40/100G UDP interface

Features

This UDP implementation provides UDP communication using two UDP ports: for register access and for upstream channel. The 40/100G Ethernet interface uses a 10/25G capable quad MGT as well as 4 lanes optical transceiver (QSFP+, Firefly etc.).

This UDP implementation supports following protocols with some restrictions:

- UDP with headCRC=0
- IPv4 without packet fragmentation (UDP packets limited to 1472 bytes)
- no ICMP support (e.g. ping)
- ARP reduced to ‘reply to request’ packets only *)
- Ethernet
- Static network configuration (no DHCP):
  mac address AA:BB:CC:DD:EE:FF **ip address 192.168.1.100** **port: 5000**
- Autonegotiation is disabled

*) -> The PC must initiate the communication by sending a UDP packet first for each UDP port, before FPGA can use it.

Jumbo Frames

yes

Register Access

This service is used for FPGA parameterization as single read/write acess to an internal 32bit register space based on simple acsii-hex syntax.

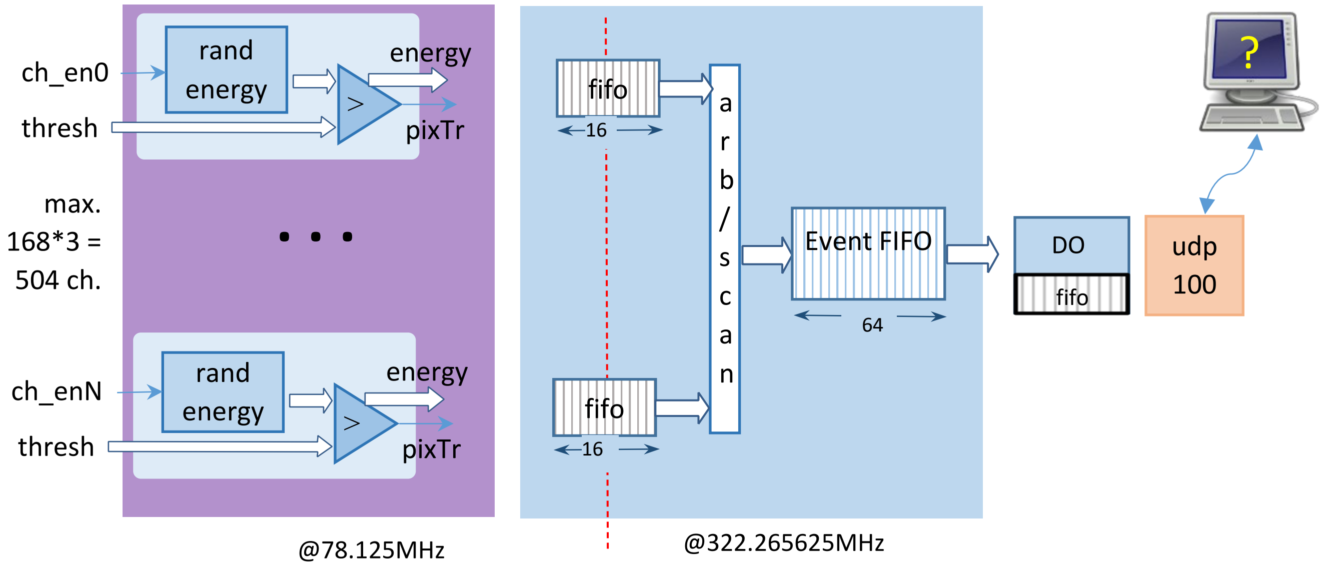
WRITE command: wAAAAAAAA_DDDDDDDD
e.g. w00000007_12345678 writes 0x12345678 data to address 0x00000007

READ command: rAAAAAAAA
e.g. r00000007 reads from the address 0x00000007, FPGA answers with 12345678<CR>

Upstream Channel

This channel uses binary 512bit data stream to transmit (triggered/event) data to PC. The length of such packets is either fixed or pre-defined by user. Note that PC should initiate a (dummy) transfer to the streaming port before to announce its mac+ip+port addresses/number.

Validation of SW-Histogramming



Register Map

#	hex	size	name		description	def.
0		uint32	Version	ro	project, version, revision, geao address etc.	
1		uint32	Command	wo	self-cleared bits; [0]- sw_reset; [31]- sw_trigger	0
~~2~~		~~uint32~~	~~UserIntrptMask~~	~~r/w~~		~~FF~~
~~3~~		~~uint32~~	~~UserIntrptSource~~	~~ro~~		
4		uint16	s_streamPort[15:0]	r/w	starting port number for the upstream	5002
5		uint32	Threshold	r/w	common Threshold[23:0]	max
6		uint16	N_size	r/w	packet length in bytes	3200
7		uint32	chann_n	r/w	number of channels [8:0]	0
8		uint32	N_frames[31:00]	r/w	N_frames[63:0] number of eth. frames to be send before stop	0
9		uint32	N_frames[63:32]	r/w	N_frames = 0 means send continuosly (don’t stop)	
10		uint16	e_streamPort[15:0]	r/w	last port number for the upstream *)	5002
11		uint32	RunControl	r/w	[0]- run trigger flag [1]- Behaviour of hist_mask: 0 – constant first, 1- sequencially [2]- run transmission	0
16		uint32	TriggerRate00	ro	trigger rate of channel 0 in kHz	
17		uint32	EventFIFO Status	ro	[15:0] fill status [16] FIFO full [24] FIFO empty	
18		uint32	Frame_Cnt[31:00]	ro	current frame counter [63:0]	
19			Frame_Cnt [63:32]			
20		uint32	EventCount[15:0]	ro	current event counter 0..2^{16}-1	

Parameters

parameter	type	descr.	def.
thresh[23:0]	r/w	common, setting 0x00ff_ffff stops triggering	-1
channN	r/w	number of running channels 0...1023, setting 1 enables one channel etc.	0
udp_size	r/w	udp payload size 64...9000 bytes	
sw_reset	wo	stops everything, clears all fifos	0

Event Format

8byte frame

evtID 16bit	channel 16bit	energy 24bit	histgrm_mask 8bit
i	x	Energy(x)	b’01000001
i+1	y	Energy(y)	b’10000100

16byte frame

evtID 2byte	channel 2byte	energy 3byte	histgr_mask 1byte	TriggerInfo 2 bytes	timestamp 6 bytes
i	x	Energy(x)	b’01000001	xAAAA	seconds, subseconds
i+1	y	Energy(y)	b’10000100	xAAAA	seconds, subseconds

subseconds are in 16ns ticks (= ADC sample counts) in range of 0 to 62.5M-1.

subseconds[25:0]

seconds[21:0]

20byte frame

packet type 1B	packet ID 1B	eventID 2Byte	channel 2Byte	energy 3Byte	aux. info 3Byte		timestamp 8Byte
‘H’	j	i	x	Energy(x)	histgrm_mask	flags	seconds, subseconds
‘E’, ‘T’	j+1	i+1	y	Energy(y)	adc_offset	flags	seconds, subseconds

32byte frame

packet type 1B	packet ID 4B	eventID 4Byte	channel 2Byte	energy 4Byte	aux. info 9Byte		timestamp 8Byte
‘H’	j	i	x	Energy(x)	histgrm_mask	flags	seconds, subseconds
‘E’, ‘T’	j+1	i+1	y	Energy(y)	adc_offset	flags	seconds, subseconds