Technical Progress Report for:

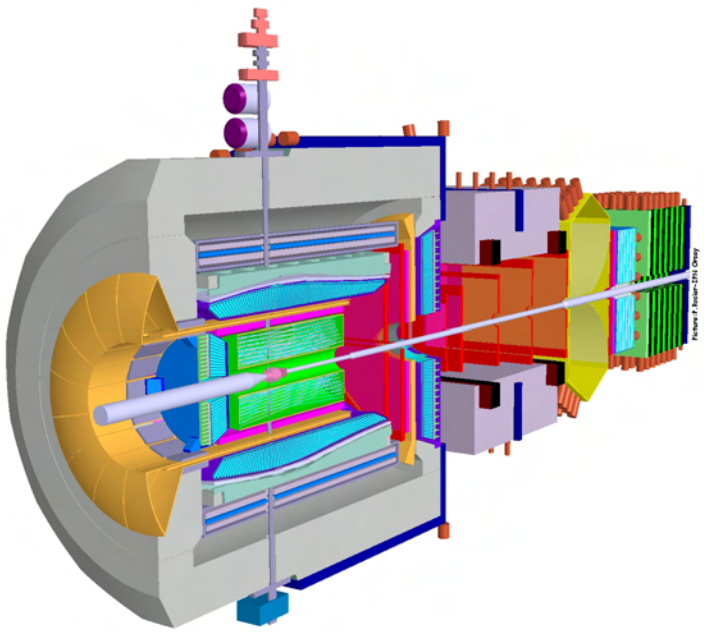
$\overline{\text{P}}$ANDA

(AntiProton Annihilations at Darmstadt)

Strong Interaction Studies with Antiprotons

$\overline{\text{P}}$ANDA Collaboration

We propose to study fundamental questions of hadron and nuclear physics in interactions of antiprotons with nucleons and nuclei, using the universal $\overline{\text{P}}$ANDA detector. Gluonic excitations and the physics of strange and charm quarks will be accessible with unprecedented accuracy thereby allowing high-precision tests of the strong interaction. The proposed $\overline{\text{P}}$ANDA detector is a state-of-the-art internal target detector at the HESR at FAIR allowing the detection and identification of neutral and charged particles generated within the relevant angular and energy range. This task will be shared by the combination of a central and a forward spectrometer of modular design where both are optimized for the specific kinematics of the antiproton-nucleon annihilation process.



The $\overline{\text{P}}$ANDA Collaboration

Universität Basel, Switzerland
M. Kotulla, B. Krusche, F. Zehr

Institute of High Energy Physics, Chinese Academy of Sciences, Beijing, China
J.J. Xie, B.S. Zou

Universität Bochum, I. Institut für Experimentalphysik, Germany
A. Golischewski, K. Götzen, T. Held, H. Koch, B. Kopf, B. Lewandowski, H. Nowak, H. Schmücker, M. Steinke, P. Wieczorek, A. Wilms, J. Zhong

Helmholtz-Institut für Strahlen- und Kernphysik, Bonn, Germany
F. Hinterberger

Università di Brescia, Italy
A. Zenoni

Dipartimento di Fisica e Astronomia dell'Università di Catania and INFN, Sezione di Catania, Italy
M. De Napoli, G. Raciti, E. Rapisarda

Instytut Fizyki, Uniwersytet Jagiellonski, Cracow, Poland
P. Hawranek, B. Kamys, St. Kistryn, A. Magiera, P. Moskal, B. Piskor-Ignatowicz, C. Piskor-Ignatowicz, Z. Rudy, P. Salabura, J. Smyrski, M. Wojciechowski

Gesellschaft für Schwerionenforschung mbH, Darmstadt, Germany
U. Lynen, J. Lühning, H. Orth, K. Peters, T.R. Saitoh, C. Schwarz, C. Sfienti

Technische Universität Dresden, Germany
K.-T. Brinkmann, H. Freiesleben, R. Jäkel

Veksler-Baldin Laboratory of High Energies (VBLHE), Joint Institute for Nuclear Research ([a]), Laboratory of Particle Physics (LPP) ([b]), Laboratory of Information Technologies (LIT) ([c]), Laboratory of Nuclear Problems (LNP) ([d]), Dubna, Kabardian-Balkarian State University ([e]) and Institute of Applied Mathematics and Automation ([f]), Nal'chik, Russia
V.M. Abazov[d], G. Alexeev[d], A. Arefiev[a], M.Yu. Barabanov[a], B.V. Batyunya[a], D. Bogoslovski[a], T.Yu. Bokova[a], V.V. Borisov[a], V.A. Budilov[a] Yu.V. Bugaenko[a], V.Kh. Dodokhov[a], A.A. Efremov[a], O.I. Fedorov[a], A.A. Feshchenko[b], A.S. Galoyan[b], G. Ivanov[a], E. Jafarov[a], V.I. Kaplin[a], A. Karmokov[e], E.K. Koshurnikov[a], V.Ch. Kudaev[f], V.I. Lobanov[a], A.F. Makarov[a], L.V. Malinina[a], V.L. Malyshev[d], K.V. Mikhailov[a], B. Morosov[a], G.A. Mustafaev[e], A.M. Nakhushev[f], P.V. Nomokonov[a], I.A. Oleks[a], V. Pismennaya[a], T.A. Pocheptsov[a], A. Polanski[c], G. Pontecorvo[d], A. Povtoreyko[a], Yu.N. Rogov[b], I.A. Rufanov[a], S. Ryabtsun[b], Z.Ya. Sadygov[a], R.A. Salmin[b], A.G. Samartsev[d], M.G. Sapozhnikov[b], T. Sereda[a], G.S. Shabratova[a], A.A. Shishkin[d], A.N. Skachkova[d], N.B. Skachkov[d], E.A. Strokovsky[b], R.Sh. Teshev[e], V. Tikhomirov[a], V.V. Tokmenin[d], E.P. Ustenko[a], V.V. Uzhinsky[c], N.V. Vlasov[b], A.S. Vodopianov[a], S.A. Zaporozhets[a], N.I. Zhuravlev[d] A.I. Zinchenko[a]

University of Edinburgh, United Kingdom
M. Aliotta, D. Branford, K. Föhl, D. Watts, P. Woods

Friedrich Alexander Universität Erlangen-Nürnberg, Germany
W. Eyrich, A. Lehmann

Northwestern University, Evanston, U.S.A.
K. Seth

Università di Ferrara and INFN, Sezione di Ferrara, Italy
D. Bettoni, R. Calabrese, P. Dalpiaz, E. Luppi, M. Savriè

Johann Wolfgang Goethe-Universität Frankfurt, Germany
R. Dörner, R. Grisenti, M. Kaesz

INFN-Laboratori Nazionali di Frascati, Italy
P. Gianotti, C. Guaraldo, O.N. Hartmann, M. Iliescu, V. Lucherini, E. Pace, C. Petrascu, D. Sirghi, F. Sirghi

INFN, Sezione di Genova, Italy
R. Ballantini, M. Macri, R. Parodi, A. Pozzo

Justus Liebig-Universität Gießen, II. Physikalisches Institut, Germany
M.G. Destefanis, W. Döring, P. Drexler, M. Düren, I. Fröhlich, D.G. Kirschner, W. Kühn, K. Makonyi, V. Metag, M. Nanova, R. Novotny, F. Ottone, C. Salz, J. Schneider, B. Seitz, G.-C. Serbanut, H. Stenzel, U. Thöring M. Thiel

University of Glasgow, United Kingdom
J. Annand, A. Borissov, D. Ireland, R. Kaiser, J. Kellie, K. Livingston, C. McGeorge, D. Protopopescu, G. Rosner

Institut für Kernphysik ([a]), Zentralinstitut für Elektronik ([b]), Forschungszentrum Jülich, Germany
M. Drochner[b], W. Gast[a], A. Gillitzer[a], D. Grzonka[a], V. Hejny[a], G. Kemmerling[b], H. Kleines[b], W. Oelert[a], D. Prasuhn[a], J. Ritman[a], S. Schadmand[a], A. Sibirtsev[a], A. Sokolov[a], T. Stockmanns[a], H. Ströher[a], A. Ucar[a], P. Vlasov[a], P. Wintz[a], P. Wüstner[b]

Uniwersytet Slaski, Katowice, Poland
J. Holeczek, J. Kisiel, B. Kłos,

Institute of Modern Physics, the Chinese Academy of Science, Lanzhou, P.R. China
R. Chen, L. Duan, Z. Hu W. Li, Z. Sun, G. Xiao, Z. Xiao, H. Xu, H. Xu

Institut für Kernphysik, Johannes Gutenberg-Universität Mainz, Germany
P. Achenbach, J. Pochodzalla, A. Sanchez-Lorente

Politecnico di Milano ([a]), Physics Department, Università di Milano ([b]) and INFN, Sezione di Milano ([c]), Italy
P. Alberto[b,c], R. Bassini[c], C. Boiano[c], I. Iori[b,c], S. Riboldi[a,c]

Research Institute for Nuclear Problems, Belarus State University, Minsk, Belarus
V.I. Dormenev, G.Y. Drobychev, A.A. Fedorov, A.E. Korneev M.V. Korzhik, A.R. Lopatik, O.V. Missevitch

Technische Universität München, Germany
B. Ketzer, I. Konorov, A. Mann, S. Neubert, S. Paul, L. Schmitt, Q. Weitzel

Westfälische Wilhelms-Universität Münster, Germany
D. Frekers, A. Khoukaz, A. Täschner, J. Wessels

Budker Institute of Nuclear Physics (BINP), Novosibirsk, Russia
E. Baldin, V. Malyshev, A. Maslennikov, S. Peleganchyk, G. Pospelov, A. Sukharev, Yu. Tikhonov

Institut de Physique Nucléaire, Orsay, France
M. Guidal, T. Hennino, M. Mac Cormick, S. Ong, B. Ramstein, J. Van de Wiele, J. Pouthas, P. Rosier, T. Zerguerras

Dipartimento di Fisica Nucleare e Teorica, Università di Pavia ([a]), INFN, Sezione di Pavia ([b]), Italy
G. Bendiscioli[a,b], G. Boca[a,b], A. Fontana[a,b], P. Genova[a,b], L. Lavezzi[a,b], P. Montagna[a,b], A. Panzarasa[a,b], A. Rotondi[a,b], P. Salvini[b]

Institute for High Energy Physics (IHEP)([a]), Protvino;
Tomsk State University (TSU)([b]), Tomsk, Russia;
and National Center of Particle and High Energy Physics (NCPHEP)([c]), Minsk, Belorussia
E. Ardashev[a], Yu. Arestov[a], G. Ayzenshtat[b], G. Britvich[a], B. Chuiko[a], S. Golovnya[a], S. Gorokhov[a], A. Kholodenko[a], V. Lishin[a], V. Parakhin[a] V. Pikalov[a], V. Shelikhov[a], N. Shumeiko[c], A. Solin[c], O. Tolbanov[b], A. Tyazhev[b], A. Vorobiev[a]

Petersburg Nuclear Physics Institute of Academy of Science (PNPI), Gatchina, St. Petersburg, Russia
S. Belostotski, G. Gavrilov, Y. Naryshkin, O. Miklukho, A. Sarantsev, V. Vikhrov

Kungliga Tekniska Högskolan (KTH), Stockholm, Sweden
B. Cederwall, A. Johnson

Stockholms Universitet, Sweden
C. Bargholtz, K. Lindberg, P.E. Tegnér, I. Zartova

Università del Piemonte Orientale Alessandria, Torino and INFN, Sezione di Torino, Italy
M.L. Colantoni, L. Fava, D. Panzieri

Dipartimento di Fisica Generale 'A. Avogadro', Università di Torino and INFN, Sezione di Torino, Italy
M. Alexeev, A. Amoroso, F. Balestra, R. Bertini, M.P. Bussa, O. Denisov, A. Ferrero, L. Ferrero, V. Frolov, R. Garfagnini, A. Grasso, A. Maggiora, M. Maggiora, G. Pontecorvo, G. Piragino, F. Tosello, G. Zosi

Dipartimento di Fisica Generale Università di Torino (^{a}), Dipartimento di Fisica Sperimentale, Università di Torino (^{b}), INFN, Sezione di Torino (^{c}), IFSI, Sezione di Torino (^{d}) and Politecnico di Torino (^{e}), Italy
M. Agnello[c,e], E. Botta[b,c], T. Bressani[b,c], L. Busso[a,c], D. Calvo[b,c], P. De Remigis[c], A. Feliciello[b,c], F. Ferro[c,e], A. Filippi[b,c], F. Iazzi[c,e], S. Marcello[b,c], G. Mazza[c], O. Morra[c,d], A. Rivetti[c], R. Wheadon[c]

INFN, Sezione di Trieste and Università di Trieste, Italy
R. Birsa, F. Bradamante, S. Dalla Torre, M. Giorgi, A. Martin, P. Schiavon, F. Tessarotto

Physikalisches Institut, Universität Tübingen, Germany
H. Clement, E. Doroshkevitch, K. Ehrhardt, P. Gonser

The Svedberg Laboratory, Uppsala, Sweden
H. Calén, C. Ekström, K. Fransson, A. Kupsc, P. Marciniewski

Institutionen för Strålningsvetenskap, Uppsala Universitet, Sweden
F. Cappellaro, B. Höistad, T. Johansson, I. Lehmann, A. Lundborg, Y.-N. Rao, Ö. Nordhage, J. Nyberg, H. Pettersson, K. Schönning, P. Thörngren Engblom, U. Wiedner, J. Zlomanczuk

Universitat de Valencia, Dpto. de Física Atómica, Molecular y Nuclear, Spain
J. Diaz

Stefan Meyer Institut für Subatomare Physik, Österreichische Akademie der Wissenschaften, Vienna, Austria
M. Cargnelli, H. Fuhrmann, P. Kienle, J. Marton, E. Widmann, J. Zmeskal

Soltan Institute for Nuclear Studies, Warsaw, Poland
Z. Guzik, M. Kisielinski, T. Kozlowski, D. Melnychuk, J. Wojtkowska, B. Zwieglinski

Warsaw University of Technology, Institute of Atomic Energy, Otwock-Swierk, Poland
B. Slowinski

Spokesperson: Ulrich Wiedner Email: ulrich.wiedner@tsl.uu.se
Deputy: Paola Gianotti Email: paola.gianotti@lnf.infn.it

Contents

11 Trigger and Data Acquisition

11.1 System Overview

In many contemporary experiments the trigger and data acquisition system is based on a two layer hierarchical approach. A subset of especially instrumented detectors is used to evaluate a first level trigger condition. For the accepted events, the full information of all detectors is then transported to the next higher trigger level or to storage. The time available for the first level decision is usually limited by the buffering capabilities of the front-end electronics. Furthermore, the hard-wired detector connectivity severely constraints both the complexity and the flexibility of the possible trigger schemes.

In our approach all detector channels are self triggering entities. They autonomously detect signals and pre-process them to extract and transmit only the physically relevant information. The data related to a particle hit substantially reduced in the pre-processing step are marked by a precise timestamp and buffered for further processing. The trigger selection finally occurs in so called compute nodes which access the buffers via a high bandwidth network fabric.

11.1.1 Key Requirements

Since the physics of $\overline{\mathsf{P}}$ANDA is frontier physics, high statistics is needed to detect rare processes or small deviations in physical distributions. Therefore the $\overline{\mathsf{P}}$ANDA experiment plans to operate at interaction rates of the order of 10 MHz (with increases up to 20 MHz at a later stage). Even with pre-processing on the detector electronics for a substantial reduction of the data volume, typical event sizes are in the range of 4 to 8 kB. This amounts to total raw data rates in the order of 40 GB/s and up to 200 GB/s later on.

Furthermore $\overline{\mathsf{P}}$ANDA has a rich physics program with many different topics based on varying physical selection criteria. This means that different detectors contribute to the selection process for the measurements, which in turn should be performed in parallel wherever possible, to maximize the physics yield.

A conventional approach to read out the experimental data is not sufficient to match both criteria – high flexibility and selectivity on one side and very high data rates on the other – at the same time. Therefore the concept of a continuously sampling data acquisition in which event selection takes place in programmable processing units is planned for $\overline{\mathsf{P}}$ANDA.

11.1.2 Major Components

Key technologies to be exploited within the DAQ framework are high speed serial links (10 Gb/s per link and beyond) and high-density FPGA with large numbers of programmable gates and more and more advanced embedded features. There are three major ingredients of the new architecture:

The basic building blocks of the hardware infrastructure, which can be combined in a flexible way to cope with varying demands, are the following:

- Intelligent front-end modules capable of autonomous hit detection and data preprocessing (clustering, hit time reconstruction, pattern recognition ...) are needed.
- A very precise time distribution system is mandatory to provide a clock normal from which all timestamps can be derived. Without this, data from subsystems can not be correlated.
- Concentrators/buffers provide point-to-point communication, typically via optical links, buffering and on-the-fly data manipulation.
- Compute nodes aggregate large amounts of computing power in a specialized architecture rather than through commodity PC hardware. They may employ FPGAs, DSPs or other computing units and have to deal with feature extraction, association of data fragments to events and finally event selection.

A major component providing the glue between all others is the network fabric. Here special emphasis lies in cascadable switches which can be reconfigured to reroute traffic for different physics selection topologies.

Finally, the various algorithms for front-ends and selection levels have to be developed and tested. It should be attempted to develop a common high level programming environment for all components so that algorithms can be coded by physicists rather

than engineers in more abstract programming languages than HDL and DSP assembler.

11.1.3 Data Flow Overview

The detector front-end electronics reads and digitizes the data and performs the first level of data reduction by determining valid hits, combining them to physical information like clusters, energies or even tracklets.

Each group of data are marked by a precise timestamp which allows the association of information belonging to one interaction. This data association has to proceed in a detector specific way allowing for different signal characteristics (length, delay, ...) and some data may have to be assigned to multiple interactions. Thus, each interaction corresponds to a detector specific time slice in the data stream.

According to the settings of the respective physics measurements, the data of a detector are either buffered or passed directly to a network of compute nodes. These are supposed to extract a first simple physical signature which allows a decision on which time slices have to be transferred to the next processing level. The decision is broadcast through the network so that the next level's compute nodes can address the relevant buffered data of other detectors. Some detectors with a particularly difficult reconstruction may buffer data across several levels, contributing only at the last event selection step.

The first processing level and the concentrators/buffers are attached to a configurable cascaded high speed network. This means that some parts of the data stream bypass the processors at lower aggregate bandwidth while others feed into the processors with high bandwidth. Which detectors send to how many processing units and which ones wait for the first selection broadcasts is fully programmable and requires no hardware intervention. Like this computing power can be assigned dynamically based one the computing requirements of the respective physics measurements.

The last network level is attached to the online reconstruction farm which performs the final event selection based on the reconstruction of complete events and applying a comprehensive set of selection criteria. This network can be more traditional and just has to provide the required aggregate bandwidth.

11.1.4 Benefits of the Concept

The new concept provides a high degree of *flexibility* in the choice of trigger algorithms. It makes trigger conditions available which definitely are outside the capabilities of the standard approach, an obvious example being displaced vertex triggering.

In addition, all sub-detectors can contribute to the trigger decision on the same footing, and there are no restrictions due to a hard-wired trigger setup. Different physics can be accessed either in parallel or after software reconfiguration of the system with the same hardware setup.

Apart from these fundamental advantages the system offers some further benefits. Its *modularity* allows to connect building blocks in different ways through network interfaces and to realize small test systems as well as large high-rate systems.

Furthermore the system is *scalable*, both by implementing more components and by providing more powerful ones, e.g. faster processors, higher bandwidth interfaces, larger buffers *etc.*.

In addition, the system should be *cost efficient* since it is largely based on mass market components and consists of a small number of different building block designs which are used in large quantities. After all, using the same setup for a large number of measurements allows the most efficient usage of the equipment.

11.2 Requirements for DAQ and Trigger

11.2.1 Design Goals

11.2.1.1 Data Acquisition

The DAQ system must provide electrical and logical interfacing to front-end electronics and collect the data produced by the $\overline{\text{P}}$ANDA detector front-end systems. Tight but flexible integration of trigger and data acquisition ensures adaptation to the entire range of the $\overline{\text{P}}$ANDA physics programme. The processed data will be assembled into event-correlated data streams (event building).

Tagging of subsystem data will ensure proper event building at all levels. The final output stream will be archived for further offline analysis. Multiple event data streams will be provided for online analysis. Integration with slow control will ensure that no detector parameters are modified during archiving runs. Flow control with propagation of

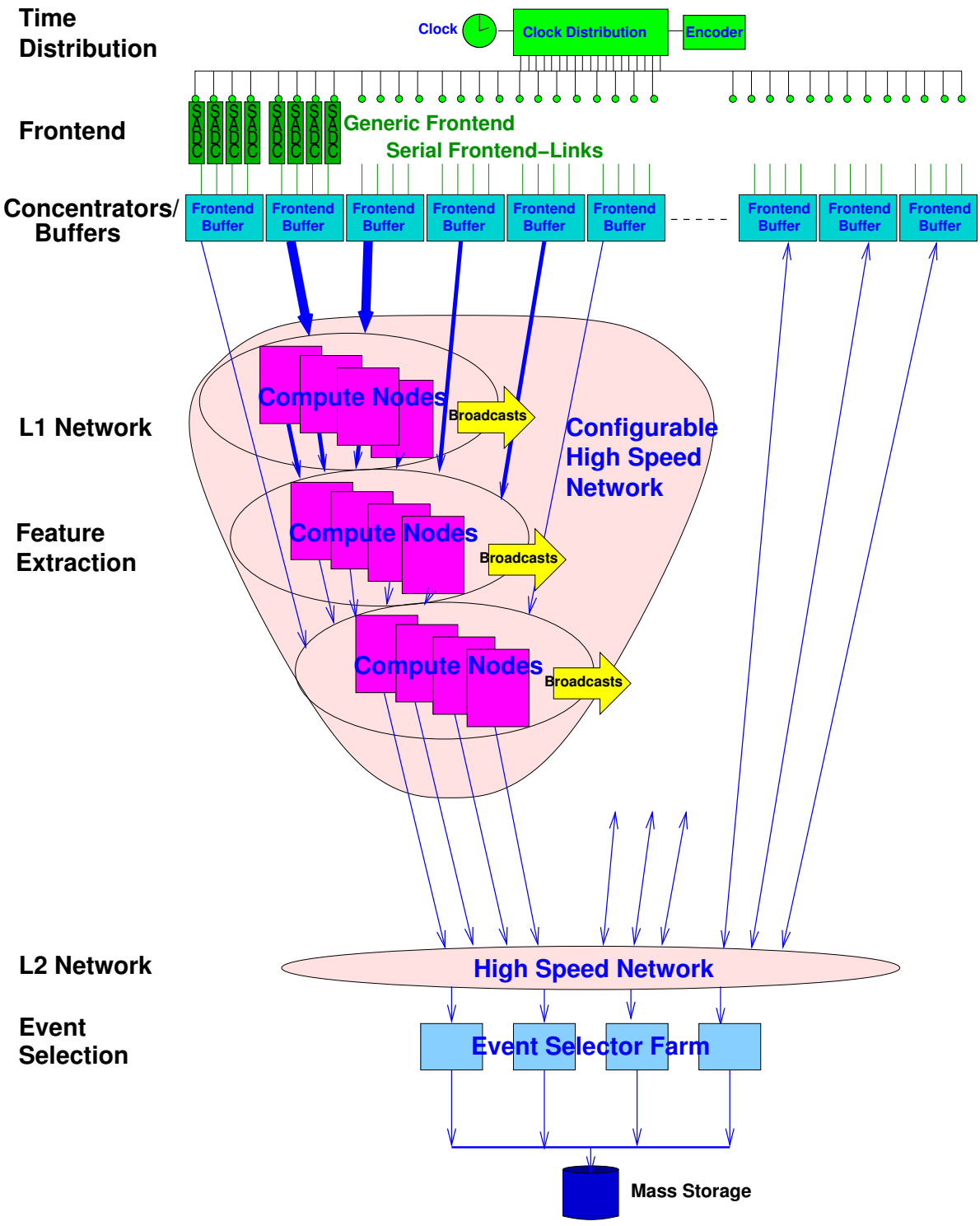


Figure 11.1: Scheme of the DAQ system

busy signals throughout the chain of DAQ/Trigger components is provided. The system has to be able to run in different modes including normal running with/without archiving, calibration runs, test/debugging modes. Flexible partitioning will support parallel debugging/calibration of detector subsystems. The system must guarantee high availability which can be achieved via techniques such as fault tolerance and self monitoring.

The system has to operate at an interaction rate of $10^7\,/s$. Data produced by the detector subsystems which is below predetermined thresholds will be suppressed by the detector front-end electronics. Moreover, dedicated feature extraction at the front-end level reduces the raw data rate down to a maximum of $3 \cdot 10^{11}$ bits per second which must be processed by the data acquisition system. Event selection in the trigger system will further reduce the data rate down to 10^{10} bits per second which need to be archived online.

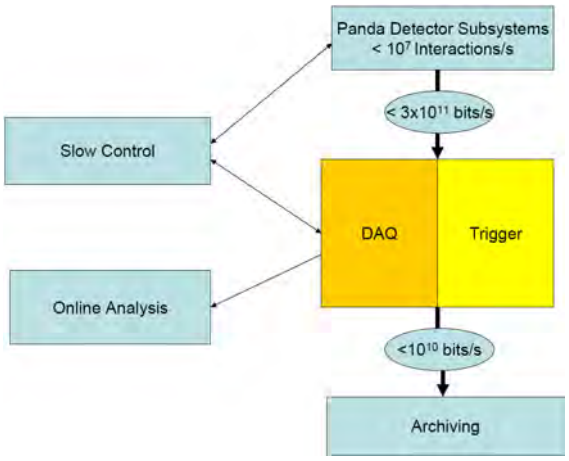


Figure 11.2: Schematic view of the data flow to/from the data acquisition and trigger system.

11.2.1.2 Trigger System

The trigger system will select a subset of the primary data stream based on physics signatures. The ability of the trigger system to reduce the primary event rate is strongly dependent on the physics to be studied. A significant part of the PANDA physics programme involves rare channels such as open charm production. Here, selective trigger algorithms such as the search for displaced vertices in the silicon pixel tracker can be devised which lead to a reduction of the primary event rate by more than 2 orders of magnitude. In contrast, other parts of the programme focus on channels with large cross sections. Here, the trigger system will be much less selective, requiring a reduction of the primary rate. This is compensated by the fact that channels with large cross sections can be investigated with sufficient statistics at reduced event rates.

The diversity of the physics programme requires a very flexible approach to triggering. Thus, conventional solutions with hard-wired trigger decisions are not optimal. Instead, massively parallel processing based on FPGA and DSP technology interconnected by a configurable high speed network is foreseen.

Data produced by the detector which is below predetermined thresholds will be suppressed by the detector front-end electronics. Moreover, dedicated feature extraction at the front-end level reduces the raw data rate down to a maximum of $3{\cdot}10^{11}$ bits per second which must be processed by the data acquisition system. Event selection will further reduce the data rate down to 10^{10} bits per second which need to be archived online.

11.2.2 Rate Requirements

In order to estimate the expected data rates, we have assumed operation at 10^7 interactions per second at a beam momentum of 15 GeV/*c*. GEANT4 simulations have been performed employing the actual detector geometry and a realistic event generator for the dominant annihilation channels (see Sec. 12.3.1 for details). The data rate will strongly depend on the choice of the central tracker. Two options are currently discussed which include a straw tube tracker (STT) and a TPC. A comparison of the expected data rates for both tracking solutions is shown in Table 11.1. For the TPC solution (bottom table), the resulting data rate is larger by a factor of 3 as compared to the STT solution (top table). In this estimate, front-end zero suppression and feature extraction as well as data compression using Huffman encoding for the TPC was assumed. Resulting data rates of 15 and 41 GB/s are obtained for the STT solution and the TPC solution, respectively.

This does not include electronics noise, pileup and background particles. Although the intelligent front-end electronics should be capable to distinguish unwanted hits from good ones, a safety factor of two should be foreseen giving an upper limit of about 80 GB/s.

11.2.3 Trigger System

Triggering is most effective for rare channels with clear physics signatures. A significant fraction of the physics channels of interest does allow very selective triggering. As an example, the decay of J/ψ into lepton pairs can be considered. A selective trigger would combine information from tracking and particle identification to reconstruct the invariant mass of lepton pairs. Candidates within the J/ψ mass region can be used as trigger signature for

15 GeV/c	Multiplicity	#DataBits/hit	#HeaderBits/hit	B/event	MB/s
FSMDC	45	20	16	203	2 025
FSEMC	102	24	14	485	4 845
FSHC	9	10	14	27	270
MVD	16	16	20	72	720
EMC	90	24	14	428	4 275
STT	69	16	20	311	3 105
					15 240

15 GeV/c	Multiplicity	#DataBits/hit	#HeaderBits/hit	B/event	MB/s
FSMDC	45	20	16	203	2 025
FSEMC	102	24	14	485	4 845
FSHC	9	10	14	27	270
MVD	16	16	20	72	720
EMC	90	24	14	428	4 275
TPC	850	12	16	2 975	29 750
					41 885

Table 11.1: Expected $\overline{\mathsf{P}}$ANDA data rates from GEANT4 simulations. Events representing the dominant annihilation channels at 15 GeV/c are considered for the two different tracking options (top: straw tube tracker, bottom: TPC). The primary interaction rate is 10^7/s.

events relevant to charmonium spectroscopy. The implementation of such triggers requires a highly parallel architecture which allows independent lepton candidate searches in the relevant subsystems. In a second stage, lepton candidates of opposite charge can be combined to calculate the pairs' invariant mass.

As another example for a highly selective trigger signature, we consider the decay of D mesons. Here, the signature is a displaced secondary vertex. Due to the small $c\tau$ of a few 100 μm, the relevant tracking information can only be obtained from the silicon tracker. As a second step, a calculation of the invariant mass of the decay products can be performed to further suppress background.

In order to efficiently utilize allocated beam time, it is crucial that the trigger system is able to handle multiple trigger conditions in parallel. The triggers should provide high selectivity without introducing a physics bias within the channels selected. In a typical run, triggers might be applied which differ in rate by several orders of magnitude. In order to avoid dead-time due to bandwidth limitations and archiving, appropriate downscaling factors for the individual triggers have to be introduced.

Moreover, effective methods for online evaluation of the trigger efficiency have to be available. It should be noted that trigger operation is strongly correlated to detector performance and stability.

Trigger processing will occur mainly on 3 levels. A first level, where only local information of a single subsystem is considered, will be able to identify signatures of rare events based on the local particle identification properties of the subsystem. Examples include pattern recognition for the Cherenkov detectors or the search for displaced vertices in the tracking detectors. Depending on selectivity, efficiency and fake rate of the local trigger, reduction factors between 3 and 10 could be expected. The second level will utilize the information derived from local trigger processing. Here, the aim is to reconstruct the invariant mass of rare particles using fast tracking methods which have limited momentum resolution. Reduction factors of the order of 10 can be expected. The third level trigger would then utilize more sophisticated tracking algorithms and could take into account other global observables such as kinematic constraints, depending on the reaction. This could further reduce the accepted event rate by one order of magnitude.

11.2.4 Data Handling

Latency and buffer sizes will strongly depend on the hardware implementation of the various algorithms under consideration. The First Level Buffers must be capable of holding data until a first level trigger decision is made. Assuming a latency of 3 μs for first level decisions, buffer space for 30 events must

be made available to cope with the interaction rate of $10^7\,/s$. This latency depends on the underlying algorithm and may extend up to $100\,\mu$s.

The data volume to be processed per event is approximately 4 kB (TPC scenario). In order to estimate the required processing power, we assume that on average 100 operations per byte are required. As a result, $4 \cdot 10^5$ operations per event or $4 \cdot 10^{12}$ operations per second are required for the first level trigger. This can be realized by implementing parallel algorithms in compute nodes based on large FPGAs, each assumed to operate at 0.5–1 Tops/s. Several hundred FPGAs will be required. The total number of FPGAs does not only depend on the desired processing power but also on boundary conditions imposed by data transfer. Modern FPGAs with high speed serial links are very well suited to implement such architectures. Moreover, the local topology of the subsystems has to be considered when designing the compute nodes.

Assuming a reduction of 3 by the first level trigger, the data rate to be handled by the second level trigger is approximately 14 GB/s. For the second level trigger, a latency of $100\,\mu$s is assumed, assuming FPGA - based compute nodes. After further reduction by one order of magnitude, the third level trigger has to handle a data rate below 2 GB/s. Here, larger latency of the order of several 10 ms for sophisticated trigger decisions including calibrations and sophisticated tracking are required.

11.3 System Design

11.3.1 Integrated Concept of Sampling DAQ

As lined out in the overview, the requirements should be met by a continuously sampling data acquisition system. This concept can only work, if massive parallel processing of the digitized data is performed on the way throughout all levels of data transport from the detector front-end to the final event selection. One has to achieve a sufficient data reduction and physics selectivity to arrive at a tolerable data rate to mass storage. In this section we outline the various levels of processing on the data path and which tasks and algorithms have to performed.

11.3.1.1 Data Reduction

This term denotes all the processing on the detector side which is aiming at the effective reduction of data by treating locally available information. It may include physics based data processing but not yet active selection of time slices across the experiment, i.e. beyond the local noise reduction.

First step in any detector readout is the noise reduction and zero suppression. This step includes of course the conversion of analog detector signals to digital data.

The second step is the evaluation of the signal time from data. Depending on the detector type a fitting of detector signals across several clock cycles can be necessary. Other detectors with intrinsically good time resolution are read e.g. by TDCs to give directly hit times. This is an important step in the preparation of event association, i.e. the combination of overlapping timeslices throughout all detectors belonging to one physical interaction.

Clusterisation of hits is performed in most detectors. In this way information is grouped and even improved (by calculating centroids) to minimize data size. Here overlapping signals from pileup have to be distinguished correctly, best by using refined hit times.

Pattern recognition is the last step in the preprocessing phase in which actual physical patterns are extracted from the digitized detector data. This includes e.g. track segments in tracking subsystems, energies in calorimeters or ringlets in Cherenkov detectors.

In the end, data concentrators with buffers store the preprocessed data which is reduced to the physically relevant information, ready for transport to more global processing levels.

At the same time, processing power on these concentrators can already group and address data according to the timing information as a further preparation for the event association. Here, the higher levels of data reduction (e.g. pattern recognition, clusterisation) can be performed as well.

On receiving broadcasts with data requests from the next processing levels the specified time slices are passed on to the indicated destination.

For the effective treatment of data on the front-end side, with respect to an accurate description of physical signatures, a constantly updating online calibration including alignment *etc.* is a prerequisite. The logical consequence is, that front-end processors perform this calibration autonomously and save their constants in regular intervals in a persistent database. Typically two modes can be distinguished, one where a fresh calibration database is built from some starting values and a sec-

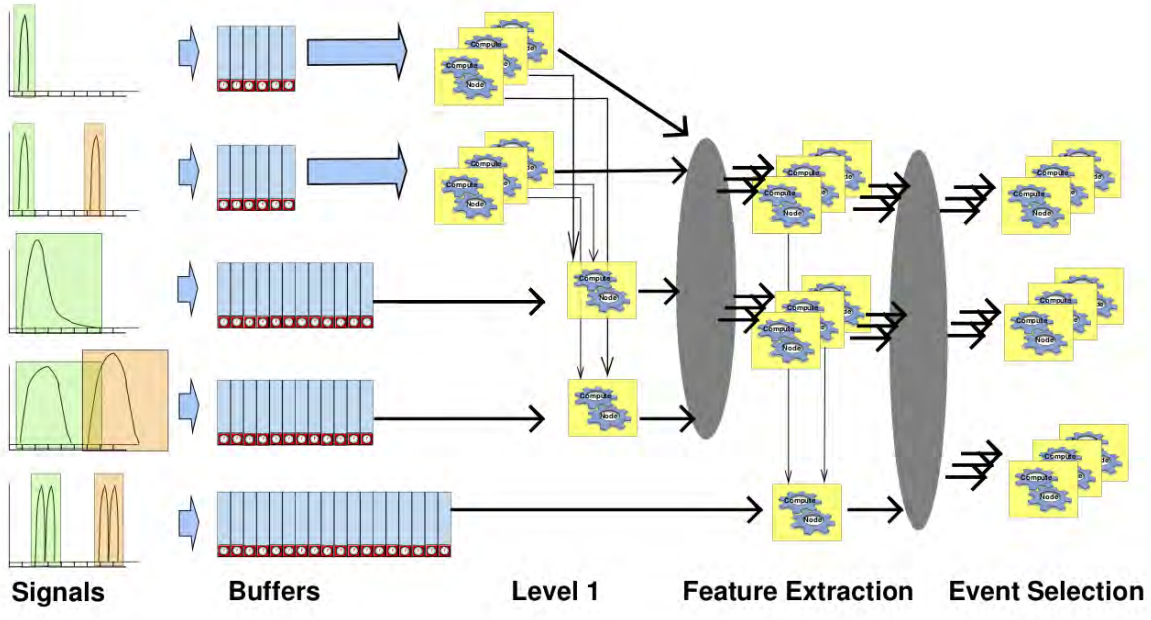


Figure 11.3: Different view of the data flow and selection.

ond one performing calibration updates throughout the measurement during actual physics data taking. The first mode needs feedback from more powerful processing levels e.g. including full track reconstruction which precludes physics data taking.

11.3.1.2 Feature Extraction

The next step in processing the data is to combine detector information to extract physical signatures relevant for the actual measurement of interesting processes. Examples are the fast determination of a muon track, the combination of two calorimeter clusters to find neutral pions and the verification of calorimeter clusters against tracking data to distinguish neutral and charged particles. Other examples are a fast determination of time-of-flight, the detection of tracks with large impact parameters w.r.t. the interaction point and the enrichment of kaons using PID detectors.

The various selection tasks can be grouped internally in several levels, for simplicity we consider here only two. The first level is based on signatures which are fastest to extract and require the smallest fraction of the total data stream. The corresponding detector front-ends deliver these data at very high bandwidth to processing nodes. These nodes determine interesting time slices from the signatures and broadcast the corresponding timestamps to the next processing level. This second level combines already processed data with still buffered data to achieve a more refined selection which then is the base for the final event building. The decision at which level a selection takes place is based among other criteria on the complexity of the selection algorithm, the amount of data to process and the number of detectors involved.

As described in the overview the feature extraction is embedded in a configurable cascaded network. This allows to change dynamically the bandwidth assignments between detector front-ends and compute nodes for different physics measurements.

11.3.1.3 Event Selection

The last step in the data processing before mass storage is the event selection based on completely assembled event data blocks. Here, the goal is to achieve the highest possible level of selectivity by performing a nearly complete event reconstruction with momentum information, invariant mass determination of decaying particles, vertex reconstruction and particle identification. This level can be performed at a large processor farm connected to the previous levels through an ordinary (i.e. static) high bandwidth network. The event reconstruction should be greatly facilitated by the fact, that front-end preprocessing already provides physically relevant information so that data decoding is trivial at this level and most processing power can be spent

for physics reconstruction.

11.3.2 Data Flow

In this subsection the flow of data is described. Two issues are important in this context, on one hand the actual architecture of the network and on the other hand the management of data flow.

11.3.2.1 Data Network

There are essentially two network classes present in the DAQ structure. A fairly low bandwidth class serves the need of controlling and programming the readout system. This comprises both detector control and programming networks as well as the time distribution network. Depending on the choice of the underlying hardware these two may be combined or not as explained in more detail in the section on network implementation (see Sec. 11.4.2).

The further focus in this paragraph however is the class of high bandwidth networks for data distribution. The total bandwidth as follows from the requirements is in the order of 40–200 GB/s. The data flow is governed by the respective physics selection scheme. This scheme implies the selection algorithm and along with it buffer latencies and maximum storage times. The buffers are located on the concentrator modules and are dimensioned big enough to hold event data for a maximum of 10 s, corresponding to a full reconstruction. If lower latencies can be kept by a faster selection this figure may be reduced. The processing nodes of the first selection level receive data from a specific subset of detectors, e.g. with 10% of the data volume. The transfer blocks are grouped in larger time frames to have bigger block sizes and thus more efficient transfers, which are assigned to single processing units. In addition this facilitates the combined analysis of adjacent, potentially overlapping time slices. The processing units of the first level must provide buffer and processing power to process the received data in a short time of the order of up to 1 s. This amounts to buffer sizes of around 50 GB. Once a node has processed a time frame it broadcasts the interesting time slices to the next processing level via a scheduler keeping track of which time slice is processed by which second level node. The reduction factor of the first level is in the order of 10 and with the same amount of buffering 10 s processing latency can be covered. Here a higher selectivity can be achieved, but a conservative estimate of the reduction is again 10. The final step is to transmit all remaining data blocks of selected time slices to the final level where a full reconstruction and high level selection is performed before mass storage so that in the end the desired 100–200 MB/s are reachable.

11.3.2.2 Flow Management

The flow of data through the processing network has to be controlled in order to avoid congestion and direct it to available processing nodes. In our system we consider several different options which shall be briefly discussed here.

In general one can distinguish between two classes, "pull" and "push" schemes of data transfer. In the pull-scheme a decision is taken which destination takes care of which part of the data and requests this from the data sources when it is ready. This scheme is especially attractive for the higher filtering levels since here the corresponding compute node just requests the time slice it has to process from the concentrators. The disadvantages of this scheme are, that in total more messages are sent and that some central intelligence ("scheduler") has to direct the assignment of slices to nodes. The information on available nodes can come from notifications of the nodes to the scheduler or from a modified round robin trial and error scheme, in which the scheduler asks certain nodes for their availability.

In the push-scheme data sources send data to the destination assuming that the destination is in principle always ready to receive data. They may be some buffering available on the source side and XON/XOFF signals can control, when the destination is ready to receive. But in average the destination has to digest the full rate.

A third way is a push-scheme where the destination is not a single compute node but a chain of nodes where each node either processes a data block or forwards it to another member of the chain. The chain can be constructed either physically from nodes with two interfaces connected one after another or logically by defining the passing of unprocessed data. The advantage is, that no scheduler is needed but an excess of compute nodes has to be made available to ensure the processing of all data and nodes later in the chain may be idle for a larger fraction of time then the first nodes. A variation of this would be a collection of star configurations with a central node keeping track of the availability of processing nodes assigned to it and distributing data accordingly. The hub node should be able to buffer data and reset processing nodes to overcome processing time-outs.

Finally, an additional aspect is the management of

buffers. A compute node broadcasts information on selected time slices to all concentrators/buffers, which mark the corresponding buffer areas for further processing. Unmarked buffers are erased or overwritten after a programmable time-out corresponding to the processing latency plus some safety margin.

11.3.3 Definition of Time Slices and Events

In a traditional track-and-hold system a detector is read out after a hardware trigger signal has arrived. Only after everything has safely been read, the system is cleared and ready for a new trigger input. More modern pipelined systems are able to process new triggers while previous ones are still being read. Here, a unique event number is needed to distinguish data blocks belonging to different events. Triggers are nevertheless - as in the old-fashioned way - read in a strict sequential order. However, in the concept of a continuously sampling data acquisition system this notion of trigger is lost. As a consequence it is not any more trivial to define which data belongs to each other.

However, this is not simply a feature (or weakness) of the sampling DAQ, but has its physical reason in the large rate of interactions and consequently detector pulses following one another. In fact, they may not even follow nicely any more but be smeared and overlapping. The problem outlined here has to be solved in the process of what we call event association. This process has in its realization two steps, first, the grouping of data into time slices, and second, the association of time slices to events, i.e. single primary interactions, which should be measured by the experiment.

The definition of time slices as the first step is a detector specific task. Here it has to be taken into account how the signal is generated in the detector and which are its characteristics. The crucial parameters are signal shape, signal duration, propagation or delay, amplitude and noise. It is also important to consider features of the specific readout electronics like double pulse resolution and deadtime. Based on these items a processing module on or close to the front-end electronics should characterise the (zero-suppressed) detector signals and group them into appropriate time slices. A detector with very good time resolution like a scintillator may have very short time slices whereas e.g. a gas-based detector might need to define longer and thus potentially overlapping time slices. This is graphically represented in the left part of Fig. 11.3.

The second step is even more elusive than the first: While one detector specific timeslice should correspond to one single physical signal it is likely that more than one time slice or physical signal can be created from one single primary interaction. Disentangling potentially correlated physical signals from background and other reactions can therefore depend also on the respective physics selection scheme. This means that, depending on the signature a selection of primary interactions is based on, more or fewer time slices might have to be selected for further processing and analysis.

This may then also lead to a non-unique assignment of time slices to events, to potential retransmissions of data and to the necessity of a wider latency window within which data cannot yet be safely discarded. A physical reason can again be overlapping physical pulses or too close primary interactions.

The proper way to deal with these issues in the design of the system is to come to the concept of time-addressed buffers which are typically located on the data concentrators. After the detector-specific assignment of time slices the data can be stored in buffers which can be addressed by external processing resources for feature extraction and event selection on the basis of time. The selection based addressing is then simply mapped on the proper definition of address ranges to access during a selection cycle. In this way the detector part and the selection part are decoupled logically, and may be so also physically (being handled by different processing units).

11.3.4 Data Filtering Stages

After processing on the front-end level the data are reduced to the minimum corresponding to the physically relevant information. This fact was already considered in the estimation of event sizes in Sec. 11.2. This still amounts to raw rates in the order of 40–80 GB/s (at 10 MHz), which is far too much for any storage system. However, up to now no selection of specific physics processes has been done. The storage requirement was set to 100 MB/s, so in total a reduction factor of 1 000 has to be achieved. A second constraint is the number of nodes per level, which has to be in a manageable order of magnitude. Furthermore the latency of selection algorithms determines the total buffer size. Here the fraction of data rate needed to be processed at one level multiplied by the latency gives the buffer size per level.

11.3.4.1 Bandwidth

Starting at a safe value of 100 GB/s and assuming that about 10% of all detector data has to be considered at the first level, 10 GB/s have to be transferred to the first processing level. Possibly less data would be required in case of simpler, cleaner signatures. With a latency of 100 μs per event 1 000 processing nodes would serve to digest the required input event rate of 10 MHz.

Similarly at a latency of 1 ms in the second level another 1 000 nodes would fit to read a first level rate of 1 MHz. Here data bandwidth is not an issue, since only at maximum only 10% of the initial events are selected plus a possible overhead of pileup. At the final step before event building 10 ms per event match to a 1% input from the second level.

However the description given here should not be taken literally, but only as an example of the evolution of rates. Different physics runs may require a different assignment of rates and levels and moreover different selection paths could be operated concurrently, e.g. a very clean signature leading to a big reduction factor may need only two levels whereas more difficult patterns may require more processing steps. Another issue is the possibility to combine two fairly simple signatures arising from two separate selection trees. This can be done already at the transfer level, e.g. by requiring in the scheduler that time slices are transferred only then to the second level if two subsystems from the first level have both marked them as interesting.

11.3.4.2 Processing Nodes

The various options for processing platforms are discussed in more detail in Sec. 11.4.3. Here the logical structure of nodes should be discussed. Basic features of the nodes are as follows:

- The basic network fabric of the DAQ system, e.g. Ethernet, is the common interface to all processing cards. Furthermore the nodes have to adhere to the common scheduling protocol in which processed time slices are marked by one level, assigned to the next level and addressed on the buffers.

- A common control and programming environment should be available. This comprises on one hand an (embedded) platform on which control software is executed and on the other hand a common high level language to describe the algorithms. In this way building blocks of a chain of algorithms can be assembled in a consistent way.

Beyond these common points the choice of the platform may depend on the actual type of algorithm. A simple pattern matching algorithm which has to digest a big amount of data at high speed is best implemented in FPGAs whereas more complicated floating point calculations are better represented in CPUs or DSPs. Therefore a possible scenario is a mixed pool of processing nodes based on FPGAs and DSPs/CPUs to which assignments are adapted according to the respective processing profiles.

11.3.4.3 Algorithms

In the following a brief overview of possible physical signatures and their implementation in algorithms is given.

Decay vertices. In the case of open charm decays, the most powerful signature is the observation of a decay point clearly separated from the interaction point. A reconstruction of a such a secondary vertex can give a reduction factor in the order of 100 or more. However to achieve this it might be more convenient to first simply require the presence of tracks not pointing to the interaction point as a first selection. The combination of all tracks for the actual vertex reconstruction is then performed at a higher level so that the total event rate for this more complicated task is reduced. At yet a higher level momentum assignments to the tracks would allow the determination of invariant masses.

J/ψ decays. A very clean signature already employed in many experiments are the decays of $J/\psi \to \mu^+\mu^-$ or $\to e^+e^-$. Easiest to realize is the trigger on muons because of the characteristics in the interaction with matter. But also electrons when combining to high energy clusters in the calorimeter and making an invariant mass cut provide a clean selection. Both can be done at the first selection level.

Lepton tags are more general signatures which are simple to realize but which have to be seen in conjunction with other filters to achieve appropriate reduction factors. Here triggering on muons from Charm decays, but also high energy electrons may provide an enrichment of D decays.

PID tags. A very important signature is the occurrence of a specific type of charged particle. This signature is covered by the Cherenkov detectors and the $\mathrm{d}E/\mathrm{d}x$ measurements, most notably in the TPC. Since there single detector systems are concerned, a preprocessing of ring patterns (RICH) or hit train curvatures and energies (TPC) would provide an early enrichment of interesting particles like kaons or electrons.

V^0 decays and hyperons. A whole class of interesting particles is based on the detection of V0 decays, i.e. the decay of neutral strange hadrons into two charged tracks. The V-shaped decay signature is easy to detect. The TPC hit pattern algorithm is well adapted to this task, but also other trackers can give reveal these decays with reasonable data processing. Based on this signature mother particles like Ξ can be reconstruction by adding further pions to Λ candidates at a higher processing level.

EMC correlations. Final states with photons or electrons are detectable by the electromagnetic calorimeter. The task of selecting physical signatures here is concerned with the combination of data from two opposite hemispheres. Since in $\overline{\mathsf{P}}\mathsf{ANDA}$ the hit multiplicity is fairly low simple combinatorial tests of photon or e^+e^--pairs can give an enrichment of there mother states (e.g. η, π^0 or $J\!/\!\psi$). Again, this should be feasible at the first processing level dealing with single detectors.

Missing energy. A more complex example is the detection of missing energy. First of all, this requires a good hermiticity of the detector. To extract this signature several detectors have to be considered and even a momentum balance of tracks has to be calculated to a certain accuracy.

If enough processing power is available, many of these selection tasks can be performed in parallel and more complex *selection masks* combining several criteria can yield high reduction factors at high efficiency.

The second level will then combine the signatures (particles) and perform computations like vertices or invariant masses. The bottom of the line is, that our system is flexible enough to achieve very high selectivity simply by anticipating typical data analysis steps in an act of processed data transmission. In this way a large number of detectors can participate in the extraction of the actually looked for physics at an early stage.

11.3.5 Discrete Event Simulations

In the proposed scheme of a continuously sampling data acquisition system with multiple layers for processing, proper buffering will play a crucial role in places where sampled and partially processed data have to wait for further processing steps before being finally stored or discarded. First of all we plan to use discrete event simulations to obtain indications on the amount of buffering needed at various stages and the optimal location where the buffers should be installed. As the amount of buffering is a function of available throughput and speed of implemented algorithms, we plan to construct models of various interconnections between buffers and processing nodes and use them to find the optimal architecture with indications on the placement of various components and required parameters (like processor speed or link throughput).

The placement of buffers has a direct impact on system latency, traffic patterns and congestion points. Modelling gives us a possibility to look into the internals of the architecture, usually not easily accessible with other tools, and follow the dynamics of the data flow. During the analysis of various buffering topologies we plan to monitor all aspects of data flow and look for an optimal combination leading to a stable, working system. The results from these models should be considered when taking a decision on the final architecture.

The type of behavioural simulation used here is known as "discrete event simulation". Basically, the simulation program maintains a time-ordered list of "events" i.e., points in time at which the simulated system changes state in a way implied by the type of "event" occurring. Only when an event occurs the modelled system is allowed to change its state. In most cases only a small part of the states of the simulated system needs to be updated. The state change can result in the generation of new events for a later time. These events are then entered at the correct position in the event list. The simulation program executes a loop in which the earliest event is fetched from the event list and then handled. We plan to organize modelling efforts around four activities in the sequence described below, with possible iterations of some or all steps in case we encounter significant discrepancies in demonstrating the feasibility of our model:

Parametrization: The components of the designed architecture may be quite complex. In our approach we will identify a limited set of parameters to keep the model as simple as possible but which

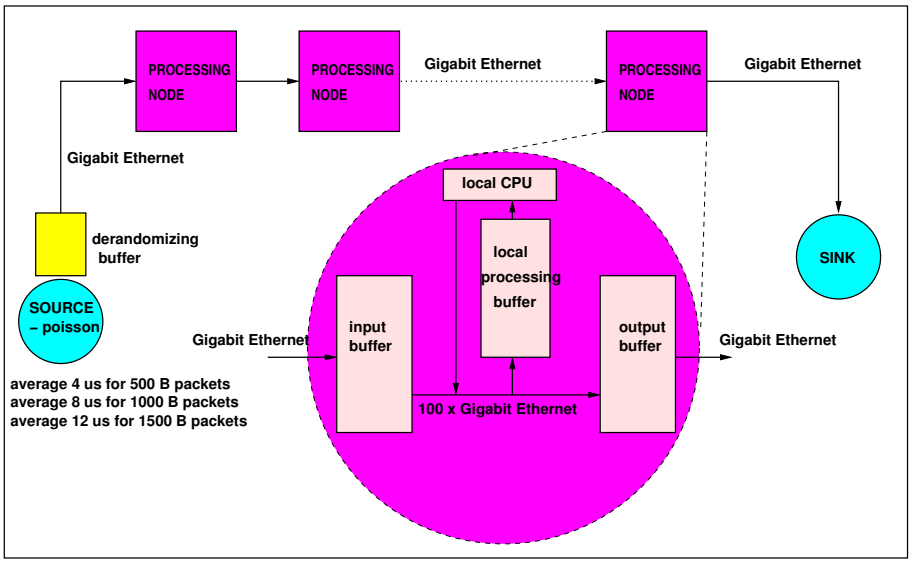


Figure 11.4: Schematics of a simple test case for discrete event simulations.

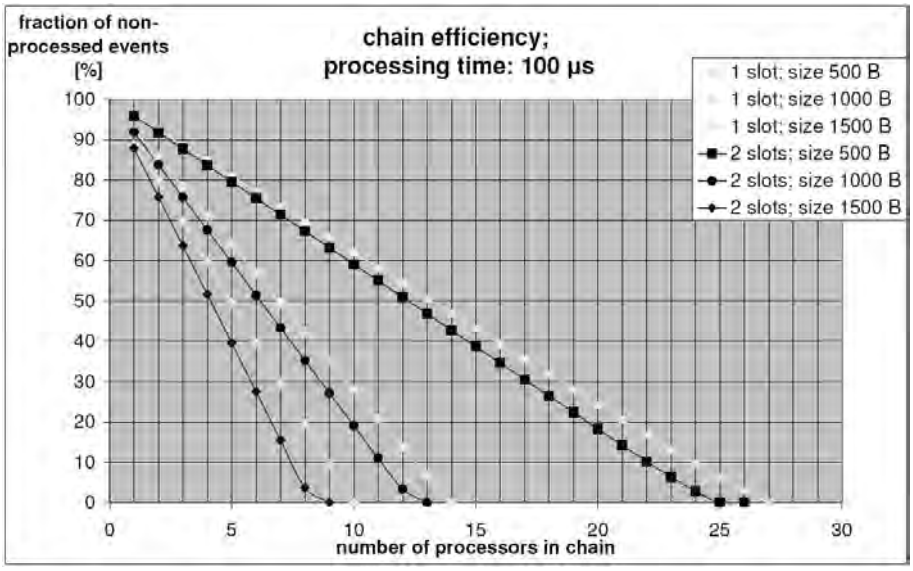


Figure 11.5: Efficiency of different chain architectures.

is sufficiently detailed to reproduce behavioural aspects relevant to the issues studied.

Calibration: The values for the identified parameters will be measured in dedicated setups. Hardware prototypes will be monitored using electronic equipment. Models of software will be calibrated by introducing timestamps into the code and taking measurements during the software execution.

Validation: Calibrated, parametrised models will be used to simulate small-scale test-beds where various prototypes of the components of the designed architecture will be put together to test the inter-operation and measure the performance. The capability to reproduce with our model the same results as measured in the test-bed will give credibility to modelling the full scale system.

Prediction: This is the final step of our modelling effort where models of various architectures

will be run and produce results (throughput, latencies) which will be the base for making architectural decisions for the final system. We considered and evaluated two tools (modelling environments) as platform for our model development. The Ptolemy environment, developed by the University of Berkeley, which has been closed and now becomes obsolete and the SystemC, developed by OSCI (Open SystemC Initiative) which offers free access to a set of libraries supporting hardware-oriented constructs implemented as the C++ classes. We selected SystemC for our development as it offers the same performance as Ptolemy in terms of memory usage and modelling speed but has now much better support.

A simple test architecture of daisy chained processing nodes, depicted in Fig. 11.4, has been used as example to demonstrate applicability of discrete event simulation for behavioural analysis. This is one model addressing the problem of handling busy nodes and data flow management in a simple way (s. Sec. 11.3.2). The links interconnecting nodes provide Gigabit throughput. The source of data (for example detector front-ends) produces packets of raw data according to the Poisson distribution with an average depending on the packet size in order to keep the Gigabit Ethernet link saturated. The packets are put into a de-randomizing buffer and stay there during the time the output link is occupied by the previous packet. Once placed on the output link they travel through the chain to reach the sink. If the processing node at which a packet arrives has free processing power, it uses the processing power for a fixed time and changes its contents from raw data into processed data. Upon conversion from raw to processed data, the size of the packet shrinks by half. Once processed, data packets do not need to be processed any more by the downstream compute nodes. In our model we investigated how the distribution of buffers within the processing nodes improves the performance of the architecture. We compared the performance of two chains: One with processing nodes having only one buffering slot for the local CPU and the other with 2 slots, where an additional packet can wait for the local CPU if it is busy. The rate at which the packets were generated follows a Poisson distribution with an average equal to the time a given packet size is being transferred over the Gigabit link (a 64 B packet takes 0.512 μs whereas 1500 B takes 12 μs).

Fig. 11.5 shows efficiency of the two architectures for three data sizes measured as a fraction of non-processed data arriving at the sink for a given length of the chain. The architecture with processing nodes having two buffering slots for the local processing requires a shorter chain to reach the same level of non-processed packets than the architecture with only one slot for the local processing. This observation is confirmed in Fig. 11.6, where CPU usage is plotted for the shortest chain with all data packets arriving at the sink as processed. A single additional buffering slot allows to use the CPU very efficiently – close to 100% and more slots are not necessary. In both cases processors close to the chain's end are used less efficiently as most of the traffic passing through these nodes is already processed, and the simple relaying of processed packets between input and output becomes the main task of these nodes. A possible optimization would be to use shorter chains with the possibility to re-direct non-processed packets from the sink back to the chain at some place close to the chain's end.

11.4 Implementation

11.4.1 Guidelines for Front-end Design

The concept of a continuously sampling data acquisition has major implications on the design of any detector front-end electronics. While in pipelined, but triggered systems a small dead-time can still be accommodated by regulating the minimum time between triggers, in a sampling DAQ dead-time is translated directly into detector inefficiency. Therefore any dead-time should not exceed the length of physical signals so that inefficiencies are only of the same order as pileup and signal overlap. Naturally, pipelines and buffering have to be implemented at all levels to avoid congestion and allow maximum de-randomization so that only bandwidths and transport and processing latencies have to be considered.

11.4.1.1 Data Reduction Levels

Throughout the front-end electronics several levels of data reduction have to be implemented. The first level concerns the zero suppression of raw detector pulses. This comprises noise reduction by common mode noise suppression and threshold comparison after proper pedestal subtraction. Furthermore, pileup, delta electrons and secondary interactions should be dealt with at this level. This first level has to be performed on the detector front-end. It may

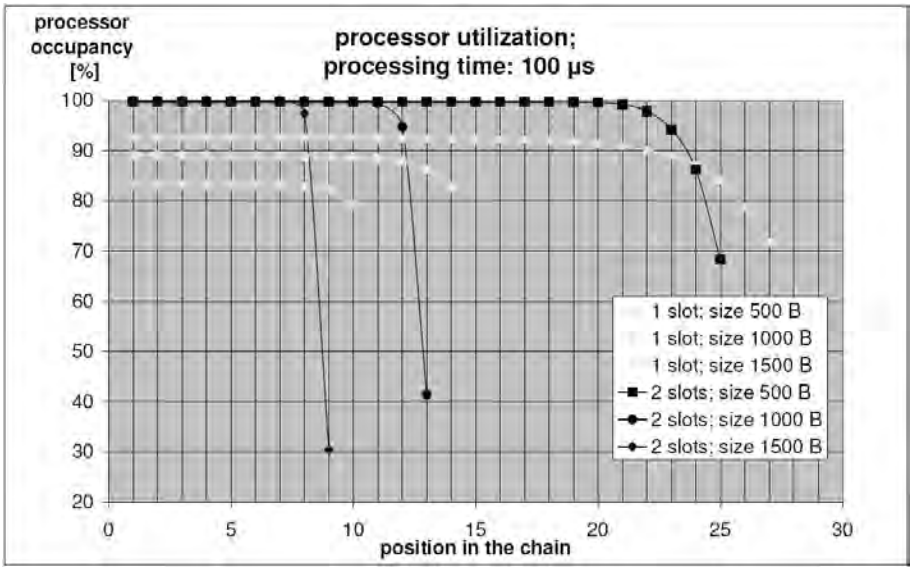


Figure 11.6: CPU usage for different chain architectures.

be located on the readout card directly connected to the detector channels or on a readout driver multiplexing several readout cards at a small distance.

On the next level, digitized data is treated to extract first physical signal features. Here clusterisation and time reconstruction take place. Amplitudes are treated to combine channels in a weighted summary information (cluster centroids, cluster amplitude sums). This level can be either performed on the detector front-end or on a subsequent processing unit.

On the third processing level, physical information is extracted from the data. Here amplitudes and centroids are converted into energies and coordinates. A first step in pattern recognition to form tracklets, rings, *etc.* can be performed at this level as well. It also requires access to online calibration constants. The proper place for the processing are data concentrator/buffer modules with attached processing units which interface between the detector level and the general cascaded high speed network.

11.4.1.2 Standard Interfaces

For the integration in the overall processing and data transport infrastructure several standardized interfaces have to be defined.

Time Distribution Network: The time distribution system described in Sec. 11.4.4 provides a stable clock to all readout systems with a jitter in the order of 25 ps. The signal is generated by a single source and transported via optical fibers and a network of passive optical splitters to all readout boards. From this clock normal all clocks relevant for timing and digitization are derived in order to achieve a synchronous readout with consistent timestamps.

Control and Configuration Network: Simple Ethernet will be employed as control and configuration network. Readout concentrators must have a small embedded CPU and an electrical Ethernet interface. Through this interface they receive asynchronous control signals and data for programming and configuration of readout processors attached to them. Local network switches will have optical uplinks to ensure electrical decoupling to avoid ground loops and limit noise.

Serial Data Interface: Wherever practical a standardized serial interface for data transmission should be implemented. This would be the standard input to default concentrator boards. The serial link should also implement a backwards direction for setup and programming of the attached front-end boards. Depending on detector requirements and cost an electrical or an optical version can be selected.

Data Network: The concentrators send their output to a cascaded high-speed network. This can

be (10-)Gigabit Ethernet or another switching network as will be specified during the design phase. If detector-specific concentrators are needed this interface must be implemented as this would be the last level where deviations from general purpose components may be allowed for special detectors. The PANDA Front-end Committee will define the standards for these interfaces which must be implemented by all types of front-end electronics. The strict adherence to these interfaces is mandatory to achieve a homogeneous infrastructure minimizing losses and ensuring data coherence.

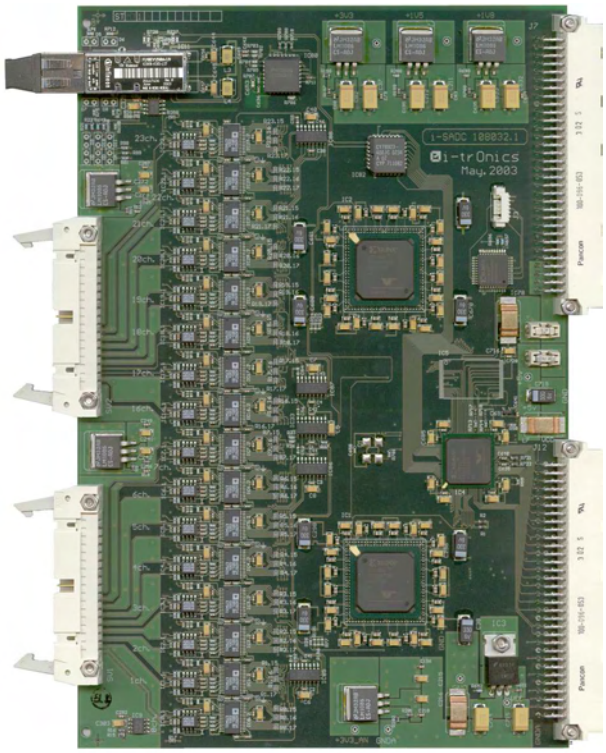


Figure 11.7: Sampling ADC VME module from TUM.

11.4.1.3 Example Front-ends

The goal of an early standardization of readout components is to reduce overhead by multiple parallel developments both in the design phase and the operation phase. With a standard set of front-end boards only the actual coupling to the detector including proper pre-amplification and shaping are open detector-specific tasks. The Front-end Committee will define a minimal set of front-end boards. Example front-ends are given as examples in the following, but the detector- specific parts are described in the detector chapters.

The first implementation of a small scale sampling DAQ is the Sampling ADC (SADC) from TU Munich. The SADC comprises on one VME card 32 differential analog to digital converters and some FPGAs as a first processing stage. The input range and impedance for each channel can be adapted to different detector signals by preamplifiers preceding each ADC. The ADC resolution can also be changed between 8, 10 and 12 bits to meet different precision and power requirements. The selectable sampling frequency up to 100 MHz, allows also an ideal adaptation to either raw detector signals or already preprocessed signals from a front-end ASIC. A first processing of the raw ADC data is then performed in FPGAs on the SADC module. This can include for example data buffering, pedestal subtraction, noise suppression, peak detection or signal time reconstruction. As the FPGA firmware can be reloaded during operation, it is possible to run different physics programmes with ideal matched firmware versions. The preprocessed data are transmitted to the next readout level via a bidirectional optic fibre link. To ensure also a precise time information between different SADC cards and channels, the fibre uplink to the SADC is used to distribute a low jitter clock signal to the module which then drives the ADCs synchronously.

Certain classes of detectors require a readout with very good time resolution rather than analog amplitude measurement. These are high time definition devices like scintillators or RPCs on one hand and detectors where time is a measurement defining coordinate like drift chambers on the other. For these detectors traditionally TDCs are used for readout. To avoid the necessity of a custom sampling TDC an alternative is the conversion of time into charge with subsequent analog-to-digital conversion. Like this only a pre-stage to the sampling ADC is needed. The time resolution that can be achieved in this way is with few tens of picoseconds quite excellent and beats most high frequency PLL devices. A design-critical issue is however the conversion dead-time at the first step. This can be reduced by fast switching parallel input stages.

Nevertheless where large channel counts and very low noise requirements are stringent, ASICs are unavoidable. To minimize the number of different front-end classes it is desirable to find a device that can serve many detector systems. For future experiments at CERN and GSI an initiative is under way to design a multi-purpose ASIC for TRDs, calorimeters, RICH detectors and TPCs. Its purpose is an accurate dead-time-free measurement of charge and time. It includes signal filters and

shapers, analog-to-digital conversion, noise reduction and signal processing for data reduction. It shall be able to run in a trigger-less as well as in triggered mode.

11.4.2 Network Infrastructure

The DAQ system utilizes a number of different networks to transmit the various types of information needed during full scale operation having different requirements on bandwidth, flexibility, stability and added functionality.

In general there are several possible choice in each case and the final decision will be based on design criteria as well as the market situation. The basic choices are between more customized networks with additional functionality and off-the-shelf high-bandwidth networks where functional overheads are simply absorbed by extra bandwidth.

In the following we outline the different network topologies for the required networks.

The simplest network is the control and configuration network. It serves to program components of the DAQ system, upload run condition data, upload or download calibration data and monitor the operation. This can easily be taken over by simple Ethernet.

The next task is the distribution of a precise clock normal along with short synchronous messages and potentially some additional asynchronous information. There are various options to implement this time distribution network:

The first option is a custom optical network as used in COMPASS and at LHC. This is unidirectional and can therefore be operated best by using passive optical splitters which provide a natural method to broadcast messages and deliver a very low time jitter in a simple and stable manner. The bandwidth of asynchronous information is however limited as data addressed to subsystems is received by all connected devices and has to be filtered according to a tag identifying the actual destination.

Another option is the usage of slightly standard optical networks, potentially even Ethernet. The essential modification here is the implementation of a standard clock used by all emitters and passed on from level to level with proper phase recovery. In this way a bidirectional link is possible and data can actively be routed to single destinations. The distribution of a stable clock is however a more complicated task and has not yet been done in such a framework. Nevertheless more economic components could be used and the number of different networks may be reduced by merging control and time distribution.

An intermediate way is the use of a passive optical distribution system with the capability of allowing directed back-links from receivers to the controller.

The most demanding network is the data processing network. The requirement is to be able to assign at full raw front-end data rate (after data reduction) processing nodes to any detector, to be able to configure arbitrary first level selection schemes. Of course, processing power and buffer sizes must match the latencies of the respective selection algorithms and the resources are limited within the full network. To achieve this, two roads can be chosen.

The brute force method is to over-assign bandwidth to all attached nodes in a away that almost all possible connection requirements can be met in any selection scheme. Then the limiting resources are only buffers and processing units, but bandwidth should not play a role any more.

The second approach requires a configurable network hierarchy. Here switches are connected in several layers and only the actual connections to processing nodes and concentrators/buffers have full physical bandwidth. The network itself is not a full cross-bar but switches have to be assigned to different levels to have sub-networks with higher bandwidth for first level selection and others with more nodes but less traffic for higher levels. This is depicted in Fig. 11.1.

The final network connection the feature extraction network with the event building network where the last selection step is performed after assembly of complete event blocks. This network is also hooked up to the storage network. Here standard computer center technology is fully sufficient.

For the time distribution network some customization is needed in any case. If this is done based on standard (Ethernet) components or with simpler (partly passive) optical networks is a choice to be made.

For the more demanding task of the high-speed processing network the selection of network technologies at hand or at least predictable now is given in the following.

Clearly the most obvious choice is to follow the evolution of Ethernet. As of now Gigabit Ethernet is fully mass-market, and even optical components for this become cheap. The standardisation of electrical 10-Gigabit Ethernet is at hand, the optical standard is in production in computer centres and network backbones. The formulation of 40-Gigabit

Ethernet is on the way and will be certainly available when $\overline{\text{P}}$ANDA turns on. Thus Ethernet offers a rather safe road to higher bandwidths at reasonable prices.

A new emerging technology is the advanced switching layer for PCI-express. Initially being a serial high-speed replacement of PCI and meant for usage only within computers the serial interface as such offers the possibility to have point-to-point connections over longer distances. With the introduction of advanced switching of PCI-express more complex networks are feasible at very high bandwidths. The new technology offers high scalability and the possibility to achieve high integration of large bandwidths. Clearly the aim of this technology is a high-performance mass market with strong interconnection of communication and computing infrastructures. On top of PCIe-AS multiple protocols can be implemented by means of encapsulation offering added functionality for a custom DAQ network.

Other network technologies in the high-performance sector are Infiniband and GSN (aka HIPPI-6400). They are both targeted at future high performance data centers and offer scalability and separability of I/O and processing resources. At the moment Infiniband has the broader industry base and bandwidths up to 100 Gb/s are under development.

11.4.3 Computing Infrastructure

The computing infrastructure of the DAQ system has to provide the necessary resources to treat in real-time data generated and distributed at all levels of the readout and selection process. In the implementation of the full system there are two competing tendencies: On one side, processing power has to be localized as close as possible to the place where data treatment is required at a specific level in order to reduce data traffic and bandwidth requirements. On the other side, the full system should have a reasonably homogeneous structure to be able to move single tasks from one level to another without having to surmount big architecture dependencies.

11.4.3.1 Processing Levels

A number of separate processing levels can be identified throughout the DAQ system. Boundary lines are of course not sharp but are defined by requirements as well as platform choices.

Front-end Board: The first processing step concerns the efficient data reduction on the detector front-end. After zero suppression the signal time has to be determined, hit clusters have to be found and online decoding with preloaded calibration constants should take place. These steps take place at the front-end board level.

Detector Processing Unit: At the next level data is formatted and may be buffered on concentrator modules. A processing unit attached to a concentrator/buffer module performs more complex cluster algorithms and first steps in pattern recognition. It also performs the grouping of data into timeslices for time addressed access.

Compute Node: General purpose processing units placed in the cascaded high speed network take part in the feature extraction, filtering physically relevant event signatures for further processing.

Farm Node: The final high level event selection and online reconstruction takes place in powerful computer farms attached to the storage network.

11.4.3.2 Platform Options

The collection of useful platforms depends very much on the evolution of the IT sector with time. However, different architectures, their advantages and disadvantages can be outlined and at the time of implementing a prototype architecture it will crystallize, which platform is used at which levels. Here we describe the options in a hierarchy supposedly going from devices closest to the detector to the end of the processing chain.

Only the very first level in the readout system, if at all, has to be implemented in ASICs. Here careful electrical design not to deteriorate analog signals, radiation tolerance and highly optimized designs are crucial issues.

Modern FPGAs offer a large flexibility and power and implement more and more additional features like integrated DLLs or even PLLs for highly stable clock pulses and time measurement. On the other hand, embedded CPU cores (MIPS, PowerPC, ARM, ...) allow for standardized communication and control. In addition, having ample amounts of gates available, complex functionalities previously deemed too costly like e.g. arithmetic units can be implemented in custom designs. Chip road-maps indicate that these devices are getting

more and more powerful. However, up to now proprietary tools for programming and even software development are needed.

DSPs allow to parallelise complex arithmetic operations of digital signals. As such they provide high but specialized processing power. Their drawbacks are the need for custom software development tools and specific interfaces to the computing and network environment. Most DSPs have high-performance serial or parallel data interfaces but with vendor-specific designs and protocols making it more difficult to communicate in a standardized, platform-independent manner. A powerful solution is their usage as co-processors for high-speed processing. As processing device attached to a standard platform they can perform simple but time-consuming and frequently routines.

A very promising platform are System-on-Chip devices, i.e. systems which implement all necessary peripheral interfaces, memory controller and bootstrap devices on the same chip as the CPU itself. These devices are more fault-tolerant and power-efficient than ordinary computers and can therefore be embedded on smaller form factors and at larger numbers. Their application can be seen in two directions, coupling to more specialized hardware to provide a control and programming platform or in massively parallel computing. They may supersede in future a large amount of more standard computing platforms.

The domain of high-performance multi-purpose CPUs will be the highly computing-intense tasks where multi-Gigahertz processing with fast floating point and integer pipelines is required as well as high-speed mass-storage interfaces and other high-level periphery.

Clearly, the choice of the platform for each level not only depends on performance and flexibility but also on economic boundary conditions. General market trends can therefore play an important role outside of pure design considerations.

11.4.3.3 Operating Environment

Since our system will comprise many thousands of processing nodes at the various levels, there is basically no alternative to royalty-free operating environments. The goal of a homogeneous environment across all levels can be best achieved by using Linux wherever possible: Linux is available as embedded variant on small scale micro-controllers as well as full fledged networked operating system on almost all standard types of CPUs.

An interesting strategy in the field of reconfigurable logic is the use of Linux on CPUs embedded on-chip to program FPGA resources on the same die. The FPGA is driven by Linux as a "device".

In addition there are Linux ports for most System-on-Chip devices. A seamless integration from front-ends to compute nodes to high level farms is therefore feasible and a common environment can be achieved allowing the interoperability of tasks and processes across all levels. In addition the development can start at an early stage on less powerful standard platforms and then be migrated at the time of deployment.

11.4.3.4 Programming Tools

By using standard tools like high level programming languages and compilers, algorithms can be ported across the system. Moreover, non-engineering personnel can participate in the development and implementation of algorithms making the direct use of physics based know-how possible.

Promising tools in the context of simulation and synthesis of FPGA based algorithms are e.g. HandelC and SystemC. In particular SystemC on one hand is connected to a growing Open Source community and has a strong industry base on the other.

In both cases a high-level language interface via C/C++ is provided with additions and libraries specific to hardware implementations. Core algorithms can - in an ideal case - be simply translated into hardware code to run on an FPGA or DSP.

11.4.4 Timing and Fast Control

Timing and synchronisation are common problems of high rate experiments in the field of particle physics. In order to synchronize the data and provide a time reference for high precision timing measurements the Time Distribution System (*TDS*) is needed. The system provides an absolute time by distributing the reference clock down to the level of the front-end electronics, i.e. very close to the detectors. The front-end electronics measures the timing of the detector signal respectively to the clock, adds an absolute timestamp and sends this information to the next level of the DAQ.

11.4.4.1 Requirements

- The maximum jitter of the time reference is 20 ps.

- The system is scalable from few hundred to few thousand destinations.
- The components of the TDS are mounted on the front-end modules very close to the detectors where radiation conditions exceed normal radiation level, therefore the components should be radiation tolerant.
- The time distribution system is flexible in order to follow the development and new requirements of the experiment.

11.4.4.2 Architecture

The architecture of the proposed Time Distribution System is based on existing systems like the TTC for LHC experiments [1], and the TCS [2], built for the COMPASS experiment at CERN [3]. The precise time reference is provided by distributing an encoded clock and data from a single source via a passive optical network to a large number of destinations. The basic architecture of the TDS is shown in Fig. 11.8 and includes only unidirectional data transfer, but we plan to study the possibility of using passive optical networks with a bidirectional interface. The bidirectional network will allow to combine the functionality of the time distribution system with the front-end interface.

We are considering two exploit one of two possible methods to implement the bidirectional network using the TDS architecture:

- One can use passive splitters as a concentrator to transfer the light signals from all destinations back to the central master module. This is possible because the light splitter has an asymmetric attenuation parameter depending on the direction of light propagation. The disadvantage of this method is that only one TDS receiver at a time may power its laser diode and switching from one TDS receiver to another requires extra time and a safe protocol.
- One can use active concentrators which combine information from e.g. 32 sources to one destination. This solution is certainly feasible but the reliability of the system has to be evaluated.

The Time Distribution System includes three main components:

- TDS master module with integrated laser transmitters;

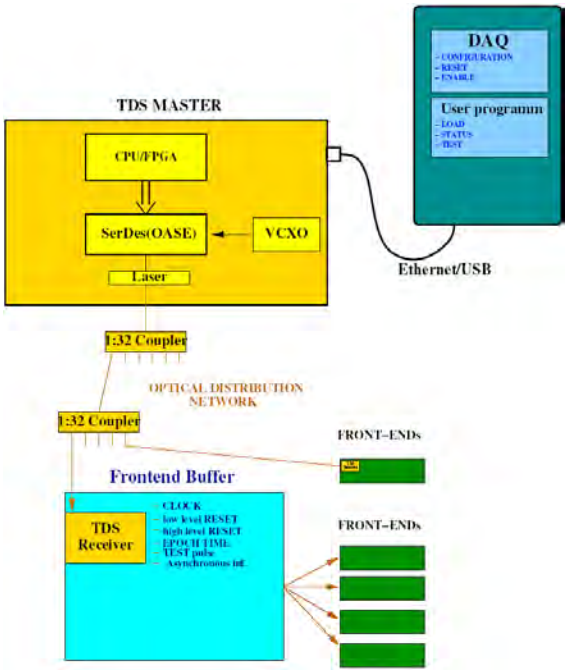


Figure 11.8: The TDS system, functional diagram.

- passive optical network with e.g. 1:32 splitters;
- TDS receivers.

The TDS master module is a central module of the system which includes:

- the processor and/or an FPGA to control the system;
- the temperature compensated oscillator crystal,
- OASE serializer/deserializer chip,
- the interface to the control software, DAQ and user programs;
- the control signal inputs,
- input/outputs for extending the system to a bigger number of destinations.

The TDS receiver can be a small mezzanine card with the OASE chip and an FPGA. The pin and laser diodes are integrated into the SerDes chip. The FPGA receives data from the SerDes chip, extracts control signals and provides asynchronous information to the destination module via the serial link. Every TDS receiver has its own unique ID for addressing and configuration via the optical link.

11.4.4.3 Clock and data Encoding

The clock and data are encoded in a single serial line by using standard 8 bit/10 bit encoder. Using a serializer/deserializer (SerDes) at a high speed of 2 Gb/s, which becomes a standard in these days, will eliminate the problem of clock jitter.
The most suitable candidate for the SerDes chip is the OASE chip which is being developed by the University of Mannheim in cooperation with the company ULM Photonics (Ulm). The functional diagram of the chip is shown in Fig. 11.9. The development of the OASE chip is part of the FutureDAQ project and gives the possibility to implement features which are specific for the time distribution system:

- radiation tolerant behaviour of digital electronics by designing the state machines and data transfer interfaces with a single bit error recovery logic;
- clock phase adjustment to compensate the fibre length and the position of the detector respectively to the target;
- clock divider to provide the front-ends with desired clock frequency;

11.4.4.4 Synchronous and asynchronous information

The TDS distributes two classes of information: synchronous and asynchronous. The synchronous information is a time critical control information, which is distributed with fixed latency. The synchronous information is decoded at the destination and provided as control signals to the front-end modules.

The synchronous information or control signals:

- low level RESET- reloads FPGAs, resynchronises PLL and DLL ;
- high level RESET - resets all data buffers, counters and state machines, it also defines the TIME ZERO in the experiment;
- GLOBAL ENABLE - enables/disables data taking in all front-ends ;
- EPOCH TIME - the signal is distributed with a fixed time interval, generates the EPOCH TIME information on the data stream and allows to verify the synchronisation of the destination module;
- TEST PULSE - initiate generation of the test pattern on the data stream, the test patterns allow to verify the integrity of the front-end DAQ interfaces.

The asynchronous information is distributed for configuration of the TDS receivers.

11.4.5 Control Software

To operate a high speed readout system it is necessary to have control over all aspects of the experiment to ensure stable measurement conditions and good data quality. To achieve this a control system must address several issues discussed here briefly.

Run Control. Data taking conditions have to be defined by means of a graphical user interface. These conditions comprise the setup of detector front-ends to read, triggering and selection conditions, run types (physics data taking, alignment, calibration, test, ...). Data flow should be controllable from this interface to view data rates and buffer levels to detect congestion or other problems at an early stage. It should be controllable which nodes or node-networks are participating in the readout monitoring host load (network, CPU, buffer) and responsiveness. Furthermore one has to define which types of data are to be recorded on mass storage and the state of the storage resources should be monitored.

Monitoring. During data taking the quality of the data has to be monitored constantly. This monitoring system extends not only to detector channels (wire maps). It also has to report readout errors (data inconsistency, missing front-ends, ...) and give access to online detector calibration constants. It should monitor physical signals, e.g. by means of showing standard particle yields and filter efficiency. Furthermore the beam has to be monitored to obtain its quality and keep record of the integrated luminosity.

Detector Configuration. Important running conditions are the proper configuration of the detectors and their readout. This includes thresholds and other parameters for data reduction. Of relevance is also the programming of front-ends with firmware, calibration constants, algorithms and lookup tables.

Detector Control. Finally, the control of detector parameters like high voltage, low voltage and

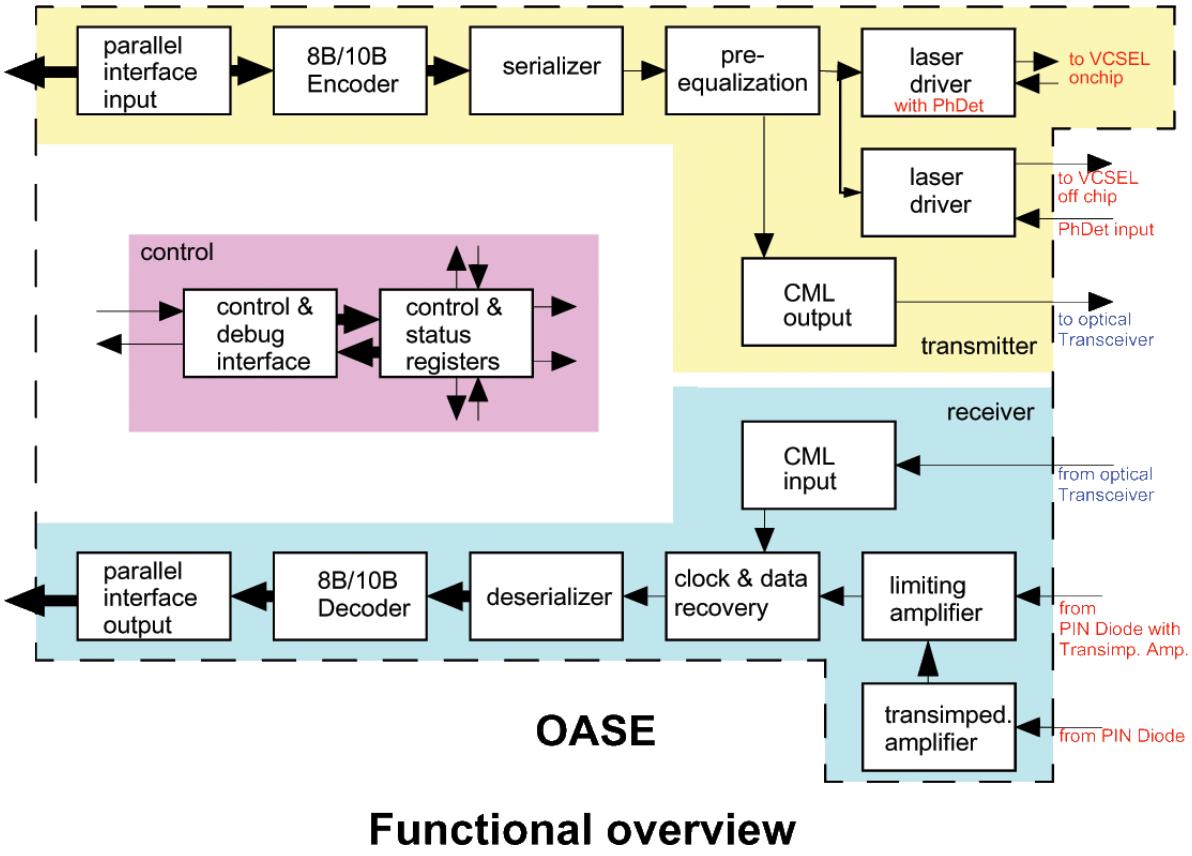


Figure 11.9: OASE chip, functional diagram.

currents, gas pressures, flows and mixtures has to be performed by the system. In addition environmental conditions like temperatures, atmospheric pressure and the readout of various probes (magnetic fields, switches and relays, ...) have to be monitored. At last, critical conditions have to be detected and announced and emergency procedures have to be executed.

A common database for all issues described above is desirable as is a unified user interface covering all aspects of experiment control. A joint experiment controls group at GSI shall work out the framework of software and hardware with which each experiment can realize its goals.

11.5 Planning

The development of the DAQ/Trigger system for $\overline{\text{P}}$ANDA is characterized by the fact that it is operating on the technological frontier in IT. To achieve the desired results not only technical aspects have to be addressed but the evolution of the IT market has to be taken into account, since it is impossible to develop all necessary base technology within experimental physics alone. For this reason even though general concepts and specifications have to be spelled out early enough to allow for a proper interfacing with all parts of the $\overline{\text{P}}$ANDA detector most feasible technical options should be kept open until the final prototyping starts.

11.5.1 Relation to FutureDAQ JRA

The key technologies of the relevant components will be explored within FutureDAQ, a Joint Research Activity within the EU 6th Framework Integrated Infrastructure Initiative on Hadron Physics (I3HP) aimed at the development of a data acquisition concept which is much better matched to the high data rates and to the complexity of the next generation of experiments.

The deliverables of FutureDAQ include the prototyping of a data concentrator, a time distribution system, a cascadable switch, compute and filter nodes and various algorithms. Since FutureDAQ is a joint project of the $\overline{\text{P}}$ANDA and CBM experiments not all results may suite our needs one-to-one but

may have to be adapted for reasons of functionality or budget.

FutureDAQ is supposed to finish at the latest at the end of 2007 assuming a one year prolongation due to the late start of funding.

After FutureDAQ is completed, a final evaluation and selection of designs shall take place and a prototype design for the PANDA DAQ and Trigger system will start. This prototype will be tested extensively and then the designs are updated for the subsequent initial production and deployment phase.

.

11.5.2 Responsibilities

Table 11.2 represents expressions of interest of groups involved in PANDA concerning the realisation of the DAQ/Trigger system is given. This implies that the corresponding institutes apply for funding for manpower, R&D and later on the construction of the system. Since this funding is however not yet guaranteed at this time, no legal commitment can be made yet.

Item	**Person/Institute**
Time distribution system	TU Munich
Concentrator/Buffer	Krakow
Data reduction	Torino
Compute Nodes	Gießen
Cascaded high-speed network	GSI & FZJ
Control Network	FZJ & GSI
Algorithms	Gießen, TUM, GSI
	Katowice, Pavia, Warsaw
Event filter farm	GSI

Table 11.2: Expressions of interest in work for the DAQ/T system.

References

[1] B. Taylor et al., IEEE Tran. Nucl. Sci. **45**, 821 (1998).

[2] I. Konorov et al., The trigger control system for the COMPASS Experiment, in *IEEE Nuclear Science Symposium Conference Record*, IEEE, 2002.

[3] L. Schmitt et al., IEEE Tran. Nucl. Sci. **51**, 439 (2004).

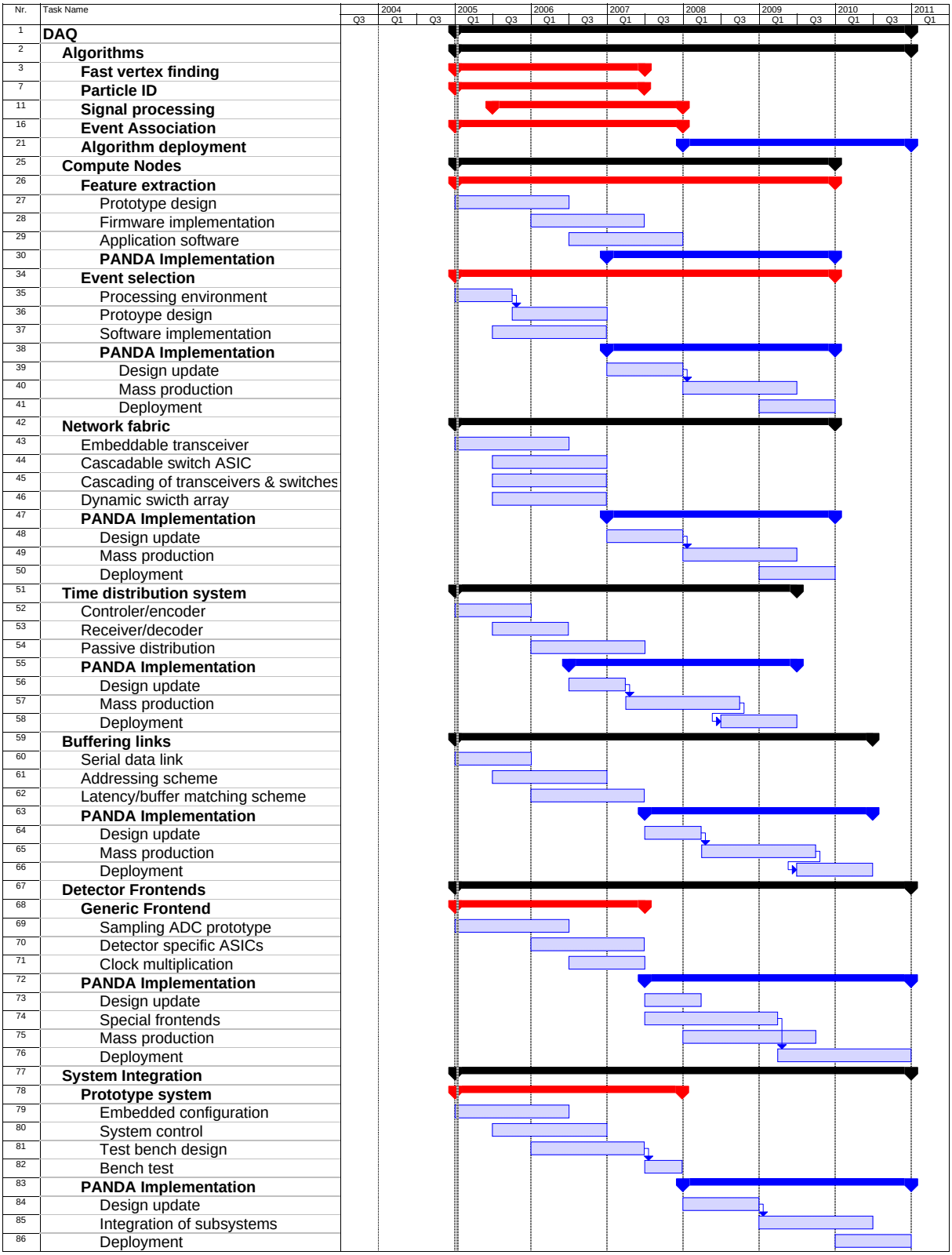


Figure 11.10: Estimate of timeline for development and deployment of the $\overline{\text{P}}$ANDA DAQ.