PXD DAQ Discussion
München HLT & PXDDAQ
Workshop
22.02.2011

More algorithms for PXDDAQ ATCA

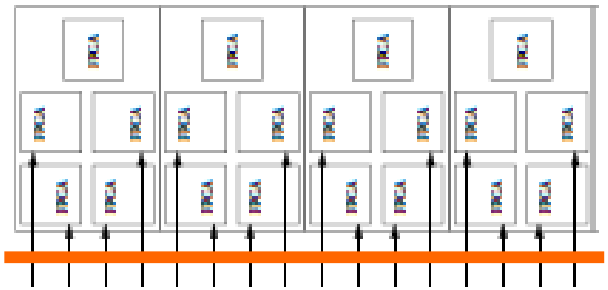
- cluster finder
  - remove 1-pixel cluster for large ϑ
    no problem, as inside ROI
  - Low p_T tracks = high dE/dx
    cluster finder
    problem, as outside ROI !
    requires extra pass of all PXD hits > time consuming!
- pedestal algorithm
  plan: last event before injection burst might contain all pixels
  not clear yet, if ATCA
  (because JTAG connection to frontend connection is needed anyway,
  maybe from extra PC, which then can run pedestal algorithm)
- alignment algorithm (i.e. solving coupled differential equations)
  -> construction site, nobody yet
  (although required by SVD ROI algorithm)
- PXD-only subevent Building
  RAM access and sorting algorithm (time consuming)
  if we need to run the other algorithms,
  can this probably be shifted to 2nd event builder farm?

PXS Subevent Builder

Note: backplane link is also RocketIO

40 FPGAs (data receive optical links) + 10 FPGAs (routing to backplane)

40 FPGAs for storing complete events



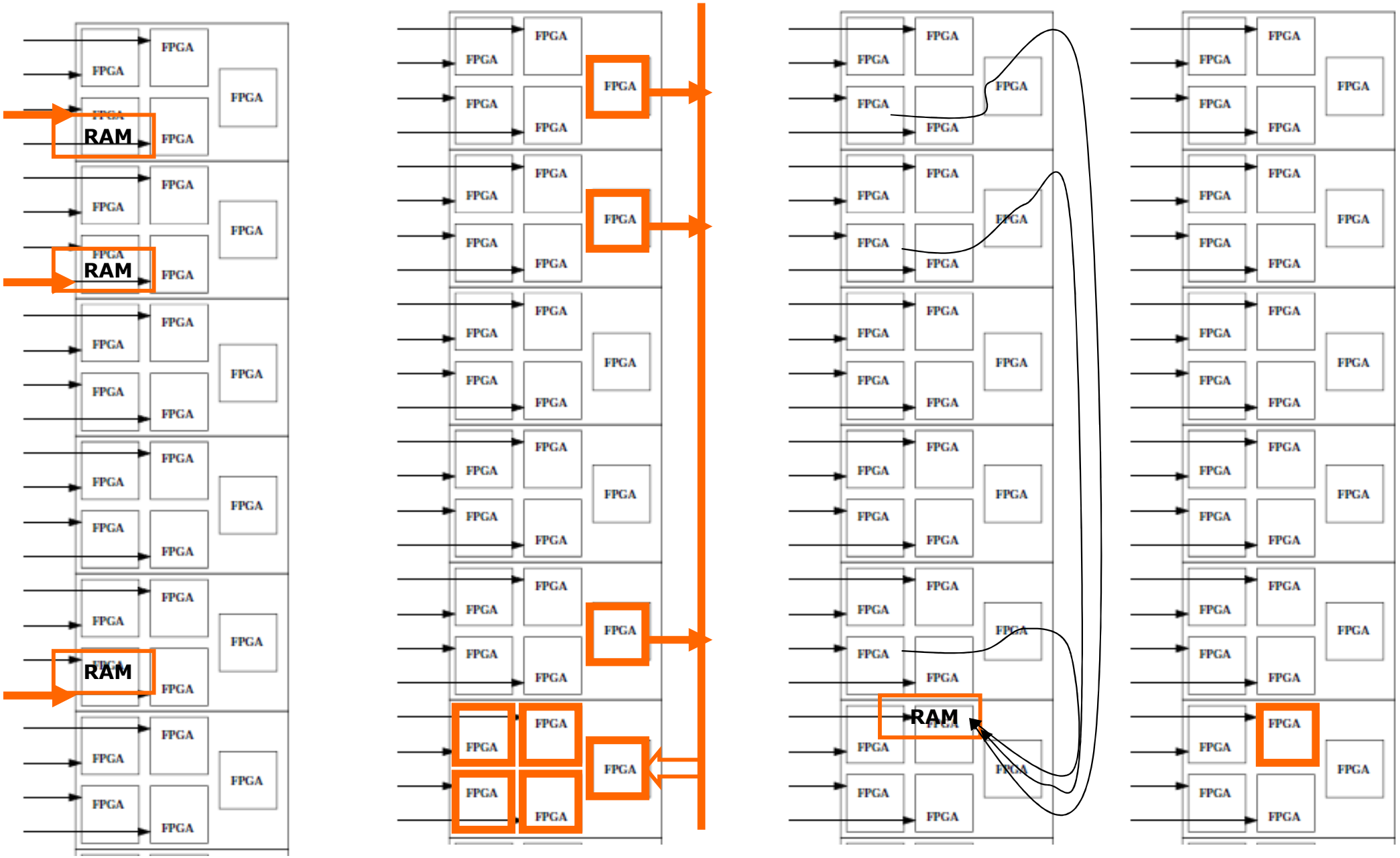
optical links here not connected

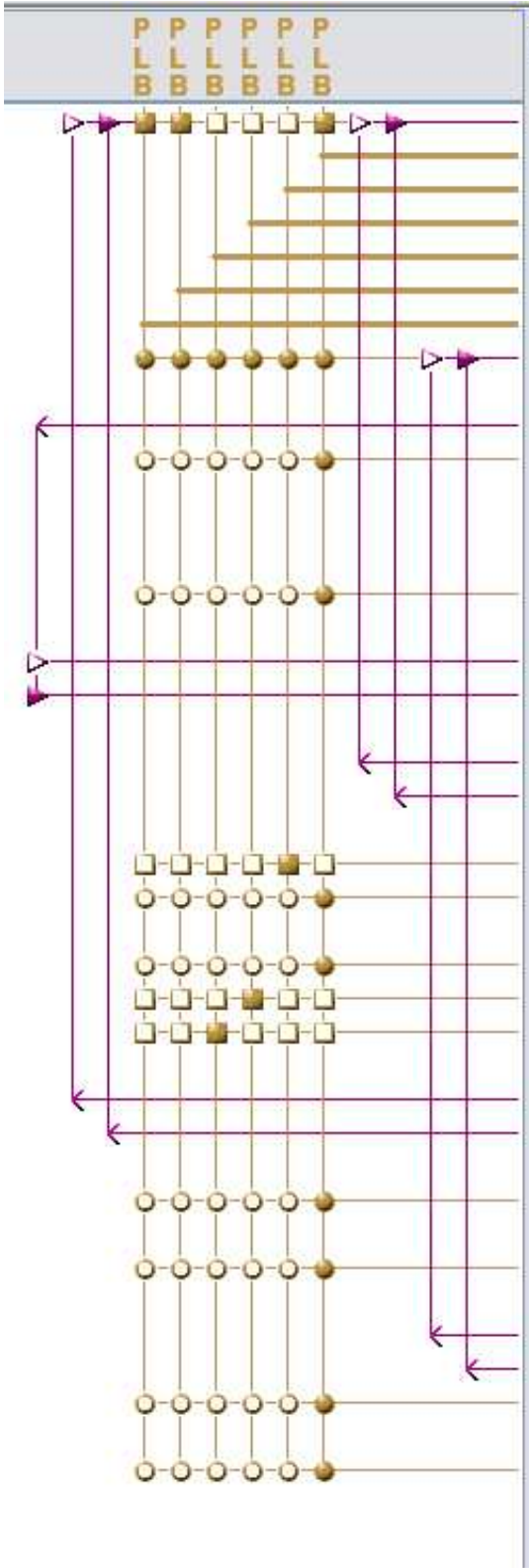
receive data by optical links

Notify all bookkeeper FPGAs

Move (copy and delete) via ATCA backplane

wait for HLT





Bus Interfaces | Ports | Addresses

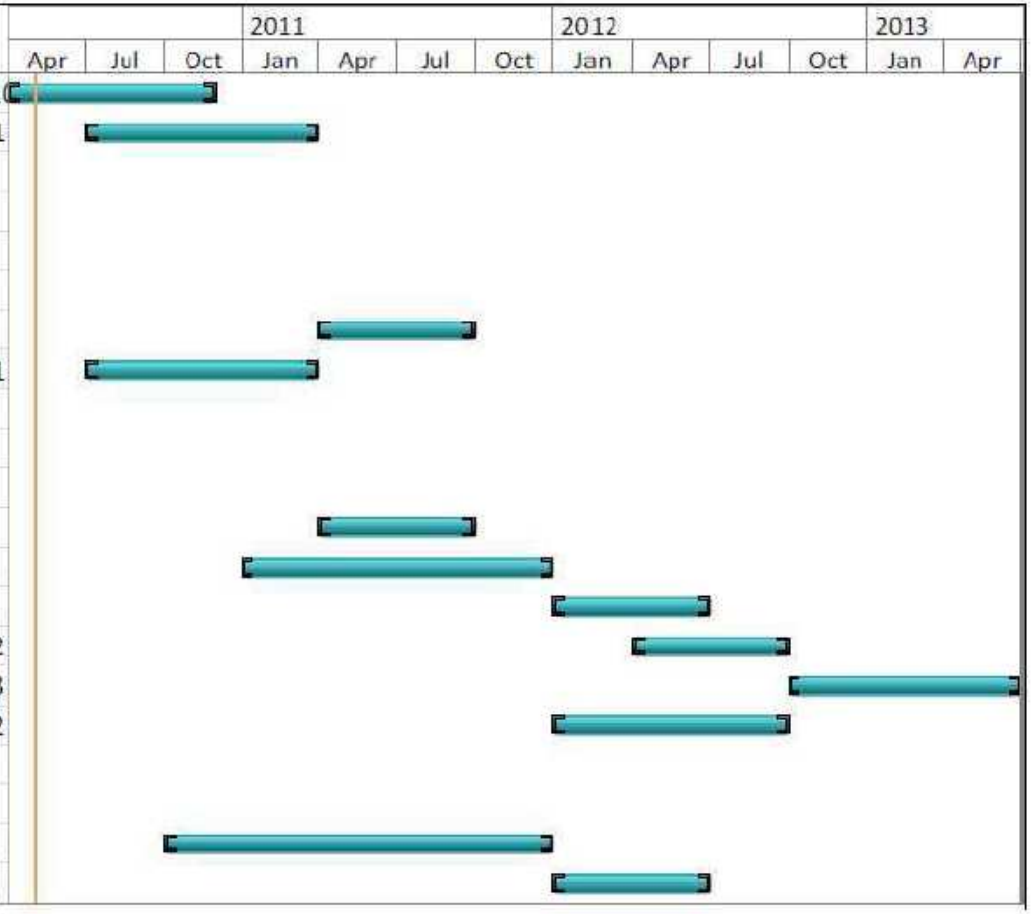
Name	Bus Connection	IP Type	IP Version
ppc405_0		ppc405_virtex4	2.01.a
plb		plb_v46	1.03.a
plb_v46_0		plb_v46	1.03.a
plb_v46_1		plb_v46	1.03.a
plb_v46_2		plb_v46	1.03.a
ppc405_0_dplb1		plb_v46	1.03.a
ppc405_0_iplb1		plb_v46	1.03.a
DDR2_SDRAM_W1D32M72R...		mpmc	4.03.a
xps_bram_if_cntlr_1		xps_bram_if_cntlr	1.00.a
PORTA	xps_bram_if_cntlr_1_port		
SPLB	plb		
FLASH		xps_mch_emc	2.00.a
MCH1	No Connection		
MCH0	No Connection		
SPLB	plb		
plb_bram_if_cntlr_1_bram		bram_block	1.00.a
PORTB	No Connection		
PORTA	xps_bram_if_cntlr_1_port		
jtagppc_cntlr_0		jtagppc_cntlr	2.01.c
JTAGPPC1	jtagppc_cntlr_0_JTAGPPC1		
JTAGPPC0	jtagppc_cntlr_0_0		
ll_plb_optical_0		ll_plb_optical	1.00.a
MPLB	plb_v46_0		
SPLB	plb		
plt_test2_fifo_0		plt_test2_fifo	1.00.a
SPLB	plb		
MPLB2	plb_v46_1		
MPLB	plb_v46_2		
proc_sys_reset_0		proc_sys_reset	2.00.a
RESETPPC1	proc_sys_reset_0_RESETPPC1		
RESETPPC0	ppc_reset_bus		
roi_core_test_0		roi_core_test	3.00.a
SPLB	plb		
xps_intc_0		xps_intc	1.00.a
SPLB	plb		
Hard_Ethernet_MAC		xps_ll_temac	1.01.b
LLINK1	Hard_Ethernet_MAC_LLINK1		
LLINK0	Hard_Ethernet_MAC_LLINK0		
SPLB	plb		
RS232		xps_uartlite	1.00.a
SPLB	plb		
clock_generator_0		clock_generator	2.01.a
FLASH_util_bus_split_0		util_bus_split	1.00.a

More algorithms: what‘s the problem?

- FPGA resources are limited
  not only ROI algorithms on FPGA, but
  PLB bus, FIFOs, optical link, RAM memory controller, …
  it is already using >50%
- possible solution: 2nd ATCA shelf
  4 additional CN
  (e.g. for SVD, running the SVD-only ROI algorithm)
- Main problem:
  was not foreseen neither in the project milestone plan
  nor in the project financial plan

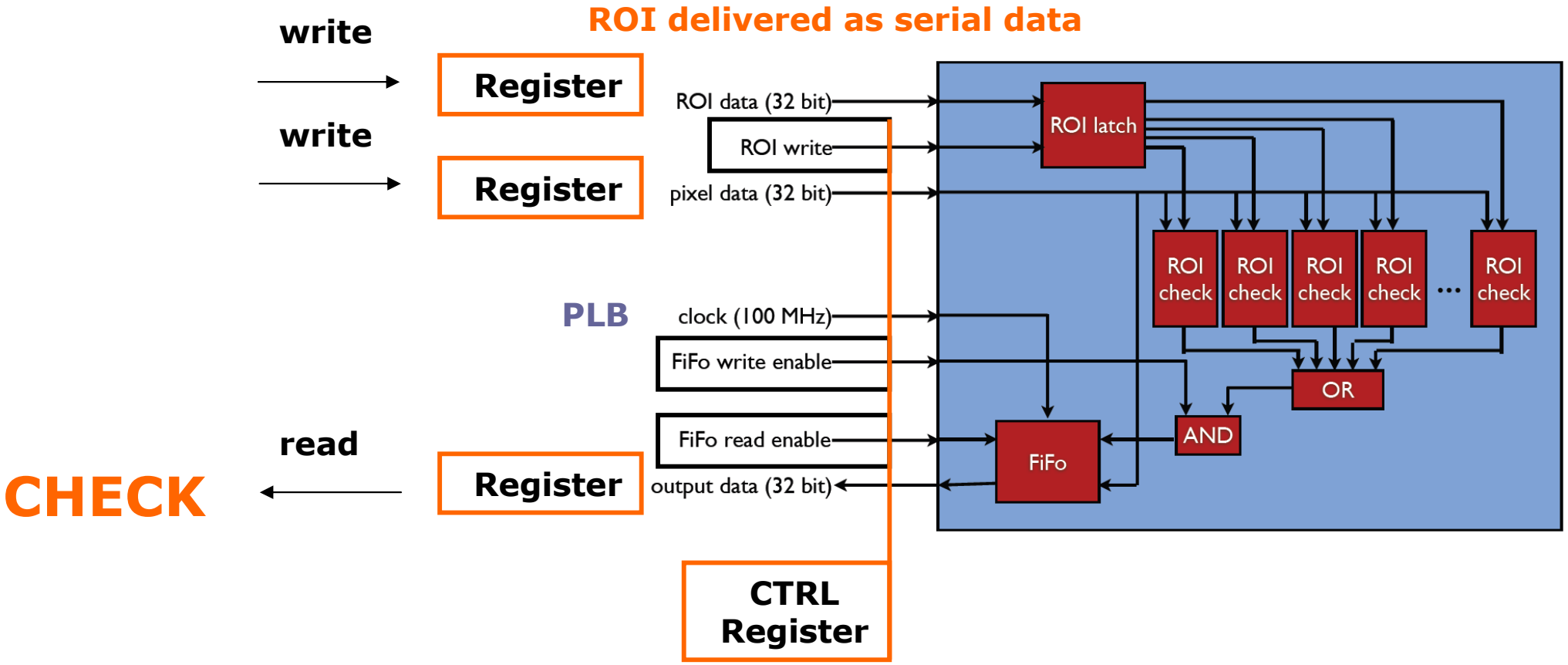
Cluster finder was not considered yet for the project plan (neither pedestal algorithm)

ID	Task Name	Start	Finish
1	Data Receive on FPGA	Thu 01.04.10	Wed 01.12.10
2	Subevent Builder on FPGA	Thu 01.07.10	Thu 31.03.11
3	1. ATCA backplane communication		
4	2. Master/Slave protocol		
5	3. implement BUSY (RAM full)		
6	4. Bookkeeping for (partially) built events		
7	Subevent Sorting for HLT	Thu 31.03.11	Fri 30.09.11
8	Data Reduction on FPGA	Thu 01.07.10	Thu 31.03.11
9	1. SVD+PXD Track finder		
10	2. Helix Extrapolation for HLT track, ROI		
11	3. Alignment/calibration parameter handling		
12	Interface to HLT farm	Fri 01.04.11	Fri 30.09.11
13	Data Output on FPGA (GB Ethernet), TCP/IP Stack	Fri 31.12.10	Sat 31.12.11
14	Prototype System (Readout of 1 Half-Module)	Sun 01.01.12	Sat 30.06.12
15	PCB Mass Production	Sun 01.04.12	Sun 30.09.12
16	1 Full System (Test at KEK)	Mon 01.10.12	Sun 30.06.13
17	MC Simulations	Sun 01.01.12	Sun 30.09.12
18	1. Efficiency Study for physics benchmark channels		
19	2. Offline Implementation of Data Reduction		
20	New PCB (Carrier-Board, AMC Card w/ Virtex 6)	Fri 01.10.10	Sat 31.12.11
21	Testing of new PCB (Softcore PowerPC)	Sun 01.01.12	Sat 30.06.12



ROI Algorithm

parallel in ROI



Register #5 (write ACK, read ACK) is not needed anymore (removed now)

cluster finder

- Andreas Wassatsch, ASIC on DHH?
- which algorithm?
  - walking 2-dim and updating,
    averaging ADC of neighbour pixels
  - sliding window, e.g. sum ADC 3x3 pixel
  - find high ADC, then sum around it

    -> in any case needs additional arithmetic operations,
    so probably computational bottleneck

- RAM bottleneck question:
  how to do it, when only touching each pixel 1x

- if we need the cluster finder on CN,
  we would need an something like a „manifest“