FRC Pronouncements

H.Evans **Columbia U.**

Hardware

1. J3 Backplane

Data Formats & Timing

1. T/R Data
2. L3 Data
3. Monitoring Data

Protocols

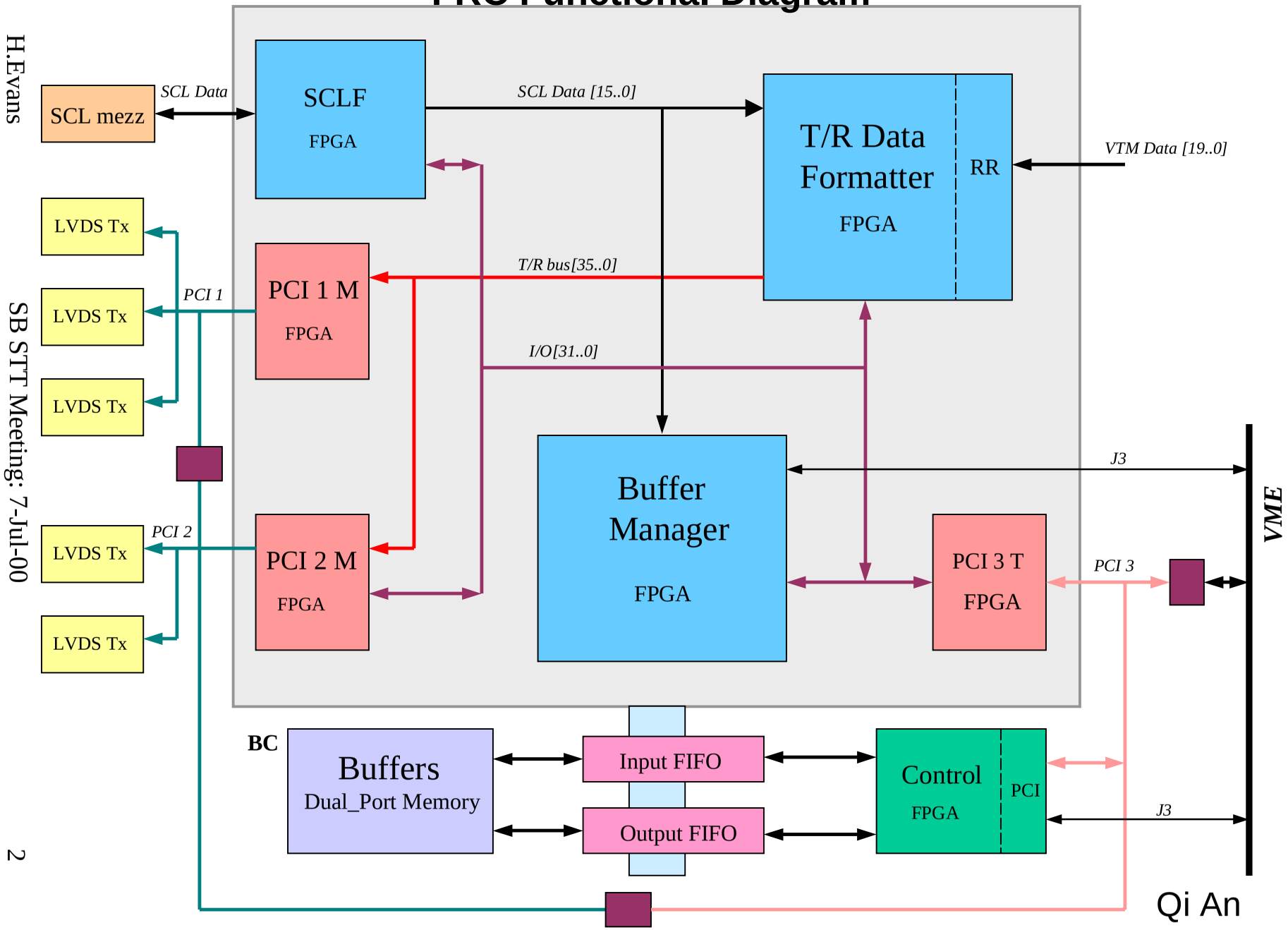
1. VME Arbitration
2. Errors & Busy
3. SCL Init
4. Monitoring
5. BM – BC Communications

Status & Plans

The Columbia/Nevis Team

- **Physicists:** Tulika Bose, Hal Evans, Georg Steinbrück
- **Engineers:** Qi An, Bill Sippach

FRC Functional Diagram



H.Evans

SB STT Meeting: 7-Jul-00

2

Qi An

J3 Backplane

- **Crate Layout**
  - **Where should we put the J3 terminator?**
- **J3 Pinout (mod's from VRB flavor)**
  - **All in row C**
  - **Use two VRB-reserved bussed lines as extra status lines: *stat*[10-11]**
  - **Status lines now TTL/LVTTL totem pole**
    - **were open-collector**
  - **Move VBD comm. Lines (*srdy**, *done**)**
    - **from C45-C47 ⇒ C42-C44**
  - ***All J3 pins checked for compat. w/ VTM***
    - ***using rev B. J3 (1/11/00) & VRB (10/22/99)***
- **Open Questions**
  - **Ground pins ⇒ Feedthroughs ?**

Crate Layout

Slot	Module	Trans. Mod.
1	CPU	—
2-3	unused	
4	CPU	—
5	Terminator ?	
6	unused	
7	Terminator ?	
8	TFC	—
9-12	STC	VTM
13	FRC	VTM
14-18	STC	VTM
19	TFC	—
20	unused	
21	Terminator	

Bussed Lines on J3

		B'plane	Signal	Direction
J3 Pins	**Signal Name**	**Op**	**Type**	**BM – BC**
C1-C8	FRC_MESSAGE[0-7]		TTL	⇒
C9-C12	FRC_COMMAND[0-3]		TTL	⇒
C13	FRC_STROBE		TTL	⇒
C14	PUT_DONE	AND	TTL #	⇐
C15	PUT_DONE*	OR	TTL #	⇐
C16	GET_DONE	AND	TTL #	⇐
C17	GET_DONE*	OR	TTL #	⇐
C18	L1_BUSY*	OR	TTL #	⇐
C19	L2_BUSY*	OR	TTL #	⇐
C20	L1_ERROR*	OR	TTL #	⇐
C21	L2_ERROR*	OR	TTL #	⇐
C22-C23	free		TTL #	⇐
C24-C25	free (reserved in VRB)		TTL #	⇐

Signal is a TTL input to the FRC. Other modules should provide as a totem-pole TTL/LVTTL output.

J3 Pinout: STC/TFC Slots

Pin	Row A	Row B	Row C	Row D	Row E
1	L0_D0	GND	MESSAGE[0]	GND	L2_D0
2	L0_D1	L0_D2	MESSAGE[1]	L2_D2	L2_D1
3	GND	L0_D3	MESSAGE[2]	L2_D3	GND
4	L0_D5	L0_D4	MESSAGE[3]	L2_D4	L2_D5
5	L0_D6	GND	MESSAGE[4]	GND	L2_D6
6	L0_D7	L0_D8	MESSAGE[5]	L2_D8	L2_D7
7	GND	L0_D9	MESSAGE[6]	L2_D9	GND
8	L0_D11	L0_D10	MESSAGE[7]	L2_D10	L2_D11
9	L0_D12	GND	COMMAND[0]	GND	L2_D12
10	L0_D13	L0_D14	COMMAND[1]	L2_D14	L2_D13
11	GND	L0_D15	COMMAND[2]	L2_D15	GND
12	L0_D16	L0_D16	COMMAND[3]	L2_D16	L2_D16
13	L0_D17	GND	STROBE	GND	L2_D17
14	L0_D17	L0_D18	PUT_DONE	L2_D18	L2_D17
15	GND	L0_D18	PUT_DONE*	L2_D18	GND
16	L0_D19	L0_D19	GET_DONE	L2_D19	L2_D19
17	L0_CAV*	GND	GET_DONE*	GND	L2_CAV*
18	L0_DAV*	L0_LNKRDY*	SCL_L1BSY*	L2_LNKRDY*	L2_DAV*
19	GND	L0_LNKRDY*	SCL_L2BSY*	L2_LNKRDY*	GND
20	L0_STRBOUT	GND	SCL_L1ERR*	GND	L2_STRBOUT
21	L0_SIG_DETECT	L0_ERROR	SCL_L2ERR*	L2_ERROR	L2_SIG_DETECT
22	GND	L0_ERROR	status[8]	L2_ERROR	GND
23	GND	GND	status[9]	GND	GND
24	GND	GND	status[10]	GND	GND
25	GND	GND	status[11]	GND	GND
26	L1_CAV*	GND	GND	GND	L3_CAV*
27	L1_DAV*	L1_LNKRDY*	GND	L3_LNKRDY*	L3_DAV*
28	GND	L1_LNKRDY*	GND	L3_LNKRDY*	GND
29	L1_STRBOUT	GND	GND	GND	L3_STRBOUT
30	L1_SIG_DETECT	L1_ERROR	GND	L3_ERROR	L3_SIG_DETECT
31	GND	L1_ERROR	GND	L3_ERROR	GND
32	L1_D0	GND	n.c.	GND	L3_D0
33	L1_D1	L1_D2	n.c.	L3_D2	L3_D1
34	GND	L1_D3	n.c.	L3_D3	GND
35	L1_D5	L1_D4	n.c.	L3_D4	L3_D5
36	L1_D6	GND	GND	GND	L3_D6
37	L1_D7	L1_D8	GND	L3_D8	L3_D7
38	GND	L1_D9	GND	L3_D9	GND
39	L1_D11	L1_D10	GND	L3_D10	L3_D11
40	L1_D12	GND	GND	GND	L3_D12
41	L1_D13	L1_D14	n.c.	L3_D14	L3_D13
42	GND	L1_D15	n.c.	L3_D15	GND
43	L1_D16	L1_D16	n.c.	L3_D16	L3_D16
44	L1_D17	GND	n.c.	GND	L3_D17
45	L1_D17	L1_D18	SERIAL	L3_D18	L3_D17
46	GND	L1_D18	RESET*	L3_D18	GND
47	L1_D19	L1_D19	MODID	L3_D19	L3_D19

J3 Pinout: FRC Slot

Pin	Row A	Row B	Row C	Row D	Row E
1	L0_D0	GND	MESSAGE[0]	GND	L2_D0
2	L0_D1	L0_D2	MESSAGE[1]	L2_D2	L2_D1
3	GND	L0_D3	MESSAGE[2]	L2_D3	GND
4	L0_D5	L0_D4	MESSAGE[3]	L2_D4	L2_D5
5	L0_D6	GND	MESSAGE[4]	GND	L2_D6
6	L0_D7	L0_D8	MESSAGE[5]	L2_D8	L2_D7
7	GND	L0_D9	MESSAGE[6]	L2_D9	GND
8	L0_D11	L0_D10	MESSAGE[7]	L2_D10	L2_D11
9	L0_D12	GND	COMMAND[0]	GND	L2_D12
10	L0_D13	L0_D14	COMMAND[1]	L2_D14	L2_D13
11	GND	L0_D15	COMMAND[2]	L2_D15	GND
12	L0_D16	L0_D16	COMMAND[3]	L2_D16	L2_D16
13	L0_D17	GND	STROBE	GND	L2_D17
14	L0_D17	L0_D18	PUT_DONE	L2_D18	L2_D17
15	GND	L0_D18	PUT_DONE*	L2_D18	GND
16	L0_D19	L0_D19	GET_DONE	L2_D19	L2_D19
17	L0_CAV*	GND	GET_DONE*	GND	L2_CAV*
18	L0_DAV*	L0_LNKRDY*	SCL_L1BSY*	L2_LNKRDY*	L2_DAV*
19	GND	L0_LNKRDY*	SCL_L2BSY*	L2_LNKRDY*	GND
20	L0_STRBOUT	GND	SCL_L1ERR*	GND	L2_STRBOUT
21	L0_SIG_DETECT	L0_ERROR	SCL_L2ERR*	L2_ERROR	L2_SIG_DETECT
22	GND	L0_ERROR	status[8]	L2_ERROR	GND
23	GND	GND	status[9]	GND	GND
24	GND	GND	status[10]	GND	GND
25	GND	GND	status[11]	GND	GND
26	L1_CAV*	GND	GND	GND	L3_CAV*
27	L1_DAV*	L1_LNKRDY*	GND	L3_LNKRDY*	L3_DAV*
28	GND	L1_LNKRDY*	GND	L3_LNKRDY*	GND
29	L1_STRBOUT	GND	GND	GND	L3_STRBOUT
30	L1_SIG_DETECT	L1_ERROR	GND	L3_ERROR	L3_SIG_DETECT
31	GND	L1_ERROR	GND	L3_ERROR	GND
32	L1_D0	GND	n.c.	GND	L3_D0
33	L1_D1	L1_D2	n.c.	L3_D2	L3_D1
34	GND	L1_D3	n.c.	L3_D3	GND
35	L1_D5	L1_D4	n.c.	L3_D4	L3_D5
36	L1_D6	GND	GND	GND	L3_D6
37	L1_D7	L1_D8	GND	L3_D8	L3_D7
38	GND	L1_D9	GND	L3_D9	GND
39	L1_D11	L1_D10	GND	L3_D10	L3_D11
40	L1_D12	GND	GND	GND	L3_D12
41	L1_D13	L1_D14	n.c.	L3_D14	L3_D13
42	GND	L1_D15	SRDY (4-C08)	L3_D15	GND
43	L1_D16	L1_D16	res (4-C09)	L3_D16	L3_D16
44	L1_D17	GND	DONE (4-C10)	GND	L3_D17
45	L1_D17	L1_D18	SERIAL	L3_D18	L3_D17
46	GND	L1_D18	RESET*	L3_D18	GND
47	L1_D19	L1_D19	MODID	L3_D19	L3_D19

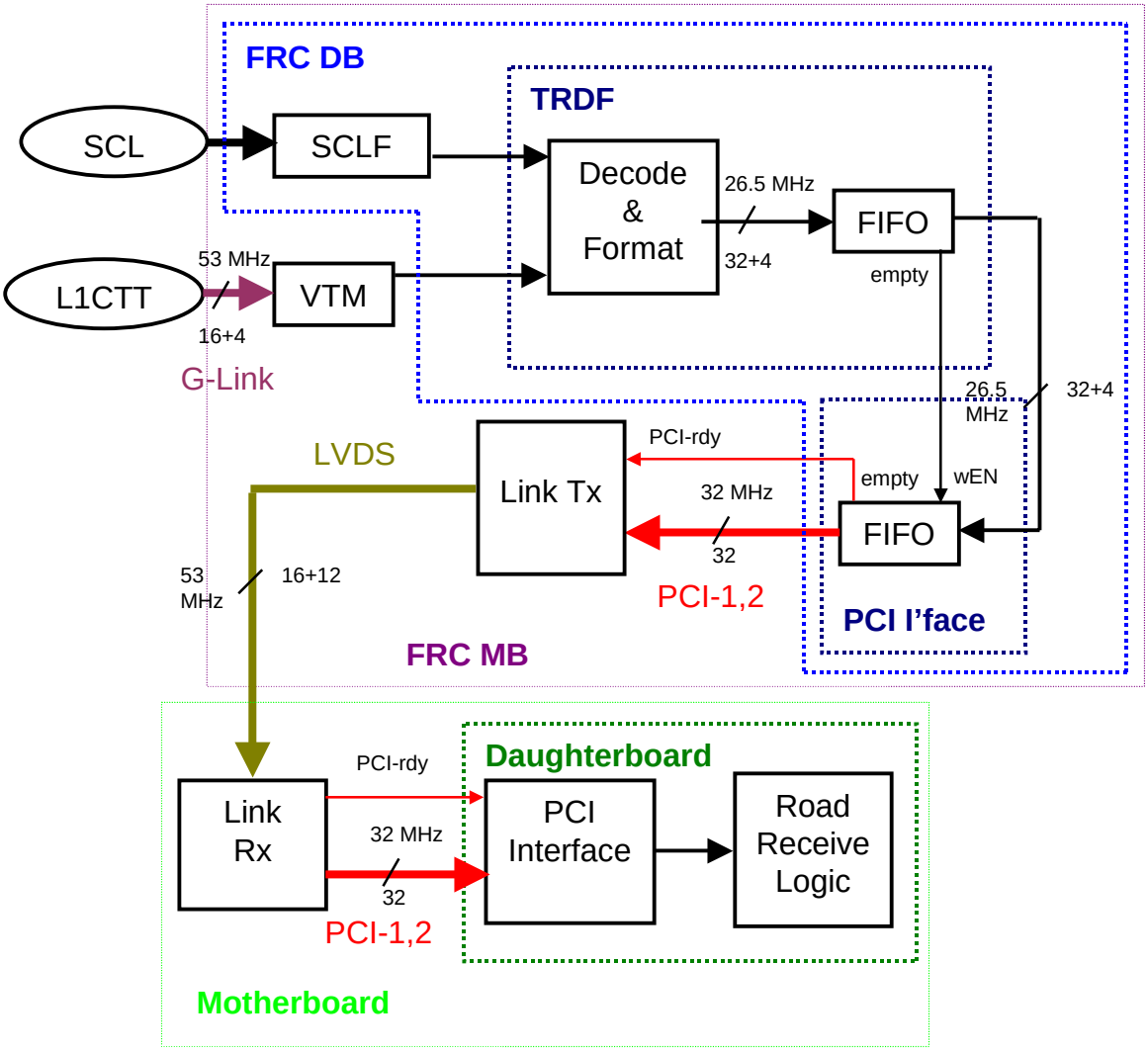
T/R Data Format

31 24	23 16	15 8	7 0
L1_QUAL		reserved LTB	L1_BX
Head Length	No. Objects	H/Obj Format	Object Length
L1_BX	Data Type	L1_TURN	
Algo min ver	Algo max ver	Status Bits	Firmware ver
P N Ntrk-Pt2	P N Ntrk-Pt1	P N Ntrk-Pt4	P N Ntrk-Pt3
S Ext/Pt H/L	Err R P	Rel Φ P	D Sector Addr
S Ext/Pt H/L	Err R P	Rel Φ P	D Sector Addr
S Ext/Pt H/L	Err R P	Rel Φ P	D Sector Addr
Data Type	L1_BX	Longitudinal Parity	
Pad	Pad	Pad	Pad
Pad	Pad	Pad	Pad
Pad	Pad	Pad	Pad
free		T/R Errors	L1_BX

- **New Information**
  - L1CTT data padded to multiples of 4 words
    - ⇒ Min T/R Data Size: 10 words
    - ⇒ Max T/R Data Size: 54 words
- **Open Question: Endianess**

VTM-Frame	VTM-Byte		T/R Byte
1	2 (15-08)	⇒	3 (31-24)
1	1 (07-00)	⇒	2 (23-16)
2	2 (15-08)	⇒	1 (15-08)
2	1 (07-00)	⇒	0 (07-00)

T/R Data Transfer



T/R Transfer Times

Latencies

L1CTT	2.90
VTM	?
TRDF	>0.08
FRC PCI	0.25
Link Tx	?
Link RX	?
STC/TFC PCI	0.25
Total after L1 Accept	>3.5 μs

Arrival Times

	Data Size	1st Word	All Words
Normal			
min	10	3.5 μs	3.9 μs
ave	10	3.5 μs	3.9 μs
max	54	3.5 μs	5.5 μs
No BoE	2	*var*	*var*
No EoE	*var*	3.5 μs	5.5 μs

VME Arbitration by BM

- **VME Uses**
  1. **Download & Initialization**
  2. **Testing**
  3. **Readout of L3 Data to VBD**
  4. **Readout of Monitoring Data**
  5. **SCL Init**
- **Annoyance: VBD Hogs the Bus**
  - **Use an Arbitration Scheme where BM only passes requests to Tasks that use VME when the bus is free**
  - **BM must know who is using the bus**
- **Arbitration Scheme in BM**
  - *i. Grant(scl_init) = Req(scl_init) & !(srdy* || mon_req)*
  - *ii. Grant(monitor) = Reg(monitor) & !(srdy* || scl_req)*
  - *iii. Grant(vbd) = Req(vbd) & !(scl_req || mon_req)*

L3 Data Format

31 24	23 16	15 8	7 0
unused	*unused*	Source *	BX
Data			
....			
Data			
Total Word Count (?) †			
32-bit Checksum (?)			
unused *	*unused* *	*unused* *	Errors *

* Added by BC

† Modified by BC

- **Tortuous Communication Path**

BM – BC	J3 msg & status	Control of Storage/Readout
BM	Arbitration	Advertise Data Ready to VBD
DB	L3 Data Buffer	Storage while waiting PCI-3
DB – BC	PCI-3 + spare-*i*	Block Xfer in Disconnect Mode
BC	BC DP Memory	L1 Store while waiting L2
BC – VBD	Output FIFO	Buffer for L3 hangups
	PCI-3 – Univ.II	Interface to VME
	VME	VBD hogs the bus

BM – BC Communications

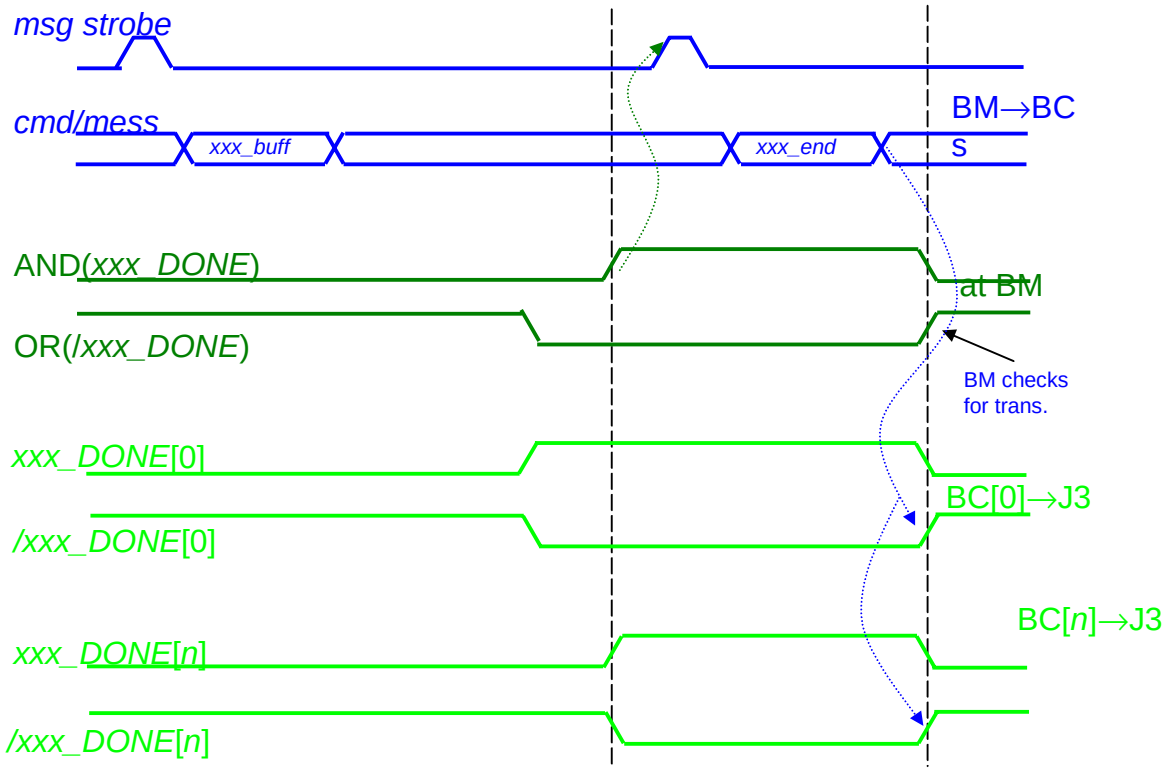
Messages

cmd [0-3]	*mess [0-7]*	L1/ L2	Purpose
0			free
1	*PUT_BUFF*	L1	Number of next buffer in BC Mem to put data from DB
2	*PUT_BX*	L1	BX for cross-check with data put to BC Mem
3	*PUT_END*	L1	All BCs done putting data from DB to BC Mem
4	*GET_BUFF*	L2	Number of next buffer in BC Mem to get data
5	*GET_BX*	L2	BX for cross-check with data got from BC Mem
6	*GET_END*	L2	All BCs done getting data from BC Mem to Output fifo
7-15			free

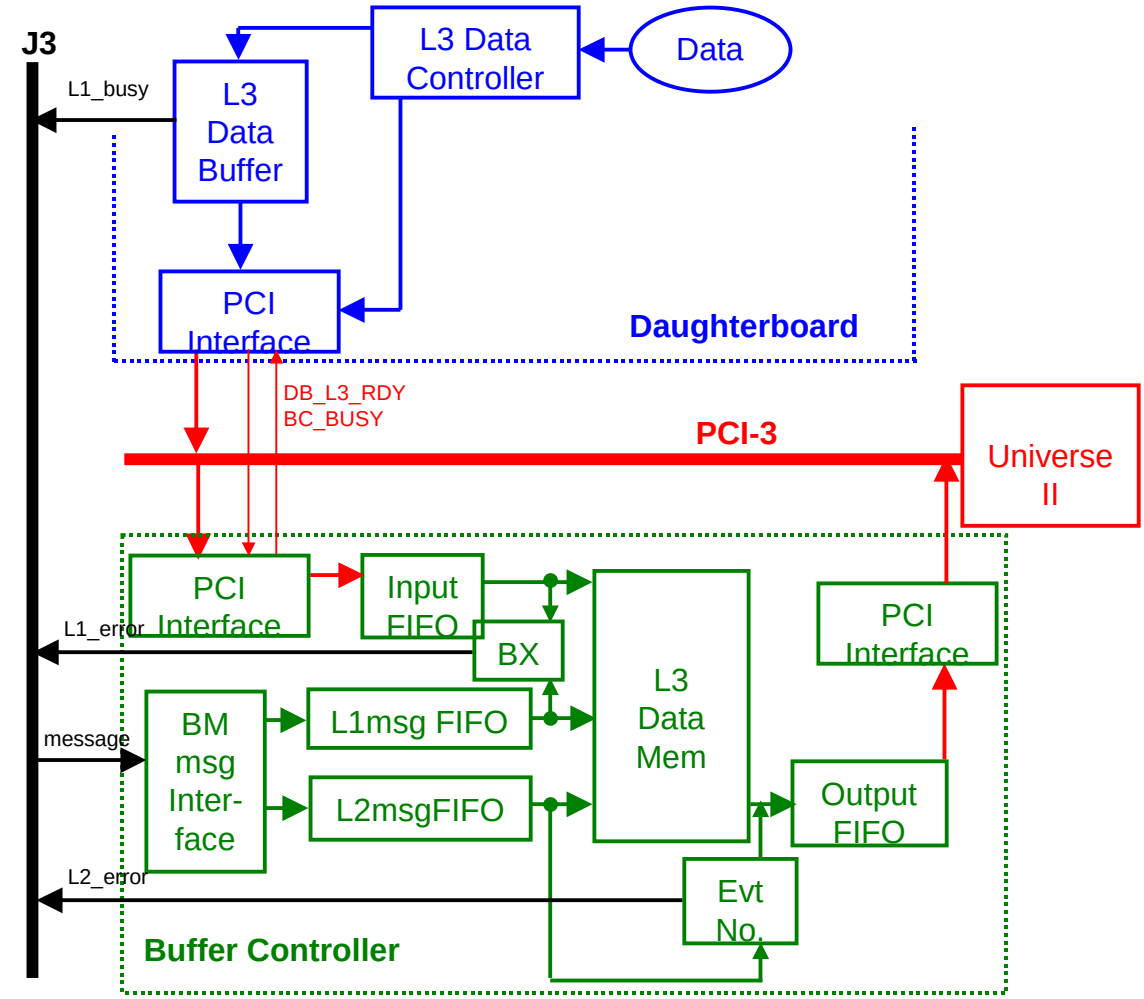
Status

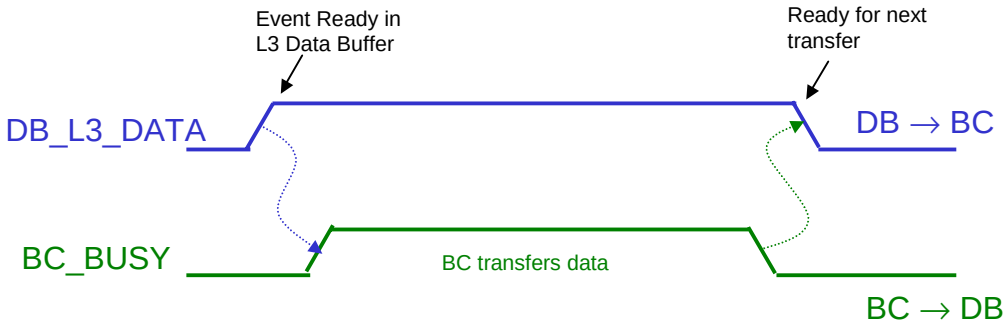
stat	Name	L1/ L2	Act	Purpose
0	*PUT_DONE*	L1	AND	Status BC putting data to Mem (see Fig. 1)
1	*PUT_DONE**	L1	OR	*put_done* inverted for wire OR
2	*GET_DONE*	L2	AND	Status of BC getting data from Mem
3	*GET_DONE**	L2	OR	*get_done* inverted for wire OR
4	*STT_L1BUSY**	L1	OR	L1 Busy (request holdoff of L1 Accepts)
5	*STT_L2BUSY**	L2	OR	L2 Busy (request holdoff of L2 Accepts)
6	*STT_L1ERROR**	L1	OR	L1 Error (request SCL_INIT)
7	*STT_L2ERROR**	L2	OR	L2 Error (request SCL_INIT)
8-12				free

BM – BC Timing

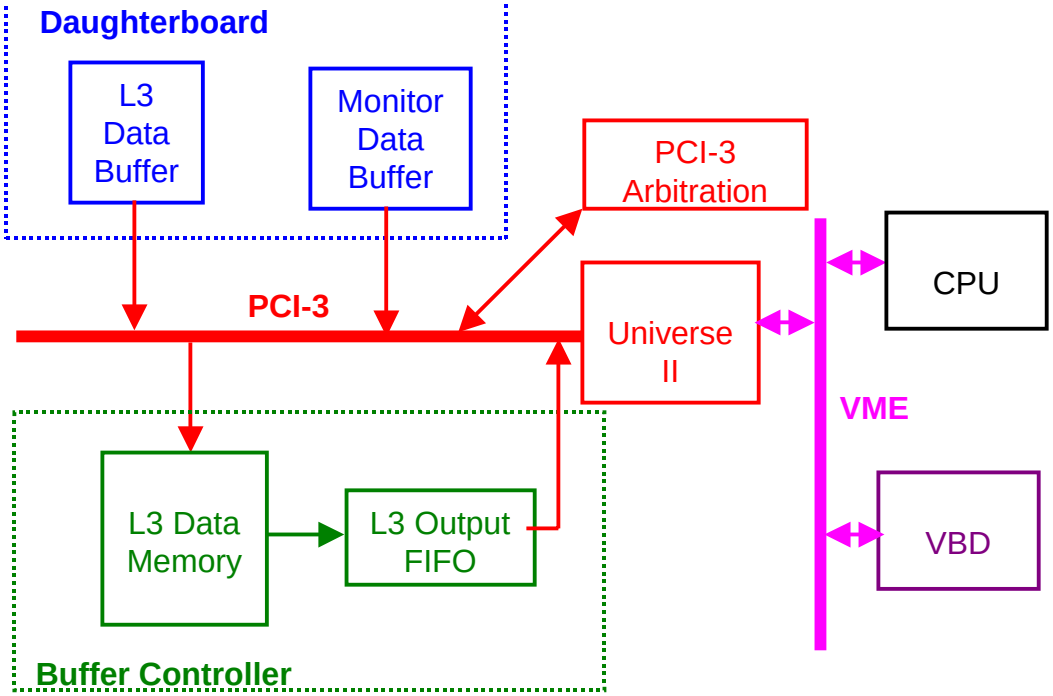

msg strobe
cmd/mess
xxx_buff
xxx_end
BM→BCS
AND(xxx_DONE)
at BM
OR(/xxx_DONE)
BM checks for trans.
xxx_DONE[0]
BC[0]→J3
/xxx_DONE[0]
BC[n]→J3
xxx_DONE[n]
/xxx_DONE[n]

BC – DB Communications





Data Buffer Sizes



- **Buffer Depths set (mainly) by:**
  1. **PCI-3 Xfer Times for other data**
  2. **Data Sizes**
  3. **Time b/w Actions that fill Buffer**

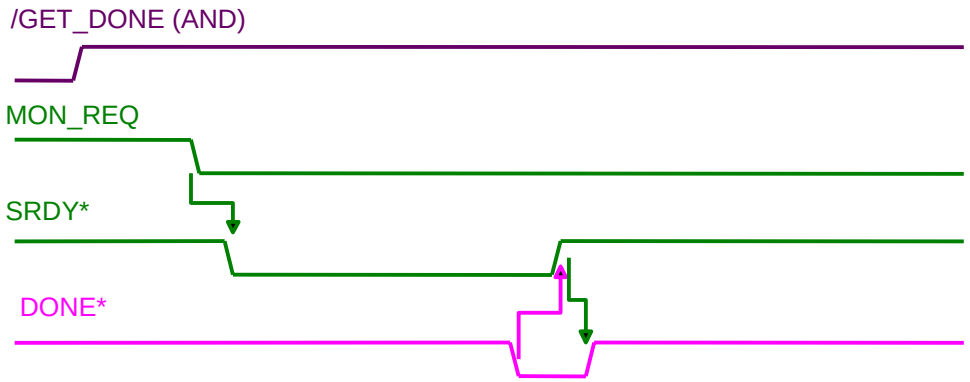
Buffer	Where	Data Size		Min. Depth
L3 Data	FRC	0 54	ave max	154
	STC	40 5847	ave max	9216
	TFC	6 6173	ave max	6217
L3 Out FIFO	BC	8 x 32k		256 k
Monitor	DB	?		1 evt

Busy Conditions (partial list)

- **Two Levels of Busy**
  1. **L1_BUSY** **stops L1 Accepts**
  2. **L2_BUSY** **stops L2 Accepts**
- **Implementation**
  - **L1/L2_BUSY are wire-OR status lines on J3**
  - **Must be set directly by element requesting**

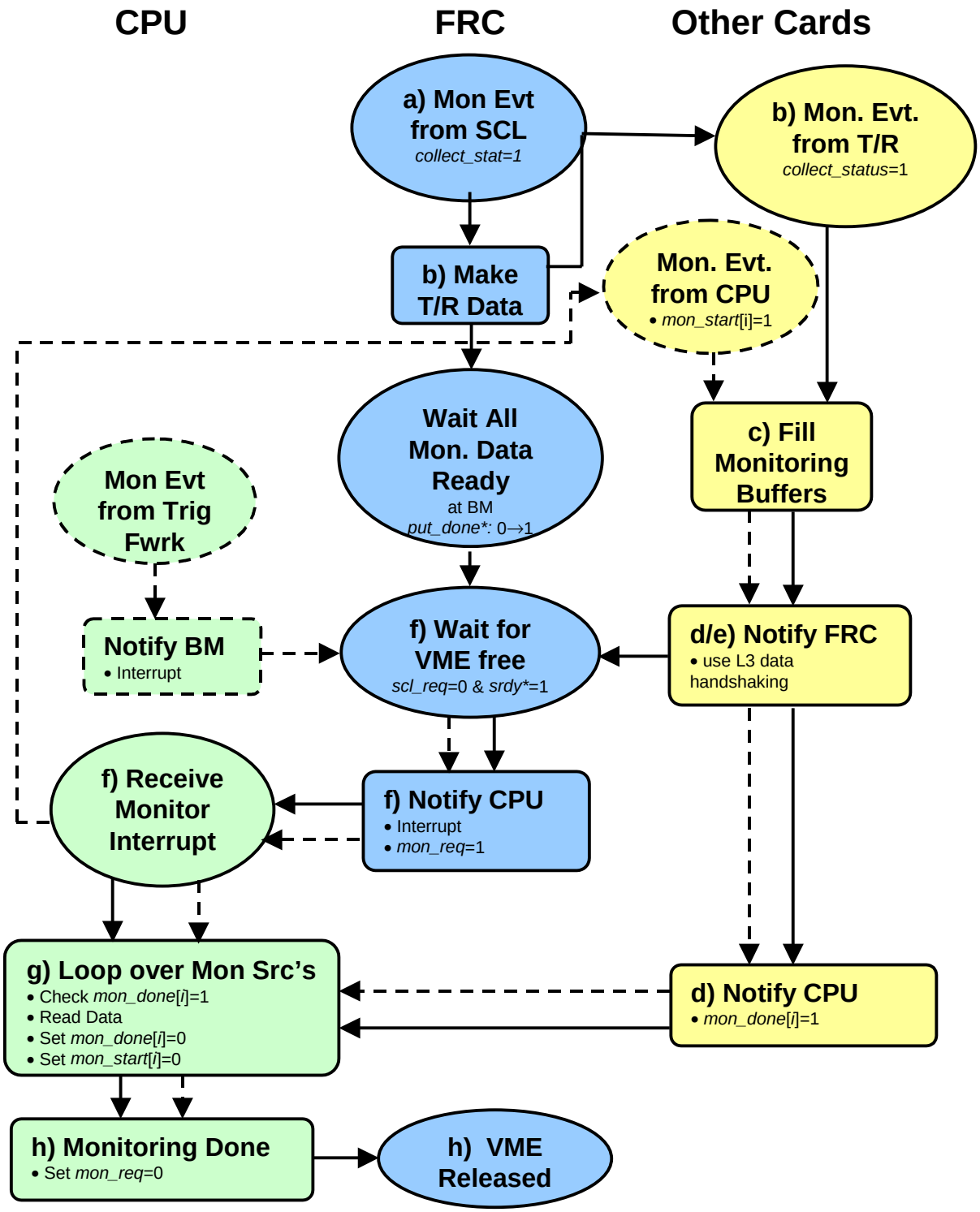
Who	What	When
DB	L1	L3 Data Buffers Full
DB	L1	Mon Buff Full – *Not Possible*
BC	L1	BC Memory Full
BC	L2	L3 Output FIFO Full
STC/FRC	L1	Lose G-Link Synch (?)

L3 Data to VBD



- **Open Issues**
  - **PCI-3 arbitration during data transfer to VBD**
    - **Universe II has limited size buffer**
  - **When does VBD release VME**
    - **Between blocks?**
    - **Between word-count and block?**

Monitoring Protocol



Monitoring Communications

BM BC	*PUT_BUFF* *PUT_DONE**	Handshake b/w BM and other cards to indicate Mon Data Ready
BM	*MON_REQ*	VME used by Monitoring
BM	Interrupt	Notify CPU to start monitoring
CPU	Interrupt?	Notify BM of CPU-Init monitoring
DB	*MON_DONE[I]*	Mon Data Ready (for CPU)
DB	*MON_START[I]*	Start CPU-Initiated Monitoring
DB	*MON_START[I]*	Mon Data Block (VME-accessible)

- **Open Issues – Interrupts**
  1. **PCI Interface has only one Interrupt Line – but need two interrupts (Monitor, SCL Init)**
     - **CPU reads register on BM to determine source?**
  2. **How to Implement CPU→BM Interrupt?**

Error Conditions (partial list)

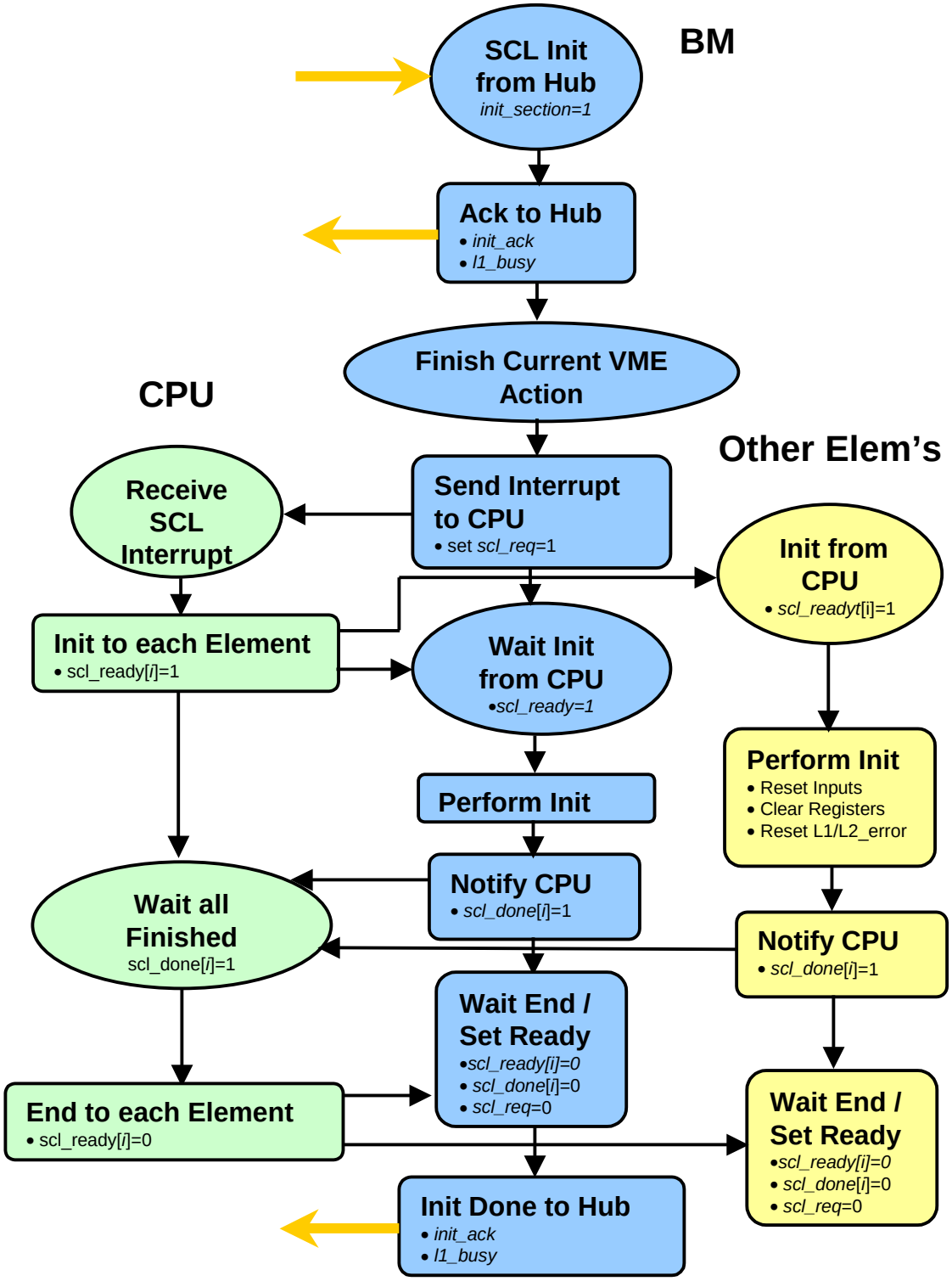
- **Two Severity Levels in the FRC**
  1. **Fatal** — **pull SCL Init immediately**
  2. **NON-Fatal** — **pull SCL Init only after a few of these in a row**
- **Two Types**
  1. **L1_ERROR**
  2. **L2_ERROR**

Error	Sever	Condition	Action when Fatal
INP_BX_ERR	F	BX(SCL) ≠ BX(L1CTT)	*L1_ERR*
INP_TURN_ERR	F	TURN(SCL) ≠ TURN(L1CTT)	*L1_ERR*
L3_BXIN_ERR	F	BX(DB L3 Data) ≠ BX(BM message)	*L1_ERR*
L3_BXOUT_ERR	F	BX(BC Output) ≠ Evt No.(BM message)	*L2_ERR*
NO_BOE	N-F	2 EoE's in L1CTT data from VTM seen with no BoE in between	*L1_ERR*
NO_EOE	N-F	no EoE seen in L1CTT data from VTM within a number of clock cycles equal to the max allowed number of L1CTT words (downloadable number, 52 by default)	*L1_ERR*
CTT_ERR	N-F	Error Flags set in L1CTT data Trailer (will not be dealt with by FRC)	—

SCL Init

- General Strategy
  - Element sending L1/L2_ERROR stops all work until it receives INIT
  - When *init_section* received by BM – wait until current VME activity finishes (but no more)
    - SCL Init has slightly higher priority than Monitoring or VBD
  - CPU informed of SCL Init by Interrupt from BM
  - CPU signals start of Init to all elements by VME registers
- Open Issue
  - SCL Init Logging in BM
    - Straightforward – but needs to be added to design

SCL Init Flow



FRC/BC Power Estimates

Chip	Current (mA)	Voltage (V)	Chip Power (mW)	No.	Approx Total Power (W)
FRC					**5**
10K50E-int i/o	90	2.5 3.3	~300 ~200	6	3
Others					2
BC					**7**
DP RAM	230	3.3	760	4	3
FIFO	230	3.3	760	2	1.5
10K50E			~500	2	1
Others					1.5

Status & Plans

Status

Comp	Blk Diag	Chips	Logic	FPGA code
SCLF	✔	✔	✔	start
TRDF/RR	✔	✔	✔	
BM	✔	✔	parts	
PCI-1/2	✔	✔	✔	start
PCI-3		✔		
BC	✔	✔		

Short Term Plans:

- Clock Architecture
- Configuration Scheme for FPGAs
- Schematics for each FPGA
- Assign Pins
- Plan Test Points
- Build!