

H1 Second Level Fast Track Trigger Design Description

Project Manager:David MüllerAuthors:David Müller, David Meer, Jörg Müller, Martin HeimlicherReviewers:Daniel Erhardt, Jean Christophe Dumeril

Revision list:

Version	Date	Author	Remarks	Visa
0.0	31.10.00	dm	Preliminary	—
2.3	17.11.00	Dm	To be reviewed	
3.0	1.12.00	Dm	Includes Feedback from Review	
3.1	8.12.00	DM	Cosmetics DE	
3.3	27.12.2000	Dm	Approved by IPP and SCS	

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
2/53	28.12.2000, 15:53

Copyright reminder

Copyright © 2000 by Supercomputing Systems AG, Switzerland.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed and published in Switzerland.

While Supercomputing Systems AG believes the information included in this publication is correct as of the date of publication, it is subject to change without notice.

All cited trademarks and registered trademarks are the property of their respective owner.

Non-disclosure reminder

All information in this document is strictly confidential and may only be published by Supercomputing Systems AG, Switzerland. The permissions of the reader are defined in the non-disclosure agreement. Any violation of the non-disclosure agreement terms will be handled as described in the agreement.

Distribution list:

IPP
DESY
RAL
Manchester

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	3/53

1 Index

1	Index	(5
2	Introduction			7
	2.1	Status of	f this document	7
	2.2	Status of	f the project	7
	2.3	Reference	ces to use	7
	2.4	Technica	al / 'Physical' terms used	
	2.5	Project S	Scope	
3	Cons	straints		10
	3.1	Timing		
	3.2	Interface	25	
		3.2.1	FTT Data transmission	
		3.2.2	Input data from other H1 subsystems	
		3.2.3	I/O from/to the central trigger	
		3.2.4	Data output to the L3 system	
		3.2.5	Data output to the other L2 systems	
		3.2.6	VME interface	11
4	Solut	tion		12
	4.1	Hardwar	re	
		4.1.1	Overview	
		4.1.2	Multi-purpose L2 FTT card	
		4.1.3	Starting up	
		4.1.4	DATA Controller	
		4.1.5	Piggyback IO connector / cards	
		4.1.6	Local Bus	
		4.1.7	FPGA Interconnection	
		4.1.8	DSP Controller	
		4.1.9	VME Interface	
		4.1.10	DSP	
		4.1.11	Clocking	
		4.1.12	Power Supply	

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	5/53

4.2	SW Syste	m Functions		22
	4.2.1 Message System		22	
		4.2.1.1	Fitter / Decision Card / L3	24
		4.2.1.2	FEM \rightarrow Merger \rightarrow L2 Linker	25
	4.2.2	Control		25
	4.2.3	DSP Interfac	Ce	26
	4.2.4	Readout		26
4.3	Linking			26
	4.3.1	Description	of the algorithm	27
		4.3.1.1	The algorithm in detail	27
		4.3.1.2	Filling CAMs	31
		4.3.1.3	Searching links	34
		4.3.1.4	Implementation	34
		4.3.1.5	Binning of the κ - ϕ histogram	35
		4.3.1.6	Resource usage	36
		4.3.1.7	Timing linking	36
4.4	Fitting of T	Fracks		37
4.5	Forming L	2 Decisions		37
4.6	L2 Timing	g		38
	4.6.1	Transmissio	n delays	38
		4.6.1.1	$FEM \rightarrow Merger \rightarrow Linker$	38
		4.6.1.2	Data delay in the daisy chain	38
	4.6.2	Overall timir	ng	39
	4.6.3	Usage of the	e daisy chain	40
4.7	Controlling	g the L2 FTT		41
4.8	Space req	uirements		41
	4.8.1	Physical lay	out of a single card	42
	4.8.2	Front panel	(This section is quite preliminary and shows the state as of end of December 2000)	43
Using	g the Cards f	or an L1 Trigg	er	45
5.1	Introductio	Introduction: Description of the Problem		45
5.2	Solution: S	Solution: System Overview		45
5.3	Changes	Changes to the Hardware Design		48
	5.3.1	Speed	·	48
	5.3.2	Synchronou	s Clock	48
5.4	Operation			50
5.5	Timing			51

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
6/53	28.12.2000, 15:53

2 Introduction

2.1 Status of this document

The design description / spec of the hardware must be detailed enough to allow the implementation to start now. The description of the software (VHDL) is (except for the linker) only 'top level' and needs to be defined in more details before the implementation starts.

The document has been reviewed SCS internally and been approved. Both, SCS and IPP have accepted the design as it is described here.

The understanding of this document may be difficult without knowledge of the H1 trigger system as well as the information given in previous documents about the FTT and the datasheets listed in the references.

Both documents, design description and specification are accepted by SCS and IPP. In case of a mismatch between the two, the specification has higher priority.

2.2 Status of the project

Specification is agreed on by SCS / IPP.

Design phase has been finished with the review of this document.

Schematics of Piggyback cards have been reviewed and layout has started.

Schematics for the Main Board are being entered.

Critical components have already been ordered. All critical components for the prototypes of the PB Cards are already @ SCS.

2.3 References to use

(1) Feasibility study SCS: 'Parent' of this document, outdated.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	7/53

- (2) Project Book SCS: Administrative / financial aspects of the project as seen in spring 2000
- (3) Contracts
- (4) Specification SCS: Interfaces of SCS cards
- (5) Feasibility H1 & Addendum: Feasibility overall project, needed to get founding...
- (6) Specification H1: base of complete documentation, to be improved continually
- (7) Spec VME IF (Scott Kolya)
- (8) Spec Crates (Scott Kolya)
- (9) Spec Service Module (Scott Kolya)
- (10) Table FPGA Resource (SCS)
- (11) Table Addresses (SCS / H1)

2.4 Technical / 'Physical' terms used

- HERA: Hadron Elektron Ring Anlage: Accelerator at DESY
- H1: Name of the experiment
- DESY: Institute in Hamburg where H1 is situated
- IPP: Institute for Particle Physics (ETH Zürich)
- FTT: Fast Track Trigger
- L1 / L2 / L3 / L4: Trigger levels (increasing level means decreasing input rate and increasing processing time per event)
- L2NN / L2TT: L2 trigger components: Neuronal Network and Topological Trigger
- CJC: Central Jet Chamber used to track particles
- Layer1 ... Layer4: Layers of signal wires of CJC used in FTT
- FEM: Front End Module (FTT L1 processing cards)
- CTL: Central Trigger Logic. Combines 'Trigger Elements' from various subsystems to active trigger decisions ('subtriggers'). Distributes all synchronisation signals to all subsystems.
- HClk: HERA-Clock: 10.4 MHz signal synchronous with packets of particles colliding inside the detector.
- L1Keep: stops all readout-pipelines in case of an active subtrigger, initiates L2 operation
- Aclr: End of readout / abort readout: start of L1 operation

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
8/53	28.12.2000, 15:53

- Run start / run End
- PB: Piggy Back
- LVDS: Low Voltage Digital Signalling
- FPGAs (Field Programmable Gate Arrays):
 - DataCtrl: Data Controller
 - DspCtrl: DSP Controller
 - LvdsCtrl: LVDS Controller (on PB cards)
 - VmeIF: VME Interface (not developped by SCS)
 - DpramIF: DPRAM Interface between VmeIf (synchronous) and DpramIf (asynch)

2.5 Project Scope

Very short overview:

- SCS will develop cards used for the L2 FTT.
- SCS develops system level DSP code and VHDL code for embedding the DSPs into the system.
- IPP/H1 develops 'user code' DSP
- IPP develops VHDL code Data controller / LVDS controller with support from SCS.
- Card production will be handled by SCS.

Based on an additional contract, SCS will build the L2 cards such that they can also be used for L1 merging and L1 linking (see section 5)

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	9/53

3 Constraints

This section should make part of the specification. The information give here is intended to help the reader to understand the system design and not for specification purposes.

3.1 Timing

Operation of the L2 trigger starts after the L1 keep signal. An L2 trigger decision has to be delivered to the central trigger after 19.8 us. It is still open, whether the FEMs will send the first segements imediately after L1Keep or only somewhat later.

The final overall timing is to a large extent in the hands of IPP/H1 and not guaranteed by SCS.

3.2 Interfaces

3.2.1 FTT Data transmission

Track segments are received from the L1 system. Three inner layers consist of 6 FEMs each, on the fourth layer, 12 FEMs are needed. A maximum of 128 valid track segments is delivered per layer. The ordering of the track segments is arbitrary; if more then 128 track segments are delivered, only the first 128 segments will be processed on L2, ignoring the rest of the data entirely. No assumptions on the distribution of the track segments relative to the L1 cards are done (i.e. it does not matter whether they are uniformly distributed or whether all segments originate from one single L1 card).

Physically, the LVDS standard using the DS90C483 / DS90C484 serializer chipset by National Semiconductors will be used. The chipset allows to transmit 48 bit wide data plus clock at 33 to 112 MHz. For the FTT, the speed will be 52-104 MHz, yielding 2.6-5.2 Gbit/s transmission rate.

SCS has decided to use the 14526-EZHB-200-0QC type cable and the 10226-1210VE board connectors as recommended by 3M. A very similar cable by the same manufacturer is successfully used by SCS with a different LVDS chipset. Halogen free cables of this type have been found to be available, even though at higher cost.

Data transmission between the different L2 FTT cards and to L3 will be done using the same serializers.

In order to allow the PLL of the receiver to stabilize, the transmitter cannot be switched of completely when no valid data is transmitted. Valid data has therefore to be clearly identified (a valid 'zero' must

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
10/53	28.12.2000, 15:53

differ from 'no valid data'). Data formats are defined in (11) . Some motivation for the chosen format will be given in the section 4.2.

3.2.2 Input data from other H1 subsystems

It is assumed that additional input is needed for the calculations on the DSPs and on L3. The timing of this data is not critical as almost the entire linking step has to be finished before fitting starts. This data (32 bit words in strict order per event) is received by the L2 FTT through additional PB cards (DESY-PB) and fed into the LVDS chain, completed with the according addresses.

Similarly, any data output can be realized from the standard L2 FTT card, adding additional piggy back cards.

3.2.3 I/O from/to the central trigger

Even though it seems feasible to run the L2 FTT without input signals from the CTL (deducing e.g. the L1 keep signal from the data delivered by the L1 FTT), it seems to be useful, to foresee a few digital inputs e.g. to finish all tasks and to soft-reset the system by the 'AClr' signals, or to reset internal counters at run start. The physical definition of these inputs is fixed in (4), (9).

To provide the L2 FTT trigger elements to the CTL, LVTTL signals are driven from the Data Ctrl to the backplane.

3.2.4 Data output to the L3 system

Adapter cards in order to transform LVDS data to some standard which can be received by the L3 VME PPC Boards will be built by H1. SCS PB cards will be used there to receive the signals and eventually to build a short daisy chain.

3.2.5 Data output to the other L2 systems

Output to other L2 systems (L2NN / L2TT) can be done on any fitter card, if needed / wished. The data format needed here is not yet defined.

3.2.6 VME interface

The usage of a VME rather than a CompactPCI interface is unavoidable in order to fit into the H1 readout system. The interface must allow to configure the FPGAs, boot the DSPs, to download constants to the DSPs (e.g. the vertex location) and to read data from FPGA and DSP after an event is accepted. Readout is done only after an event is kept by L2, download of constants only at runstart.

In addition to the VME interface, the possibility to configure the FPGA via serial interfaces (JTAG) will be implemented anyway, in order to allow easy configuration in the testing / commissioning phase. Configuration without VME access is not possible any more, when the cards are installed at their final location

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	11/53

4 Solution

4.1 Hardware

4.1.1 Overview

An overview of the data flow is given in . The L2 FTT consists of four types of cards:

- The MERGER card is needed to collect the data from the different L1 cards of each layer. The main characteristics are:
 - 6 LVDS inputs from the L1 cards.
 - 1 LVDS output to the linker board.
 - An FPGA to multiplex all input data to the output.
- The LINKER card collects all track segments and scans all layers for coincidences in the kappa-phi plane. The coordinates of the linked track segments are communicated to the fitter cards.
 - 5 LVDS inputs from the merger cards.
 - As many LVDS outputs to the fitter cards as possible (tree / daisy-chain).
 - An FPGA for the linking task.
 - VME interface for readout.
- Each FITTER card carries 4 DSPs to calculate the parameters of the tracks. The accuracy of the kappa-phi determination can be increased significantly on this step. The parameters of the fitted tracks are transmitted via daisy-chain.
 - 1 LVDS input
 - 1 LVDS output
 - 1 Input for additional data
 - Eventually outputs to L2TT / L2NN.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
12/53	28.12.2000, 15:53

- The DECISION card is similar to the fitters. Output of all fitter cards is collected here and an L2 decision based on this information is formed by DSPs. Calculation of e.g. momentum sums starts as soon as the first track data is available.
 - 1 LVDS input
 - 1 Output to the CTL L2.
 - 1 (or more) outputs to L3.

All cards have a large number of IO connections which must be handled by an FPGA. The functionality of the FPGA is very simple, except for the linker. DSPs are necessary only on the fitter/decision boards. In spite of these differences it is still advisable, to use the same card layout for all three purposes: initial costs for design, layout and production would be much higher if different designs are used. The economisations due to smaller prints and cheaper components in the case of three different designs play a much smaller role because no more than approximately 20 boards are needed in total.

Using one single layout, the choice on whether to populate all boards identically (making bookkeeping with FPGA software and with spares simpler) or not (saving cost for large FPGAs and DSPs), was with the customer. It was decided to have three differently populated versions (see specification section 4.2 for more details on this):

- Merger (no DSP)
- Linker (no DSP)
- Fitter (DSP)

The boards will be carry 2 connectors for piggyback cards on each side. Every piggyback card can provide

- two LVDS inputs or (SCS)
- one LVDS input and one output or (SCS)
- IO channels to interface with other trigger systems (DESY).

Each connector will provide 60 parallel IO connections, attached to the IO-Controller FPGA.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	13/53



Figure 1 Overall view of the L2 FTT data flow.

Multi-purpose L2 FTT card

An overview of the functionalities of the main board is given in . The task of interfacing the IO piggybacks and the DSPs has been split on to two FPGAs in order not to have to go to the very largest devices available (number of available IO pins). The VME interface will be implemented in a third, small FPGA with its configuration permanently stored in a EPROM based memory

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
14/53	28.12.2000, 15:53



Figure 3 Block schematic of the L2 FTT multi-purpose card.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	15/53

4.1.3 Starting up

After power up, the VME controller will configure quickly. Download of the application dependent code for DSP controller and IO controller is then possible via VME. After initializing the FPGAs, the controlling PC writes the DSP code into the dual port RAM. On completion of this, a command is written to the DSP controller which starts one DSP after the other: the according bus switch is enabled and the reset signal is released. The DSP will load its code from the DPRAM.



Figure 5: Schematics for FPGA configuration

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
16/53	28.12.2000, 15:53



Figure 7: Schematics local bus and VME IF

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	17/53

4.1.4 DATA Controller

Depending on the card function, this FPGA has to fulfill completely different tasks:

- Merger: Multiplex data coming from 6 data streams to 1 output stream. One or two input streams must be read at a time, the output stream operated at the same time. The code needed is very simple; FIFOs, Multiplexer; 52-104 MHz.
- Linker: Receive data from all 4 layers in parallel, write to RAM and CAM at the same time. As soon as all data is received, the linking process starts and linked track segments are sent to the daisy-chains. The code is very complex and must reach high speed (104 MHz); see section 4.3 for more about the algorithm.
- Fitter: Receive data from the daisy-chain, retransmit with minimal delay; recognize data needed by the DSPs of the board, send fit results into the daisy-chain.
- Decision card: Receive data from fitters, communicate to DSP controller. Transmit trigger decision to CTL; transmit data to L3.

The number of IO pins needed is identical for all cards, see (10).

The ALTERA 20K400E type FPGA in the 672 pin FBGA case provides 488 user IO pins which would perfectly match our needs here for mergers and fitters. The spare IO pins make layout easier and reduce the risk of cross talk / ground bouncing in the FPGA.

The fastest speed grade (-1X) is necessary in order to reach the requested 104 MHz bi-directional data connections to the PB cards.

On the other side, the layout-compatible 20K600E to 20K1000E devices in the 672 pin FBGA case can be used for the linker board. The fastest speed grade will certainly make sense there.

4.1.5 Piggyback IO connector / cards

Four identical connectors will be mounted on the main board, two on the top side, two on the bottom. 48 data bits must be transmitted bi-directionally at 104 MHz. In addition 12 control signals (select IO channels, signals from the FIFOs) plus clocks are necessary. In total a 120 pin connector will be sufficient (see (4)).

As several additioal IO piggybacks may be needed in order to integrate the FTT into H1, the risk of plugging a wrong piggyback card to a main board cannot be neglected. If an input card is placed on a board where the FPGA is programmed to interface to an output card, serious damage of the FPGA as well as of the input card may be expected. A coding system pulling e.g. 4 pins differently to GND / +3.3V on each type of piggyback card would allow the FPGA to check at startup whether the correct cards are mounted and to keep the IO pins tri-stated in case of a mismatch.

4.1.6 Local Bus

During configuration, both FPGAs are accessed by the VME controller via their configuration pins which make part of the configuration bus.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
18/53	28.12.2000, 15:53

During readout or data download, the local bus is controlled by the VME controller, providing 16 data bits and 18 bit address space. IO controller, DSP controller and DPRAM (two chips providing together 32 bit words to the DSPs) are treated as four memory banks, adding 4 additional bits (chip selects).

4.1.7 FPGA Interconnection

Data transfer between DspCtrl and DataCtrl has to be possible in both directions. In order to make things simpler, two uni-directional connections, 24 bit wide and running at 104 MHz, will be used, splitting the standard 48 bit FTT 'tripple words' into two 'triple-half-words ©'. For uni-directional connections, the clock-lock feature of the ALTERA 20KE family is not needed, because of the shorter clock to out times, thus a 20K200EFC484-1 can be used for the DspCtrl.

4.1.8 DSP Controller

This FPGA provides the interface for the four DSPs and is connected to the VME bus via the local bus. It receives data from the IO controller and sends the DSP output back there. In addition this FPGA will be responsible to put the data needed for the readout to the VME controller or DPRAM.

Again, pin count shows that at least the 20K200 device is necessary. From experiences with ALTERA 10K devices, we can predict that the available logic cells will be sufficient and the slowest 20KE devices would be fast enough for that task, but because of the required IO speed, a 20KE-1 device is needed. Pin count DSP controller (including test-signals etc.): see (10).

4.1.9 VME Interface

Only a subset of the VME functionalities is needed and will be implemented, see (7).

4.1.10 DSP

Four DSPs of the TMS320C6701 series by Texas Instruments will be placed on the fitter boards

The DSPs are connected through the 'extended memory interface' (EMIF) to the DSP controller as well as to the DPRAM. As multi processor busses are poorly supported by that type of DSP, each of them is connected to a separated bus. Interrupt and flag lines allow communication with the FPGA (e.g. 'new track segments available' or 'fitting procedure finished').

4.1.11 Clocking

A 10.4 MHz HERA clock synchronous on all subsystems is received by the FTT L2 as well (from service module).

It is multiplied by two subsequent PLLs (x2, x5) in order to achieve the 104 MHz necessary for data processing. The synchronicity makes dealing with 'slow' signals HClk, L1Keep, etc simpler and is especially useful for the L1 option. As the 104 MHz clock is multiplied by another factor of 7 in the LVDS transmitter, a jitter on the 10.4 MHz clock of less then 1 ns might already seriously decrease the reliability of the LVDS data transmission.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	19/53

As fallback and for commissioning purposes, local 10.4 and 104 MHz oscillators are foreseen, with jumpers to chose between the different options.



Figure 9: Areas of different clocks.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
20/53	28.12.2000, 15:53



Figure 11: Clocking scheme.

For the 41.5 MHz needed by the DSPs, a local oscillator will be used.

4.1.12 Power Supply

Supply of +5V, 3.3V, 1.8V and -5V will be provided via the backplane. For the DSPs a 1.9 V supply will be generated on board in addition.

The total power consumption of a fully equipped board is estimated to be in the order of 30 W for 3.3 V and 12 W for the 1.8 V supply.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	21/53

4.2 SW System Functions

SCS is responsible for the design of all system level software on the L2 cards.

This includes VHDL code for control of the hardware and of the DSPs as well as a message system necessary to transmit data between the different devices / logical blocks on the card. (Implementation by David & David)

'User software' for FPGA (Linker) and DSP (Fitter) will be developed by H1. The linker is described by David Meer in section 4.3, the fitter code is not described in this document.

4.2.1 Message System

All data is transmitted in 48 bit ('triple-word') messages which basically consist of 16 bit of address and 32 bit of data (exceptions see 4.2.1.2). This format is motivated by the LVDS link and used at all locations. For the 16 bit wide local bus / VME IF the triple-words are split on three 16 bit words. The data format is defined in (11).

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
22/53	28.12.2000, 15:53



Figure 13: Block schematic of the L2 FTT message system.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	23/53



Figure 15: Schematic of the address decoding in the MessUnit of the DspCtrl.

4.2.1.1 Fitter / Decision Card / L3

On the L2 fitter cards, decision card and on the connection to the L3 system data ('messages') is exchanged between a large number of devices. Via the LVDS daisy chain data can be sent to any device 'downstream' the chain (e.g. from the linker to any fitter DSP or from the latter to the decision unit). In addition commands can be sent to any device from the VME IF and vice versa every device can send data to the VME readout FIFOs.¹

The message system on a fitter card is show in figure 7. Message units (MSG_UNIT / MsgUnit) are present in every FPGA where the same entities will be instantiated. All incoming data is fed via a Fifo

¹ Apart from the data sent actively by the entities on the card to the readout system, only control information can be accessed by the control system 'at free will'. This means that most of the data stored inside the L2 FTT is not visible to the control system.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
24/53	28.12.2000, 15:53

(to buffer data if several sources deliver data at the same time) into the message unit, where the receiver address is analyzed and the data is forwarded accordingly. The operation of the MsgUnit is shown for the case of the DspCtrl in figure 8. The input Fifos of DSPs, DataCtrl and local bus (VME) (plus eventually the CtrlUnit) are checked and served sequentially. If a message is to be processed, the ID of the destination card (byte 5 bit 7 downto 4) is checked. If the data is to be sent to a different card or to all cards, it is transferred to the interface of the FPGA interconnection and from there to the MsgUnit on the DataCtrl. If the data is addressed to the very card or to all cards, it is passed to the next level of the DataCtrl, it is again passed to the FPGA interconnection IF, otherwise it can be written to some of the output Fifos on the DspCtrl, according to the sub-address (to a DSP or to the VME Fifo). Byte 4 of the triple-word is used as an 'internal address' i.e. it is used by the target device in order to determine which kind of data has been sent.²

The message units on DataCtrl and LvdsCtrl work very similarly, whereby on the LvdsCtrl shortest possible latency for data travelling along the daisy chain has to be achieved (this will probably ask for a priority steered processing of messages).

4.2.1.2 FEM \rightarrow Merger \rightarrow L2 Linker

Data transfer rates between FEM and linker exceed the bandwidth which can be achieved with the message system. On the other hand, the data exchanged between these cards is essentially always the same and much less variety has to be handled, in specific, only very few destinations must be reachable³.

The first five bits contain all information which is necessary for a 'system level understanding' of the L2 FTT. If all bits are '0', invalid data is transmitted, if all bits are '1' control information is being sent. For all other cases (coded values 1 to 30) these bits code the index of the cell of CJC where the data originates from (followed by all other data needed to process a hit in the linker and the fitter).

Control information is sent whenever the status of the FEM system changes (and hence probably even the meaning of the data arriving after this control word will be different (see section on the L1 option)). It is obvious, that bit errors (faking control words on normal data or making genuine control word useless to the receiver) will be fatal to the system. Sensitive control words will therefore most probably have to be sent several times and include 'magic numbers' which can serve as 'keys' to validate them. Error counters can count the number of missing control words and control words with bad keys, which will increase the reliability and allow to estimate the rate at which really 'fatal' errors still can occur.

4.2.2 Control

Control blocks (CtrlUnit) are present on every FPGA and allow the control system (PC) to steer the status of the card (e.g. 'reset dsp', write card-ID into FPGA) or to read status information (e.g. identification of PB cards, status of system (L1 operation, run status etc)).

³ Data format and destination change between L1 and L2 operation. The switching between these two modes of operation therefore needs special care.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	25/53

² In some cases, the last two bits of byte 6 are already used as internal addresses (if device=all DSP).

The CtrlUnits of DspCtrl and DataCtrl are directly connected to the local bus. Sub addresses within the control block can be defined. It is still open, whether the CtrlUnits should be connected to the message system or not.

For the DataCtrl, the control block as well as the control block of the attached PB cards must be accessible. To access the control block of the PB cards, a specific command will first have to be written into the control block of the DataCtrl, interrupting the operation of the selected PB card and redefining all data lines between the two FPGAs as control signals, which can then be accessed by the local bus through the control block of the DataCtrl.

The control blocks can be accessed at any time after configuration of the according FPGA. The set of command-addresses (less then 8bit) and data (8 or 16 bit) is not yet defined.

4.2.3 DSP Interface

The control signals (Reset, IRQ, TIN/TOUT) of the DSPs can be accessed through the DspIf of the DspCtrl. Actions to be taken in case of changing TOUT signals will have to be defined here, e.g. sending messages to the readout Fifos.

Only 32 'data' bits of messages will be transmitted this way to a DSP , the 16 bit FTT addressing overhead will be discarded.

4.2.4 Readout

From the local bus a few addresses can be accessed on every FPGA of the main board.

On the DspCtrl, four addresses can be accessed from the local bus (with eventual sub-addresses):

- Control block
- VME Input (write only) where messages to any receiver can be entered into the message system
- VME STD output: Data written by the message system into this Fifo is cleared whenever an event is rejected by the L2 or L3 trigger.
- VME ERROR output: serious error messages may have to be kept even in the case that the event they originate from is not kept (maybe exactly due to this error). The VME ERROR Fifo is therefore not cleared by the Aclr signal, but must be read by the readout.

4.3 Linking

Linking track segments to one track is done on the linker card. The algorithm will run in the *DataCtrl* FPGA. The main tasks of the algorithm is:

- Receive track segment of L1 layer 1 L1 layer 4
- Linking corresponding track segments to one track
- Send linked track to fitter card

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
26/53	28.12.2000, 15:53

Track segments that belong to the same track will have approximately the same value of kappa and phi like the original track. This property is used as the basic idea of the linking algorithm.

4.3.1 Description of the algorithm

The linking is performed in a 2dimentional array, which is representing the kappa-phi plane. We use the convention to plot phi on the horizontal, kappa on the vertical axis. In a first step, this array is filled with track segments form L1 layer1 to L1 layer4. A track segment corresponds to an entry in this array.

In a second step, the algorithm is looking for multi entries in a bin. If a bin is found which has entries from different L1 layers, then a link was found and these track segments are sent to the fitter cards.

There will be a limitation of 128 track segments per L1 layer. It would be inefficient to map the kappaphi array in a RAM and to check it for all track segments. Therefore, the algorithm is working with a virtual array, which is represented by CAMs.

This algorithm, which is expected to be best, has partially been modelled in VHDL and fitted onto a 20K100EQC240-1 device (fastest speed grade of a chip which can only hold some parts of the logics needed)

4.3.1.1 The algorithm in detail

The 2dimensional kappa-phi array consists of a certain number of bins. With the *bin number*, each bin of the array can be identified. This bin number will already be generated for each track segment in the L1 part of the FTT and is part of the track segment data. Track segments arriving from the L1 have to be temporally stored in the *DataCtrl* FPGA. The bin number is stored in a CAM. In parallel to this CAM, a RAM works as a tag field. The content of this RAM points to another RAM (segment RAM), which contains the complete data of the track segment. The bin number is also written in a third RAM (seed RAM).

A simple algorithm would start with the first track segment in the seed RAM. With the bin number of this segment, the CAMs can be searched for other track segments in the same bin. If other track segments from other L1 layers with the same bin number are found, the rest of their data is accessible through the tag field. The data of these track segments can be sent afterwards to the fitter cards. The algorithm would repeat the same search with the second segment of the seed RAM and so on. It will stop if the last segment is processed. In such an algorithm, the adjacent bins of the search bin are ignored.

To improve this, it is necessary to consider a 3x3 window of bins. The first segment, from which the search is started (search seed), doesn't have to be necessarily in the centre of this 3x3 window. Therefore, a *sliding window* is needed to allow the nine possible positions of the first segment in the 3x3 window. However, this results in a bigger search window of 5x5 bins. This window is called *frame neighbourhood*.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	27/53



Figure 17: Naming convention for linking

A fast search can only be guaranteed, if a frame neighbourhood can be searched in parallel. Therefore, the kappa-phi array is divided in 5x5 sub array. A bin of a sub array is represented by a CAM. This is leading to 25 CAM per layer.

shows how the tracks segments are filled in the virtual $\kappa\text{-}\phi$ histogram consisting of

- 25 CAM (with sub array number)
- Tag RAM (with pointer to data)
- Segment RAM (rest of track segment data)
- See RAM (sub array number an CAM number)

For each L1 layer, there is one virtual histogram.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28/53	28.12.2000, 15:53



Figure 19: kappa-phi array for one layer

The sub array number is written to a CAM. The number of the CAM is given by the position of the segment in the sub array. The sub array number is also written to the seed RAM, from which the segment search later will be started. A tag RAM helps to find the segment data afterwards.

In case of multiple entries in exact the same bin of a layer a random selection is unavoidable. In this case, only the first segment is taken into account while the following segments are overridden.

The search goes through all segments of all four layers. Every single search starts with the sub array number from the seed RAM. This number is passed to all 25 CAMs in all 4 layers. All CAM are searched in parallel for a matching entry. Because the start seed will not always lie in the centre of a sub array, the sub array number has to be slightly adapted for each CAM (see).

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	29/53



Figure 21: Adaptation of sub array number

A 'link-quality' can be calculated for each sliding window. table 1 shows the weighting factors for a sliding window. A track segment in a bin of a sliding window is weighted by this factor.

1	2	1
2	4	2
1	2	1

table 1: Weighting factors for a sliding window

In a neighbour hood frame, there is space for 9 different sliding windows. For each of them, a weight sum can by calculated. It is the weighted sum of all track segments in the sliding window. For this sum, only one track segment per layer is taken into account (the closest to the search seed). If there is more than one track segment from one layer, the selection is defined by a priority scheme see Table 2: track segment priority in a sliding window in case of multiple entries.

7	4	9
3	1	2
6	5	8

Table 2: track segment priority in a sliding window in case of multiple entries

This 9 numbers are written in a matrix *LQM* (Link quality matrix). See also **Fehler! Verweisquelle** konnte nicht gefunden werden.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
30/53	28.12.2000, 15:53



Figure 23: Linking track segments

A link is found if:

- The weighted sum of the search seed is greater or equal than 5 (that means at least 2 track segments in a sliding window)
- The maximum of the 9 weighted sums is in the search seed bin. If there are other bins with the same entry, a well-defined selection has to be done.
- The search seed is the inner most track of the central bin. If not, this link was already found in a previous step.

After a link was found, the track segments belonging to this link, are sent to the fitter card.

4.3.1.2 Filling CAMs

Data arrives from L1 at 100 MHz - however every word has to be written only into one out of 25 CAMs. Writing to a CAM requires two clock cycles. A FIFO on the piggy back FPGA would therefore allow to

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	31/53

stall the writing operation whenever two segments following each other are to be written into the same CAM. The probability to have *n* succeeding segments in the same CAM is $25^{(1-n)}$. For 2 succeeding segments, this probability is 4%, for 3 it is already 0.16%. As long as the segments are to be written to different bins, the FIFO can be read at 100 MHz until it is empty again (no new data from L1 available yet). The only way, delay could arise from this filling procedure is, if the last segments of a layer are all to be written to the same CAM (delaying by 20 ns or 30 ns per segment).

operating modes:

 Mode
 Speed
 Problem

The exact procedure of writing depends on the operating mode of the CAMs. There are 3 possible

Mode	Speed	Problem
Single-match	~ 200 MHz	Doesn't work if 2 entries have same value
Multiple-match	~ 90 MHz	Too slow
Fast multiple-match	~ 190 MHz	Only 16 instead of 32 entries per CAM

table 3: CAM operating modes

Before writing in the *single-match mode*, an additional check as to be performed to guarantee that there is only one entry with the same data. This test can slow down writing speed.

The other reasonable operation mode is the *fast multiple-mach mode* with the disadvantage of a shorter storage depth. Because the number of track segment is limited to 128, the overflow probability can be calculated. The result of this calculation is shown in as a function of track segments. The calculation was made under the assumption, that the track segments are randomly distributed over the κ - ϕ histogram.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
32/53	28.12.2000, 15:53



Figure 25: overflow probability for a single CAM with a depth of 16 respectively 32 data words. The total overflow probability for the linker is a factor 100 (= 4 layers \cdot 25 CAM) higher.

Because writing is independent for every layer, also the segment and seed RAM have to be separately for each layer. Filling a track segment will cause the following operations:

- 1. Write the sub array number to corresponding CAM
- 2. Write segment RAM pointer to tag RAM
- 3. Write sub array and CAM number to seed RAM
- 4. Write data to segment RAM
- 5. Increase segment RAM pointer

This procedure has to be repeated for every track segment in each layer.

When the linking algorithm is finished, all RAMs and pointers are reset to zero, so that the linking algorithm is ready for the next particle event. CAMs don't have a clear input. But they can be filled with *never bits*. This way, any search in a CAM wouldn't be successful.

For one layer this procedure has been fitted onto a 20K100EQC240-1 type. The intended functionality can certainly be reached at 100 MHz. Less then 300 logic cells have been necessary to implement that model (including 'write' and 'read' operation).

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	33/53

4.3.1.3 Searching links

Searching through the CAMs (32 or 16 word deep) is possible at up to 175 MHz, one search per clock cycle. This has been tested together with the writing sequence for the 25 CAMs of one layer.

For the evaluation, a second test program was written. Starting point was an array 5x5 of logicals ('segment found'), the expected as output from the CAMs. The following 5 steps have been defined for the evaluation pipeline:

- 1. sum number of hits of all layers of each bin (resulting number 0..4; for each of the 25 bins in parallel)
- 2. For each of the 9 bins in the 'frame', sub-sums of 3 bins forming one row (including weighting) have been built. (3x9 times in parallel).
- 3. Sum the three sub-sums to the full sum ('link-quality'). (9 times).
- 4. Compare central bin to each of its 8 neighbors ('neighbor > central bin',8 times in parallel).

(+ compare central bin to threshold for 'good link' + check whether link has already been found before; not simulated)

5. ORe result of comparisons.

That algorithm consumed 679 logic elements but no ESBs. 139 MHz were predicted for the 20K100E-1 device. Together with the determination of correct bin-sub-addresses and searching through the CAMs and vetoing in case of segments in lower levels, a total pipeline length of 10 steps seams feasible.

4.3.1.4 Implementation

The implementation of the linking algorithm into the DataCtrl FPGA is shown in .

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
34/53	28.12.2000, 15:53



Figure 27: Logical blocks on the DataCtrl FPGA.

Track segments are received through the piggyback connector and are written to the different RAM / CAM by the *write CTRL* block. Because data are received in parallel for all 4 layers, they also have to be written in parallel to the FPGA memories.

The linking algorithm is started by the *link CTRL* block. It takes the first sub array number from the *seed RAM* and sends it to the CAMs. They will provide the track segments in the neighbour hood of the search seed. Form here, the process path is spitted up. One part of this path takes the track segments in the neighbour hood frame and check, whether there was a valid link or not (*link decision*). On the other path, the data to the track segments are searched in the *segment RAM* with help of the *tag RAM*. This has to work for all layers in parallel. After that, data of the track segments were written to a register in the message unit, where they will be deleted or put in the output FIFO of the message unit, depending on the link decision.

Finally, the message unit will send the linked track segments to a fitter card.

4.3.1.5 Binning of the κ - ϕ histogram

The binning of the κ - ϕ histogram has to be adapted to the resolution of the central jet chambers CJC of the H1 detector. If the binning is too coarse, the facilities of the CJC aren't exhausted and the probability for wrong linked tracks arises. On the other hand, a too fine binning will reduce the linking efficiency.

Actually, the binning is fixed to

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	35/53

- 40 bins for κ,
- 640 bins for φ.

From above it becomes clear, that the binning can be a good tool to optimise the effectiveness of the linking algorithm. Therefore, the number of bins shouldn't be fixed in hardware implementation.

4.3.1.6 Resource usage

25 CAMs per layer are necessary with the scheme described. The most important point whether the algorithm fits in a FPGA or not is the number of used ESB (Embedded system blocks). If a CAM occupies exactly one ESB, it can store either 32 or 16 lines, depending on the operation mode of the CAM (see also 4.3.1.2). Also RAM will be implemented in ESB. table 4: Used ESB resources on DataCtrl FPGA shows the ESB usage.

Block	Size	ESB
САМ	4 x 25 x (16 / 32 lines, 10 bit) ⁴	100
Tag RAM	4 x 25 x (16 / 32 lines, 7 bit)	8 /16
Segment RAM	4 x (128 lines, 40 bit)	12
Seed RAM	4 x (128 lines, 15 bit)	4
FIFO message out	4 x (128 lines, 40 bit)	12
Total		136 /144

table 4: Used ESB resources on DataCtrl FPGA

The 20K600E device with 152 ESBs seems to be sufficient. This device will contain 24320 logic elements - far beyond of what is needed for the tests in 4.3.1.2 and 4.3.1.3.

4.3.1.7 Timing linking

An overview on the maximal linking time is shown in table 5. It is based on the assumption, that there are 128 track segments in all 4 layers. This situation will happen in less than 1% of all events.

Tack	Clock s	teps	Total	Fraguancy	Time	
TASK	Repetitions	Pipeline	steps	Frequency		
Receiving	-	3	3	100 MHz	0.03 us	
Filling⁵	2 x 128	4	260	100 MHz	2.60 us	

⁴ Depending on the CAM mode: fast multiple-match: 16 lines; single-match and multiple-match: 32 lines.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
36/53	28.12.2000, 15:53

Search/Link	4 x 128	15	527	100 MHz	5.27 us
Total	6 x 128	22			7.90 us

table 5: Estimation of linking time

There is no large potential to improve this time. Writing to CAM requires in every operation mode two clock cycles. On the other hand, the linking algorithm is based on the idea that every track segment becomes a search seed for one cycle of the algorithm. This results in maximum of 512 repetitions.

The only possibility to improve the linking time is to save pipeline clocks. But saving 3 pipeline cycles – which can be hard – would reduce the time only about 1%.

4.4 Fitting of Tracks

The entire 'fitting business' is outside the SCS responsibility and will not be discussed here any more.

4.5 Forming L2 Decisions

No specific code to form L2 trigger decisions has been tested so far.

Evaluations of the track information can be done in two ways:

- On the DataCtrl of the decision unit. Even for large numbers of fitted tracks, sums of track properties can be calculated in very short time in fixed-point arithmetics.
- On the DSPs of the decision unit. It is very tempting, to do algorithms like mass combinations in floating point DSPs.⁶

Both schemes can be operated in parallel, each of them defining part of the trigger elements.

As the trigger elements will be driven to the backplane from the DataCtrl FPGA, decisions from the DSPs will have to be indicated to the latter device either by using messages or by using the TOUT pins of the DSP which for this purpose will be connected not only to the DspCtrl, but to the DataCtrl as well.

 $^{\scriptscriptstyle 5}$ Assumption: Writing only to 1 CAM. Writing to alternate CAMs - which is more realistic - will half this time.

⁶ As the number of combinations grows quadratically (or stronger) with the number of tracks, it seems to be possible, that certain masses are reconstructed only if a given number of tracks is not exceeded in an event.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	37/53

4.6 L2 Timing

4.6.1 Transmission delays

4.6.1.1 FEM \rightarrow Merger \rightarrow Linker

This time is of crucial importance for the L1 option, see section 5.5.

4.6.1.2 Data delay in the daisy chain

The manufacturer quotes a latency of 1.5 clock cycles + 4.5 ns latency for the LVDS transmitter, 3 clock cycles + 4.5 ns for the receiver. At 100 MHz this sum up to 55 ns. With the FIFO, another 75 ns are used up on the receiver side. 50 ns are assumed for the message processing; allowing another 20 ns for contingency, the data delay per card in the daisy chain amounts to 200 ns, excluding cable delay.

Note, that this time does not include delays due to the transmission between piggyback and main board, as it is foreseen, to use the two ports of the same IO PB card at every step of the chain.

Step	Delay
trans LVDS	20 ns
rec LVDS	35 ns
FIFO	75 ns
LvdsCtrl internal (message processing)	50 ns
Contingency	20 ns
TOTAL	200 ns

table 6: Timing delay in the daisy chain running @ 100MHz. Having 6 fitter cards between linker and decision unit, this time enters seven times in the total timing calculation.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
38/53	28.12.2000, 15:53

4.6.2 Overall timing

The times required for the tasks as described before are given in the two following tables. All numbers are considered to be conservative estimates.

		CC	freq (MHz)	time (us)	finished at
Latency L1-L2	1	42	104	0.404	0.404
Linking: receive data	2	256	104	2.462	2.865
Linking: fill CAMs (pipe + stall)	3	10	104	0.096	2.962
Linking: check CAMs	4	532	104	5.115	8.077
Latency daisy chain	5	147	104	1.413	9.490
Data Delay Fitting 1 (DataCtrl,	6	32	104	0.308	9.798
Data Delay Fitting 1 (Dsp)	7	8	41.5	0.193	9.991
Fitting 1	8	350	166	2.108	12.099
Data Delay Fitting 2 (DataCtrl,	9	32	104	0.308	12.407
Data Delay Fitting 2 (Dsp)	10	8	41.5	0.193	12.600
Fitting 2	11	350	166	2.108	14.708
Sums / eval	12			3.000	17.708
link/fitting overlap	13			0.000^{8}	17.708
SUM	Α			17.708	
SPARE TIME	В			2.092	
TOTAL TIME	C			19.800	
TIME USED	D			89.4%	

table 7: Timing overview; 2 us of spare time are available. Overlap between different tasks (e.g. fitting during linking) are not taken into account and will reduce the time needed in reality. On the other hand 'collisions' on the LVDS link may lead to some additional delay due to data stalls. If the FEMs transmit data at 42 rather then at 104MHz, an additional delay of approximately 180 ns is to be expected.

⁸ As fitting can start as soon as the first tracks are linked, some time can be saved due to the overlap of these actions. This effect has not been included in the present calculation.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	39/53



Figure 29: Plot of the L2 timing (see previous table).

4.6.3 Usage of the daisy chain

Additional delay can arise, if at some time too much data needs to be transferred via the LVDS daisy chain.

At the output of the linker card, a maximum of 4*48 triple words (track segments) are to be transmitted. If the link runs at 104 MHz, this corresponds to a total of 1.85 us transmission time (3.7 us at 52 MHz). Compared to the total linking time of 5.1 us, this time seems to be sufficiently short, if we assume that the linking of tracks is distributed more or less randomly over that period.

A worst case situation would be, if all the linked tracks are found at the very end of the search. In the last 48 cycles of the search, 4*48 segments to transmit might be found, leaving 144 segments in the output FIFO of the linker card at the end of the linking time. Another 48 segments would have to be sent until the last of the DSPs can start with processing of its first fit. The according delay is 0.46 us.^{9 10}

At the end of the daisy chain again 4 triple words per fitted track will be received. As the fitting of the tracks takes approximately a fixed time, the track information will be delivered with a similar temporal distribution as the track segments. (naja)

 $^{^{10}}$ If the LVDS link between linker and fitter is operated at 52 MHz, this additional delay may be as much as 1.38 us.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
40/53	28.12.2000, 15:53

⁹ For the segments used in the second fit performed on each DSP the additional delay in the output FIFO of the linker can be neglected since this data is only needed after the first fit on the DSP is finished.

No significant amount of additional data ('PQZP') will be entered into the daisy chain, but this data will be received by a piggy back on the decision unit.

4.7 Controlling the L2 FTT

Being embedded into a complex system, different flows of control signals can be defined. Possible sources of commands are:

- VME
- CTL signals via dig. input on main boards
- LVDS

Every card receives a set of control signals from the CTL (e.g. Runstart, L1Keep, AClear) which initiate tasks on the boards. Specific commands (e.g. 'end of L1 operation') can be transmitted using the LVDS link.

4.8 Space requirements

For board and connector size see specification.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	41/53





Figure 31:Top view of the physical layout of a main board with piggyback cards on both sides. Components: 1: VME connector; 2: Piggyback connector; 3: LVDS receptacle; 4: LVDS plug; 5: active components main board (with passive components on opposite side of the PCB); 6: components piggyback board (no components are mounted on the opposite side)

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
42/53	28.12.2000, 15:53

An approximate view of the geometrical layout of the card is given in Figure 31. The board-to-board distance is given by the PBConnector and amounts to 16 mm. 1.6 mm thick PCBs have been assumed.¹¹ While the main board is equipped on both sides (active components on the top side only, passive components (R, C, through-hole pins) on both sides), the PB cards are populated on one side only. On the figure, active components are indicated as 5mm high band, the passive ones as 2mm thick. The remaining gap between MB and PB is 6 mm or 9 mm respectively and should be sufficient in order to allow a sufficient air flow to cool the components. No components are foreseen on the bottom side of the PB cards. As you can see from the following table, less then 2 mm per board are left between two adjacent PB cards, if a 'sandwich' is installed in every second VME slot. Therefore every through-hole component or screw which covers some space underneath the PB must be taken into account carefully!

PCB MB	1.6 mm	1 x = 1.6 mm
PCB PB	1.6 mm	2 x = 3.2 mm
PB Connector (free space)	16 mm	2 x = 32 mm
TOTAL		36.8 mm
Width one VME Slot	20.32 mm	2 x = 40.64 mm
Space between two 'sandwiches'		40.64 – 36.8 = 3.84 mm

Table 8: Width of a 'sandwich' consisting of a MB and PB cards on both sides.

If additional MBs or PBs are developed to fit into the present system, the developers should try to restrict themselves to the given constraints. It is not excluded, that some devices (DSP, Linker) will need additional heat sinks which cover some of the free space in the Figure 31 - if that space is covered by additional components from some PB board, that option may be lost.

4.8.2 Front panel (This section is quite preliminary and shows the state as of end of December 2000)

A front panel does not only look nice but it is very important in order to guide the flow of the cooling air. A preliminary layout is given in Figure 32. The support structure (according to ELMA spec.: http://www.elma.ch/Pdf/English/EP/Frontpanels.pdf) for the front panel is shown in the left figure in yellow. While the inner two supports can be removed completely, the outer ones are needed and risk to get in conflict with the PB cards. It will have to be checked whether the PBs will therefore have to be moved more to the center of the MB – and therefore the LEDs of the MB from the center to the top or to the bottom. Similarly, the handles (marked in white on the right drawing) are very close to the LVDS connectors but the proper handling of the connectors is still guaranteed.

¹¹ As the PB cards need less electrical layers, it is not yet clear, whether these will be somewhat thinner.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	43/53



Figure 32: Front view of a MB with four PB cards and suggestion for the according front panel.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
44/53	28.12.2000, 15:53

5 Using the Cards for an L1 Trigger

5.1 Introduction: Description of the Problem

The L2 trigger described in the previous chapters is active only about 1000 times per second (which means for less then 3% percent of the time during data taking). An L1Keep signal is needed in order to start the L2 processing of the data. This L1Keep is generated by the central trigger, based on L1 trigger elements delivered by various systems. It is not absolutely necessary, but from a physicists point of view highly desirable, to derive L1 trigger elements from the track segments reconstructed in the FEMs.

In order to avoid dead time, all detector data is stored in ring buffers during L1 operation. All pipelines are designed to keep the data of the last 2.4 us, which reduces the time left to generate a trigger element down to approximately 2.2 us. The logic for the computation of a trigger element needs to be pipelined, as an L1 trigger decision is to be taken every 96 ns.

The trigger algorithm proposed for the FTT L1 is very similar to the L2 linker. Track segments from all FEMs need to be collected on one card, where a search for coincidences between the different layers is performed. A significantly coarser binning will be chosen for L1, such that the coincidences can be implemented as 'hard wired' AND functions in the FPGA. Based on the proposed solution, it was found that slightly less then 480 kappa-phi bins are feasible and deliver satisfactory results.¹²

A simple evaluation of the matrix of coincidences will allow to generate the trigger elements (depending on multiplicity, spatial distribution, etc.). The trigger elements have to be delivered to the central trigger with a fixed phase relative to the Hera clock, 22 clock cycles after the collision of the particles took place.

5.2 Solution: System Overview

In order to keep the latency of the data transfer as low as possible, a 'one step' transmission – every FEM is directly connected to the L1 linker – would be desirable. This means that a board with 30 independent high speed inputs would have to be built! While a solution using similar LVDS cables as used for the L2 FTT is forbidden by the size of the connectors, scenarios based on a card with 30 optical inputs have been put aside because of the development effort.

It has therefore been decided, to modify the L2 FTT cards such that they can be used for the L1 trigger as well, whereas – exactly as for L2 – two steps are performed to collect all data in order not to have to go to

¹² This number corresponds to ten triple words per layer and event – in total it's 19.97 Gbit/s (what a pity we didn't reach the 20Gbit/s threshold \circledast).

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	45/53

more than eight connectors on every board. The same merger cards as for the L2 trigger are used, receiving data from the FEMs through the same LVDS link, operated most probably at four times Hera clock.¹³ On the merger cards, these 6 times 80 bits are collected and combined into the ten tripple words which in turn have to be sent to the L1 linker / L1 decision unit. As in addition to the bare data some control information (data valid / synchronisation 'start/end of event/data block') has to be transmitted, at least one (can't we get any more?) bit cannot be used for data, reducing the size of the linking histogram to $10^{*}(48-n)$ bins.

¹³ Every FEM delivers one sixth of the 480 bits needed on the linker, i.e. 80 bits every 96 ns. Transmitting 4 triple words on the LVDS link during that time, there is plenty of 'space' left for control information.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
46/53	28.12.2000, 15:53



Figure 33: Overall view of the FTT data flow, including L1 and L2.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	47/53

5.3 Changes to the Hardware Design

A few changes to the original HW design are necessary in order to make the use of the L2 cards possible on $L1.^{14}$ The additional effort for SCS will be subject of an additional contract which is to be signed in December 2000.

5.3.1 Speed

While the L2 trigger feasibility study required no more then 50 MHz on the boards and on the LVDS link, the data needed in the L1 linker FPGA can only be received if we go to 104/105 MHz.¹⁵

These increased requirements for data transmission concern the LVDS link, the LvdsCtrl FPGA, the connection between LvdsCtrl and DataCtrl (Piggyback connector) and the DataCtrl FPGA.

5.3.2 Synchronous Clock

The L1 trigger elements have to be delivered by the L1 FTT with a fixed phase relative to Hera Clock, after a very well defined time. Two scenarios have been proposed:

- Synchronous: 'traditional' solution¹⁶. The 10.4 MHz clock which has a fixed phase relation to the particle bunches colliding in the detector is used to drive the complete trigger logic. Using a 10x PLL, 104 MHz can be generated.
 - Advantages:
 - As the number of triple words to be transmitted every 96ns is given, only little additional synchronisation is necessary once the system has started.
 - As the delay for the whole trigger decision is a fixed time span, no additional mechanism is necessary in order to force the trigger elements to be sent with a fixed phase relative to Hera Clock.
 - Disadvantages:
 - The 10.4 MHz input signal is multiplied by a factor of 70 (including the PLL in the LVDS driver). This sets very hard requirements on the stability of the clock signal, which are difficult to measure in advance and might jeopardise the reliability of the LVDS link.

¹⁶ So far many different L1 triggers have been in operation, all of them running absolutely synchronous to Hera Clock. As all electronics work with 'copies' of this one single clock, data transmission at 10.4 MHz is usually done completely synchronous, adjusting a clock delay at each receiving system individually, such that setup and hold times are met.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
48/53	28.12.2000, 15:53

¹⁴ These modifications of the concept are already included in the previous chapters, the following explanations will show the motivation for these features.

¹⁵ Consider that 4*48=192 IO pins of the DataCtrl FPGA constantly receive data at 104MHz!

- Local oscillators have to be placed on the boards nevertheless in order to allow testing and commissioning without additional equipment.
- Asynchronous FIFOs will be necessary in order to receive the data on the LvdsCtrl FPGA nevertheless (adjusting delays of the LVDS link such that data can be fed from the receiver into the LvdsCtrl synchronously is out of discussion).
- Asynchronous: Every card (mergers, L1 linker) uses its local 105 MHz clock oscillator¹⁷. For every 96 ns a 'block' of valid triple words is received by the linker and collected into a 'ten triple word block' © which is sent to the L1 linker at 105 MHz. As after some time the merger will 'run out of data', invalid (empty) data words will have to be sent to the L1 linker from time to time.¹⁸ No matter whether the oscillator on the linker is slightly faster or slower than the ones on the merger cards, it still receives its 10*(48-n) data bits every 96 ns and can compute its trigger elements.

As the trigger elements have to be delivered with the correct phase relative to Hera Clock, they may have to be delayed for a few clock cycles until the locally available Hera Clock – which is used as a 'slow enable signal' – indicates when it is time to send them. *Synchronicity is achieved by running slightly too fast and waiting on the finishing line until the specified time has expired.*

- Advantages:
 - Running with a 'clean' local oscillator reduces the risk of problems due to clock jitter significantly.
- Disadvantages:
 - Synchronisation words ('invalid data') have to be introduced during operation. As the synchronisation mechanism at start-up is considered to be much more important and critical, this does not seem to be a major argument against this solution.
 - The FEMs and other boards with analogue signals are situated next to the Mergers and the Linkers. If noise is induced on these signals by our asynchronous system, it will be much more harmful than the effects of a synchronous system.

As both scenarios bear certain risks and advantages, both solutions are foreseen in the design, with the possibility to choose the preferred solution by jumpers.

¹⁸ If necessary, it can easily be guaranteed, that invalid words are inserted between different blocks and not 'inside' of a block.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	49/53

¹⁷ The FEMs have to run synchronously to a multiple (4x, 8x) of Hera Clock due to their analog input.

5.4 Operation

Using the same merger cards for L1 and L2 means, that very different data will have to be handled by these previously 'dumb' cards. During L1 operation, incoming data words have to be merged and sent continuously to the L1 linker. During L2 operation the incoming data may have to be stored in FIFOs and is the to be multiplexed (without touching the triple words at all) onto the link to the L2 linker. As it is necessary to use more or less the full bandwidth of the LVDS link for payload data, it cannot be indicated on the data, whether it is for L1 or for L2. Switching from one modus to the other has therefore to be done by additional control signals. Two scenarios are foreseen in the hardware:

- The L1Keep signal which is distributed by the central trigger is available on the main boards (received from the backplane via LVDS) exactly as on the FEMs. Checking this signal is a very safe way to switch from one modus to the other. It will however be hard (if not impossible) to calculate all delays for the L1Keep signals and the LVDS data such, that the merger card can decide for every single triple word, whether it has been sent by the FEM in L1 or in L2 operation modus.¹⁹
- Every FEM sends control words such as 'end of L1 operation' when switching from one modus to the other. This scenario has the advantage, that it will be trivial, to treat every triple word correctly. A risk arises however in case of bit errors on the LVDS link. If a control word is missed or if 'normal data' is interpreted as control word, the system will completely fall over. Most probably it will have to ask for the whole data taking to be stopped in this case. The risk of this kind of accident can be reduced by the following measures:
 - Control words include parity bits and/or magic numbers in order to be validated by the receiving device which can reject almost any false control words this way.
 - Every control word is sent several times. If it is lost once due to a bit error at the receiver, its copies still should be received correctly.

Additional measurements of the quality of the used links can be achieved in the following ways:

- When the links are not needed for data (e.g. after all L2 track segments have been delivered) control words including parity information are sent and checked by every receiver (Bookkeeping to be read via VME).
- Whenever L1 operation starts, for every of the 48 bits a parity counter is stared, running until L1 operation is terminated. After the 'end of L1 operation' control word itself is sent over the link and compared to the expected value calculated by the receiver.²⁰

²⁰ In contrast to the block diagram shown in section 4.2.1, this would require the presence of a control unit on each LvdsCtrl FPGA, which would send the according messages to the VME FIFO.

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
50/53	28.12.2000, 15:53

¹⁹ This effect seems to be much less of a problem, if an 'arbitrary' number of invalid triple words is inserted whenever the FEM change from one modus to the other. (\rightarrow How much time is left there?)

5.5 Timing

As the trigger algorithm as well as the track finding in the FEM are outside the SCS responsibility we could estimate the delay to be expected for the transmission.

Device	Step	Delay
FEM	trans LVDS ^{α}	45 ns
Input-PB Merger	rec LVDS [•]	80 ns
	FIFO ^β	85 ns
	$LvdsCtrl-DataCtrl^{\gamma}$	20 ns
Merger	DataCtrl internal	20 ns
	DataCtrl-LvdsCtrl ^y	20 ns
Output-PB Merger	LvdsCtrl internal	50 ns
	trans LVDS'	20 ns
Input-PB Linker	rec LVDS [•]	35 ns
	FIFO'	75 ns
	LvdsCtrl-DataCtrl	20 ns
Linker	DataCtrl internal	20 ns
TOTAL		490 ns

^α @ 40 MHz

 $^{\gamma}$ @ 100 MHz

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	51/53

 $^{^\}beta$ write-side @ 40 MHz, read-side @ 100 MHz

VISA, 18.12.00:

For IPP:

For SCS:

(Dr. A. Schöning)

(Dr. D. Müller)

P:\FTT\Doku\SCS\Design Description\DesignDesc33.doc	
28.12.2000, 15:53	53/53