L2βeta Design Report
A replacement and upgrade path for DØ Level 2 processor cards

Bob Hirosky
University of Virginia
Drew Baden
University of Maryland
Philippe Cros, Bernard Lavigne, Pierre Petroff
Laboratoire de l'Accélérateur Linéaire

version 1.4

April 26, 2001

Contents

1 Introduction

This document outlines an alternate model for the construction of the Level 2 processor cards. The new cards would be fully compatible with the present system. Additionally, this design provides a path to upgrade Level 2 with increased processing power and possibly greater data throughput. The new design offers a minimum of engineering effort, and, through use of standard industrial processor cards and modern FPGAs, will provide a more flexible, upgradable, and more easily commissioned system.

2 Motivation

There are a number reasons motivating this new design of the Level 2 processor cards:

1. to develop a source of spare devices for long term maintenance of the trigger;

2. to provide a fall back position in case a full complement of the current alpha boards are not deliverable;

3. to provide processing performance improvements during Run II for nodes that may suffer in performance at peak luminosities;

4. and to develop a platform campatible with future enhancements for the higher luminosity running of Run IIb.

The original approach to the Level 2 Alpha processors built on the model of earlier custom designed fastbus-based single board computers used in the CDF experiment [1]. The design relies on a complete layout of a single board computer[1] that includes an industry standard VME interface chip [2] and three custom PCI devices for access to the Magic Bus and output to trigger control frameworks. These PCI devices are implemented via various FPGAs, FIFOs, buffer/drivers, a CPLD, and a variety of support logic. The functionality of the overall design is suited to the initial needs of the experiment. However, the complexity of its construction (large component count and integration of all CPU and I/O functions in a custom-manufactured board) lends difficulty to debugging/commissioning and maintaining the boards. One of the difficulties involved with attempting a custom layout of the processor itself is evident in the observed unreliable functioning of the Level 3 (off CPU) cache [3] on the Alpha boards. Limited prototyping cycles did not permit a full verification and optimization of this section of the processor boards. As a result DØ has planned to run the boards with Level 3 cache disabled. Manufacturing difficulties with the first production runs of the alphas are well known to the experiment and the manpower cost in the debugging effort has impeded software and system integration work on the trigger. Such costs will are to be carefully weighed in any upgrade or replacement scenario.

The lifetime of the present processor boards is clearly limited by the growing obsolescence of components. In particular, a number of components crucial to the function of the PC164 motherboard design have already fallen well into obsolescence. These include much of the "core logic chip set" (composed of a PCI controller and data switches for memory access), the ISA bus interface, and even the particular RAM chips used. These devices essentially compose the body of the on-board computer. Supplies of these components are scarce and a long term source of spare components to support the original trigger processors is unlikely to be found. Any modernization or upgrade path using a monolithic processor card model would require an engineering effort equally large as the original, if a more modern,

[1] The computer is an implementation of the COMPAQ PC164 motherboard [3]

but similar approach were pursued. Any such implementation would obviously suffer the same fate of rapid obsolescence[2] as all modern computing devices. Furthermore, the prototyping and verification of the design would prove difficult (especially when considering faster clock speeds in more modern CPUs), expensive (in terms of engineering and prototyping), and taxing of manpower resources.

While the 500 MHz Alpha CPUs were superior in performance at the genesis of the project they no longer dominate the alternatives and are now surpassed by the latest processors. Our current processor boards as implemented cannot benefit from this march of technology.

At highest luminosities, the current system may be limited by CPU performance as more channels cross thresholds for clustering or fitting. Anticipating little room to improve the CPU efficiency of our trigger algorithms, there are several possibilities for dealing with data requiring with greater CPU performance in the present system:

- Increase seed thresholds. For example, in the Calorimeter Electron and Jet Preprocessors [4], seed thresholds are fixed in hardware at Level 1, but higher software thresholds could be imposed in the processor code. The effect of this would be to reduce efficiency, especially for lower P_T objects.

- Increase the Level 1 trigger threshold or prescale. Again efficiencies are significantly lowered.

- Remove functionality from the algorithm. This further taxes the Level 3 trigger by decreasing background rejection at Level 2. The resulting functionality loss may combine with input bandwidth limitations to Level 3 to force raising thresholds at Level 1 or Level 2, also lowering efficiency.

- Add more Worker CPUs. It is possible to add additional workers to the Level 2 crates. In the case of certain detector preprocessors, the algorithm may be parallelized in a fairly straightforward manner by splitting the detector into distinct regions. This is a viable solution in certain cases and some contingency for extra parallelism has been built into the number of processor boards manufactured. However, this approach is ultimately limited as an upgrade path for Level 2,

[2] In fact, given the rapid evolution of computer electronics, it would be difficult to complete a prototyping to production cycle without finding that various components have gone obsolete prior to final assembly.

because of the finite numbers of processor boards available and also due to space limitations in the Level 2 crates.

Not all processing can benefit in direct proportion to the amount of parallelization in the hardware. This is the case where an algorithm must correlate information across an event. The Level 2 global worker may be duplicated so that each worker processes separate events or a single event's processing may be partially parallelized between workers. The advantages in the former case are limited by the need to report trigger decisions in the order in which events arrive ("worst of 'n' performance," see Fig. 1) and in the latter case by Amdahl's law [5]. In such cases direct gains in single CPU processing power are the most effective means to improve performance[3].

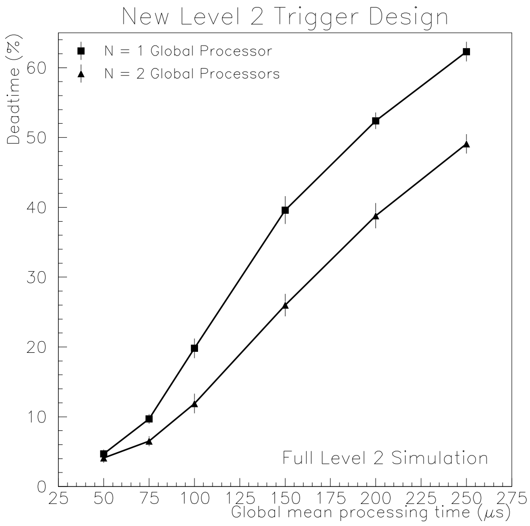

Figure 1: Timing results for a scenario with two Global Level 2 workers handling alternating events compared to a single worker system (from Level 2 Global TDR [5]). Since this is an $N = 2$ serializing farm, a full factor of 2 gain is not attainable.

[3]Using another example taken from the Calorimeter Preprocessor, in the case of Level 2 jet clustering, greater CPU resources would allow for running a seedless (movable window) algorithm within the Level 2 time budget. This would offer the benefit of higher efficiency for low E_T jets even at lower luminosities and is a desirable feature for multi-jet triggers.

3 The L2βeta Strategy

In order to buttress the long term performance and reliability of the DØ Level 2 trigger, we propose a processor system compatible with future performance upgrades and allowing for long term replacement of system components. Rather than attempt a redesign of the entire processor board, we instead prefer to implement a system using a more modular approach, taking full advantage of industry-standard components and upgrade paths. We follow the same conceptual design for the function of the board in our Level 2 system (moreover we require both hardware and software compatibility between both Alphas and βetas). However, this implementation replaces the CPU and assorted computer peripheral functions with an commercially produced single board computer (SBC). This SBC will reside on a 6U VME card and will also contain an integrated Universe II interface to the VME bus. Such cards are readily available "off the shelf" from numerous vendors (SBS-BIT3, Raytheon, VMIC, Altatech, Concurrent Technology, etc., ...). The remaining functionality of the board will be implemented primarily in a single FPGA mounted on a 6U to 9U VME adapter card as shown in Fig. 2. The adapter card will contain all DØ-specific hardware for Magic

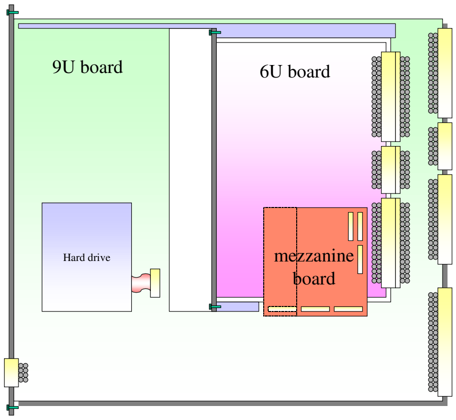

Figure 2: Physical model for the L2βeta processor card.

Bus and trigger framework connections. The SBC, in the adapter, will sit close to the VME backplane, but J1 and J2 signals may be picked off or assumed by the adapter card before entering the SBC. This is necessary in the case of our user-defined J2 signals. A bridge card connects the SBC to the adapter via a standard PCI Mezzanine Connector (PMC) on the SBC card. A short (flex)cable will provide the electrical connection between the cards. The bridge contains a commercial PCI interface, decoupling the PCI bus from the 9U card and providing a simplified add-on bus protocal. This implementation offers several clear advantages:

- The modular design, incorporating existing CPU cards, greatly reduces the engineering, debugging, and prototyping needed.
- The modular design provides a clear path for CPU performance upgrades by simple swapping of SBC cards.
- Compatibility allows for a phased-in replacement of the Alphas.
- A single slot design frees space in the trigger crates and allows greater flexibility for future SBC selections.
- Higher or lower performance CPUs may be selected. For example, lower performance, less expensive CPUs may be used in Administrator-type processor cards, while more powerful CPUs can be used where physics algorithms are run.
- The use of a much smaller number of components to implement the DØ-specific functions of the board greatly reduces the parts count, increasing reliability and ease of hardware debugging.

The last point above is of particular importance. The current implementation of the processor boards is composed of over 170 individual ICs mounted on a board composed of 10 printed circuit layers with approximately 9000 through-hole connections (vias) [7]. By contrast the custom-built components of the above implementation can be constructed with approximately 30 ICs (most of which are simple buffers or logic converters) on a vastly simplified circuit board layout. This is a direct result of using a manufactured CPU and VME interface module and greater integration of functions into a more modern FPGA [4].

[4]Note: The only other cards in the Level 2 system that communicate via the MBus are the Magic Bus Transceiver (data broadcast) [8] cards. These cards already implement their primary MBus functions in a single FPGA. The addition of an interface with added programmability to the processor cards would also allow for future flexibility in the utilization of this bus.

4 Hardware Components

The L2βetas will be composed of three components: a commercial single board computer with integrated VME interface, a custom 6U-to-9U adapter card, and custom PMC card to interface the two. Each component is described in the following sections.

4.1 L2βeta SBC

We recommend using the VMIC Model 7740 6U SBC [9] for the processor and VME interface in the L2βetas (Fig. 3). This board includes an 850 MHz Pentium and an integraded UII VME interface. It is a single slot board hosting a (5.0 V) PMC [10] connector for expansion hardware. The 7740 remains a single slot device with a standard PMC card attached. Competing manufacturers with very similar products include SBS Technologies and Concurrent Technologies. The VMIC board has two advantages over SBS, namely it is more mature in its production cycle and it can accomodate a PMC card while staying within a single slot. An advantage of VMIC over both of these alternatives is that an improved version (the 7750) will be available before we go into production. The 7750 will offer a marginally faster CPU (933 MHz), but it will offer a significantly faster memory bus (133 MHz as opposed to 100 MHz).

The 7740 supports a hard drive, CDROM, floppy, etc. through an IDE controller accessed via user-defined J2 pins. Our 9U adapter will have to intercept these signals and route them to a hard drive mounted on the card to prevent conflicts with J2 lines in use by the trigger.

The 7740 is supported under RedHat Linux 6.2. And the KAI C++ compiler is readily available for this platform and OS. After inspecting PowerPC and other options in the VME SBC market, we have found that the software tools we require to implement the system in a straightforward way are not readily available for these choices. Furthermore, we expect the Pentuim flavor of Linux to more closely parallel our current Alpha version, thus reducing efforts in completing our necessary operating system modifications.

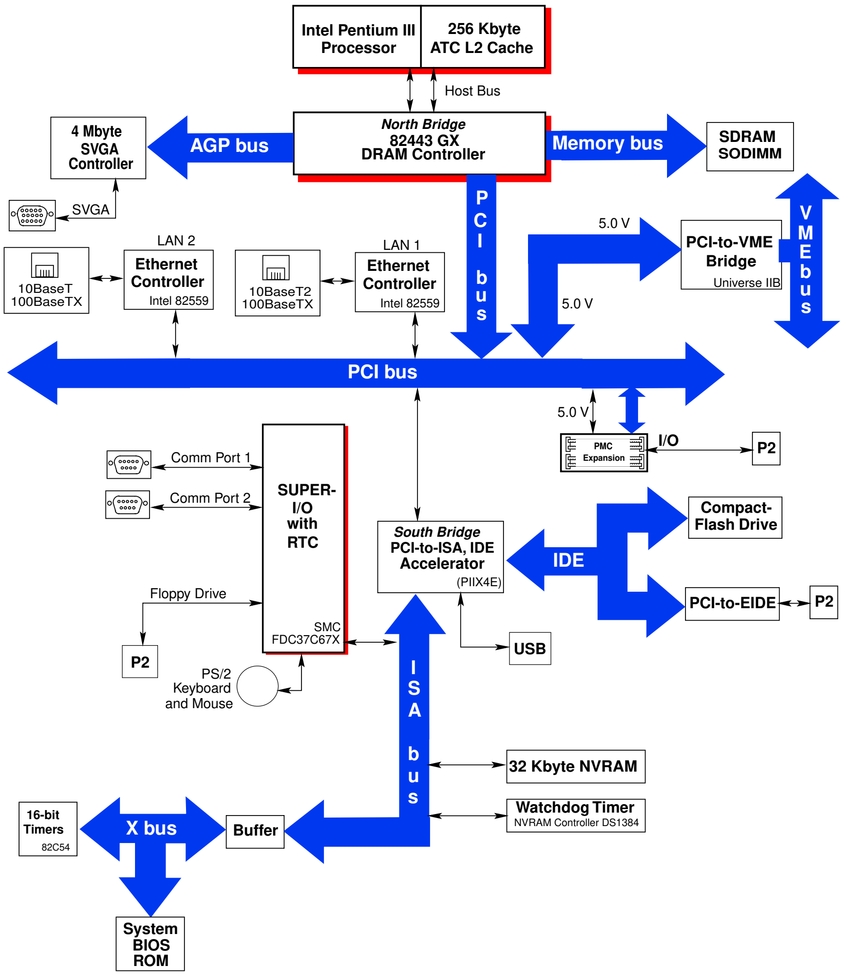

Figure 3: VMIC 7740 SBC block diagram (from VMIC manual).

CPU Type	Specint95	Specfp95
Alpha 500 MHz	~ 15	~ 21
Pentium III 800 MHz	~ 38	~ 29
Pentium III 850 MHz	~ 41	~ 35
Pentium III 933 MHz	~ 45	~ 39
Pentium III 1000 MHz	~ 48	~ 41

Table 1: Performance comparisons [11] for several modern CPUs available on mass produced VME single board computers. The current Level 2 Alpha CPU is shown at the top of this table. The integer performance of the CPUs is of primary importance for most of the operations in DØ's trigger.

Table 1 compares the performance of various CPUs available today on VME SBCs to the Alpha PC164 benchmarks. The integer performance of the CPUs is of primary importance for most of the operations in DØ's trigger. Each modern CPU compares quite favorably in this category. (The 1 GHz PIII boards were excluded from consideration because of their form factors. These boards are two slots wide at a minimum and require an additional slot to add PMC cards.)

4.2 L2βeta 9U Adapter

The 9U card will both adapt the 6U SBC card to the 9U crate form factor and provide hardware for all custom I/O required of the processor cards. This includes all Magic Bus I/O, an interface to user-defined J2 lines, and outputs to trigger scalers.

Details of the 9U adapter card are shown in Fig 4. All functions on this card will be implemented in a single FPGA plus assorted logic converters and drivers. The FPGA of choice is the Xilinx XCV405E [12]. This device is particularly suited to our application, because of its large amount of available Block RAM. 70 KB of RAM (in addition to > 10 K logic cells) is available to implement data FIFOs and Mapper (address translation counter) tables, thus greatly reducing the complexity of the 9U PCB. The replacment of various Alpha hardware components by the Xilinx 504E is further discussed in section 5.3.1.

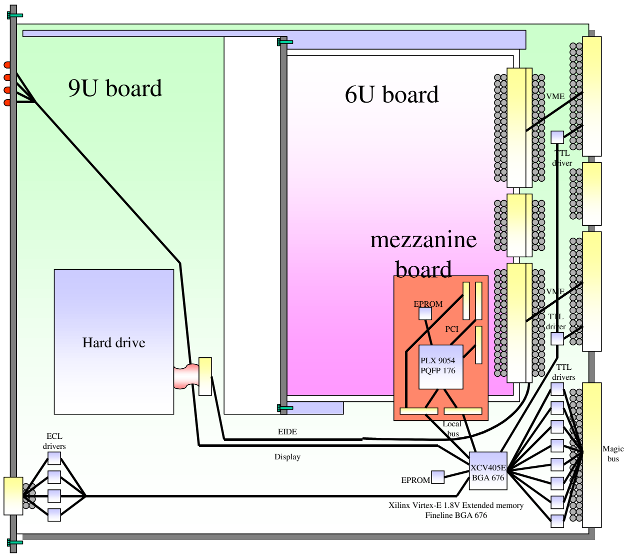

Figure 4: 9U adapter card with major electrical connections shown.

4.3 L2βeta PMC / PCI Interface

The third component to the L2βeta processors is a bridge card connecting the custom I/O on the 9U adapter to the PCI bus on the SBC. This is accomplished (as shown in Figs. 4 and 5) by use of a PMC card designed for this purpose. The PMC card will contain a commercial PCI interface (PLX 9054 [13]). This device provides a 32-bit interface fully compliant with PCI V2.2 specifications on one end and a simplified (protocal) add-on bus is provided on the other end. We have opted to use a hardware interface rather than an FPGA core to handle PCI in order to simplify our firmware project. It will also serve to buffers signals routed to/from the PCI bus. This device will free the FPGA from any direct connection with the PCI protocal and with our Alpha experience as a guide, we expect this to be a significant advantage in developing our firmware. This card will attach to the SBC at its PMC connectors. The vertical location of the connectors may vary somewhat by board manufacturer and model (or possibly by production run of an individual model). However, the transverse orientation is fixed relative to the SBC front panel by the PMC standard. Because only the verticle degree of freedom is required to support future SBCs, the Orsay group has proposed that this card be constructed with an integrated flex circuit connection to the 9U card, thus reducing physical connectors and possible impediance mismatches in the system.

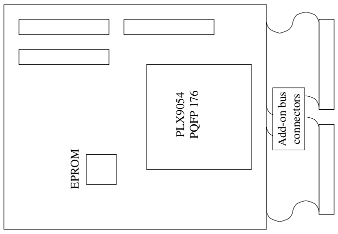

Figure 5: PMC card shown with a flex circuit connection.

5 System Requirements

This section summarizes the performance features and specifications of the L2Alpha processors and describes how the L2βetas will satisify these requirements.

5.1 Processing Power

At the heart of the L2Alphas is a 500 MHz COMPAQ 21164 CPU supported by a PC164-style motherboard. The 21164 provides:

- 2 integer and 2 floating pipes (64-bits wide)
- 96 kB on-chip L2 cache (enough for event data + 12 K (8 byte) instructions of non-looping code execution (These characteristics **seem** to allow operation of the system with the (4 MB) L3 cache disabled.)
- **execution of proposed L2 algoritms in under 50 μs/event**
  Performace benchmarks: Specint95 15, Specfp95 21

As shown in Table 1 the 850 MHz PIII compares very well to the ALPHA 21164 in terms of performance. It should also be noted that the PIII contains a larger on-die cache (256 KB versus 96 KB, this Level 2 cache runs at the clock rate of the CPU). The Pentium instruction set is more complex than the ALPHA set. Instruction sizes can range up to a maximum of about 16 bytes in size [6], however from reference [6], typical integer and branching instructions should be on the order of 8 bytes or less. Conservatively, with a 4 KB event in its cache, the PIII should be able to hold approximately 32K instructions.

The PIII SBCs typically do not have L3 cache, however DØ already expects to run with the L3 cache disabled in the Alphas and it should be noted that the new PIII boads offer significantly faster RAM access than the alphas. The local RAM bus on the PIII's run at 100 to 133 MHz at 64 bits width. This speed is comparable to the PC164 L3 cache (although at half the data width).

5.2 Linux/RT Unix Operating System

RedHat Linux 6.2 is already in use on the VMIC card and other PIII-based SBC models. The needs of the experiment require two modifications to standard Linux to provide real time behaviour and to allow direct mapping of physical memory within user programs. The latter problem is expected to be relatively straightforward for two reasons: the solution employed in the Alphas is a direct adaptation from code taken from Intel Linux and the UII drivers now available for Linux must already address this specific need. In order to provide real time behavior, a special interrupt handler must be implemented to sit below the Linux Kernal and to service interrupts, such as new event interupts, in a timely manner. We do not expect new difficulties here with respect to the Alphas.

Intel Linux offers the added advantage of the KAI compiler. The L2βetas will in fact be more easily integrated into the DØ software environment than the Alphas. Intel Linux is fully supported at DØ and KAI is the compiler choice.

5.3 VME, TSI, MBus Devices

The Alphas contain four seperate PCI devices to accomadate our special I/O needs: the UII VME interface; the TSI which handles a variety of MBus and J2 signals and data output to the trigger framework through a front panel ECL connection; and two Magic Bus data interfaces: A PIO FPGA and a CPLD responsible for initiating DMA transfers from FIFOs receiving MBus broadcasts.

The UII interface hardware will be identicle on the L2βetas. The remaining device functionality will be contained in the Xilinx FPGA and the PLX 9054. The functionality of the devices will be simplified from their Alpha counterparts, because they will not require firmware blocks to impliment

PCI compliant devices. Situating the functions in a single device, using a single programming tool, will reduce support and development overhead.

5.3.1 Hardware Requirements

The Alphas contain 128 bits $\times$ 4 K of data FIFOs plus an address FIFO of the same length for MBus broadcast addresses. Broadcast addresses are restricted to the 10-bit MBus address range, [0:1023]. The Mapper contains address translation (MBus to PCI) information for each broadcast address. Its registers are incremented as the FIFO's are drained and at any given time provides the PCI target location for data at the head of the FIFO. The Mapper registers may be read to determine how many words were written to each broadcast address. The nominal data storage requirements for these devices is:

$$\text{Data FIFO: } 128\,\text{bits} \times 4\,\text{K} = 64\,\text{KB}$$
$$\text{Address FIFO: } 10\,\text{bits} \times 4\,\text{K} = 5\,\text{KB}$$
$$\text{Mapper: } 32\,\text{bits} \times 1\,\text{K} = 4\,\text{KB}$$

$$\text{Total: } 73\,\text{KB}.$$

This results in a 3 KB shortage of memory compared to the 70 KB available in the Xilinx XCV504E. Xilinx produces their model XCV812E with twice the block RAM in the same package at the increased price of $\sim$ \$1100 compared to $\sim$ \$500 for the XCV405E. However, there are a number of other solutions to the shortfall.

Rather than dedicating a large number of logic elements to fill the 3 KB shortfall, it is possible to FIFO only address changes. In this case we would dedicate 1 bit $\times$ 4 K to create an address change flag FIFO (or 129^{th} data bit). The remaining 7 bits $\times$ 2 K of block RAM could be used to construct a reduced address FIFO of 10 bits $\times$ 1 K.

Alternate solutions[5] include reducing the depth of the data FIFO capacity to 3.75 KB. This would require the addition of an explicit 'hold off' mechanism in the DMA protocal to prevent FIFO overflows. The DØ MBT cards are compatible with such a mechanism. This is accomplished by use of the DDONE handshake line on the MBus backplane.

Finally, given the limits on numbers of broadcast sources and strict ordering of source broadcasts within the DØ crates, is plausible to use a set of registers instead of an address FIFO. The implications of such an approach on firmware complexity and higher level software has yet to be explored.

5.3.2 Functional Requirements

Various requirements must be met by the MBus devices:

- The 9U board is required to send data to memory in the SBC at a rate of around 80-100 MB/s.

  There are three possible bottlenecks to explore for this requirement: the FPGA/FIFO's, the PCI interface, the PCI controller. All of these devices are capible of delivering data at the full 32-bit PCI bus bandwith of 132 MB/s. We have learned of tests conducted at CERN [14] that have surpassed 100 MB/s of data transfers using the PLX 9080 (a precurser to the 9054) for large size transactions and feel confident that we can match this performance. The PCI controller is fully capible of delivering data throughput at the PCI limit and the bandwidth of the memory bus far exceeds PCI. (Note: The Alpha PCI controller has only delivered $\sim 80 - 120$ MB/s of DMA throughput despite its 64-bit width, because its buffers frequently fill while sending data to memory.) Clocking the Xilinx at sufficient speeds to deliver this bandwith poses no difficulty.

  Following are some conservative effective bandwidth estimates. The relevant numbers for DMA performace in DØ can be thought of as

[5] All solutions consider the allowable configurations of block RAM elements in the XCV405E. Block RAM elements may be configured in the following width and depth combinations [12]:

Width	Depth
1	4096
2	2048
4	1024
8	512
16	256

follows: The maximum size for a data source is about 320 bytes (Data sources transmit sequentialy to to build an event of up to 4 KB in size). Assuming about 300 ns of startup time for a DMA transaction due to Add-on bus arbitration, it would take 2.9μs to put a source's data into memory (an effective bandwidth of 110 MB/s). Additional overhead is incurred if a PIO transaction interrupts the DMA. The timescale for such interruptions is on the order of the event processing time. Assuming that the PIII's are twice as fast the Alphas, we might expect an average processing time (very roughly) on the order of 20μs/event. With three worker cards in a crate, a message might be sent to the Administrator roughly every 6μs. Therefore it is unlikely that the DMA from an individual source would be interrupted more than once before it completes (the rate would be lower for Administrator to Worker messages[6]). In the worst case scenario, rearbitrating the add-on and PCI buses for each 32-bit word in a 128-bit interrupting MBus transaction, we would add about 2.3μs to the DMA time (effectively reducing the bandwidth to 62 MB/s). Doing a burst transaction on the Add-on bus could decrease the additional overhead to about 0.8μs (effective bandwidth of 86 MB/s). These numbers are particularly conservative since the time spent in the PIO transaction was charges against the DMA. The maximum data rate (4 KB events at 10 KHz) into Level 2 would require 41 MB/s. This is more than twice the present data rates.

- Programmed I/O precedence over DMA. It must be possible for the DMA to be preempted under two conditions:

  1. FIFO's are being read out and the receiving board needs to send a PIO transaction to the Magic Bus. This would be accomplished by programing the PCI latency times in the PLX 9054. In accordance with PCI protocal, a bus master must reliqush bus ownership after its PCI latency times expires, if its GRANT has been removed by the PCI controller. This would be the case if a PIO transaction were pending from the CPU. The latency time is fully adjustable in by setting the appropriate register in the PLX 9054. This is a important matter, because latency in reporting a finished event is charged directly against the time budget for a Worker Processor. We would set the latency timer

[6]Interrupts due to VME transactions would happen at a much lower rate. Their PCI budget would be charged against the PCI bus bandwidth remaining after broadcast DMA.

to allow no more than 2-3 μs of delay.

2. The second condition could occur if the FIFOs are being read out and the Administrator Processor initiates a PIO transaction over the Magic bus. In this case the PLX can be caused to halt DMA from the add-on bus side and to re-arbitrate the PCI bus to complete the incoming PIO transaction. After this transaction, the DMA transfer would be reinitiated.

- DMA destinations must be configurable by the CPU for all MBus broadcast addresses. The full functionality of the DMA Mapper as used in the Alphas can be implimented inside the Xilinx, maintaining the required functionality. The PLX fully supports writing data to targets with a large range of local addresses on the add-on bus.
- It must be possible to interrupt the PCI bus in accordance with events on the local bus (FIFOs empty, new event, etc). The PLX explicitly supports generation of interrupts from the add-on bus.
- Magic Bus PIO must be able to support a number of data modes: Master/Write, Master/Read, Slave/Write, Slave/Read. In our design this is a matter for firmware only, bi-directional drivers are provided to send or receive MBus data.
- Fast MBus arbitration ($\sim$15ns). This is consistent with realizable gate delay times in the Xilinx FPGA.

5.4 Mechanical and Electrical

While the L2Alphas require two VME slots, the L2βetas are single slot devices. This will free two to four slots in each Level 2 processor crate for other uses. Figure 6 shows an assembly drawing for the L2βetas. The SBC will sit inline with the 9U adapter card. A cutout in the 9U card will be reinforced with aluminum bars to guide the SBC and to stiffen the adapter. Physically the SBC will sit $\sim$2-3 cm from the VME bus. User J2 signals will be intercepted by the Xilinx FPGA instead of being routed to the SBC. IDE signals provided on User J2 lines by the SBC will be routed to a hard drive mounted on the adapter (Fig 2) (A header for connecting a floppy drive could also be provided.).

Table 2 provides a rough estimate of the power requirements on the L2βeta as compared the L2Alpha. The power and heat dissipation requirements are significantly reduced in the L2βetas.

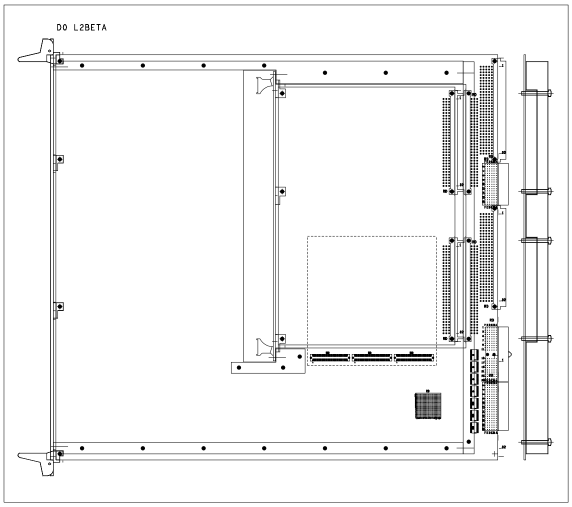

Figure 6: Mechanical assembly of a L2βeta processor.

	Supply Voltage					
Device	12 V	5 V	3.3 V	-5.2 V	-12 V	Total Watts
L2Alpha	2 A	20 A	10 A	1.25 A	1 A	176 W
L2βeta SBC	0.4 A	7 A			.0025 A	36 W
L2βeta 9U						≈10 W
L2βeta PMC						≈1 W
Hard Drive		0.5 A				2.5 W

Table 2: Approximate power consumption per processor component.

6 Software and Firmware

In order to minimize the demand on the experiment's resources to bring the L2βetas to completion it is vital that this project tread as lightly as possible on other trigger groups' projects. We have stated at the onset that the L2βetas are to be both hardware and software compatible with the Alphas. Hardware compatibility will largely be a product of proper firmware design. By software compatibility we are refering to high level software. It should be possible to recompile online code with little or no changes to run on the L2βetas. This compatibility will be enforced by the hardware interface layer, or device driver software.

6.1 Device Driver Software

The coding tasks for this project can be displayed as in Fig 7. The device driver code will require a rewrite of the hardware interface layer to communicate with the PCI bridge. The user interfaces for the device drivers will be unchanged from the present system. This is necessary for source code compatibility.

The basic structure of the device drive code is to start with a generic PCI device and to build specific functionality into classes derived from this device. All hardware interface functions are hidden in this lowest level base class.

The UII driver will require the least amount of change, since this device is replicated on the SBC. In fact, the current device drive code is expected to function "as is" provided the PCI base classes it depends on are replicated for the PIII boards.

The remaining device driver code will have to be completly redesigned, since three former PCI devices will now live inside one FPGA. Different functions of this FPGA will be treated as pseudo-PCI devices to maintain a software interface compatible with the Alphas.

6.2 Firmware

The firmware will be composed in loosely coupled blocks similar to those shown in Fig 7. The most complex blocks will be the add-on bus interface and the MBus I/O block. A more detailed summary of the firmware blocks and FPGA resources required is shown in Fig. 8.

The necessary L2βeta functions can be implimented using 337 I/O pins. This includes "utility pins" allocated for configuration settings and logic analyzer 'spy' channels for assistance in firmware debugging. An additional 66 I/O pins are available for additional functionality we may wish to add at a later time. The inclusion of the MBus data FIFOs in the FPGA has resulted in a great economy of I/O pins required.

6.2.1 Add-on Bus Firmware

In order to minimize development time and to maximize chances of success for the L2βetas, our pilosophy has been to use commerical devices where custom devices can be replaced, to simplify the electrical design of the boards as much as possible, and to push as much of functionality as possible into firmware to reduce hardware prototype cycles.

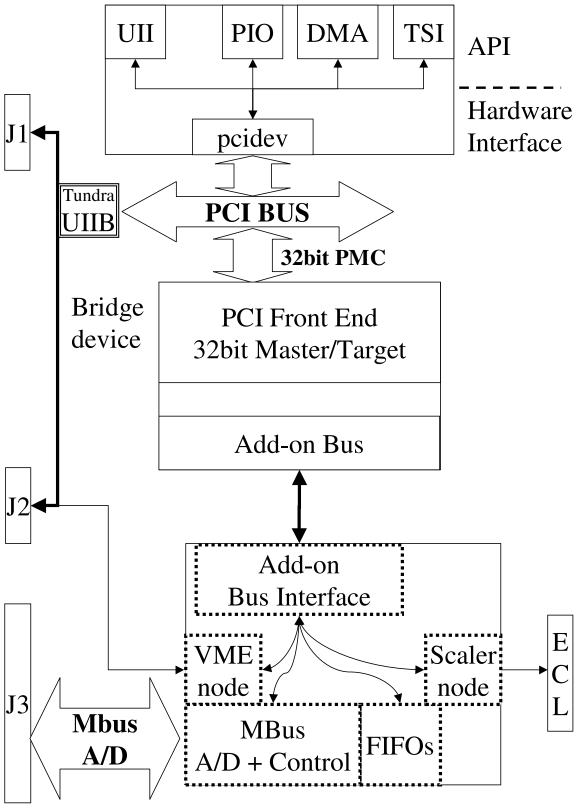

Figure 7: Block diagram of L2βeta software and firmware components.

Altough we have sought to develop a system that is implemented in firmware to as large an extent as possible, we have choosen to use a hardware PCI interface rather than to complicate our firmware project with a PCI implementation. We plan to take full advanage of the PLX's features to simplify the firmware design.

Perhaps the most interesting feature of the PLX9054 is its ability to become a Master on both the PCI and add-on buses and to fully control DMA transactions (Figure 9 gives a brief overview). Thus we may simplify the DMA block of the firmware to a protocal that provides data words at the 9054's request. This removes all need for the Xilinx to directly support PCI timing, because the PLX completely decouples the two buses. Furthermore,

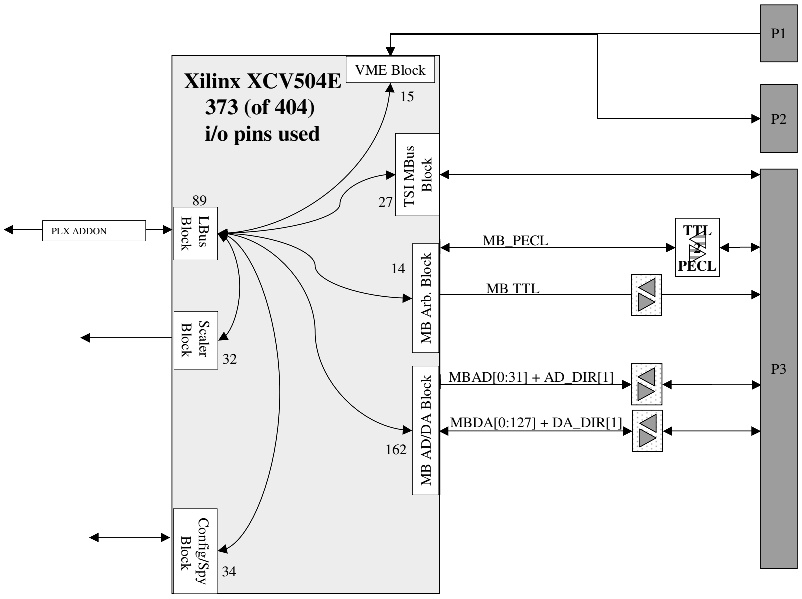

Figure 8: Overview of firmware blocks. I/O pin requirements are shown with each block.

the PLX can run its add-on bus faster than PCI (up to 50 MHz) to ensure that its internal buffers do not run out of data while feeding the PCI bus.

The PLX also offers a local bus protocal with seperate address and data lines. Using this interface all transactions can be thought of as one step responses based on address decodes. This is conceptually easier to deal with than protocals that multiplex address and data lines where addresses and data states must be considered in the protocal.

A possible approach to the Add-on bus firmware is the following:

- Provide add-on bus targets for PIO and Mapper registers, PIO writes (MBus master), PIO reads (MBus Master), and TSI (equivalent) writes/reads. Each of these Target would be specified by different local address.

- Provide add-on bus masters for PIO writes (MBus Slave) and PIO reads (MBus Slave).

- Provide an add-on bus master for interrupt generation. (Note: Pending local bus requests have higher priority than ongoing DMA, therefore the PLX protocal conveniently allows a mechanism for the local bus to halt DMA as needed.)

- Provide an add-on bus Master to set up DMA registers in the PLX and initiate the transfer.

- Provide an Add-on bus target to feed data to the PLX's DMA engine.

It is likely that several (or all) of the Add-on masters would be combined into one module to reduce complexity of the Add-on bus operations.

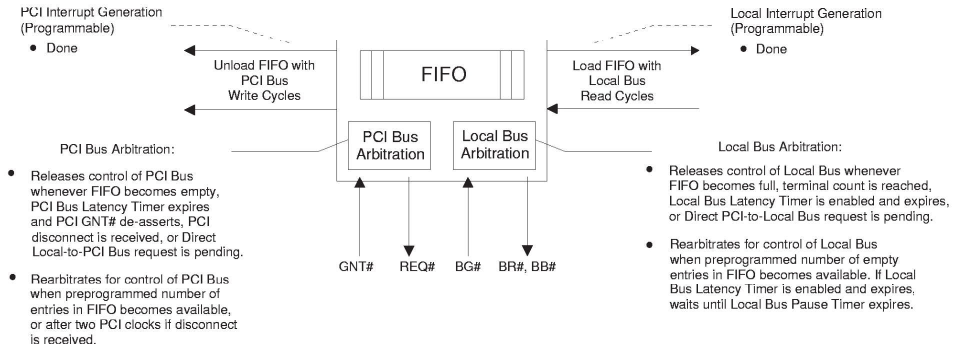

Figure 9: Diagram of PLX 9054 DMA control.

6.2.2 MBus I/O and Arbitration Firmware

The MBus arbitration firmware block will implement the MBus arbitration logic. This essentially amounts to a series of combinatorial logic elements switching at speeds on the order of the (few ns) gate delays of the Xilinx.

The MBus I/O firmware (Fig. 10) is responsible for controling the MBus Data and Address drivers and the (internal) Data FIFOs. It must respond to a requests from the PLX to send or fetch data (PIO Master), it must request that the PLX fetch or receive data (PIO Target), and it must provide data to the PLX for DMA transfers (FIFO readout). The latter function must also provide a halt mechanism (perhaps a local bus abort) to end the DMA when the FIFOs have cleared and to send an interrupt to the PCI bus. A simple mechanism for generating an interrupt would be to write to a properly configured mailbox register in the PLX. The PLX may be configured to send an interrupt upon completing DMA, however, this may be less straightforward, since the DMA begins before the ultimate size of the data is known. An add-on bus master may be used to configure the DMA registers in the 9054 before beginning to transfer data from a source.

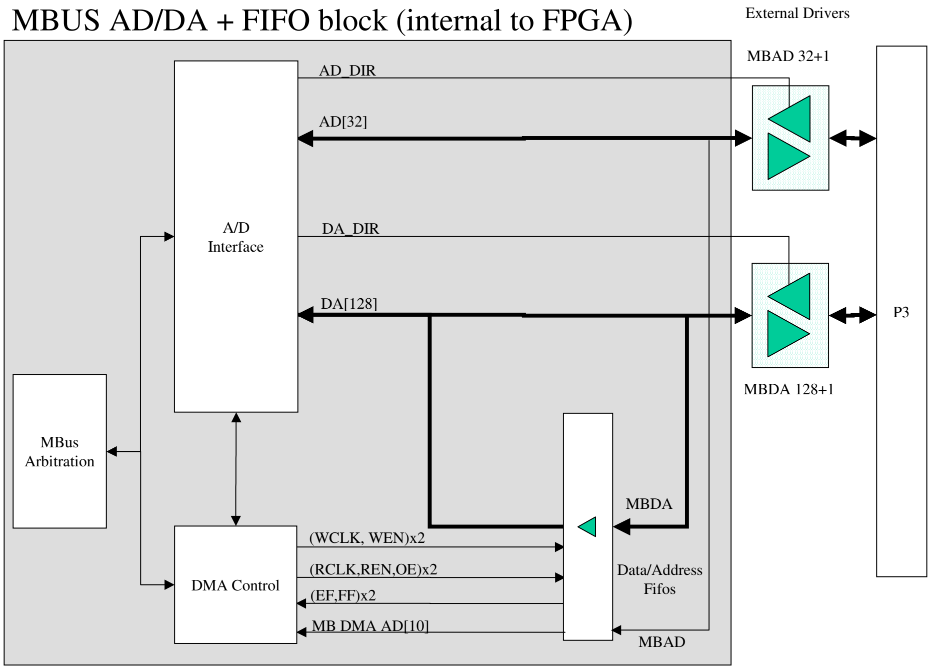

Figure 10: MBus I/O firmware blocks.

6.2.3 VME and Scaler Firmware

The VME and scaler firmware block is NOT required to provide any VME functionality. It will provide the access to User J2 lines and VME GEO lines as does the TSI device in the Alphas. The scalar block will simply drive front panel ECL drivers.

6.2.4 Spy and Configuration Channels

Sixteen channels (I/Os) are reserved to drive a header or additional front panel connector for the purpose of examining signals internal to the FPGA.

Signals would be selected by firmware settings and used for debugging purposes. Other pins are provided to connect status LEDs and a configuration switch to control firmware functions. More I/O pins may be added to this block if desired.

7 Project Collaboration and Orgnization

Institutions in the L2βeta now include the University of Virginia, the University of Maryland, and the Laboratoire de l'Accélérateur Linéaire at Orsay.

Virginia will take responsibility for general project organization and for setup of the SBCs, development tools and device driver coding. The Virgina group will include Bob Hirosky (Project Leader) at approximately 0.75 FTE and postdoc/student support for the project at approximately 0.33 FTE.

Maryland will concentrate on implementing the Alpha MBus functions in the L2βeta firmware. Drew Baden (Firmware Coordinator) will be available at approximately 0.20 FTE throughout the prototyping and commissioning phases. Additional postdoc/student support may add another 0.33 FTE. Engineering support from Maryland will also be available on an hourly rate basis. (Funds are allocated for approximately two months of firmware engineering assistance.)

Orsay will lead the hardware effort and provide and support the add-on bus interface firmware. Two engineers, Philippe Cros (Technical Coordinator) and Bernard Lavigne will spearhead the hardware at approximately 1.5 FTE through the prototyping phase. We expect to maintain 0.50-0.75 FTE engineering support through the commissioning of the system. Two Orsay physicists (Laurent Duflot and Pierre Petroff) can be available to assist with component testing. Orsay also commands a PCB design staff and mechanical group that will support the prototype phase of the project.

Ultimate verification that the prototypes perform all Alpha functions will fall to Virginia and Maryland in coordination with Level 2 software experts and State-side Orsay personel. Orsay has expressed a desire to take a significant share of the responsibility in integrating the production L2βetas into the trigger.

Integration of the system will require close cooperation among the groups. We will plan for video meeting on a weekly basis. Personal meetings will be required in France or the US later this spring to setup working platforms for firmware development. Additional travel at crucial junctures to test and integrate components will be supported by the collaborating institutions.

Component	prototype/processor	production/processor
9U PCB + Mechancal	$1,700	$1,100
9U Components	$1400	$800
9U Assembly	$400	$200
9U Total	$3,500	$2,100
PMC PCB	$300	$200
PMC Components	$100	$50
PMC Assembly[7]	$0	$50
PMC Total	$400	$300
SBC	$3,400	$3,000
Totals (3 boards)	$21,900	
Totals (30 boards)		$162,000

Table 3: Hardware cost estimates to construct L2βeta processors.

8 Schedule

The project schedule is summarized in Fig. 11. Our goal is to deliver prototype hardware this summer and to verify the L2βeta functionality at the DØ Level 2 Test Stand by mid-fall of this year. Barring any major difficulties a production run would be ordered and integration of the L2βetas into the trigger could begin by the end of the year.

9 Costs

A summary of the project costs is shown in Tables 3 and 4. The total estimated cost to build and fully commission three prototype L2βetas is $110K. The breakdown of available funding is as follows:

Orsay: $65K (equivalent) for engineeering and assembly, ~$12K components and assembly. Total: $75K.

Fermilab/DØ operating funds: Total: $50K. These funds will cover purchace of the SBCs, engineering costs at Maryland, leaving roughly ~$17K of contingency funds.

[7] The PMC card can be fully assembled at Orsay's Electronics shops

	Orsay[8]	Maryland[9]
9U card prototype	$50K	$3K
System commissioning	$15K	$20K

Table 4: Engineering cost estimates to construct L2βeta processors.

The funding strategy for a L2βeta production run will depend on the ultimate decision of the collaboration on how and when to proceed with the trigger integration.

10 Future Upgrade Paths

The baseline design does not include the roughly 40 J2 pins and their respective drivers which would be necessary for use of the βeta by the CDF collaboration, with their custom J2 backplane. It is likely that the L2βetas design could be generalized to include this functionality.

An upgrade from a SBC with 32-bit PCI to a SBC with 64-bit PCI would require revision of the mezzanine board. Some attention to this possibility in the final layout of the 9U card should allow this type of future upgrade with no hardware changes to this device.

11 Acknowledgements

We greatfully acknowledge the keen eye and assistance of Jim Linnemann in our pursuit of the better βeta.

12 Document History

version 1.4
Update schedule: move forward date to order SBCs for prototypes (VMIC has 60 day lead time), adjust prototype production schedule to Orsay's shop schedule; minor cost updates; typo fixes.

[8] The numbers represent equivalent US Dollar values for Orsay's engineering commitment based on the FTEs listed in section 7.

[9] The prototype funds cover costs incurred at Maryland for work on the initial project specifications. An added $20K will be earmarked for assistance with firmware engineering from Maryland's Instrumantation Shop.

References

[1] Nucl.Instr.Meth.A269:51,1988.

[2] Tundra Semiconductor Corp., www.tundra.com.

[3] AlphaPC164 Technical Reference Manual,
http://ftp.digital.com/pub/Digital/info/semiconductor/literature/archives.html

[4] Level-2 Calorimeter Preprocessor Technical Design Report,

[5] Technical Design Report for the Level 2 Global Processor,
DØNote #3402.

[6] Intel Architecture Software Developer's Manual,
ftp://download.intel.com/design/mobile/MANUALS/24319101.PDF

[7] http://umwnt1.physics.lsa.umich.edu/alpha/schematics.htm

[8] DØ MBT pages: http://macdrew.physics.umd.edu/dzero/trigger/mbt.html

[9] For VMIC VMIVME-7740 infocmation see:

`http://embeddedpc.vmic.com/info_files/hw_sbc_vme_table.html`
`http://galileo.phys.virginia.edu/`
`~rjh2j/l2beta/beta_docs/SBC/VME-7740_1_Rev_A.ps`

[10] A copy of the PMC specification is available at:

`http://galileo.phys.virginia.edu/`
`~rjh2j/l2beta/beta_docs/PCI/pmc_draft.pdf`

[11] Information on Spec measurements can be fount at www.specbench.org

[12] Xilinx XCV405E, www.xilinx.com, goto VIRTEX-EM products.

[13] http://www2.plxtech.com/products/toolbox/9054.htm

[14] Attila Raz, private correspondence.

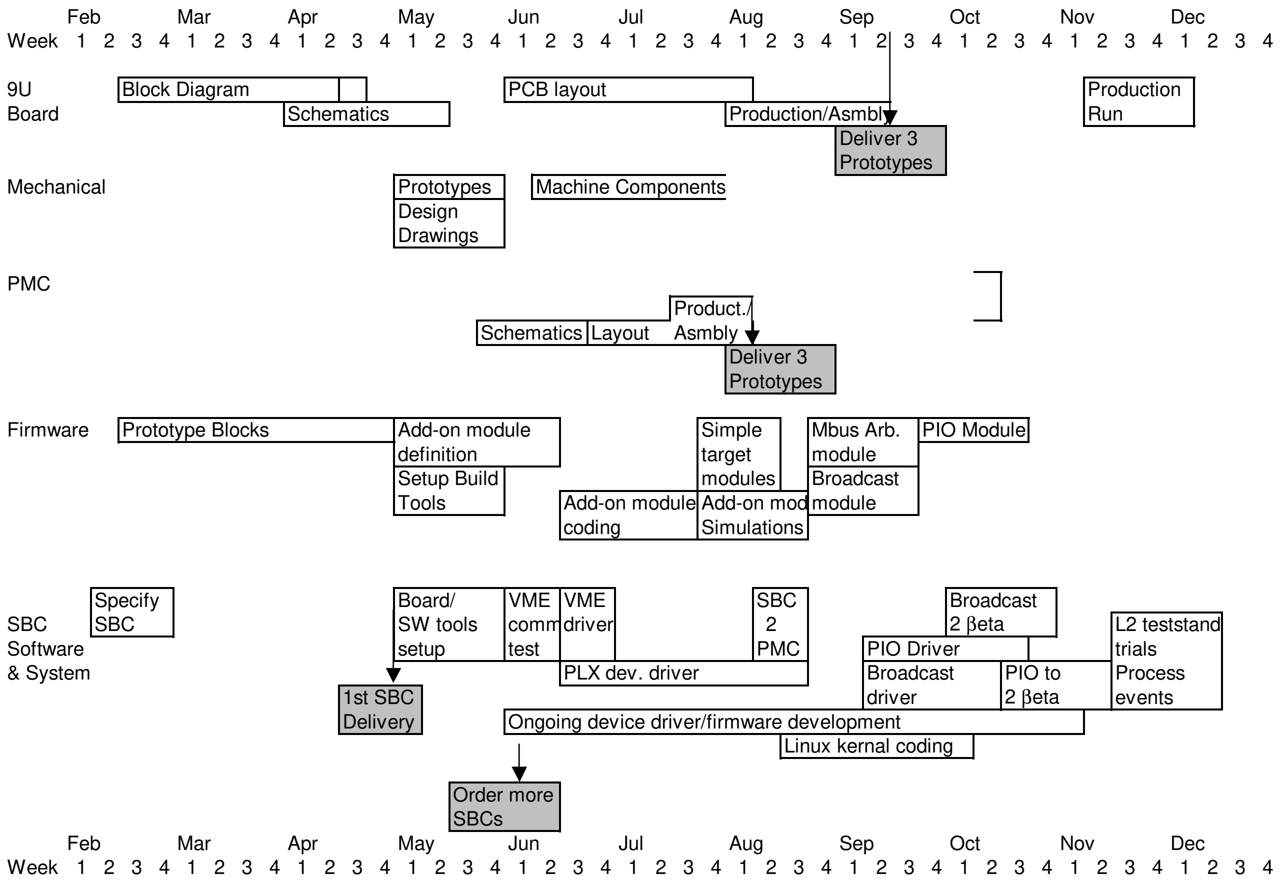

Figure 11: Timeline estimate for development of the L2βeta processors.