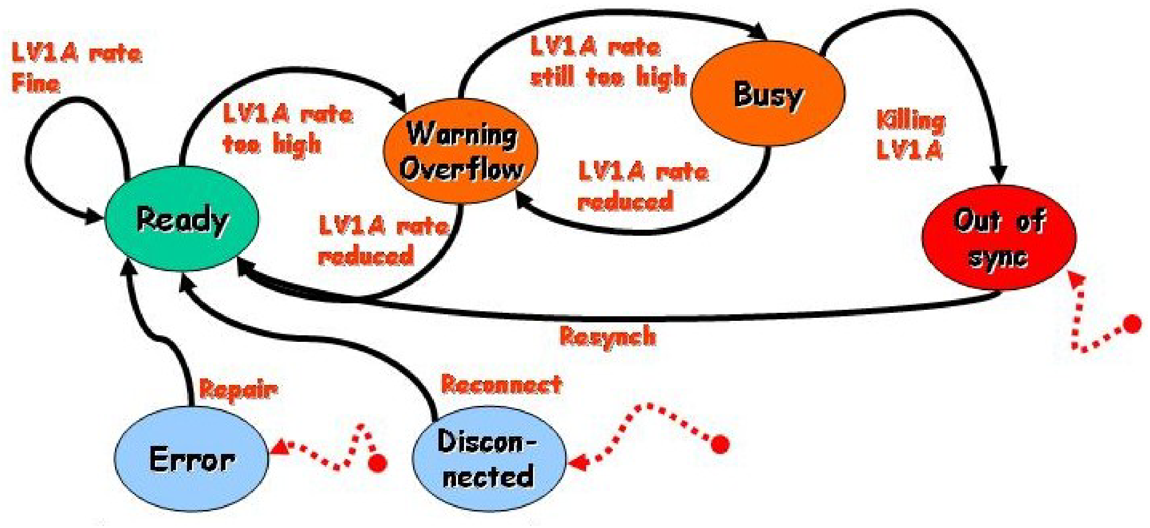
More details are in CMS-NOTE 2003/033.

The FED can be in 6 different states (Ready, Warning Overflow, Busy, Out-of-Synch, Error, Disconnected). The transitions between the different states expected from the DAQ are shown in the following picture:



List of ROB/ROS Warnigs/Errors in the data flow

WARNING OVERFLOW	**Error word**	**# sources**	**affected channels**	**Action taken by DDU**	**activation bit (if =1)**	**Channel blocked**
1) INPUT FIFO Warning overflow	0xDF(2)	1500	128	TTS logic based on the counts of errors (type 1)	14	NO
2) L1A FIFO Warning overflow	0xDF(7)	60	3200	TTS logic based on the counts of errors (type 2)	14	NO

OUT OF SYNCH	**Error word**	**# sources**	**affected channels**	**Action taken by DDU**	**activation bit (if =1)**	**Channels Blocked**
HPTDC Error	0xC	6000	32	None	-	NO
3) Link timeout Error	0xDF(0)	1500	128	Resync if n.err>N (n.err=3+4+5+6)	14	YES/NO (programmable, default = YES)
4) Event ID misalignment Error	0xDF(1)	1500	128	Resync if n.err>N (n.err=3+4+5+6)	14	YES/NO (programmable, default = YES)
5) FIFO Full Error	0xDF(3)	1500	128	Resync if n.err>N (n.err=3+4+5+6)	14	YES/NO (programmable, default = YES)
6) CEROS timeout Error	0xDF(4)	240	768	Resync if n.err>N (n.err=3+4+5+6)	14	YES/NO (programmable, default = YES)
7) Max number words Error	0xDF(5)	1500	128	Resync if n.err>N (n.err=3+4+5+6)	14	YES/NO (programmable, default = YES)
L1A FIFO Full Error	Event trailer and 0xDF(6)	60	3200	None	-	NO
Transmitter Parity Error	Event trailer	60	3200	None	-	NO

OUT OF SYNCH

Errors that will produce a transition from Ready to Out-of-Synch:

3) Link timeout error (300 sources/DDU) (the ROB is not sending data or there has been an unlock)
4) Event ID misalignment (300 sources/DDU)
5) FIFO full (300 sources/DDU)
6) CEROS timed-out (48 sources/DDU)
7) Max.number of words (300 sources/DDU)

Activation flags: bit 14 in ROS error word

Counter (**N_ERR(3)**) with 8 bits (max 256 errors) with the number of received error words.
Incremented by 1 for all errors except CEROS timed-out
Incremented by 6 for CEROS timed-out error
Reset after a Resync, never decremented

TTS logic:
Out-of-Synch if **N_ERR(3)>2^N3**
Where N3 is a 3 bits register. (0 to 7)
Possible thresholds: 1, 2, 4, 8, 16, 32, 64, 128.

5 status bits **ST_3,ST_4,ST_5,ST_6,ST_7** (1 for each error type)

Counter with the number of Resynch requests (**N_STAT(3))** common to all errors

OK

****

-We will have another counter (N_ERR(3)) per DDU that will be incremented each time you see one of our errors for resynch (link timeout, fifo full, etc). ok

- This is a unique counter for all these types of errors. ok

- This counter will be incremented each time you see one of these errors with our bit 14 to 1 in the error word and by 6 if the error is Ceros timeout. (It can be incremented more than once per event). yes

-The counter has 8 bits, that is, it counts up to 256 errors. There are 300 sources per DDU. yes

-There will be a programmable threshold in the DDU for this counter, and the DDU will ask for a resynch when this threshold is reached. yes

-After the resynch all the errors are erased (the counter N_ERR(3) is resetted) and you start from Ready status. yes

- When an EC0 is received, all the error flags in the registers will be deleted, the FIFOs will be reseted and all the channels will be enable again, recovering all the channels with problems from the error status and enabling them to continue the run.
You mean this is the ROS/ROB behaviour ? I agree if EC0 only follows a Resynch. I don't know if EC0 can be sent for other reasons. In the DDU we reset the N_ERR(3) counter after a Resynch, we don't use EC0 (in the DDU it just rest the Event Counter in the TTCrx). I imagine you use EC0 as a Resynch command.

OK

Doubts:
I understand the N_ERR(3) counter does not overflow…
Are the status bits ST_3,… reseted after a resynch? I understand not.

This counter can not overflow. We go in Out-of-Synch before, and we reset it after a Resynch.
In case bit 14 in the ROS error word is 0, we don't increment it.
I would reset these status bits only by request (not after a Resynch). Not easy, but Giulio will try.

Debugging

In order to understand why the system went into out of synch:

- Reading the data or with the DQM (slow) Only a fraction of events will be saved on disk. We should use registers to understand the reason for a Out-of-Synch (at least on a statistical bases) No, Zanetti said data integrity will be done also at 100khz

- Reading the DDU status registers (if they are not reseted after a resynch) Statistics and some status registers should not be resetted if we want to understand the reason of a Out-of-Synch..
- Masking different types of errors until we know which one caused the out of synch (slow, only for early stages). Only for debugging.
- Try to implement a register in the ROS that latches all the errors and it is not reseted by a resynch. (Still to see, difficult). It's difficult also for a DDU. From the DDU registers you will not undertand which ROS/ROB gives a problem. In case you ant to understand which part of the detector gives a problem, it's necessary to read ROS registers.

WARNING OVERFLOW/BUSY

ROS L1A FIFO

- There is a programmable threshold at the L1A FIFO occupancy at the ROS (ROS_threshold), when this threshold is reached, all the events that will be processed afterwards, will include the L1A Warning overflow error word. ok

- When the number of L1As if below the threshold, the following events will not include the error word. If the error word has bit 14 to 1, the DDU has to perform an action. we increment a counter, we don’t take an action immediatly.

- We will have one counter (N_ERR(2))at each DDU for the ROS warning L1A fifo overflow that will be incremented by 1 in each event that you see one error of this kind from the ROS (whenever we have the bit 14 to 1). yes

- It will be an OR of all the ROS, it will be incremented only once per event, no matter if there is only one ROS with errors or many. yes, but we can discuss.

- There are four thresholds programmable with registers N2 and N_HIST, when we go over the first one (READY->WO) the DDU will go to Warning overflow, and when we go over the second one (WO->Busy), the DDU will go to busy. yes
- Each time you receive an event without any error, you will decrement the counter. yes
- When you go below the Busy->WO threshold you will go from Busy to Warning overflow and when you go below the WO->Ready threshold, you will go back to Ready. yes

Ready->WO = > 2^N2+2^N_HIST
WO->Ready = < 2^N2-2^N_HIST
WO->Busy = > 2*2^N2+2^N_HIST
Busy->WO = < 2*2^N2-2^N_HIST

Maybe it's better to use the following thresholds

Ready->WO	**> 2^N2+(2^N_HIST-1)**
WO->Ready	**< 2^N2-(2^N_HIST-1)**
WO->Busy	**> 2*2^N2+(2^N_HIST-1)**
Busy->WO	**< 2*2^N2-(2^N_HIST-1)**

In this way N_HIST=0 means NO HYSTERESIS.

- We don´t need to reset anything here because the system will go to ready by itself. yes

-N_ERR(2) has 16 bits. (de 0 a 65535) Probably it does not need to be so big (ROS L1A fifo is 256 words deep, but fine.
16 bits is less than 1 second at LHC, but I agree N_ERR(2) needs much less.
Reseted after a resynch.
N2 is 3 bits
N_HIST is 3 bits

Example:
N2=0 N_HIST=0
Minimum lower threshold = 2^N2+2^N_HIST = 2 (to go from Ready to warning overflow)
To go back from Warning overflow to ready counter < 0 I guess <=0
With the new proposal (see above)
NO Ready->WO threshold is 1 if N2=0 N_HIST=0
N_ERR(2)>1 -> Ready->WO
N_ERR(2)<1 -> WO->Ready
Minimum upper threshold = 2*2^N2+2^N_HIST = 3 (to go from warning overflow to busy)
To go back from busy to warning overflow counter < 1
With the new proposal
WO->Busy threshold = 2
N_ERR(2)>2 -> WO->Ready
N_ERR(2)<2 -> Ready->WO

-How to guarantee that we will not get stuck in Busy, a possibility: (also read below)
Once the DDU is in Busy status, it should not increment the N_ERR(2) counter any more. Otherwise, if the L1As take a time to stop, the ROS can still send at least (255-ROS_threshold) events with the error and you may never get out of Busy. Yes, it's one possibility. Maybe the simpler one, but the transition BUSY->WO could be too fast.

Once we are in busy, the L1As will stop, the ROS will still send events from the L1As it has stored. At some point, the occupancy of the L1A FIFO will go below the ROS_threshold and at least we will send a number of events equal to ROS_threshold that will not have the L1A warning error.
Therefore, the thresholds have to be programmed in such a way that verify:

ROS_threshold > WO->Busy – Busy->WO =>

ROS_threshold > 2*2^N_HIST

Yes, I agree also on this possibility.

OLD PROPOSAL

N2	N_HIST	Ready->WO	WO -> Busy	Busy -> WO	WO -> Ready	ROS_THR>
0	**0**	2	3	1	0	2
1	**0**	3	5	3	1	2
1	**1**	4	6	2	0	4
2	**0**	5	9	7	3	2
2	**1**	6	10	6	2	4
2	**2**	8	12	4	0	8
3	**0**	9	17	15	7	2
3	**1**	10	18	14	6	4
3	**2**	12	20	12	4	8
3	**3**	16	24	8	0	16

4	0	17	33	31	15	2
4	1	18	34	30	14	4
4	2	20	36	28	12	8
4	3	24	40	24	8	16
4	4	32	48	16	0	32
5	0	33	65	63	31	2
5	1	34	66	62	30	4
5	2	36	68	60	28	8
5	3	40	72	56	24	16
5	4	48	80	48	16	32
5	5	64	96	32	0	64
6	0	65	129	127	63	2
6	1	66	130	126	62	4
6	2	68	132	124	60	8
6	3	72	136	120	56	16
6	4	80	144	112	48	32
6	5	96	160	96	32	64
6	6	128	192	64	0	128
7	0	129	257	255	127	2
7	1	130	258	254	126	4
7	2	132	260	252	124	8
7	3	136	264	248	120	16
7	4	144	272	240	112	32
7	5	160	288	224	96	64
7	6	192	320	192	64	128
7	7	256	384	128	0	256

NEW PROPOSAL

N2	N_HIST	Ready-WO	WO-Busy	Busy-WO	WO-Ready	ROS_THR
		N_ERR(3)>	N_ERR(3)>	N_ERR(3)<	N_ERR(3)<	
0	0	1	2	2	1	2
1	0	2	4	4	2	2
1	1	3	5	3	1	4
2	0	4	8	8	4	2
2	1	5	9	7	3	4
2	2	7	11	5	1	8
3	0	8	16	16	8	2
3	1	9	17	15	7	4
3	2	11	19	13	5	8
3	3	15	23	9	1	16
4	0	16	32	32	16	2
4	1	17	33	31	15	4
4	2	19	35	29	13	8
4	3	23	39	25	9	16
4	4	31	47	17	1	32
5	0	32	64	64	32	2
5	1	33	65	63	31	4
5	2	35	67	61	29	8
5	3	39	71	57	25	16
5	4	47	79	49	17	32

5	**5**	63	95	33	1	64
6	**0**	64	128	128	64	2
6	**1**	65	129	127	63	4
6	**2**	67	131	125	61	8
6	**3**	71	135	121	57	16
6	**4**	79	143	113	49	32
6	**5**	95	159	97	33	64
6	**6**	127	191	65	1	128
7	**0**	128	256	256	128	2
7	**1**	129	257	255	127	4
7	**2**	131	259	253	125	8
7	**3**	135	263	249	121	16
7	**4**	143	271	241	113	32
7	**5**	159	287	225	97	64
7	**6**	191	319	193	65	128
7	**7**	255	383	129	1	256

3 status bits - **ST_2** - (1 bit=error received +1 bit = warning-overflow +1 bit = busy)

Counters with the number of Warning (**N_STAT(5)**) and Busy requests (**N_STAT(6)**) (common to FIFO almost full, readable by request)

<u>**ROS INPUT FIFO**</u>

-When the occupancy in the ROS input fifos goes over a programmable threshold, an error word of warning overflow (PAF) is sent within the data flow. ok

-When we are below this PAF threshold, we don´t send this word any more. ok

-The DDU will have a counter (N_ERR(1)) for the ROB warning fifo full (PAF).
-This counter will be incremented in each event only if the number of PAF words is bigger or equal than a programmable value (N1_0). (and bit 14 is 1) ok
- The counter will be incremented only once per event. ok
- The counter will be decremented each event when the number of error words in the event is <N1_0 (I think that if we force it to be =0 and we have some fifos (N1_0) with errors we will always be in warning overflow).
Yes, they decrement the counter when the number of errors is below a threshold. (Not when the number of errors is 0).
We can fix N1_0>1. Anyway, if we have noise problems, some of the FIFOs should fill up and we don't see their warnings anymore.

N1_0 is 3 bits. If 0 to 7 ROBs send PAF, the counter will be incremented.
I see this N1_0 a little too small, it is about 1 minicrate. If one minicrate is always noisy (for example) we may not want to increase the counter all the time (At least during debugging phase).
4 bits is enough ?

Vincenzo proposes to make it 2^N1, reviser.

-N_ERR(1) counter is 16 bits (0 to 65535)
Reseted after a resynch.

-N1_1 is 4 bits (0 to 15).
Ready->WO = 2^N1_1+2^N_HIST
(Like with L1A WO maybe it's better to set the threshold as 2^N1_1+(2^N_HIST-1) (N_HIST=0 means no hysteresis)
Maximum threshold for Ready->WO = 32896

I doubt we can accept so many L1As with PAF flag on. I don´t think this counter needs to be so big, neither N1_1, but it is up to you, we can always program a small value in the thresholds.

It's better to have a big counter than a small one.

Yes, there is a risk of getting stuck in busy, for example if the last event processed is bigger than our Almost Full limit in the input fifo, you will receive a PAF error word and you will not be able to get out of Busy.
However, I think it is still important to have the possibility to go to Busy, otherwise it is very likely that we block the input fifos for fifo full and we have to ask for a resynch. Maybe it could be programmable?
Ok. We could add the option to disable the WO->Busy transition.

Other Options:
-Ask Christoph if after being in busy for more than XX time they automatically go to warning overflow=>No, I already asked him and it seems that they don´t do anything, that the systems should be the ones that have a mechanism to not get stuck into busy.
-I don´t know if you have anything already implemented for your busy errors that can be common to us. For example, that when you don´t have any more L1As pending in your L1A FIFO you get out of Busy, or after a fixed time, or after receiving XX bunch resets or after an event reset, etc.
Christoph also commented that first of all, we have to be careful with the Warning Overflow because at the Magnet test in that status they totally stopped the L1As. We have to talk to Joao Varela to make sure that this does not happen.
He also said that maybe our Out of synch errors look more like Error status, but I guess that if we can get out fine from the errors with a resynch then it should be fine to operate like that. (We have to confirm it with other people).

To go out of a Busy state when the DDU L1A FIFO is empty seems reasonable to avoid to stuck in that state. Maybe it's the best solution for our doubts.
Of course after an EC0 we could go in Ready (I think EC0 only follows a Resynch command). We can implement sucha mechanism for a WO state during the Magnet Test.
Vincenzo agrees on that.

Debbuging

- Status registers in the ROS and DDU, these bits will not be reseted so can be read through VME after the busy status has been flaged. (not easy, we will see if possible) ok no problem
- Enabling, disabling send PAF words or L1AWO words. or set bits 13,14 = 0.

I assume that if you are in Warning Overflow or Busy status and our L1A fifo goes full (and therefore you will get an event id misalignment), you can go directly to Out-of-Synch, right? In such a case the flags at the ROS will be deleted (in principle). But I understand that your status bits will not be deleted, right?

They should not be deleted, otherwise we cannot understand the origin of the Out-of-Synch transition.

As far as Christoph told me the transitions from one state to the other are pretty fast, maybe in the order of 20 bxs (25 bxs) until the Central Trigger reduces the L1A rate or sends the resynch. So the bottleneck for a fast reaction is in our system (the time for the ROS to process an event (300 bxs), fibers (12 bxs) and for the DDU to process the event and send the TTS status).

Yes.

ec0 is guaranteed in each resynch?
ec0 is only sent at the beggining of the run and in a rsynch or somewhere else? can it be a problem?