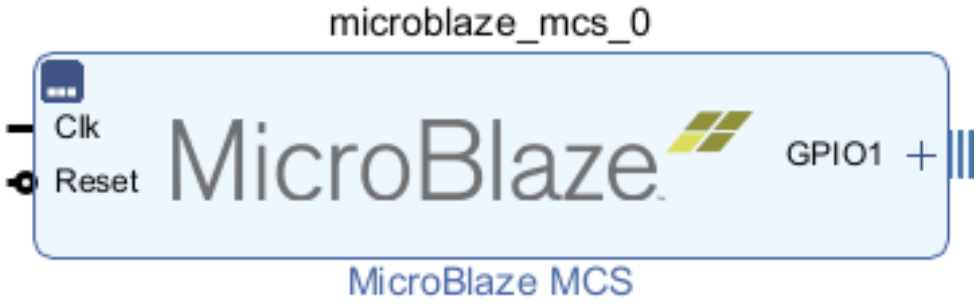
AXI-Stram port	Master	Slave	Bandwidth	Mode
AXIS_TDATA	Output	Input	32	-
AXIS_TKEEP	Output	Input	4	Active high
AXIS_TVALID	Output	Input	1	Active high
AXIS_TREADY	Input	Output	1	Active high

LocalLink to AXI-Stream

LocalLink port	Master	Slave	Bandwidth	Mode
LL_DATA	Output	Input	32	-
LL_REM	Output	Input	2	Active high
LL_SRC_RDY_N	Output	Input	1	Active low
LL_DST_RDY_N	Input	Output	1	Active low
LL_SOF_N	Output	Input	1	Active low
LL_EOF_N	Output	Input	1	Active low
AXI-Stram port	**Master**	**Slave**	**Bandwidth**	**Mode**
AXIS_TDATA	Output	Input	32	-
AXIS_TKEEP	Output	Input	4	Active high
AXIS_TVALID	Output	Input	1	Active high
AXIS_TREADY	Input	Output	1	Active high
AXIS_TLAST	Output	Input	1	Active high

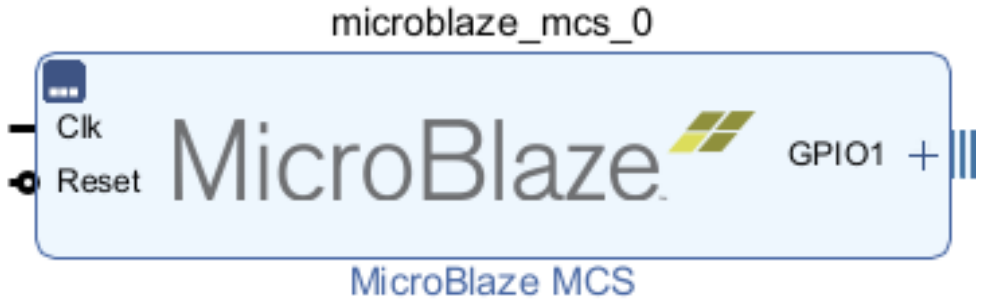
What is a MicroBlaze?

- Replaces the PPC, which was used for slow control and debugging
- A MicroBalze is **soft core processor** embedded as part of the FPGA



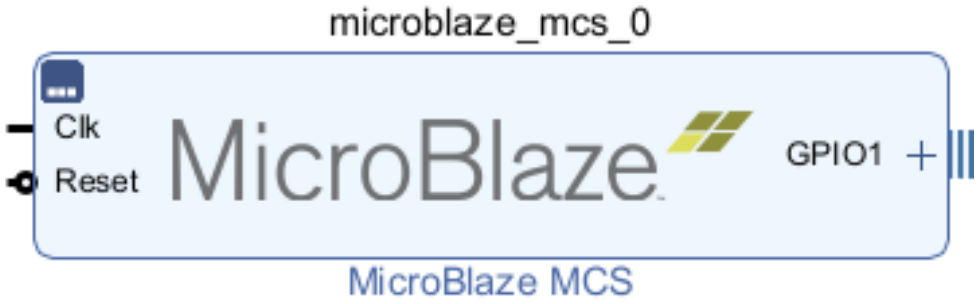
What is a MicroBlaze?

- Replaces the PPC, which was used for slow control and debugging
- A MicroBalze is **soft core processor** embedded as part of the FPGA
- High degree of **flexibility** and **configurability**
- Better optimisation and parallelisation



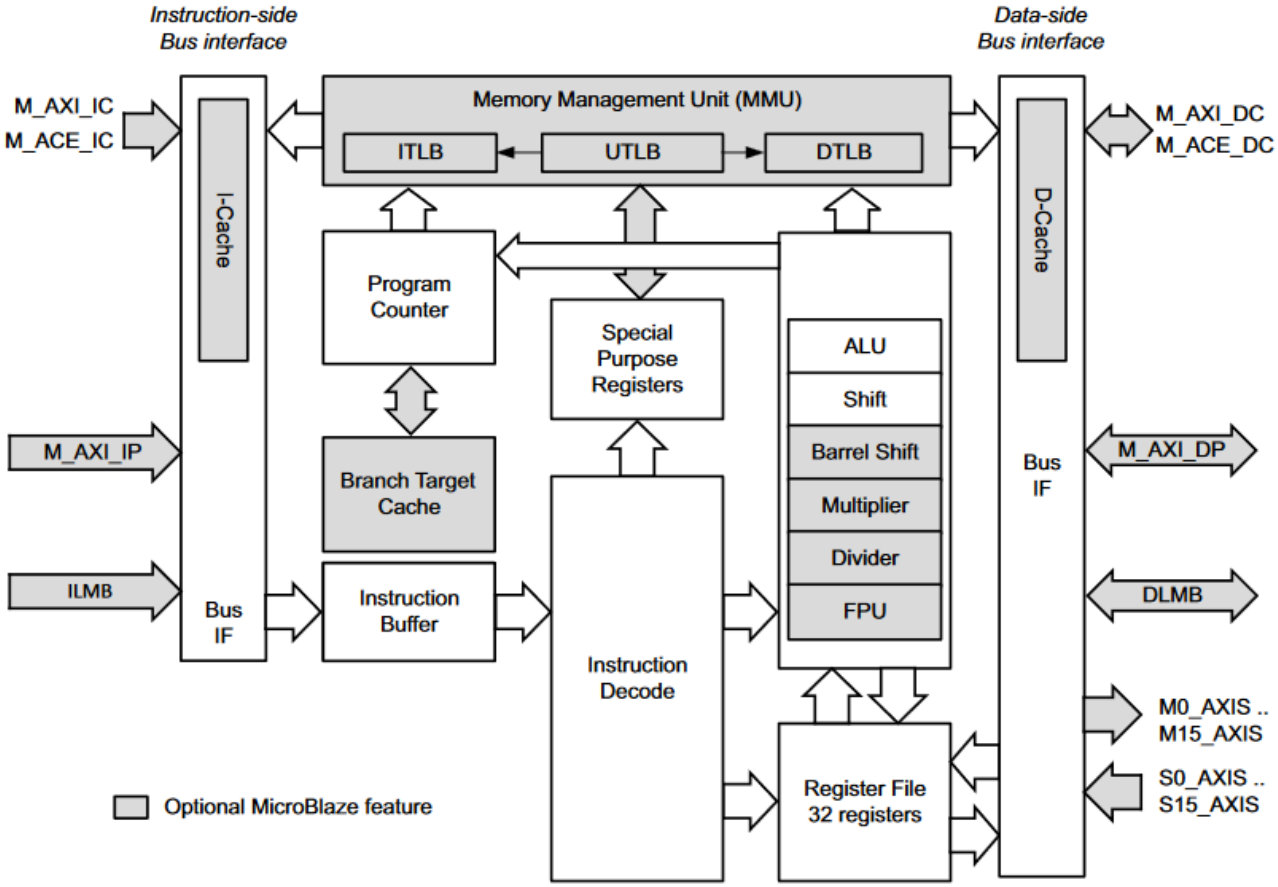
What is a MicroBlaze?

- Replaces the PPC, which was used for slow control and debugging
- A MicroBalze is **soft core processor** embedded as part of the FPGA
- High degree of **flexibility** and **configurability**
- Better optimisation and parallelisation
- Slower processing speed
- Is able to run OS (Linux)



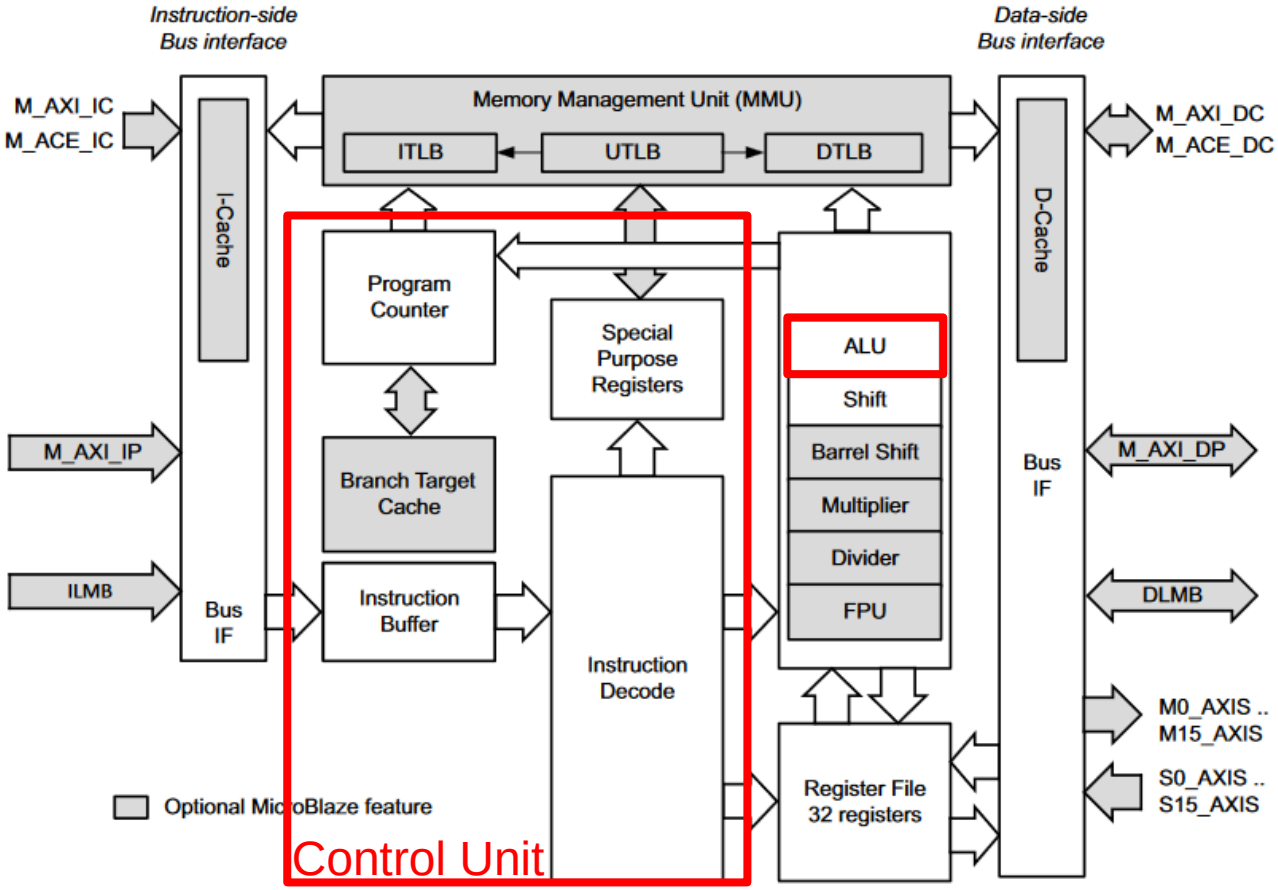
Architecture of a MicroBlaze

- The MicroBlaze uses the **Reduced Instruction Set Computer** (RISC) architecture



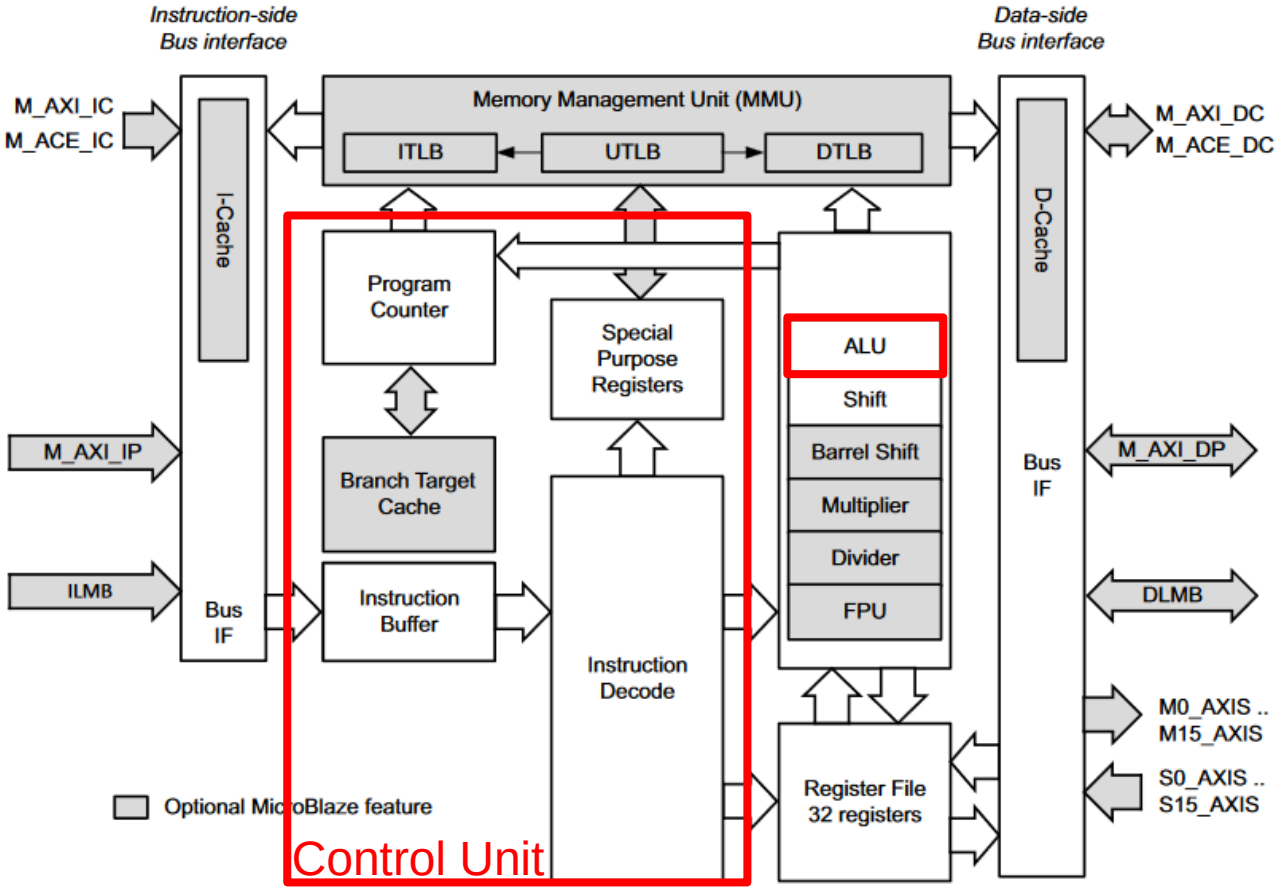
Architecture of a MicroBlaze

- The MicroBlaze uses the **Reduced Instruction Set Computer** (RISC) architecture
- Processors consists of **Arithmetic Logic Unit** (ALU) and and a **control unit** to coordinate its operation



Architecture of a MicroBlaze

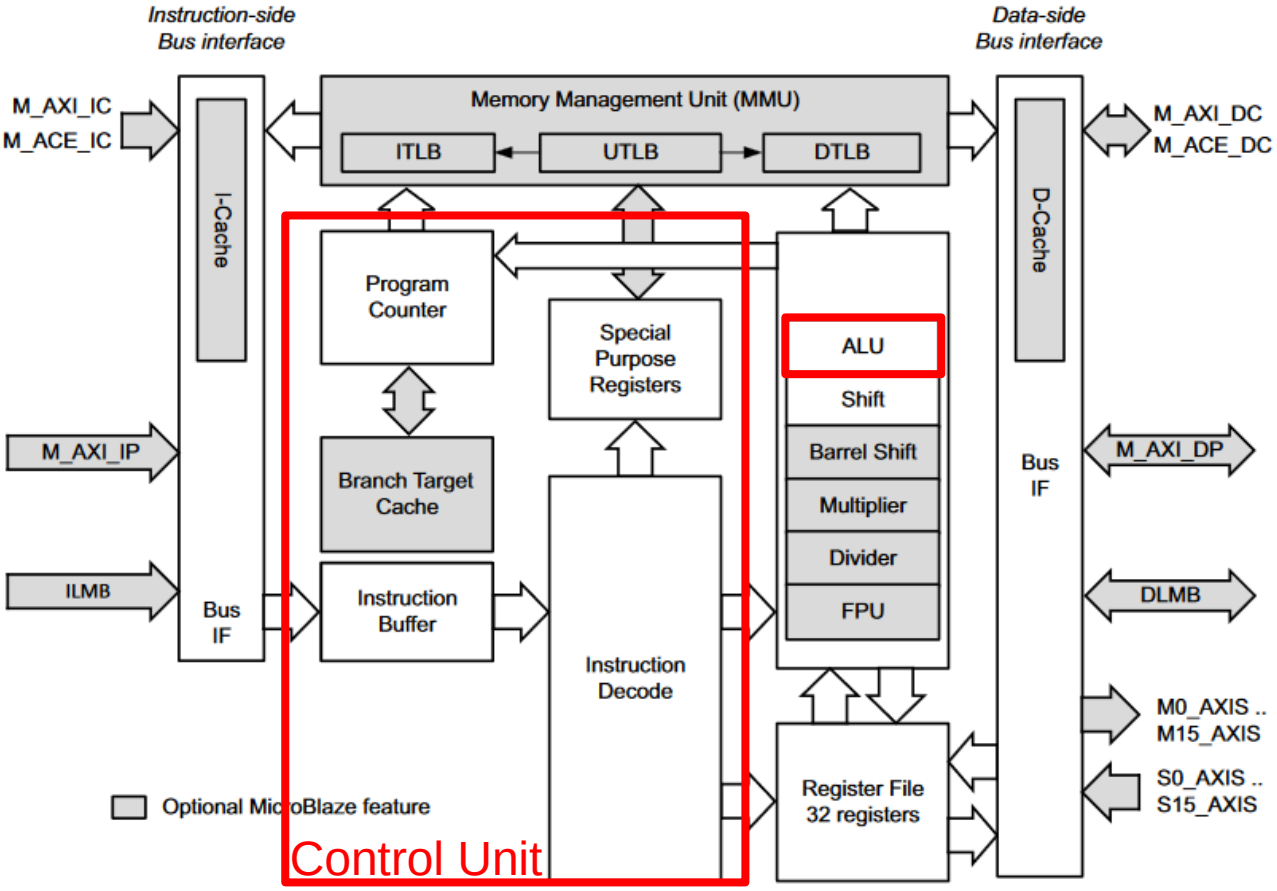
- The MicroBlaze uses the **Reduced Instruction Set Computer** (RISC) architecture
- Processors consists of **Arithmetic Logic Unit** (ALU) and and a **control unit** to coordinate its operation
- The fundamental process is the **Fetch-Decode-Execute** cycle



Architecture of a MicroBlaze

- The MicroBlaze uses the **Reduced Instruction Set Computer** (RISC) architecture
- Processors consists of **Arithmetic Logic Unit** (ALU) and and a **control unit** to coordinate its operation
- The fundamental process is the **Fetch-Decode-Execute** cycle
- Resources used by the MicroBlaze are in the range of:

LUTs:	500 – 10 000
FFs:	200 – 10 000
DSPs:	0 – 15
36k BRAMs:	0 – 20

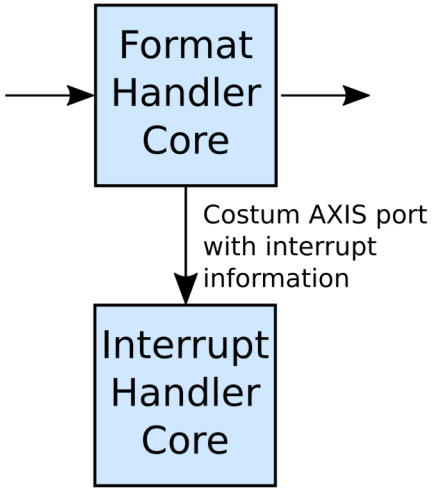


New Interrupt Logic

- Usually interrupts are treated as a counter that increases, when a interrupt is issued
- In case of multiple interrupts the information of simultaneously issued interrupts are lost and only totals are available

New Interrupt Logic

- Usually interrupts are treated as a counter that increases, when a interrupt is issued
- In case of multiple interrupts the information of simultaneously issued interrupts are lost and only totals are available
- A new IP core writes the information to FIFO as 32-bit word
- The FIFO is read and emptied via the slow control and the information is logged



Extract device tree from Vivado

Adjust slow control source code to include necessary features

Cross compile Linux using build root

Summary

- A new CNCB had to be developed, since the old components are not available any more
- The new hardware needs firmware adjustments
  - Updated protocol from **LocalLink** to **AXI-Stream**
  - Additional **reset logic** for the Aurora Core
  - Replaced **PowerPC** with **MicroBlaze**
  - New **interrupt logic** for Belle II Format Handler core