ZPD Design Overview[1]

7 April 2003

S. Bailey, G. Brandenburg, A. Eisner, N. Felt, V. Halyo, W. Innes, M. Morii, J. Oliver, U. Schwanke, K. Shen, N. Sinev, Su Dong, E. Won

For the *BABAR DCT Upgrade Final Design Review*

This document describes the design of the Z-P_T Discriminator (ZPD), including an overview of the block diagram, I/O, and the algorithms. The plans for the production and testing of the ZPD, and the expected performance based on simulation studies, are found in separate documents presented at this review.

Table of Contents

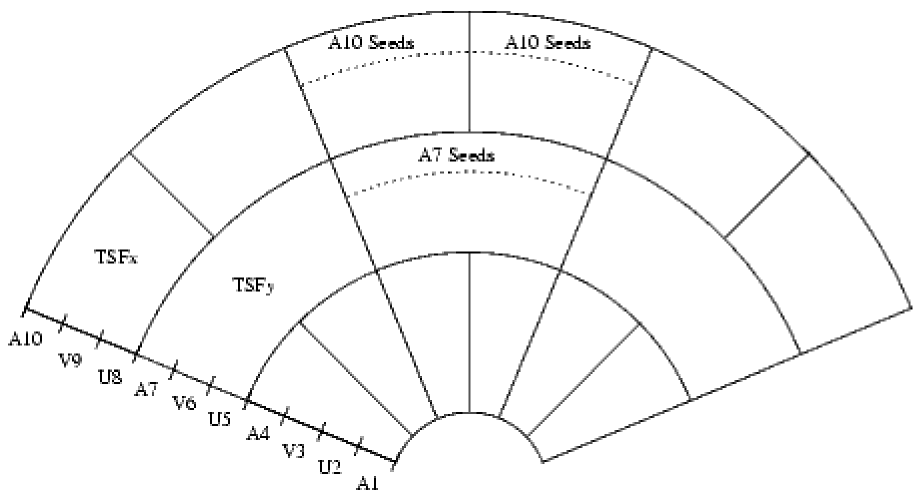
1 Overview ... 3
2 I/O ... 3
2.1 I/O Overview ... 3
2.2 Track Segment Finder (TSF) Input ... 4
2.3 Global Trigger (GLT) Output ... 4
2.4 Fast Control (FC) I/O ... 5
3 Clocks and Latency Constraints ... 5
4 ZPD Functional Description ... 5
4.1 Block Diagram Overview ... 5
4.2 Decoder/Driver and Fast Control Interface ... 6
4.2.1 Decoder/Driver ... 6
4.2.2 Fast Control Interface ... 6
4.3 The MegaBus ... 6
4.4 Track Finder/Fitter Algorithm Engines ... 7
4.4.1 Seed Track Finder ... 7
4.4.2 Track z_0 Fitter ... 7
4.5 Decision Module ... 8
4.6 Diagnostic Memories ... 9
4.6.1 Input Memory ... 9
4.6.2 MegaBus Memory ... 9
4.6.3 Finder Memory ... 9
4.6.4 Fitter Memory ... 9
4.6.5 Fit Result Memory ... 10
4.6.6 Output Memory ... 10
4.7 DAQ Memories ... 10
4.7.1 Input DAQ Memory ... 10
4.7.2 Output DAQ Memory ... 10
4.8 Internal Block I/O Summary ... 11
4.8.1 MegaBus ... 11

[1] http://www.slac.stanford.edu/BFROOT/www/Detector/Trigger/upgrade/fdr/zpd/ZpdOverview.pdf

4.8.2 Seed Track Finder to *z* Fitter ... 11
4.8.3 z Fitter to Decision Module ... 11
4.8.4 Op Control to/from Diagnostic Memory/FIFOs ... 11
4.8.5 Decision Module to GLT (via ZPD I/O Board) ... 11
5 FPGA Resources ... 11
6 Latency Summary ... 12
Appendix A Changes Since CDR ... 14
Appendix B TSF to ZPD Data Format ... 15
Appendix C ZPD Block Diagram ... 17

1 Overview

This document describes the design of the BABAR Level-1 z_0 trigger module, the Z-P_T Discriminator (ZPD). The ZPD replaces the P_T Discriminator (PTD) which selects tracks based upon their transverse momentum, p_T. The ZPD adds the ability to select tracks based upon their origin along the beamline (z_0), in addition to being able to select tracks based upon their p_T. The track-finding algorithm inherently selects tracks that originate near the beam in the transverse direction. The primary purpose of the ZPD is to find the z_0 location of tracks at the first-level (Level-1) trigger in order to accept signal tracks which come from the beam interaction region ($z = 0$) and to reject background tracks which come from beam–beampipe interactions at $|z_0| > 20$ cm. Each ZPD receives segment information from Track Segment Finders (TSFs) covering 3/8 of the drift chamber in azimuth (ϕ). The ZPD output to the Global Trigger (GLT) system is 4 to 6 bits of trigger decision information. Each of the 8 ZPD modules is responsible for triggering on tracks that have a segment in superlayer A7 or A10 within the central 45° of its coverage in ϕ. The following diagram shows the ϕ coverage of a ZPD module. Superlayers that begin with "A" are axial; "U" and "V" superlayers are stereo layers with alternating directions.



A description of the current Level 1 trigger system may be found in *BaBar Level 1 Drift Chamber and Global Trigger Implementation*[2]. That document provides a description of the drift chamber geometry, the trigger system architecture, and the basic vocabulary of segments, layers, superlayers, clocks, various acronyms, etc.

2 I/O

2.1 I/O Overview

The ZPD system will be implemented as eight 9U-sized boards, placed in a single crate with a Binary Link Tracker (BLT) and a Read Out Module (ROM). The TSF data input and trigger decision output pass through a ZPD interface board (ZPDi) in the back of the crate. The input data come from 9 TSF modules, which provide track segment information from 3/8 of the detector. The ZPD trigger decision is passed to the GLT.

[2] F.A. Kirsten *et al*, http://design.lbl.gov/BF/Trigger/specs/sysdescr/imprep-42.ps

All I/O for the ZPD is accomplished via the three connectors, J1, J2, J3, on the backplane. The pin assignment is given in a separate document[3].

2.2 Track Segment Finder (TSF) Input

The ZPDi receives input from 9 TSF modules: 6 TSFx, and 3 TSFy. Each TSF sends its data in 16 words clocked by CLK60 such that the entire TSF dataset is received every CLK4 period. Each TSF link includes a frame bit, which is high during the clock tick before the first of the 16 words. The data are passed from the ZPDi to the ZPD via the J1 and J2 backplane connectors.

The TSF data format is specified in Appendix B. Each bit carries information for one TSF segment. The first word is a mask word indicating whether a segment has been found or not. The subsequent bits contain the drift chamber cell location (loc), the 6-bit fine ϕ information with respect to that cell (ϕ), and the 3-bit ϕ error ($d\phi$). The cell location requires only 4-bit because a part of the geometrical position of the segment can be inferred from which TSF module found the segment.

The segments from the TSF modules are organized in groups of three per superlayer per sector[4]. If a TSF finds more than 3 segments in a superlayer within a sector, the 3 highest quality segments are sent. Due to geometrical constraints, not all groups of segments could contribute to a track that intersects layer A7 or A10 within the central 1/8 in coverage. Thus some of the incoming segments may be dropped immediately on the ZPDi. The baseline design requires 144 segments, but additional segments (153 total out of 180 which arrive every CLK4 period) will be sent through the backplane from the ZPDi to the ZPD to allow for design contingency and future expansion of the ZPD capabilities. The baseline design will only process 144 of these segments.

The physical TSF to ZPD link will use a National Semiconductor LVDS 287/288 transceiver pair which serializes a 28 bit bus into a 4 bit LVDS cable. The data received on these 9 TSF links will be clocked by the receiver clock and then passed through the ZPD backplane along with their frame bits. The Decoder/Driver FPGA on the ZPD synchronizes the input signals with the ZPD board clock and compares the frame bits to ensure that the 9 TSF words from the various links have been clocked together.

2.3 Global Trigger (GLT) Output

The ZPD generates 4 signals of trigger decision to send to the GLT. To allow for future upgrade possibilities, the ZPD is capable of generating 8 trigger decisions which are sent to the ZPDi. Currently only 4 of these are sent to the GLT. This document describes all 8 signals while recognizing that initially only 4 will be used. These 4 signals can be configured as 2 to 4 different triggers with coverage of 1/16 or 1/8 in azimuth.

The output to the GLT is clocked every CLK8. A ZPD trigger acceptance will be held high for 3 CLK8 periods. The exact definition of each trigger line has not yet been specified, but the base design and configurable parameters are described in *ZPD Decision Module Algorithm Definition.*[5]

[3] http://www.slac.stanford.edu/BFROOT/www/Detector/Trigger/upgrade/fdr/zpd/ZpdPinout.pdf
[4] A sector covers $2\pi/16$ radians in φ.
[5] http://www.slac.stanford.edu/BFROOT/www/Detector/Trigger/upgrade/fdr/zpd/DecisionModule.pdf

2.4 Fast Control (FC) I/O

The input clock and control signals and the output DAQ data are processed with the BABAR Fast Control interface on the J3 connector. The commands arrive on a single serial line (the CLink) clocked at CLK60. The output is similarly serialized on a single output line (the DLink) at CLK60. The ZPD uses a slightly modified version of the common Fast Control interface logic for the BABAR Drift Chamber Trigger (DCT) system, documented in *BaBar – DC Trigger Fast Control Interface FPGA*[6].

3 Clocks and Latency Constraints

The basic ZPD board clock is 59.5 MHz (CLK60) which is provided by the Fast Control (FC) interface. Various multiples and divisions of this clock are used throughout the board. Of particular importance are:

- A complete set of data arrives from the TSFs every CLK4 (CLK60 divided by 16) period. This period defines how much time each step of the algorithm has to process the data.
- The output to the GLT is clocked every CLK8 (CLK60 divided by 8).
- Internal to the ZPD FPGAs, CLK60 is doubled to CLK120 to allow more data to be processed per FPGA within the latency constraints.
- An LVDS bus, called the "Megabus," transmits the segment information at CLK120 from the Decoder/Driver FPGA to the Finder/Fitter FPGAs, which find tracks and fit them for z_0.

The ZPD has approximately 2.2 μs to make its trigger decision and pass the signal to the GLT. This corresponds to 8 CLK4 ticks, or 256 CLK120 ticks. The current implementation takes 7.5 CLK4 ticks.

4 ZPD Functional Description

4.1 Block Diagram Overview

The ZPD Block diagram is shown in Appendix C.

The data arrives at 60 MHz from 9 TSF modules on the ZPD I/O (ZPDi) board. It passes through the backplane to the Decoder/Driver (DD), which synchronizes the data with the ZPD board clock and drives the data to the rest of the board.

The data are passed from the DD to the track Finder/Fitter (FF) algorithm engines via a 75 bit-pair LVDS bus (the MegaBus) at 120 MHz. There are 6 FFs, each of which processes 2 seed tracks per CLK4. The FFs find initial seed tracks with a segment in layer A7 or A10 (12 seeds total) and then fit for their origin in z. The z_0, error on z_0, p_T, and tanλ information for each seed track is passed to a Decision Module (DM). The DM produces 8 bits of trigger decision data which are passed to the ZPDi.

[6] K. Marks, http://design.lbl.gov/BF/Trigger/specs/fast_control/TrgFcInterface.ps

Throughout the ZPD, several diagnostic memories help troubleshoot the ZPD system by recording or playing back the data stream. Additionally, DAQ memories store data to be read out for events that pass the global trigger.

The clock signal, command input, and diagnostic data output pass through the J3 connector to the BaBar Fast Control (FC) interface. The Op Control interprets Fast Control Commands for the ZPD and interfaces with the rest of the board.

4.2 Decoder/Driver and Fast Control Interface

4.2.1 Segment Decoder/Driver

Purpose: Receive incoming TSF segment data via ZPDi, synchronize data from individual TSF links, and drive the data on the MegaBus.

Input:

- 153 bits of TSF data at CLK60 frequency (but not necessarily CLK60 phase)
- 9 frame bits (one per TSF link) at CLK60
- CLK60 clock

Output:

- 75 bit LVDS MegaBus at CLK120 containing the TSF segments

The FPGA resources for the DD are dominated by the I/O pin count. Additionally, the I/O pinout should match that of the Finder/Fitter FPGAs so as to avoid routing complications of the MegaBus. XC2V3000 is the smallest Xilinx Virtex II FPGA which meets these requirements.

Each input signal line carries information for one segment with the bits arriving serially. This data needs to be “rotated” such that all bits for a given segment are simultaneously available. The minimal latency solution is for the Decoder/Driver to pass the data through as quickly as possible onto the MegaBus and allow the Finder/Fitters (FFs) to rotate the data upon receipt. This replicates the rotation logic in every Finder/Fitter FPGA. Alternatively, the Decoder/Driver could rotate the data before passing it onto the MegaBus. This incurs an additional CLK4 in latency but requires fewer FPGA resources overall. The latter design has been adopted for the ZPD. This requires one CLK4 to receive the data, and one CLK4 to reformat and send the data to the FFs.

4.2.2 Fast Control Interface

The Fast Control interface receives and decodes commands on the serial CLink, and returns data on the serial DLink. The Op Control interprets these commands and interfaces with the rest of the ZPD. This interface is used to load and read the diagnostic memories, to read the DAQ memories, configure the ZPD, and generate clock signals.

4.3 The MegaBus

The bus from the Decoder/Driver to the Finder/Fitters needs to transport 144 bits of data every CLK60. With 14 bits per segment, this could be implemented using a 70-bit wide bus running at CLK120. The MegaBus on the ZPD has 75 LVDS-pairs, which can transport 5 TSF segments per CLK120 with 1 extra bit per segment for contingency. The bus runs directly underneath the

Decoder/Driver and Finder/Fitter FPGAs and tap off to the I/O pins. It occupies 6 layers on a 12-layer board.

4.4 Track Finder/Fitter Algorithm Engines

The Finder/Fitter (FF) algorithm engines attempt to find one seed track per seed segment in layers A7 or A10, for a total of 12 possible tracks per ZPD per CLK4. The algorithms are implemented in 6 FPGAs, each processing 2 seeds per CLK4. For each seed segment, a FF attempts to find a track which passes through the seed segment, find other segments along that track, and fit for its origin in z. Other than the selection of which segment is the seed segment, the A10 FFs are all identical, as are the A7 FFs.

4.4.1 Seed Track Finder

Purpose: Find initial $1/p_T$ and $\tan\lambda$ estimate for a track which has a seed segment in layer A7 or A10, and find the closest segments to that seed track in each layer.

Input: ϕ and $d\phi$ information from every segment.

Output: 0, 1, or 2 seed tracks per Finder. Each seed track has

- $1/p_T$ (6 bits)
- $\tan\lambda$ (6 bits)
- Segment hit map (10 bits, one per superlayer)
- 10 relative ϕ values of 11 bits each. These are the difference in ϕ between the track segment and the seed segment in units of fine-ϕ bins for the segment superlayer.

The basic steps of the seed track finding algorithm are:

- Subtract seed segment ϕ from all segment ϕ values to get relative ϕ value. This allows each engine to work identically, independent of an absolute ϕ value.
- Fill a ($1/p_T$, $\tan\lambda$) Pattern Recognition Matrix which counts how many segments exist which are consistent with each ($1/p_T$, $\tan\lambda$) hypothesis.
- Find the best ($1/p_T$, $\tan\lambda$) combination in the matrix.
- Find the closest segments to a track with that ($1/p_T$, $\tan\lambda$).

The design and the implementation of this algorithm is explained in detail in *ZPD Seed Track Finder Algorithm Description*[7]. This implementation processes two seed tracks using a XC2V4000 FPGA running at CLK120. The baseline design uses 20 x 13 $1/p_T$ x $\tan\lambda$ bins. Options with different bin sizes have been studied, but these provide little improvement for the algorithm performance. The current implementation requires 3 CLK4s to receive the data, find the $1/p_T$ and $\tan\lambda$ estimates for the seed tracks, and find the nearest segments to those tracks.

4.4.2 Track z_0 Fitter

Purpose: Fit the segments of a seed track to find the z_0 origin of the track and refine the 1/pT and $\tan\lambda$ measurements.

---

[7] http://www.slac.stanford.edu/BFROOT/www/Detector/Trigger/upgrade/fdr/zpd/FinderDescription.pdf

Input: The Seed Track Finder Output

- $1/p_T$ (6 bits)
- $\tan\lambda$ (6 bits)
- Segment hit map (10 bits, one per superlayer)
- 10 relative phi values of 11 bits each

Output:

- z_0 (8 signed bits, unit of 1 cm)
- z_0 error (4 bits, unit of 1 cm)
- track curvature (proportional to $1/p_T$) (8 signed bits, unit of 2^{-12} / cm)
- $\tan\lambda$ (8 signed bits, unit of 2^{-5})
- Track segment map (which layers have segments on the track, 10 bits)

A detailed explanation of this algorithm is found at in *ZPD Fitter Design and Implementation*[8]. The basic steps are:

- Fit segments in r-ϕ to obtain improved ϕ_0 and curvature measurements. Stereo segments are combined to produce effectively axial measurements.
- Using ϕ difference between the track and the stereo segments, find the z location of the stereo segments.
- Fit stereo segments in r-z to obtain z_0 and $\tan\lambda$ measurements.

The Fitter requires one CLK4 to fit three seed tracks. Its FPGA resources are small in comparison with the Finder.

4.5 Decision Module

Purpose: Collect information from the Finder/Fitters and make a trigger decision. The nominal trigger decisions will be "is there a track with a $< z_0 <$ b?" and "is there a track with $p_T >$ c?" More complicated trigger options will consider $1/p_T$, $\tan\lambda$, and z_0 error information in conjunction with z_0.

Input from each of 12 possible seed tracks – the Finder output:

- z_0 (8 bits)
- z_0 error (4 bits)
- $1/p_T$ (8 bits)
- Track segment map (which layers have segments on the track, 10 bits)

Output: 4 trigger decisions per 1/16 wedge for 8 bits total at CLK8. Initially, only 4 bits will be used by the GLT.

[8] http://www.slac.stanford.edu/BFROOT/www/Detector/Trigger/upgrade/fdr/zpd/Fitter.pdf

The baseline algorithm for the Decision Module is described in *ZPD Decision Module Algorithm Definition*[9]. The FPGA resources for the DM are driven by I/O requirements, not by processing needs. The XC2V1000 provides sufficient resources for the trigger decision.

4.6 Diagnostic Memories

The Diagnostic Memories help with troubleshooting the ZPD system by recording or playing back the data stream. Similar memories are implemented throughout the rest of the BABAR Trigger System. The depth of the Diagnostic Memories is chosen to be 17.2 µs, or 64 CLK4s.

Each block of the Diagnostic Memories can operate either in Save or Playback Mode:

- In Save Mode, the incoming data is recorded into the memory and is later read out using Fast Control Commands (FCCs).
- In Playback Mode, the memory is pre-loaded with simulated data, which is sent downstream.

In both modes, the recording/playing is initiated with a Start Memory FCC. Stopping the recording/playing may be controlled in two ways:

- In Continuous Mode, the memory blocks are used as circular buffers and written/read repeatedly. Exiting RUN mode stops the operation.
- In One-Shot Mode, the operation is stopped after 64 CLK4s.

The following sections describe the Diagnostic Memories from upstream to downstream.

4.6.1 Input Memory

The Input Memory holds input data to the ZPD module as they are sent over the MegaBus. This can be used as the source of the simulated data when the entire ZPD module is tested.

4.6.2 MegaBus Memory

The MegaBus Memory holds the identical data as the Input Memory, but as they are received in the Finder/Fitters. Comparing them with the Input Memory tests the integrity of the MegaBus. This memory stores 64 events of 144 segments each.

4.6.3 Finder Memory

The output from the Seed Track Finder is recorded in this memory. Two seed tracks are stored for every event. Each seed track has: hitmask[9:0], curvature bin[5:0], tanλ bin[5:0], and 10 relative phi values of 11 bits each.

4.6.4 Fitter Memory

The outputs from the Z Fitter are recorded in this memory, also storing two tracks per event. Each fitted track has: z0[7:0], z0error[3:0], curvature[7:0], tanλ[7:0], and hitmap[9:0].

[9] http://www.slac.stanford.edu/BFROOT/www/Detector/Trigger/upgrade/fdr/zpd/DecisionModule.pdf

4.6.5 Decision Module Fit Result Memory

This memory stores the same data as the Fitter Memory, but as they arrive in the Decision Module. Comparing it with the Fitter Memories tests the integrity of the Fit Result Buses, which connect the Finder/Fitters with the DM.

4.6.6 Output Memory

The Output Memory stores the 8-bit trigger decision produced by the Decision Module.

4.7 DAQ Memories

The DAQ Memories store the data that are read out as a part of the BABAR event data. The size and the format of the DAQ data is defined in *ZPD Event Data Format*[10].

Each block of the DAQ Memories consists of two components:

- A circular (or FIFO) buffer that holds the data during the Level 1 Trigger latency. The buffer must be at least 12 μs long.
- Event buffers into which a 2.2 μs (8 CLK4s) worth of data is copied from the latency buffer on a Level 1 Trigger. The memory address where the 2.2 μs piece starts must be programmable to adjust for the trigger latency. There must be 4 event buffers to allow 4 Level 1 Triggers to be received before the first event is read out.

The location of the DAQ Memories and their sizes are discussed below. Note that the DAQ data will be sparsified once they are received by the ROM, as is currently done for the existing DCT system. The maximum data rate is limited by the speed of the 1 bit serial DLink which reads out DAQ memory from each ZPD. The maximum data size is 6 kbit per event per ZPD module.

4.7.1 Input DAQ Memory

The input data volume for the ZPD modules is too large to be transferred as a part of the DAQ data. To ensure the integrity of the data transmission from the TSFs, however, it is desirable to have some information for each of the 153 input signal lines. A set of 153 flags that indicate whether a TSF segment was sent through each line would satisfy this requirement.

The size of the circular buffer is 153 bits x 64 words. This can be implemented using 1 RAM block. After padding zeros to fit to the next 16-bit boundary for each CLK4 tick, each event buffer is 160 bytes.

4.7.2 Output DAQ Memory

Although the "output" from a ZPD module is only 8 bits, it is beneficial to have additional information in the DAQ to allow performance studies on the data. Recording the outputs from the Algorithm Engines, available to the Decision Module, will provide such a possibility. There are 38 bits from each of the 12 possible tracks. Adding the 8-bit output of the Decision Module makes the data size 464 bits per CLK4 per module.

The size of the circular buffer is 464 bits x 64 words, which uses 2 RAM blocks. After padding, the event buffers are 400 bytes each.

---

[10] http://www.slac.stanford.edu/BFROOT/www/Detector/Trigger/upgrade/fdr/zpd/ZpdEventData.pdf

Adding the Input and Output DAQ data the size of the DAQ data is 560 bytes per event per module. Actual event data is preceded by a 4-byte header. In addition, a 2-byte padding is needed between the Input and Output data for a technical reason. The total data size, 566 bytes, satisfies the maximum data size.

4.8 Internal Block I/O Summary

4.8.1 MegaBus

150 bits every CLK60 semi-serialized into 75 bits every CLK120 on an LVDS bus.

4.8.2 Seed Track Finder to *z* Fitter

132 bits per Finder/Fitter per seed track, internal to an FPGA:

- 10 relative ϕ values, 11 bits each
- $1/p_T$ [5:0]
- $\tan\lambda$[5:0]
- segment map (10 bits, one per layer)

4.8.3 z Fitter to Decision Module

The Fitter results are sent from each Finder/Fitter to the Decision Module on a 12-bit wide bus operating at clk30. The data sent are:

- Track segment map (10 bits)
- z_0 error (4 bits)
- $1/p_T$ (8 bits)
- z_0 (8 bits)
- $\tan\lambda$ (8 bits)

4.8.4 Op Control to/from Diagnostic Memory/FIFOs

The I/O from the CLink/DLink that feeds/reads these buses is only 1 bit at CLK60. The Op Control is connected to all the memories via a 16-bit bus (the Minibus). This bus, along with its control signals, operates at clk30.

4.8.5 Decision Module to GLT (via ZPD I/O Board)

The ZPD sends 8 bits of trigger decisions at CLK8.

5 FPGA Resources

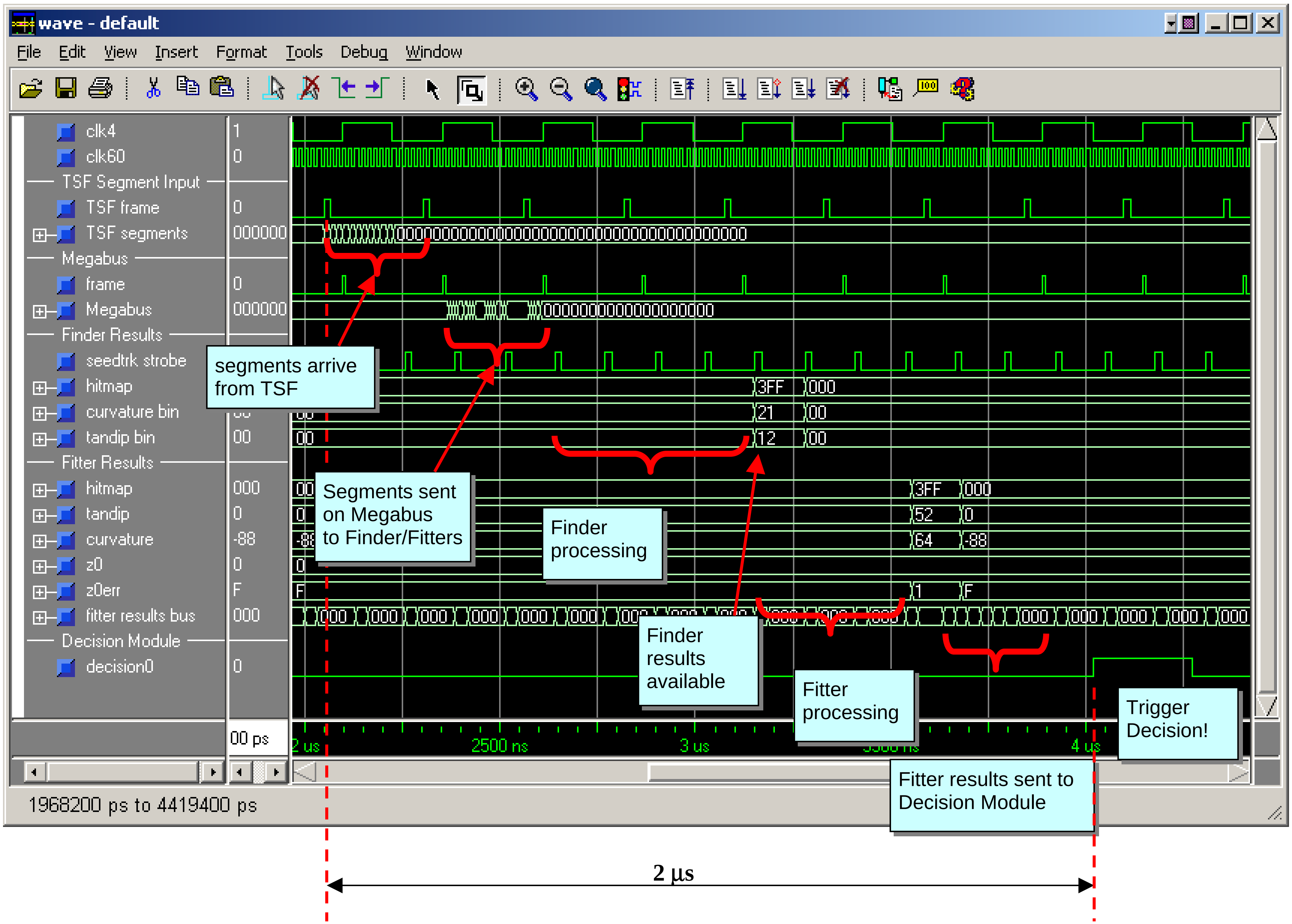
The following FPGAs have been chosen to satisfy the resource needs:

- Data Receiver and Fast Control: XC2V3000.
  The choice is driven by the need to match the I/O pinout with the Finder/Fitter FPGAs in order to achieve a clean MegaBus implementation. The data processing and input diagnostic and DAQ memories easily fit within the resources available.

- Finder/Fitter Algorithm Engines: six XC2V4000s.
  The limiting factor is the size of the Finder algorithm.
- Decision Module: XC2V1000.
  This choice has a sufficiently large number of I/O pins and plenty of processing ability to make the trigger decision.

6 Latency Summary

The ZPD latency is summarized in the following diagram, which shows a board level simulation of the ZPD finding, fitting, and triggering upon an input track. The Decoder/Driver takes one CLK4 to receive the data and another CLK4 to reformat it and send it on the MegaBus. After receiving all of the data, the Seed Track Finder requires two CLK4s – one to fill the Pattern Recognition Matrix and find its peak; and one to find the best segments on a track. The track Fitter begins fitting the seed tracks as soon as they are available and it outputs its results 1.5 CLK4s later. The Fitter results are sent to the Decision Module in one CLK4. The Decision Module must buffer these fit results until all tracks have been found, and then it makes a trigger decision and reports the result to the GLT. The total latency is 7.5 CLK4 ticks (2 µs) from the beginning of the arrival of TSF segments to the beginning of sending the trigger decision to the GLT.


wave - default
File Edit View Insert Format Tools Debug Window
clk4
clk60
TSF Segment Input
TSF frame
TSF segments
Megabus
frame
Megabus
Finder Results
seedtrk strobe
hitmap
curvature bin
tandip bin
Fitter Results
hitmap
tandip
curvature
z0
z0err
fitter results bus
Decision Module
decision0
2500 ns
3 us
4 us
1968200 ps to 4419400 ps
segments arrive from TSF
Segments sent on Megabus to Finder/Fitters
Finder processing
Finder results available
Fitter processing
Trigger Decision!
Fitter results sent to Decision Module
2 µs

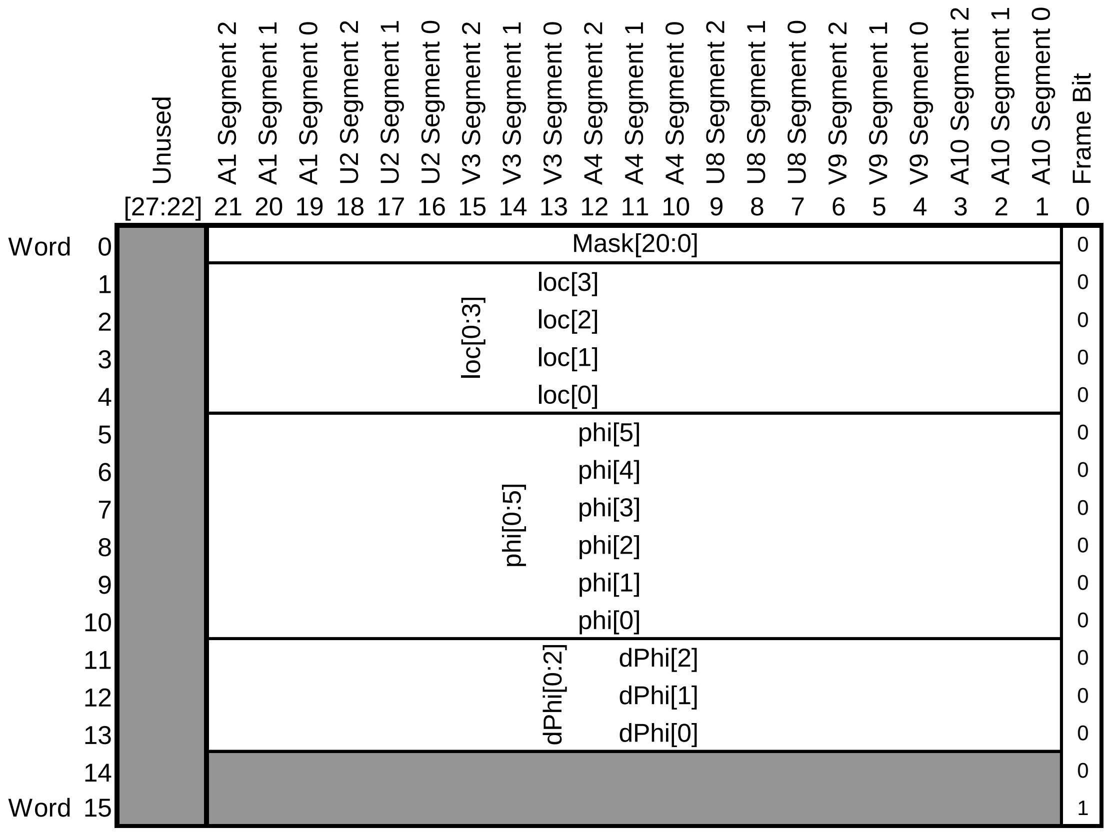
Appendix A Changes Since CDR

A number of changes have been made to the ZPD design since the Conceptual Design Review. The significant ones are summarized in the following list.

- The clock frequency for the MegaBus has been reduced from 180 MHz to 120 MHz. The width of the bus has been expanded from 54 bits to 75 bits to retain the bandwidth necessary to transport 144 TSF segments per CLK4. With 14 bits/segment, the MegaBus can send 5 segments per CLK120 (= 160 segments/CLK4) with 1 extra bit/segment for contingency.
- The clock frequency for the internal logic of the Algorithm Engine has been reduced from 180 MHz to 120 MHz to allow for a better timing margin. The increase in the latency is negligible.
- More Diagnostic Memories have been added. In particular, the MegaBus memories and the Fitter Memories have been introduced. These additions allow testing of the on-board buses, and have been used extensively in the testing of the prototype.

Appendix B TSF to ZPD Data Format

B.1 TSFx Format



B.2 TSFy Format

[27:19]	Unused
18	U5 sector 0 seg 2
17	U5 sector 0 seg 1
16	U5 sector 0 seg 0
15	U5 sector 1 seg 2
14	U5 sector 1 seg 1
13	U5 sector 1 seg 0
12	V6 sector 0 seg 2
11	V6 sector 0 seg 1
10	V6 sector 0 seg 0
9	V6 sector 1 seg 2
8	V6 sector 1 seg 1
7	V6 sector 1 seg 0
6	A7 sector 0 seg 2
5	A7 sector 0 seg 1
4	A7 sector 0 seg 0
3	A7 sector 1 seg 2
2	A7 sector 1 seg 1
1	A7 sector 1 seg 0
0	Frame Bit

Word	[27:19]	18–1		0
Word 0		Mask[17:0]		0
1		loc[0:3]	loc[3]	0
2			loc[2]	0
3			loc[1]	0
4			loc[0]	0
5		phi[0:5]	phi[5]	0
6			phi[4]	0
7			phi[3]	0
8			phi[2]	0
9			phi[1]	0
10			phi[0]	0
11		dPhi[0:2]	dPhi[2]	0
12			dPhi[1]	0
13			dPhi[0]	0
14				0
Word 15				1

Appendix C ZPD Block Diagram

The following block diagram shows the top level design of the ZPD.

ZPD Block Diagram