# **Demo Read-out protocol**

Alexander Kluge CERN-PH

**October 6**, 2008

# **Demo read-out protocol**

- Demonstrator data read-out protocol:
- inputs
  - 1 read-out clock (LVDS) up to 320 MHz
  - 9 enable\_readout (CMOS)
- outputs
  - -9 serial\_out (LVDS)

### **Demo read-out protocol**

#### • If enable\_readout is '1':

- if no hit is available in column
  - corresponding serial\_out is '0'
- if hit in column arrives corresponding serial\_out
  - goes '1' asynchronously
  - goes '1' for one clock cycle
  - goes '0' for one clock cycle
  - then 197 bits are shifted out
  - goes '0' of no hit arrived in the meanwhile, otherwise restarts with goes '1 for one clock cycle

# **Demo read-out protocol**

#### • If enable\_readout is '0':

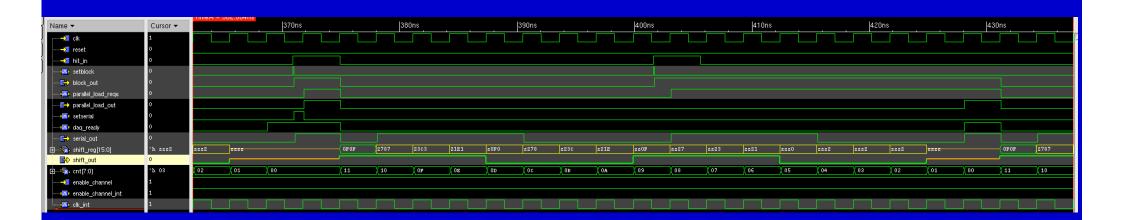
- if no hit is available in column
  - corresponding serial\_out is '0'
- if hit in column arrives corresponding serial\_out
  - goes '1' asynchronously and stays '1' until enable\_readout is '1'
  - goes '1' for one clock cycle
  - goes '0' for one clock cycle
  - then 197 bits are shifted out

 goes '0' of no hit arrived in the meanwhile, otherwise restarts with goes '1 for one clock cycle
A. Kluge

#### • 197 bits are

- 32 bits DLL data fine time leading edge
- 32 bits DLL data fine time trailing edge
- 32 bits coarse counter pos. clock edge leading edge
- 32 bits coarse counter neg. clock edge leading edge
- 32 bits coarse counter pos. clock edge trailing edge
- 32 bits coarse counter neg. clock edge leading edge
- -5 bit address
- running with 320 MHz -> 0.6 μs or 1.6 MHz

Name -	Cursor 🕶	300ns	400ns	500ns	
, elk	1				
	O				
∫→I hit_in	0				
J setblock	0				
block_out	0				
parallel_load_requ	0				
- parallel_load_out	0				
······· III · setserial	U				
daq_ready	1				
····· 🔂 serial_out ⊕⊊, shift_reg[15:0]	'h zzzz	zzzz	<u>=====================================</u>	z > z > z > z > z > z > z > z > z > z >	z z z z z z z z z z z z z z z z z z z
shift_out	z				
	'h 00				
	1				
	1				
	1				



	Cursor 🔻						1	_		laca				Luno I		
Name 👻	370ns				. 38	Ons		390ns			400ns					
j····· <b>→Σ</b> clk	1															
	0															
····· <b>→∑</b> hit_in	0															
setblock	0				I										[	
block_out	0															
parallel_load_requ	0															
····· <b>⊑→</b> parallel_load_out	0															
······• 🚾 • setserial	0															
······ <b>·¤·</b> daq_ready	0															
serial_out	0											1				
⊕ <b>√a.</b> shift_reg[15:0]	'h zzzZ	zzzZ	2222				OFOF	2787	Z3C3	21E1	zOFO	z278	z230	z21E	zzOF	zz27
Shift_out	0										1					
⊕ <b>√</b> a. cnt[7:0]	'h 03	02	01	00			11	10	OF	) OE	χ ου	) OC	Ов	0A	09	( 08
	1															
······ <b>·</b> ¤• enable_channel_int	1															
🦣 🤷 clk_int	1															

#### Name 🔻 Cursor 👻 370ns 380ns 390ns 400ns 410ns 420ns 430ns ····· → 📜 clk -----→T hit\_in - 💶 setblock -🔁 block\_out -💶 parallel\_load\_requ --- 🕞 parallel\_load\_out -- 🗖 setserial ---- 🗖 daq\_ready ---- serial\_out 'h zzzZ 0F0F 2787 2303 21E1 z0F0 z278 z230 z21E zz0F zz27 zz23 zz21 zzz0 zzz2 zzz2 zzz2 zzz2 zzz2 🕂 --- 🌆 shift\_reg[15:0] 2222 2222 📑 shift\_out 0 'h 03 01 00 00 🗄 --- 🌆 cnt[7:0] 02 (11 )(10 )(0F )(0E ) OD 00 0B 0A 09 08 07 06 05 04 03 02 01 11 10 ---**→** enable\_channel -----• - 💶 clk\_int

390ns			400ns 410ns						420ns		430ns					
															_	
		1								7					 ]	
21E1	z0F0	z278	z230	z21e	zzOF	zz27	zzZ3	zz21	zzzO	2222	2222	2222	2222		OFOF	2787
,	<u> </u>		•	v			v		L		v	·		v		
0E	<u> </u>	<u> </u>	( ов	<u> 0</u> 8	<u> (</u> 09	X 08	<u> </u>	<u>X 06</u>	<u> </u>	<u> </u>	<u> </u>	02	<u> </u>	<u> </u>	<u></u>	<u> 10</u>