Appendix A

L2 Global Worker

The DØ L2 trigger, described in Chapter 2.3, is an essential part of the data acquisition system [59]. The L2 trigger system is divided into six crates, each containing processors running identical executables (but configured differently). A processor's configuration specifies which worker runs in the processor's executable. The workers correspond to the type of data being processed in the crate.

Five of the L2 crates are preprocessor crates. These crates receive data from the L1 trigger system and the front-end crates. Each crate has a custom-built Beta processor [26] which forms L2 preprocessor objects by running an algorithm tailored to its data. For example, the L2MUC crate receives information about the central muon system and forms muon objects. The preprocessor formats the data properly, and sends the objects to the L2 Global crate.

The L2 Global worker is responsible for making trigger decisions based on the objects identified by the L2 preprocessors. Trigger decisions are made by creating Global physics objects. These objects can be based directly on the objects reported by the preprocessors or can be created by combining objects from different preprocessors. The L2 Global worker imposes cuts on the Global physics objects according to configuration information it receives from the trigger control computer based on the trigger list.

A.1 Input

Data flows from the L2 preprocessor crates into the L2 Global crate (see Figure A.1). The format of the data is described elsewhere [60]. The Beta processor in the L2 Global crate runs an executable which contains the L2 Global worker code.

The L2 Global worker code has access to all of the preprocessor objects as well as information from the L1 framework (L1fwk) for the current event. For each event, the Global worker uses the L1fwk information to decide which algorithms to run on the data from the preprocessors. It then makes a trigger decision, and returns this decision to the L2 framework code which passes it to the trigger framework. Thus, each event must either pass or fail the L2 trigger, and this decision is based on the input data and the information from the trigger list.

The trigger list specifies which trigger conditions L2 will impose for each run. The trigger list can change as frequently as every run, but attempts are made to achieve stability in the trigger list to reduce complexity at the analysis level. All of the possible configuration values are stored in the trigger database, and the trigger list is formed by choosing particular values for all of the variable parameters. The trigger list is downloaded to the L2 Global crate by the trigger control computer, which receives its instructions from COOR, the system responsible for coordinating the distribution of all the online data acquisition information to all the online crates.

A.2 Trigger Bits and Scripts

The most critical piece of L1fwk information is the L1 trigger mask. This is the record of which L1 bits fired (the reason the current event passed the L1 trigger). The L1

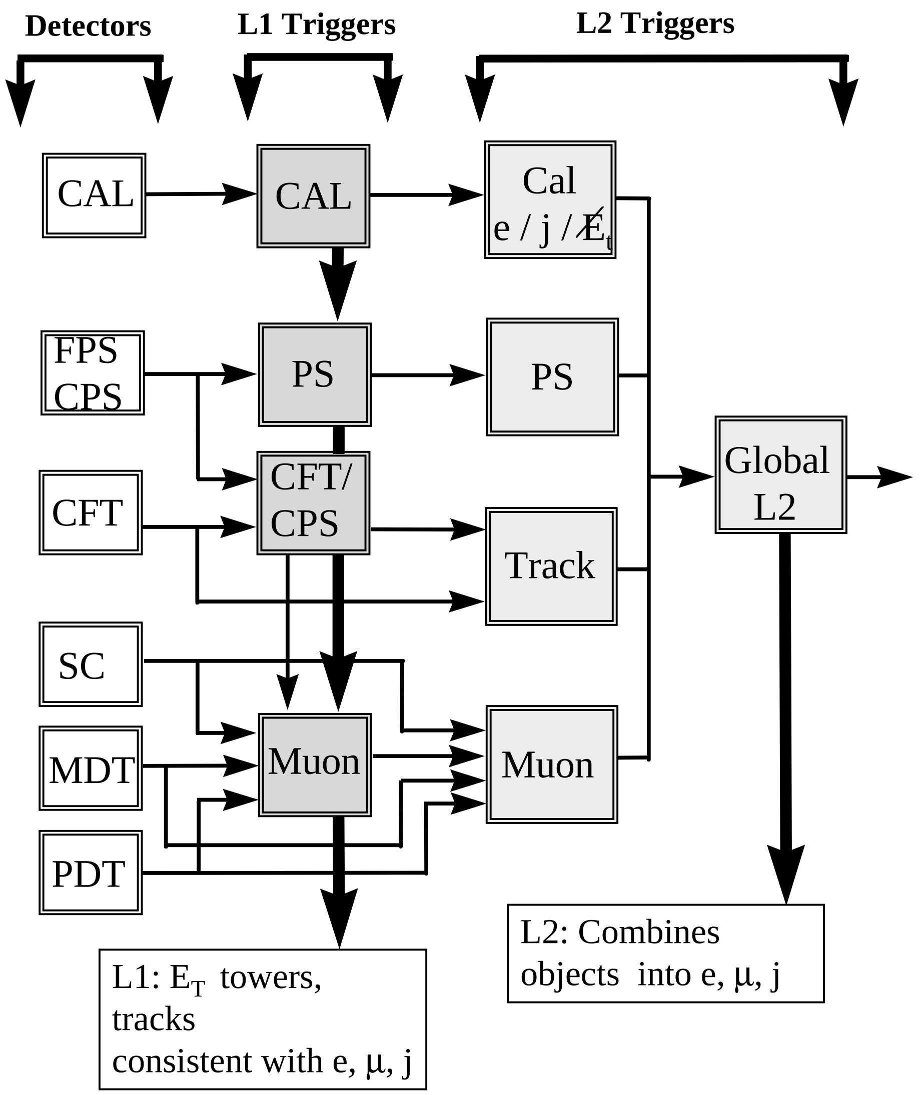

Figure A.1: Data flow from the detectors to the L2 Global worker.

trigger mask is a collection of 128 bits, one for each L1 trigger condition.

At L1, as many as 128 different conditions can be programmed to fire the L1 trigger. For each event that passes, the bits that passed are recorded and sent to L2. The L2 Global worker begins its work every event by checking which of the L1 trigger bits fired. The trigger list specifies which L2 script is associated with which L1 bit.

The L2 script is the trigger condition that must be satisfied in order for the L2 trigger to fire for a given L1 bit. There is a one-to-one correspondence between L1 bits and L2 scripts. The trigger conditions at each level are closely related, and form a coherent trigger condition. An L2 script is defined, which is related to a L1 bit. Once the L1 bit fires, the corresponding L2 script runs. For example, events with a calorimeter energy requirement at L1 may have an electron requirement at L2.

The L2 script is specified by a number of filters and a minimum number of objects required to pass each filter. The simplest script has one filter and a minimum number of objects required to pass that filter. An example is a script with an electromagnetic object filter and a minimum of two objects. This script passes if there are two electromagnetic objects in the event that satisfied the conditions of the EM filter.

If any script passes, the event passes L2, and is processed by the L3 trigger. Events are only written to tape if they satisfy L1, L2, and L3 trigger conditions.

A.3 Tools and Filters

Tools and filters are the main parts of the L2 Global worker. The filters make up the scripts described in Section A.2. The filters in turn rely on tools.

Tools are C++ classes which build a specific type of L2 Global object. The electromagnetic object (EM) tool builds L2 Global EM objects. The muon tool builds L2 Global muon objects. This building of objects is based on the preprocessor objects.

A tool starts with a list of preprocessor objects. It can then apply selection criteria to decide which preprocessor objects should be made into Global objects. It can also correlate information from two preprocessors by combining two preprocessor objects to form one Global object. An example of this is the combination of a track object and an EM object that come from different preprocessors, but refer to the same real electron that entered the DØ detector. The tools are written to be flexible, and are configurable through the trigger list. For the above example, one trigger list parameter specifies whether or not EM objects from the preprocessor should be matched to tracks from the tracking system.

The tools produce lists of Global objects. The filters then use these lists of Global objects to make a trigger decision by imposing trigger requirements on the objects. The requirements are based on the trigger list. For the case of an EM object, the electromagnetic fraction, transverse momentum, and isolation are examples of variables that can be required to have values above or below specified thresholds in order to pass the filter.

The filter generates its own list of objects which have satisfied the trigger criteria. The script then checks to see if there are at least the minimum number of objects required to pass the script.

A.3.1 Tools and Filters Currently Available

Tables A.1, A.2, A.3, A.4 and A.5 summarize the currently available tools and filters and their parameters.

A.3.2 Tools and Filters Used in this Analysis

In the analysis described in this dissertation, two different L2 trigger requirements are used. The first is the requirement of one medium L2 muon. This is implemented by configuring a L2 muon tool which produced L2 muons based on the list of muons

Tool	Parameter	Description
Muon	L1PTTHRESH	Minimum L1 p_T
	REQUIRETRACK	Is match to track required?
	TRACKWINDOWIPHI	Maximum distance to matching track in ϕ
EM	MINET	Minimum E_T
	MINNEIGHBORETACENET	Threshold for which, if central cluster η neighbor is below, it will be turned into 2 separate EM tower objects
	MINNEIGHBORPHICENET	Same for central cluster ϕ neighbor
	MINNEIGHBORETAFWDET	Same for forward cluster η neighbor
	MINNEIGHBORPHIFWDET	same for forward cluster ϕ neighbor
	MINSINGLETOWEREMFRAC	Value for which if EM fraction is greater, it will turn that cluster into two EM tower objects
	MINSINGLETOWERET	In E_T for single tower EM object. Overrides MINET if MINNEIGHBORET is true
	REQUIRETRACK	1 = Require a CFT match found (0 is false)
	TRACKFILTER	Track filter
	TRACKWINDOWIPHI	ϕ match window track
	REQUIRECPS	1 = require a CPS match (0 is false)
	CPSWINDOWIETA	η match window with CPS
	CPSWINDOWIPHI	ϕ match window with CPS
	MAXEM	Maximum number allowed to pass
Jet	MINET	Minimum E_T
MET		
MJT	MINET	Minimum E_T of jets to be included in calculation

Table A.1: L2 tools currently available. For each tool, the configurable parameters are listed, along with a brief description [61].

Tool	Parameter	Description
Track	MINET	Minimum E_T
	TRACKSOURCE	Type of input track
	REQUIREL1ISO	Require L1 isolation confirmation
	REQUIREL1PS	Require L1 preshower confirmation
	L2ISOTYPE	Type of L2 isolation required. 0 = no requirement, 1 or 2 = require 1- or 3-prongs, 3 = require 1-prong
	MAXCHISQ	Maximum χ^2 allowed
	IPSIG	Minimum impact parameter significance
InvMass	MININVMASS	Minimum mass
	NFILTERS	Number of input filters
	FILTER0	First filter
	FILTER1	Second filter
Commission		

Table A.2: L2 tools currently available. For each tool, the configurable parameters are listed, along with a brief description [61].

from the two L2 muon preprocessors.

The muon tool makes Global muons based on each of the preprocessor muons it receives. The muon tool also determines the quality of each muon. A L2 muon filter is then configured to loop over all of these muons, and pass any that are of medium quality or better.

The second L2 trigger used in this analysis is similar, except that it uses a L2 muon filter to require at least two medium muons in order to pass. In addition, an $\eta - \phi$ separation filter is used to check the separation in η and ϕ between the muons passing the muon filter. The separation filter imposes the requirement that for the event to pass, two L2 muons have to be separated by at least 0.15 in η or 0.24 in ϕ.

Filter	Parameter	Description
Muon	MINET	Minimum E_T
	QUALITY	Minimum quality (based on number of hits)
	PROMPT	Minimum timing quality (based on scintillator times)
	SIGN	Required sign of muon to pass
	TOOL	Input tool
EM	EMFRAC	Minimum EM fraction
	ISOFRAC	Maximum isolation fraction
	MINET	Minimum E_T
	MAXEM	Maximum number allowed to pass
	TOOL	Input tool
Jet	MINET	Minimum E_T
	MAXJETS	Maximum number of jets allowed
	TOOL	Input tool
Track	MINET	Minimum E_T
	QUALITY	Minimum quality
	IP	Minimum impact parameter
	IPSIG	Minimum impact parameter significance
	TOOL	Input tool
MET	MINMET	Minimum E_T
	TOOL	Input tool
MJT	MINMJT	Minimum E_T
	TOOL	Input tool
Eta	IETAMIN	Minimum ieta
	IETAMAX	Maximum ieta
	FILTER	Input filter
Phi	IPHIMIN	Minimum iphi
	IPHIMAX	Maximum iphi
	FILTER	Input filter

Table A.3: L2 filters currently available. For each filter, the configurable parameters are listed, along with a brief description [61].

Filter	Parameter	Description
PhiSep	NFILTERS	Number of input filters
	IPHIMINSEP	Minimum ϕ separation value
	IPHIMAXSEP	Maximum ϕ separation value
	FILTER0	First input filter
	FILTER1	Second input filter
EtaSep	NFILTERS	Number of input filters
	IETAMINSEP	Minimum η separation value
	IETAMAXSEP	Maximum η separation value
	FILTER0	First input filter
	FILTER1	Second input filter
EtaPhiSep	NFILTERS	Number of input filters
	IETAMINSEP	Minimum η separation value
	IPHIMINSEP	Minimum ϕ separation value
	FILTER0	First input filter
	FILTER1	Second input filter
	FILTER2	Third input filter
	FILTER3	Fourth input filter
PhiSepVeto	IPHIMINVETO	Minimum ϕ separation for veto
	IPHIMAXVETO	Maximum ϕ separation for veto
	NFILTERS	Number of input filters
	FILTER0	First input filter
	FILTER1	Second input filter
RSep	NFILTERS	Number of input filters
	RMINSEP	Minimum R separation value
	RMAXSEP	Maximum R separation value
	FILTER0	First input filter
	FILTER1	Second input filter

Table A.4: L2 filters currently available. For each filter, the configurable parameters are listed, along with a brief description [61].

Filter	Parameter	Description
HT	HTMIN NFILTERS FILTER0 FILTER1 FILTER2	Minimum HT Number of input filters First input filter Second input filter Third input filter
InvMass	MININVMASS MAXINVMASS TOOL	Minimum invariant mass Maximum invariant mass Input tool
TransMass	MINTRANSMASS MAXTRANSMASS TOOL	Minimum transverse mass Maximum transverse mass Input tool
TimeDelay		
RandomPass		

Table A.5: L2 filters currently available. For each filter, the configurable parameters are listed, along with a brief description [61].

A.4 Output

The output of the L2 Global worker is the record of what the worker did for that event. This consists first of a trigger mask that records which scripts passed L2. The trigger mask is used by the L3 trigger to decide which L3 scripts it will run on the event. The output also includes a record of all of the Global objects which caused the event to pass. The format of the Global objects is specified in [60].

A.5 Code Design and Structure

The code for the L2 Global worker is spread over several packages which are part of both the DØ offline release and the DØ L2 online release.

The offline release contains the trigger simulation code which uses a different framework than the online release. Both frameworks have the same functions; to receive the data and run the worker. The L2 Global code is identical in the online and offline release. The L2 Global code is contained in the packages l2gblem, l2gbljet, l2gbltrack, l2gblmuon, l2gbltau, l2gblgeneric, l2gblbase, l2gblworker, and l2io.

The l2gblbase, l2gblworker, and l2io packages contain the code used to contain the data, pack/unpack it, and run the tools and filters. The L1fwk bits are interpreted here. These include special bits that instruct the worker to treat the event as part of an unbiased sample by passing it regardless of the results of running the L2 scripts. Errors that have been noted by the L2 framework are also interpreted in these packages. The trigger mask from the L1fwk is decoded and the list of L2 scripts to run is prepared. The code runs all the appropriate scripts, gets the L2 decision, fills the output data, and returns the decision to the L2 framework.

The tool and filter code is contained in the remaining packages. The packages are divided only for organizational reasons. The tools all have access to all of the L2 Global data. They each have a parser interface so that they can receive L2 parser configuration data that originates in the trigger list. They use this configuration information to decide how to produce lists of Global objects.

The tools pass the list of objects to the filters. The filters can then generate lists of objects that are subsets of their input lists. It is also possible for a filter to filter the list produced by another filter. Generic filters are stored in the l2gblgeneric package and do just this. An example is the η-separation filter, which can take a list of objects from any filter as input, and filter those objects based on their separation in η.