

ATLAS Internal Note
DAQ-No-070
12 June 1997

Input Parameters for Modelling the ATLAS Second Level Trigger*

Compiled by S. George (RHBNC)[†]
J. R. Hubbard (Saclay),
and J. C. Vermeulen (NIKHEF).

June 12, 1997

1 Introduction

This document gathers together the input parameters currently in use in paper models of the ATLAS LVL2 trigger and tries to suggest a common set of inputs as a starting point for modelling activity.

Inputs covered are architecture, detector parameters, algorithms, hardware models, trigger menus, processing strategies, and the special case of TRT scans and b -jets in which all the above are considered together.

It is suggested that the information in this note forms a baseline for paper models to enable comparison between their results. It should be built on and kept up to date. It is hoped that it may be useful to other modelling, emulation and lab test activities, and informative to the ATLAS LVL2 trigger community in general.

The references section also serves as a bibliography for newcomers to ATLAS LVL2 modelling.

2 Architectures

The ATLAS demonstrator programme [1] is based around three main architectures: A, B, and C. Since their inception, the programmes have developed several variations on these initial designs. Hybrids which combine elements of two or more architectures have also been proposed. Further information may be found in [2, 3, 4, 5, 6].

The architectures themselves are not explicit inputs to paper models. They are indirectly included as part of the calculations of occupancies etc. In full modelling and emulation, the

*<http://www.hep.ph.rhbnc.ac.uk/atlas/papers/modelling-inputs.ps>

[†]e-mail S.George@rhbnc.ac.uk

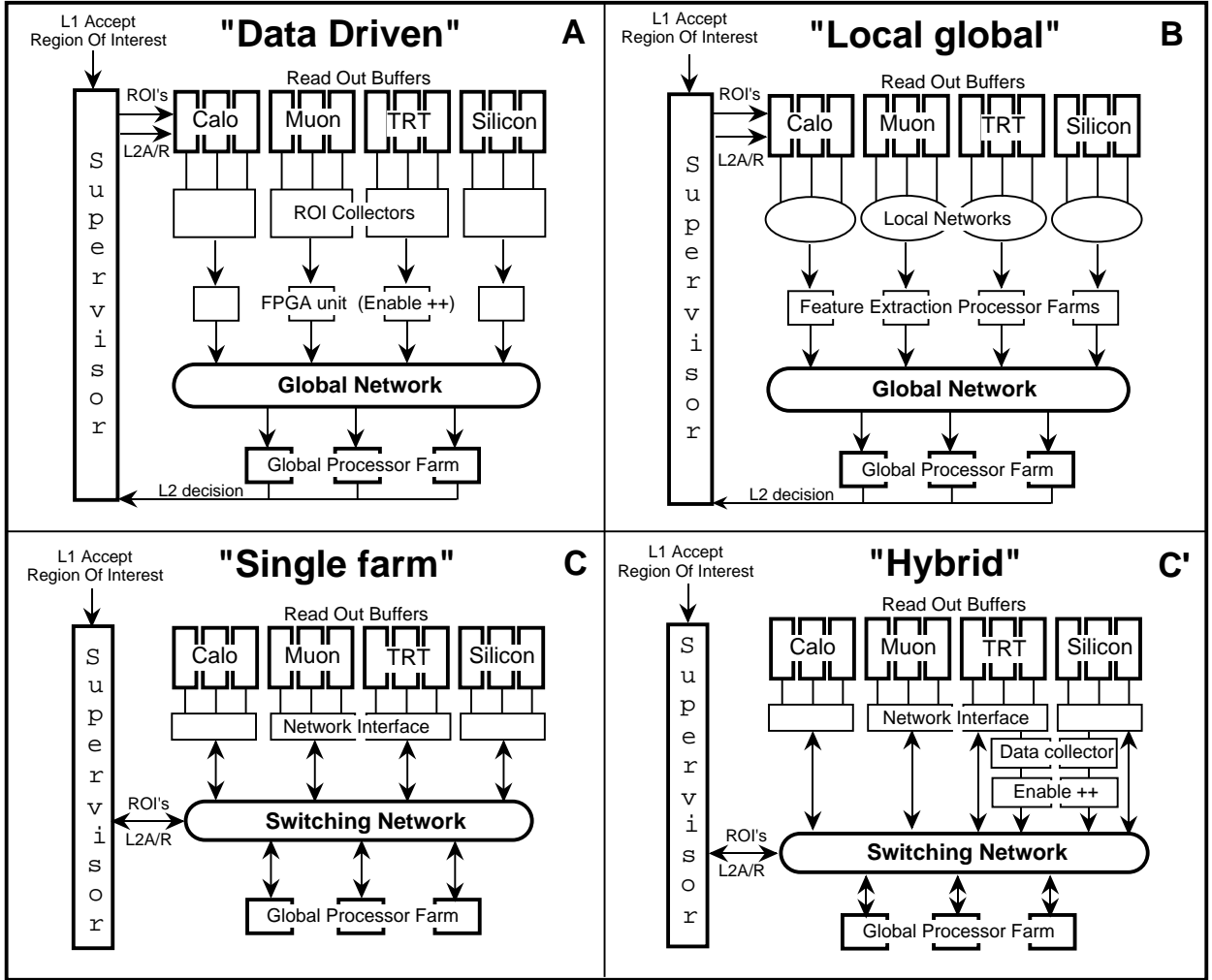


Figure 1: *ATLAS demonstrator architectures [1]*

architecture is built from objects which model the hardware characteristics or process functionality.

Architecture	farms	FEX	protocol
A	global	FPGA	push
B	local + global	local	push + pull
C	single farm	global	pull

Table 1: *Summary of the ATLAS demonstrator architectures*

3 Detector Parameters

The detector parameters are described in detail in reference [7]. The information extracted from this paper is the average number of ROBs in a RoI for each sub-detector, and the average ROB data size for each sub-detector. This is usually used in the form of averages of the whole sub-detector (barrel and end caps) for paper modelling.

Detector	μ RoI	$e/\gamma, \tau$ RoI	jet RoI
muon MDT	3.24	—	—
muon RPC	2.44	—	—
hadron cal.	2.04	2.32	3.75
e.m. cal.	6.19	6.12	16.50
TRT	12.40	6.88	—
SCT	4.44	3.94	—

Table 2: Average number of ROB's per RoI [9]

Detector	μ RoI	$e/\gamma, \tau$ RoI	jet RoI
muon MDT	1.40	—	—
muon RPC	1.16	—	—
hadron cal.	1.50	1.54	2.06
e.m. cal.	3.03	3.03	6.97
TRT	3.40	2.68	—
SCT	4.14	3.74	—

Table 3: Average number of RSI's per RoI [10]

The average number of ROBs per RoI in table 2 was originally taken from [7] but has since been subject to a few updates. The data size per ROB excludes headers and RoI information; message sizes are given later in table 13. The numbers given for muon RoIs assume that they are irregular - i.e. that their size in (η, ϕ) is not uniform. This results in a slightly higher number of ROBs per RoI than regular sizing would give, but at the time of writing the irregular sizes seemed the more likely choice. Note that the SCT readout does not have a tower structure. The grouping of wafers has been optimised using simulated annealing methods [8].

The average number of RSI's per RoI in table 3 is calculated assuming that there are four ROBs per RSI, except for the muon precision detector (MDT's), where there are three ROBs per RSI.

Table 4 shows the data volumes for each type of RoI in each sub-detector. These values also came originally from [7] but have since been updated. Some comments on the data size:

- The data volumes for the forward muon detectors are not yet known. For the moment they are assumed to be the same as the barrel.
- Lower data sizes for the calorimeter are possible due to tower summing in the electromagnetic calorimeter for jets and total RoI summing in the case of \cancel{E}_T triggers.
- The maximum data size is given for calorimeter data where the size depends on η . It is not given for any other sub-detectors because their data size is related to the occupancy of the sub-detector.
- The amount of data from the inner tracking detectors depends on their occupancy, so it varies with luminosity. The calorimeter and muon detectors have fixed data size for any given ROB, but the size can vary between ROBs.

Detector	Part	# ROBS	RoI type	Data volume per ROB (kByte)	
				average	max
muon MDT	barrel	96	all	0.60	
muon RPC	barrel	16	all	0.10	
muon MDT	forward	48	all	???	
muon RPC	forward	3	all	???	
hadron cal		48	$e/\gamma, \mu, \text{jet}$	0.89	0.94
			\cancel{E}_T	0.29	0.37
e.m. cal.		432	$e/\gamma, \mu$	1.30	1.46
			jet, \cancel{E}_T	0.06	0.06
<hr/>					
Detector	Part	# ROBS	RoI type	Data volume per ROB (kByte)	
				high	low luminosity
TRT	endcaps	384	all	0.74	0.28
TRT	barrel	128	all	0.77	0.28
SCT		256	all	1.00	0.34
<hr/>					
TOTAL # ROBS		1462			

Table 4: Total number of ROBS per detector and data volumes per ROB

4 Process Models and Algorithms

This section describes the processes (including algorithms) running on the various processors in the trigger system. Process model diagrams can be found in [11].

Only average times for the algorithms are given here. For full modelling, distributions will be required for some algorithms. It is suggested that these distributions are generated according to the following description, until they are available from measurements..

A time is generated according to an exponential distribution with a certain average, and then the minimum time is added to it. When the time calculated in this way exceeds the maximum time the time used in the simulation is set to the maximum time. So three parameters are needed:

1. minimum time;
2. average time, from benchmarking;
3. maximum time.

The average used for the exponential distribution is computed from the average time (2) minus the minimum time (1). The maximum time is set to a long time (for example 10 ms for the "normal" triggers)

4.1 Supervisor

For the purposes of paper modelling, a simple breakdown of the supervisor functions is used. We are unaware of any benchmarking of supervisor algorithms, so the times allocated to each algorithm are guestimates. The steps are based on the supervisor tasks listed in [12].

1. allocate LVL2 processors and route LVL1 RoI data:
10 μ s per event.
2. formulate ROIRs:
20 μ s per sequential processing step (A/B) or N/A (C).
3. process LVL2 decision (accept/reject event or request more RoIs):
10 μ s per seq step (A/B) or per event (C).
4. prepare decision list for ROBs (done every 100 events or 1 ms):
10 μ s per 100 events.
5. monitoring LVL2 resources and system performance:
10 μ s per event.

Note that the times given here are guestimates. It is hoped that they will be measured in the test lab. The supervisor is a complex system in its own right, so it will be the subject of detailed modelling which goes beyond the scope of LVL2 paper models.

Algorithm	Frequency	Time (μs)
formulate RoIs (C only)	per feature	20
object build	per feature	10
object decision	per seq feature	10
event build	per object	10
event decision	per seq object	10
topological decision	per event	100

Table 5: *Global decision process model.*

There is an additional overhead per message in or out of the overall supervisor system — see section 5.2 for definition and value. The time taken to broadcast the event decision list to the ROBs (following step 4) will be particularly technology dependent, as some network technologies do not support broadcasts.

4.2 Global Processor

Some work on algorithms and benchmarking has been reported in [13], and some references are also given in [15]. However, new measurements based on current ideas of trigger algorithms are needed. For the time being the processing steps and times assigned to each step below are guestimates.

The task of the global trigger processor can be split into various algorithms. We take a general trigger scenario where requests for data and decisions may be made both in parallel and in series. The global processor is used slightly differently by demonstrators B and C.

The description for demonstrator B is as follows. As features arrive in the global processor, they are used to build objects. If a sequential step is completed, and the object may be rejected, there is a decision step. When an object is complete it is built into the event. When all surviving objects have been built into the event, there is a quick decision on whether the objects require further processing; the event could be rejected at this stage if the objects do not match any trigger. The final step for an event that is still a trigger candidate is a topological event decision. In this step calculations such as invariant masses are made.

Demonstrator C uses SFIs for data reordering, which reduces the communications overhead on the global processor. Feature extraction algorithms are executed on the global processor rather than local farms. Details of these algorithms are given in section 4.7. The global processor also has to formulate RoIs, a job done by the supervisor in demo B. Apart from these differences, the same processes apply to demo C as to demo A.

A summary of processes and times is shown in table 5.

4.3 ROBs

The process model for ROBs assumes the processes and estimated execution times listed in table 6.

Process	Frequency	Time (μs)
Manage decision blocks	1/100 events	100
Look up address of RoI data	1/request	10
Extract data from memory	1/request	10
Extract data for LVL3	LVL2 accept rate	10

Table 6: *ROB process model.*

Raw data input from the RODs is assumed to be handled by dedicated hardware. Requests for data to be sent to LVL3 are generated internally by the event decision record received from the supervisor. The rate at which this is done depends on the LVL2 accept rate, which varies according to the trigger strategy. These rates are given in table 11. Input and output processes are assumed to run asynchronously, driven by interrupts. The other processes are run in a loop, so they do not cause any further context switching. See section 5.2. for details of the i/o overhead.

4.4 Pre-processing

Pre-processing is a catch-all term for any data manipulation that is done before feature extraction. It could take place in RODs, ROBs, RSIs, or in dedicated processors installed somewhere before the processor where feature extraction is executed.

Note that the ROB processing strategy and times proposed in [2] are superseded by the times given here.

RoI type	subdet	Algorithm	Time (μs)
MU	MUON	none	0
MU/EM/TAU	ECAL	calibration	50
MU/EM/TAU	HCAL	calibration	20
J	ECAL	calib + tower sums	50
J	HCAL	calib only	20
ME	ECAL	calib + E_T sum	70
ME	HCAL	calib + E_T sum	40
EM/TAU/MU	TRT	none	0
EM/TAU/MU	SCT	none	0

Table 7: *Pre-processing algorithm times*

It has become apparent from the initial paper model results that some ROBs are overloaded due to the high rate at which data is requested from them. This is especially true of the calorimeter (low p_T jets) and TRT ROBs (for full scan). The processing power in a ROB is limited and it is undesirable (i.e. expensive) to increase the number of ROBs to increase this. Therefore the amount of ROB processing has been minimised. It may also be possible to do some preprocessing in the RODs. Two options are currently considered.

In a ROD preprocessing scenario, the calorimeter calibration and sums are done in the

calorimeter RODs, with the times as estimated in the table 7, and the preprocessing times in all the ROBs are zero. In a minimal ROB preprocessing scenario, the calorimeter ROBs do the preprocessing, with the same algorithms times.

In neither case is any preprocessing done for SCT, TRT or MUON ROB data. TRT preprocessing could reduce the data size by suppressing straws without hits, but the format from the ROD is already partly suppressed. In current SCT preprocessing, strip addresses are clustered and converted to global space points; this offers no significant data size reduction, so it can be done as an initial stage of feature extraction. Muon preprocessing is likewise unnecessary. An alternative option which has not yet been modelled is to format data ‘on the fly’ in FPGAs [14], as it is sent between ROBs and processors.

Some further details of ROD data formatting and ROB preprocessing algorithms are given in [7].

4.5 ROB to Switch Interface (RSI)

The RSI has two functions:

1. receive fragments of RoI data from ROBs and merge;
2. receive a RoI data request and distribute it to the relevant ROBs.

Fragment merging is assumed to be done at 50 MByte/s. All messages received and sent incur an i/o overhead as described in section 5.2.

4.6 SFI/FEX

The Switch to Farm Interface (SFI) receives all the event fragments required for a single feature extraction step and merges them into a single fragment. Fragment merging rate and message overheads are the same as the RSI.

The local Feature Extraction (FEX) processor of architecture B is functionally equivalent to an SFI with the additional process of the FEX algorithm.

For pull models (architecture C) the SFI also receives requests for RoI data which must be forwarded to the relevant RSIs.

4.7 Feature extraction

The FEX algorithm times in table 8 were originally copied from [2]; some have since been updated. Unmeasured times (indicated by “???”) are estimated. Data formatting (DAF) times are included where known, otherwise they are assumed to be the same as the FEX time. Some of the quoted benchmarks were made on faster processors. In this case the measured times in the table have been scaled to correspond to a 100 MIPS processor. Some algorithm details are given in [15].

RoI type	sub-detector	Measured times (μs)		Extrapolated Time (μs) (total)	Reference
		FEX	DAF		
MU	MUON	220	???	100	[16]
MU/EM/TAU	CALO	100	200	100	[17, 18]
J	CALO	???	???	100	
ME	CALO	???	???	100	
MU	TRT	700	2970	590	[13, 19]
EM/TAU	TRT	700	1552	310	[13, 19]
scan	TRT	...	680000 ...	50000	[20]
MU	SCT	1500	650	500	
EM/TAU	SCT	1500	650	500	[21]
<i>b</i> -jet tag	SCT	???	???	250000	

Table 8: *Feature extraction algorithm times*

FEX algorithms run either on local processors (arch. B), FPGAs (arch. A) or in the global processor (arch. C). Sequential processing may allow smaller subsets of MU/EM/TAU RoI data to be processed in the TRT/SCT, which is not taken into account in the times in table 8.

Measured times in table 8 are extrapolated in the following way.

- The measured times given all correspond to a 100 MIPS processor. Where necessary, the results taken from the references have been normalised to 100 MIPS.
- The extrapolated execution time is then the total measured time divided by five to account for increased processor performance (100 to 500 MIPS).
- The times for TRT RoIs assume 10% occupancy. The expected occupancies for the TRT are 7% at low luminosity and 10% at high luminosity.
- The TRT full scan is reduced by an extra factor of three for optimisation of the current off-line code.
- The *b*-jet algorithm is reduced by an extra factor of two for anticipated optimisation of off-line code.

4.8 FPGA feature extraction

The algorithm times in table 9 are taken from [4]. They are based on one *enable++* [4, 14] board which contains 24 Xilinx FPGA processors. In the context of hybrid architectures, the FPGA algorithms for the full TRT scan and *b*-jet tag would just do track finding. About six *enable++* boards are needed to scan the whole TRT.

RoI type	sub-detector	Measured time (μs)
MU	MUON	10
EM/TAU	CALO	10
J	CALO	10
ME	CALO	10
EM/TAU/MU	TRT	9
EM/TAU/MU	SCT	8
full scan (low lumi)	TRT	250
<i>b</i> -jet tag	SCT	??

Table 9: *Times of feature execution algorithms implemented on FPGAs.*

5 Hardware Models

These are chosen to be as generic as possible. Obviously computer modelling requires technology specific models and numbers, but the base line for comparison of paper models should not be technology specific.

5.1 Networks & Links

The behaviour of a generic switch and link are defined by the following parameters.

Network bandwidth = 10 MByte/s/link.

Network transfer setup time, $T_0 = 100 \mu\text{s}$, suggested range 10 – 100 μs .

Network transfer time as a function of data size,

$$T_{net} = T_0 + \frac{\text{message size}}{\text{bandwidth}}$$

Simple congestion could be added by reducing the available bandwidth. With the paper models it is preferred to work the other way round and state what network bandwidth would be required. Then the network can be specified which will be able to deliver this bandwidth.

The latency for sending multiple messages with the same source or destination is the sum of all the message latencies. As a baseline it is assumed that broadcast and multicast are not possible with a generic switch, since they are not available with all switch technologies. However, these may be used in models when they are required by a trigger design, in which case it will be explicitly stated.

5.2 Processors

A "processor" is actually taken to be a board on which a CPU and i/o processor are mounted, along with memory, etc.

The elapsed time to send/receive a message is $T_{i/o}$. Thanks to the dedicated i/o processor, the CPU is only occupied for some fraction of this time.

More detailed models of specific technologies will be implemented in computer modelling; the current paper models are based on the following simplified description.

CPU time taken to send or receive a message is $\alpha T_{i/o}$. This includes a context switch per message, which may be the worst case.

$$\Rightarrow \text{total } T_{i/o} = \text{no. of messages} \times \alpha T_0$$

Here, α is the fraction of network setup time for which the CPU is occupied; suggested range for α is 0.1 – 1.0; default $\alpha = 0.5$. This results in a default of $T_{i/o} = 50 \mu\text{s}$ per message. NB Initial modelling results show that $T_{i/o}$ is a critical parameter.

An operating system overhead (for interrupts, monitoring, etc) is guesstimated to be 10% of the total processing time.

It is assumed that one message is one packet for networks, links, etc. Modelling of packetisation is too detailed for paper models.

The supervisor (SUP), global trigger processors (GTP) and local processors (where applicable) are considered to be farms of general purpose processors. Hence when paper models calculate the occupancy of these farms, this indicates the number of processors required to work at 100% occupancy.

5.3 RSI and SFI

These components act as concentrators between the ROBs and switch (RSI) and farm and switch (SFI) in architecture C. Their task is to combine data fragments to send across the switch, and fan out data received via the switch. They are modelled as CPUs with the same value of αT_0 as the processors described in section 5.2. This will be added to the load for each message received or sent. They reduce the number of messages received by the processor for a RoI, since the ROB fragments have been grouped by the RSIs and SFIs. The default configuration is to have one RSI per four ROBs and one SFI per four processors.

6 Trigger Menus

The LVL2 trigger is driven by the output from LVL1. This is modelled with trigger menus, which give the rates combinations of the RoIs expected from level one. They are estimated from physics simulations and fast simulations of the LVL1 trigger. As such, the trigger menus provide the physics input to the LVL2 paper model.

A menu item consists of a combination of types of LVL1 trigger object (RoIs) and the rate at which that combination is expected to arise. The total rate of each type of RoI gives the rates and occupancies in the trigger system. The patterns of RoI combinations are needed to give the average event latency. The notation for trigger objects comprises of a few letters indicating the object type (e.g. MU, J, EM), a number giving the threshold in GeV, and an optional 'T' indicating that isolation is required. The trigger objects delivered by the LVL1 trigger are listed in table 10.

There are currently two main variations in the trigger menus considered for the demon-

	Low luminosity	High Luminosity
Trigger RoIs		
μ	MU6, MU20	MU6+MU6, MU20
e/γ	EM15I+EM15I, EM20I, EM80	EM20I+EM20I, EM30I
lepton	MU6+EM15I	MU6+EM20I
τ	TAU80, TAU150	—
jet	J100, J200, J50+J50+J50	J150
\cancel{E}_T	ME100, ME150	ME100
Secondary RoIs		
μ	—	MU6
e/γ	EM7I, EM15I	EM10
τ	TAU40	—
jet	J15	J40

Table 10: *LVL1 trigger objects*

strator programme. Menus can be for high and low luminosity, and they can be either *minimal*, listing only LVL1 trigger RoIs, or *extended* to include secondary RoIs flagged by LVL1. This distinction is shown in table 10.

Low luminosity menus have lower thresholds. The minimal menu includes a full TRT scan at low luminosity. The extended menu also includes b -jet tags. Extended menus will clearly require some level of sequential processing. The minimal menu is derived from the ATLAS TP [22].

The menus are designed for a target rate of 40 kHz input to LVL2. However, LVL2 is required to cope with 100 kHz input rate, a safety factor of 2.5 being allowed for pp cross section uncertainties.

Menus are distinct from the processing strategy adopted to deal with them.

The full trigger menus are listed in appendix A.

7 Selection Strategy

7.1 Introduction

A selection strategy, while sometimes constrained by architecture choices, can generally be considered architecture independent. It can be limited by the trigger menu, if for example secondary RoIs are not available, as is the case for the minimal menu.

The selection strategy is closely linked to the processing strategy. If selection is parallel, it is desirable but not essential to process in this way too. The optimum processing strategy is usually dictated by the architecture. Hence for architecture B, it is recommended that parallel selection is implemented with parallel processing of all RoIs, and sequential selection

Arch.	Menu	Algorithms	Processing	Selection	LVL2 accept rate (kHz)	
					low	high luminosity
B	minimal	simple	parallel	parallel	1.5	1.4
B	extended	simple	parallel	parallel	1.0	?
B	extended	simple	sequential	sequential	1.0	?
B	extended	b -jet + E_T	sequential	sequential	0.1	?
C	extended	simple	sequential	parallel	1.0	?
C	extended	simple	sequential	sequential	1.0	?
C	extended	b -jet + E_T	sequential	sequential	0.1	?

Table 11: *Relevant combinations of trigger menus, processing, and selection. The LVL2 accept rates for the minimal menu come from the ATLAS technical proposal [22]; the extended menu accept rates are taken from [2].*

is implemented with parallel processing of the RoIs in each step, but that the steps themselves are processed sequentially. (In this context, processing refers to the time and resources taken to push/pull RoI data, data transfers, i/o and algorithms.) With architecture C, only complete sequential processing of RoIs is possible. However, all the RoI data for a given step can be requested in parallel and the RSIs will merge the ROB data fragments in parallel.

A further dimension to the selection strategy is the possibility of executing more complex algorithms on level 2 processors to reduce the bandwidth to the event filter (level 3). These algorithms — E_T recalculation and b -jet tagging — are much slower or make intensive use of the trigger system, so they can only be done on a small fraction of events. They are therefore only considered in a sequential selection model.

In table 11 the most relevant combinations for modelling are proposed. Implementations of the parallel and sequential selection strategies are suggested below. The table also gives the anticipated accept rate for each combination. Where these are unknown for high luminosity (indicated by “?”) it is suggested that the corresponding low luminosity rate is used.

7.2 Low luminosity

7.2.1 'TP' (for minimal menus or extended menus)

This is the strategy outlined in the ATLAS Technical Proposal [22]. Most RoIs are processed in parallel, except muons. For an event in which there is a muon RoI, the following sequential processing and selection procedure should be followed, while at the same time all the other RoIs are analysed in parallel.

1. Confirm muon RoI in muon detector;
2. confirm muon RoI in calorimeter and tracking (in parallel);
3. full TRT scan;
4. analyse LVL2 RoIs generated by TRT scan.

Global processing is then done sequentially.

7.2.2 'Sequential' (for extended menus)

This sequence is copied from [2].

1. Confirm LVL1 trigger using calorimeter and muon data from trigger RoIs;
2. verify non-jet triggers in inner tracking;
3. full TRT scan for confirmed muon trigger;
4. verify trigger muon isolation in calorimeter;
5. analyse non trigger RoIs (requesting all data in a single step);
6. recalculate \cancel{E}_T if required by the LVL2 trigger menu;
7. b -jet tags if required by the LVL2 trigger menu;
8. combine features for global selection criteria.

Note that verification of muon isolation (step 4) provides additional information about the muon RoI but doesn't give any further rejection.

7.3 High luminosity

7.3.1 'TP' (for minimal menus or extended menus)

All RoIs are processed in parallel.

7.3.2 'Sequential' (for extended menus)

Strategy of 7.2.2 without the TRT scan.

7.4 Accepted fractions

The selection strategies defined above are implemented in terms of messages and accepted fractions. In paper models, where the aim is just to calculate average loads, latencies, etc. the average time taken to do selection is computed from

$$T_{step,1} + a_1 \times T_{step,2} + a_2 \times T_{step,3} + \dots + a_{n-1} \times T_{step,n}$$

where a_i is the fraction of events accepted by the i^{th} step and $T_{Step,i}$ the computation time for the i^{th} , and n is the number of possible steps. The average loads on parts of the trigger system are calculated by the same principle.

Values of a are determined using simulated events and the ATLAS trigger simulation package ATRIG [27, 15]. Most of the numbers given here are guesstimates which will be updated

RoI type	FEX Algorithm	Fraction accepted
MU	MUON	0.75
	MUON + tracking	0.5
	MUON + CAL iso	0.5
EM (low lumi)	CAL	0.3
	CAL + tracking	0.04
EM (high lumi)	CAL	0.1
	CAL + tracking	0.017
J	CAL	0.5
TAU	CAL	0.05
	CAL + tracking	0.01
ME	ME	1.0

Table 12: *Accepted fractions of RoIs*

when new results become available. The overall accepted fraction of a FEX algorithm is taken at about 90% signal efficiency. It is important to realise that the events accepted by the trigger algorithms can be dominantly background events, so the nature of the signal events and the efficiency with which they are accepted are not relevant parameters for modelling the LVL2 trigger.

Rejection only occurs in the global decision algorithm, which is always run in the global processor. There are two levels of rejection.

1. RoIs can be rejected at the object decision stage; for example a muon RoI could be discounted on the basis of the muon detector feature alone, after which the inner detector feature extraction in the muon RoI would be unnecessary. This does not necessarily imply that the event will be rejected. The accepted fractions of RoIs after each feature extraction stage are given in table 12.
2. Events can be rejected at the event decision stage; for example an event flagged by LVL1 as EM20I + 2*J15 would be rejected if the EM20I RoI was not confirmed by LVL2. This would mean that the secondary RoIs (2*J15) would not be processed. Event rejection is not always so straightforward; for example an event found to be 2*EM20I + 2*J15 at LVL1 would still be a valid LVL2 trigger if one of one of the electrons was not confirmed, but 2*EM15I + 2*J15 would fail if either electron was rejected.

Generally, for events with a single trigger RoI, the accepted fraction before processing the

Message	Size (bytes)	Description
---------	--------------	-------------

ROIR	24	RoI requests sent from supervisor to RoI Distributor (B) or from processor to ROBs (C)
ROIRSF	36	send RoI records to GTP (for average 5 RoIs per event) (architecture C)
ROID	32	data sent from ROB to processor, header only; the full size includes the ROB data (see table 4)
FEXD	150	feature record sent from FEX to GTP (B only)
GPR1	20	event decision/continue from GTP to supervisor
GPR2	96	block of ROIR's produced in the FEX or GTP and sent to the supervisor for distribution to ROBs; this is the size of a message containing 20 RoIs, which corresponds to one event (B only)
T2DR	408	block of 100 event decisions sent from supervisor to all ROBs

Table 13: *Description of messages.*

secondary RoIs is given by the combined accepted fraction of the FEX algorithms for the type of trigger RoI. For events with several trigger RoIs, all the accepted fractions are combined. Assuming that the fractions of RoIs accepted are not correlated, they are combined in the following way for the two cases identified in the examples above, for a trigger menu item "A+B". Clearly there are also more complex situations.

Accept event if either RoI is confirmed:

$$a = 1 - (1 - a_A) \times (1 - a_B)$$

Accept event only if both RoIs are confirmed:

$$a = a_A \times a_B$$

8 Messages

The nature and size of messages in the system are described in [23], and a functional message passing model of some architectures is described in [24]. As far as possible, the messages in [23] are used for paper modelling.

A summary of the message sizes used in paper models is given in table 13.

9 Some notes on special triggers

9.1 LVL2 Missing E_T

Missing energy recalculation (E_T) at LVL2 is not in the minimal trigger menu. It places a heavy load on the calorimeter ROBS and network because all 480 ROBS (40 kBytes of data) must be read out and into one processor at 2 kHz. It also causes a bottleneck in the network and subsequently a long latency for events with a LVL1 E_T trigger.

Since the latency for ME events is dominated by network transfer time associated with the full calorimeter readout, it can be calculated approximately as

$$\begin{aligned} \text{Latency} &= (\#\text{ROBs} \times T_0) + \left(\frac{\text{datasize}}{\text{bandwidth}} \right) \\ &= (480 \times 100 \mu\text{s}) + \left(\frac{40 \text{ kByte}}{10 \text{ MByte/s}} \right) \\ &\approx 52000 \mu\text{s} \end{aligned}$$

The setup time dominates. This is because the network transfers are all to the same processor so they must be done sequentially.

In an architecture with local processing, the latency (but not the load) can be reduced by dividing the algorithm between several processors. An optimal value is around 16 – 32 processors. This is at the expense of a small increase to the latency in the global network by an additional T_0 per processor used. We chose $N = 16$ for modelling.

$$\begin{aligned} \text{Latency} &= \left(\frac{480 \times 100 \mu\text{s}}{N} \right) + \frac{1}{N} \times \frac{40 \text{ kByte}}{10 \text{ MByte/s}} + (N \times 100 \mu\text{s}) \\ &\approx 4850 \mu\text{s} \end{aligned}$$

This configuration is used for the architecture B paper model.

9.2 TRT scan

Preliminary paper model calculations show that the full TRT scan places a huge load on the TRT ROBS, network and processors. Assuming that it is done sequentially after confirmation of a LVL1 muon trigger, the algorithm will be called at 4 kHz with an extrapolated execution time of 50 ms. The TRT scan has so far only been timed using off-line reconstruction code. Various ways have been found to optimise this (see [25]).

The resulting event latency is dominated by this algorithm time. It can be reduced by splitting the algorithm into 16 processes which are executed in parallel in the local farm (applicable to arch B) [25] or by using an FPGA to do very fast initial track finding (applicable to architectures A,B,C) [26].

The TRT scan is usually used to generate new LVL2 RoIs. The SCT data in these regions will be analysed in many cases. The number of SCT RoIs required depends very much on the order of sequential selection. The FPGA fast initial tracking generates on average 64 RoIs, which if non-overlapping would correspond to the entire SCT (@ 4 ROBS/RoI). If the TRT data is analysed sufficiently for some of the B triggers to be done without the SCT, the average number of SCT RoIs could be only 7 (for $B_s^0 \rightarrow D_s^- \pi^+ \rightarrow K^+ K^- \pi^+ \pi^-$). The average number of LVL2 RoIs found in $B \rightarrow \mu X$ events from the "offline" TRT algorithms is about 20, which is the recommended number to use for modelling.

Studies of various options for the B physics trigger continue.

9.3 *b*-jet tag

For the *b*-jet tag, solutions similar to the TRT scan are being investigated [28]. Fast initial track finding using either the TRT or the SCT, reading out the whole detector or several jet RoIs, are possible ways of decreasing the time taken. Precision track finding using the SCT + pixels to accurately reconstruct the impact parameter is required. A method based on impact parameter tagging that just uses the pixels is also being studied [29].

Algorithm development and performance measurement is underway. The method and execution time for on-line style code need to be known before this trigger can be properly represented in models. It is currently assumed that a single global algorithm will be run regardless of the number of jet RoIs that are candidate *b*-jets, and it will work on full SCT data.

10 Acknowledgements

The authors wish to thank the ATLAS trigger community for all the work which has been collected in this note.

References

- [1] R. Blair et al, Options for the ATLAS level-2 trigger, paper given by R. J. Hubbard at CHEP97, to be published in Comp Phys Comm
<http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/APAPERS/HUBBARD970217.ps>
- [2] J. Bystricky et al, A Model for Sequential Processing in the ATLAS LVL2/LVL3 Trigger (Demo C paper model), DAQ-NO-55
<http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/NOTES/note55/daq55.ps.Z>
- [3] J. Bystricky et al, A Sequential Processing Strategy for the ATLAS Event Selection, DAQ-NO-59
<http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/NOTES/note59/daq59.ps.Z>
- [4] V. Dorsing et al, Demo A paper model
http://www-mp.informatik.uni-mannheim.de/groups/mass_par_1/projects/demo_A.ps

- [5] M. Dobson et al, Demo B paper model
http://www.hep.ph.rhbnc.ac.uk/atlas/paper_model_note.ps
http://www.hep.ph.rhbnc.ac.uk/atlas/talks/papmod_demob_24jan97/00.html
- [6] D. Calvet et al, Performance requirements of proposed ATLAS second level trigger architectures from simple models, paper given by S. George at CHEP97, to be published in Comp Phys Comm
<http://www.hep.ph.rhbnc.ac.uk/atlas/talks/chep97/index.html>
- [7] R. Bock & P. Le Du, DAQ-NO-62
<http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/NOTES/note62/daq62.ps.Z>
- [8] V. Charlton, Optimising the grouping of SCT wafers into readout buffers using simulated annealing methods, DAQ-NO-57
<http://atlasinfo.cern.ch/Atlas/documentation/notes/DAQTRIG/note57/SCTreport.ps.Z>
- [9] Jos Vermeulen, Calculation of no. of ROBs per RoI
<http://www.nikhef.nl/www/pub/experiments/atlas/daq/ROBsperRoI.html>
- [10] Jos Vermeulen, Calculation of no. of RSIs per RoI
<http://www.nikhef.nl/www/pub/experiments/atlas/daq/RSIPerRoI.txt>
- [11] Jos Vermeulen, LVL2 Process Models
<http://www.nikhef.nl/www/pub/experiments/atlas/daq/process.html>
- [12] R. Blair et al, ATLAS Supervisor for 2nd LHC workshop
<http://sgi3.hep.anl.gov:8001/hungary.html>
- [13] R. Hauser & I Legrand, Algorithms in second level triggers for ATLAS and benchmark results, DAQ-NO-27
<http://www.cern.ch/RD11/eastnotes/EAST94-37-LocalAlg.ps.Z>
- [14] Y. Gal et al, A FPGA-based Coprocessor proposed for preprocessing in ATLAS, paper given at CHEP97, to be published in Comp Phys Comm. URL not available, but see also:
<http://www.cern.ch/RD11/eastnotes/EAST94-35-Enablepp.ps.Z>
http://www-mp.informatik.uni-mannheim.de/groups/mass_par_1/projects/enable.html
- [15] Trigger Performance Working Group, Trigger Algorithms in ATRIG Version 1.30, Dec 96, DAQ-NO-60. URL not available.
- [16] S. Falciano et al, the ATLAS Muon trigger algorithm for Level-2 Feature Extraction, ATLAS DAQ note in preparation ???
- [17] R. Bock & B. Kastrup, Realistic Calorimeter Feature extraction: Algorithm, Benchmarks and Implementation Options, DAQ-NO-65.
<http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/NOTES/note65/fexbenchs.ps.gz>
- [18] B. Thoris, private communication.
- [19] R. Bock, K.-H. Noffz, M. Sessler, TRT Preprocessing — algorithm, Implementations and Benchmarks, DAQ-NO-66.
<http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/NOTES/note66/daqnote.ps.Z>

- [20] M. Smizanska, Second Level TRT trigger for B-physics, PHYS-NO-89
<http://atlasinfo.cern.ch/Atlas/GROUPS/PHYSICS/NOTES/note89/phy89.ps.Z>
- [21] S. Sivoklov, R. Dankers, J. Baines, Second Level Triggering in the Forward Region of the ATLAS Inner Tracker, INDET-NO-111
http://atlasinfo.cern.ch/Atlas/GROUPS/INNER_DETECTOR/NOTES/note111/t2ms.ps.Z
(soon to be superseded by recent SCT trigger performance studies)
- [22] ATLAS Technical Proposal CERN/LHCC-94-33
<ftp://www.cern.ch/pub/Atlas/TP/tp.html>
- [23] F Wickens, Proposed Data Records + Formats for Demonstrators
http://hepwww.rl.ac.uk/atlas/l2/demonstrator/docs/demons_data_defs.html
- [24] N. Madsen & S. Hunt, Models of Message Flow in the Atlas Level-2 Trigger
http://wwwftp.cern.ch/~madsenn/messageflow_1.ps
- [25] M. Smizanska, The LVL2 TRT algorithm optimisation
<http://wwwcn.cern.ch/msmizans/algorithm/0.html>
- [26] M. Smizanska, Histogramming method with ENABLE++ versus off-line initial tracking in LVL2 global scan
<http://wwwcn.cern.ch/msmizans/algorithm/02.html>
- [27] ATRIG ATLAS Trigger Simulation Program
<http://www.cern.ch/RD11/physics/atrig.html>
- [28] S. George & A. Belias, B-jet tagging algorithms for the ATLAS second level trigger
http://www.hep.ph.rhbnc.ac.uk/atlas/talks/t2bjet_saclay/00.html
- [29] L. Rossi, in minutes of Level2 Trigger Plenary Meeting, 4 March 1997
- [30] J. Bystricky et al, ATLAS Trigger Menus at Luminosity $10^{33} \text{ cm}^{-2}\text{s}^{-1}$ DAQ-NO-54
<http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/NOTES/note54/daq54.ps.Z>
- [31] S. George, ATLAS Trigger Menus at Luminosity $10^{34} \text{ cm}^{-2}\text{s}^{-1}$, draft document in preparation

A Trigger Menus

The trigger menus are used to simulate the LVL1 RoI rates and their distribution in events. Different menus are used for high ($10^{34} \text{ cm}^{-2}\text{s}^{-1}$) and low ($10^{33} \text{ cm}^{-2}\text{s}^{-1}$) luminosity running. In each case, two menus are shown. The minimal menus are for a LVL2 trigger which ignores secondary (non trigger) RoIs and just confirms the LVL1 trigger RoIs. The extended menus introduce secondary RoI processing.

A.1 Low Luminosity

A.1.1 Minimal menu

		Inclusive	Exclusive (Hz)
LVL1	MU6	8000	6700
LVL1.5	MU6 + MU6	400	380
LVL1.5	MU6 + MU6 + MU6	20	20
LVL1.5	MU6 + EM15I	800	760
LVL1.5	MU6 + EM15I + EM15I	40	40
LVL1.5	MU6 + TAU80	100	-
LVL1.5	MU6 + J100	100	80
LVL1.5	MU6 + J100 + J100	20	20
Tot exclusive			8000
LVL1	EM20I	10000	9700
LVL1.5	EM20I + TAU80	300	-
LVL1.5	EM20I + J100	300	200
LVL1.5	EM20I + J100 + J100	100	100
Tot exclusive			10000
LVL1	EM15I + EM15I	2500	2400
LVL1.5	EM15I + EM15I + TAU80	100	-
LVL1.5	EM15I + EM15I + J100	100	70
LVL1.5	EM15I + EM15I + J100 + J100	30	30
Tot exclusive			2500
LVL1	TAU80	5000	-
LVL1.5	TAU80 + TAU80	800	-
LVL1.5	TAU80 + J100	5000	4870
LVL1.5	TAU80 + J100 + J100	130	130
Tot exclusive			5000
LVL1	J100	8000	6000
LVL1.5	J100 + J100	2000	1800
LVL1.5	J100 + J100 + J100	200	180
LVL1.5	J100 + J100 + J100 + J100	20	20

Tot exclusive		8000
Total LVL1 rate	33500 Hz	
Total Exclusive rate		33500 Hz

A.1.2 Extended menu

		Inclusive	Exclusive(Hz)
LVL1	MU6	8000	160
LVL1.5	MU20	400	-
LVL1.5	MU20 + MU6	100	80
LVL1.5	MU20 + MU20	20	20
LVL1.5	MU20 + EM15I	40	40
LVL1.5	MU20 + TAU40	400	400
LVL1.5	MU20 + J15 + J15 + J15 + J15 + J15 + J15	75	75
LVL1.5	MU20 + J15 + J15 + J15 + J15 + J15	100	25
LVL1.5	MU20 + J15 + J15 + J15 + J15	150	50
LVL1.5	MU20 + J15 + J15 + J15	200	50
LVL1.5	MU20 + J15 + J15	300	60
LVL1.5	MU20 + J15	400	100
LVL1.5	MU20 + J40 + J40	40	40
LVL1.5	MU6 + EM15I	800	740
LVL1.5	MU6 + MU6	400	360
LVL1.5	MU6 + MU6 + MU6	20	20
LVL1.5	MU6 + MU6 + EM15I	40	40
LVL1.5	MU6 + EM15I + EM15I	40	40
LVL1.5	MU6 + J15 + J15 + J15 + J15 + J15 + J15	1500	1425
LVL1.5	MU6 + J15 + J15 + J15 + J15 + J15	2000	475
LVL1.5	MU6 + J15 + J15 + J15 + J15	3000	950
LVL1.5	MU6 + J15 + J15 + J15	4000	950
LVL1.5	MU6 + J15 + J15	6000	1900
Tot exclusive			8000
LVL1	EM80	200	170
LVL1.5	EM80 + EM80	30	30
Tot exclusive			200
LVL1	EM20I	10000	-
LVL1.5	EM20I + TAU40	3000	900
LVL1.5	EM20I + J15 + J15 + J15 + J15 + J15 + J15	1800	950
LVL1.5	EM20I + J15 + J15 + J15 + J15 + J15	2500	350
LVL1.5	EM20I + J15 + J15 + J15 + J15	3500	550
LVL1.5	EM20I + J15 + J15 + J15	5000	800
LVL1.5	EM20I + J15 + J15	7000	1200
LVL1.5	EM20I + J15	10000	2100
LVL1.5	EM20I + J40 + J40 + J40	200	200
LVL1.5	EM20I + J40 + J40 + J15 + J15 + J15 + J15	350	350
LVL1.5	EM20I + J40 + J40 + J15 + J15 + J15	500	150
LVL1.5	EM20I + J40 + J40 + J15 + J15	700	200
LVL1.5	EM20I + J100 + J100	250	250
Tot exclusive			8000

LVL1.5	EM20I + TAU40 +J15 +J15 +J15 +J15 +J15 +J15	540	450
LVL1.5	EM20I + TAU40 + J15 + J15 + J15 + J15 + J15	750	200
LVL1.5	EM20I + TAU40 + J15 + J15 + J15 + J15	1000	250
LVL1.5	EM20I + TAU40 + J15 + J15 + J15	1500	500
LVL1.5	EM20I + TAU40 + J15 + J15	2100	600
Tot exclusive			2000
LVL1	EM15I + EM15I	2500	1500
LVL1.5	EM15I + EM15I + EM7I	1000	1000
Tot exclusive			2500
LVL1	TAU80	5000	3360
LVL1.5	TAU150	1000	840
LVL1.5	TAU80 + TAU80	800	640
LVL1.5	TAU150 + TAU80	160	160
Tot exclusive			5000
LVL1	J100	8000	2750
LVL1.5	J200	250	250
LVL1.5	J100 + J100	2000	1500
LVL1.5	J100 + J100 + ME100	300	300
LVL1.5	J100 + J100 + J100	200	180
LVL1.5	J100 + J100 + J100 + J100	20	20
LVL1.5	J100 + J15 + J15 + J15 + J15 + J15	2000	1500
LVL1.5	J100 + J15 + J15 + J15 + J15	3000	750
LVL1.5	J100 + J15 + J15 + J15	4000	750
Tot exclusive			8000
LVL1	J50 + J50 + J50	3000	2400
LVL1.5	J50 + J50 + J50 + J50	600	500
LVL1.5	J50 + J50 + J50 + J50 + J50	100	80
LVL1.5	J50 + J50 + J50 + J50 + J50 + J50	20	20
Tot exclusive			3000
LVL1	ME100	2000	1970
LVL1.5	ME150	30	30
Tot exclusive			2000
Total LVL1 rate		38700	Hz
Total Exclusive rate			38700 Hz

A.2 High Luminosity

A.2.1 Minimal menu

		Inclusive	Exclusive (Hz)
LVL1	MU6 + EM20I	3000	3000
Tot exclusive			3000
LVL1	MU6 + MU6	1000	-
LVL1.5	MU6 + MU6 + MU6	200	200
LVL1.5	MU6 + MU6 + EM20I + EM20I	50	50
LVL1.5	MU6 + MU6 + J150	1000	500
LVL1.5	MU6 + MU6 + J150 + J150	300	250
Tot exclusive			1000
LVL1	MU20	4000	1680
LVL1.5	MU20 + EM30I	300	300
LVL1.5	MU20 + EM20I + EM20I	20	20
LVL1.5	MU20 + J150	2000	1400
LVL1.5	MU20 + J150 + J150	600	600
Tot exclusive			4000
LVL1	EM30I	20000	-
LVL1.5	EM30I + J150	20000	19100
LVL1.5	EM30I + J150 + J150	900	900
Tot exclusive			20000
LVL1	EM20I + EM20I	4000	-
LVL1.5	EM20I + EM20I + J150	4000	3400
LVL1.5	EM20I + EM20I + J150 + J150	600	600
Tot exclusive			4000
LVL1	J150	3000	2100
LVL1.5	J150 + J150	900	830
LVL1.5	J150 + J150 + J150	70	70
Tot exclusive			3000
Total LVL1 rate		35000 Hz	
Total LVL1.5 exclusive rate			35000 Hz

A.2.2 Extended menu

		Inclusive	Exclusive (Hz)
LVL1	MU6 + EM20I	3000	3000
Tot exclusive			3000
LVL1	MU6 + MU6	1000	-
LVL1.5	MU6 + MU6 + J40	1000	328
LVL1.5	MU6 + MU6 + J40 + J40	222	188
LVL1.5	MU6 + MU6 + J40 + J40 + J40	34	34
LVL1.5	MU6 + MU6 + EM20I	300	200
LVL1.5	MU6 + MU6 + EM20I + EM20I	50	50
LVL1.5	MU6 + MU6 + MU6	200	200
Tot exclusive			1000
LVL1	MU20	4000	-
LVL1.5	MU20 + J40	4000	2811
LVL1.5	MU20 + J40 + J40	889	752
LVL1.5	MU20 + J40 + J40 + J40	137	121
LVL1.5	MU20 + J40 + J40 + J40 + J40	16	16
LVL1.5	MU20 + EM20I	300	280
LVL1.5	MU20 + EM20I + EM20I	20	20
Tot exclusive			4000
LVL1	EM30I	20000	-
LVL1.5	EM30I + J40	20000	12223
LVL1.5	EM30I + J40 + J40	4444	3760
LVL1.5	EM30I + J40 + J40 + J40	684	604
LVL1.5	EM30I + J40 + J40 + J40 + J40	80	80
LVL1.5	EM30I + EM10	3333	3333
Tot exclusive			20000
LVL1	EM20I + EM20I	4000	-
LVL1.5	EM20I + EM20I + J40	4000	1511
LVL1.5	EM20I + EM20I + J40 + J40	889	752
LVL1.5	EM20I + EM20I + J40 + J40 + J40	137	137
LVL1.5	EM20I + EM20I + EM10	1600	1600
Tot exclusive			4000
LVL1	J150	3000	-
LVL1.5	J150 + J40	3000	2183
LVL1.5	J150 + J40 + J40	667	565
LVL1.5	J150 + J40 + J40 + J40	102	102
Tot exclusive			2850
LVL1	ME100	1000	850

LVL1.5 ME100 + J150	150	150
Tot exclusive		1000
LVL1 misc prescaled	5000	-
LVL1.5 J40 + J40	5000	4227
LVL1.5 J40 + J40 + J40	773	682
LVL1.5 J40 + J40 + J40 + J40	91	91
Tot exclusive		5000
Total LVL1 rate	40850 Hz	
Total LVL1.5 exclusive rate		40850 Hz