



Discussion of fast controls and readout system for g-2

E. Hazen

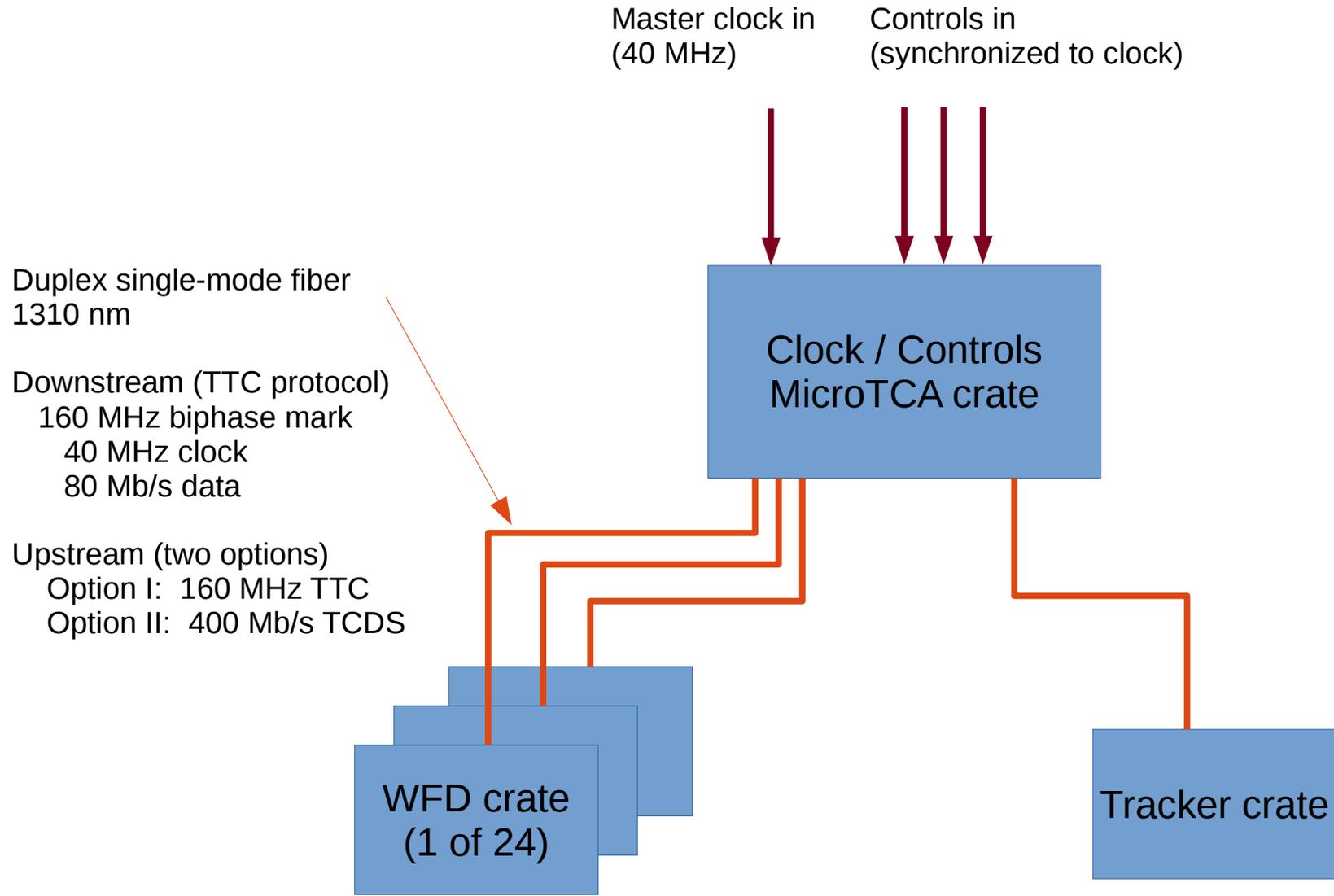


What controls?

- “Fast” controls, e.g. synch'd to machine
 - Start of spill (synch'd to clock [vernier?])
 - Set spill type (select calibration spill, etc)
 - Reset(s)(sent at start of run or other times)
- DAQ data flows automatically through the system, but what happens if buffers fill?
- “Back pressure” system
 - AMC13 says “slow down” to control system, which should discard start-of-spill signals until AMC13 says “ready” again.

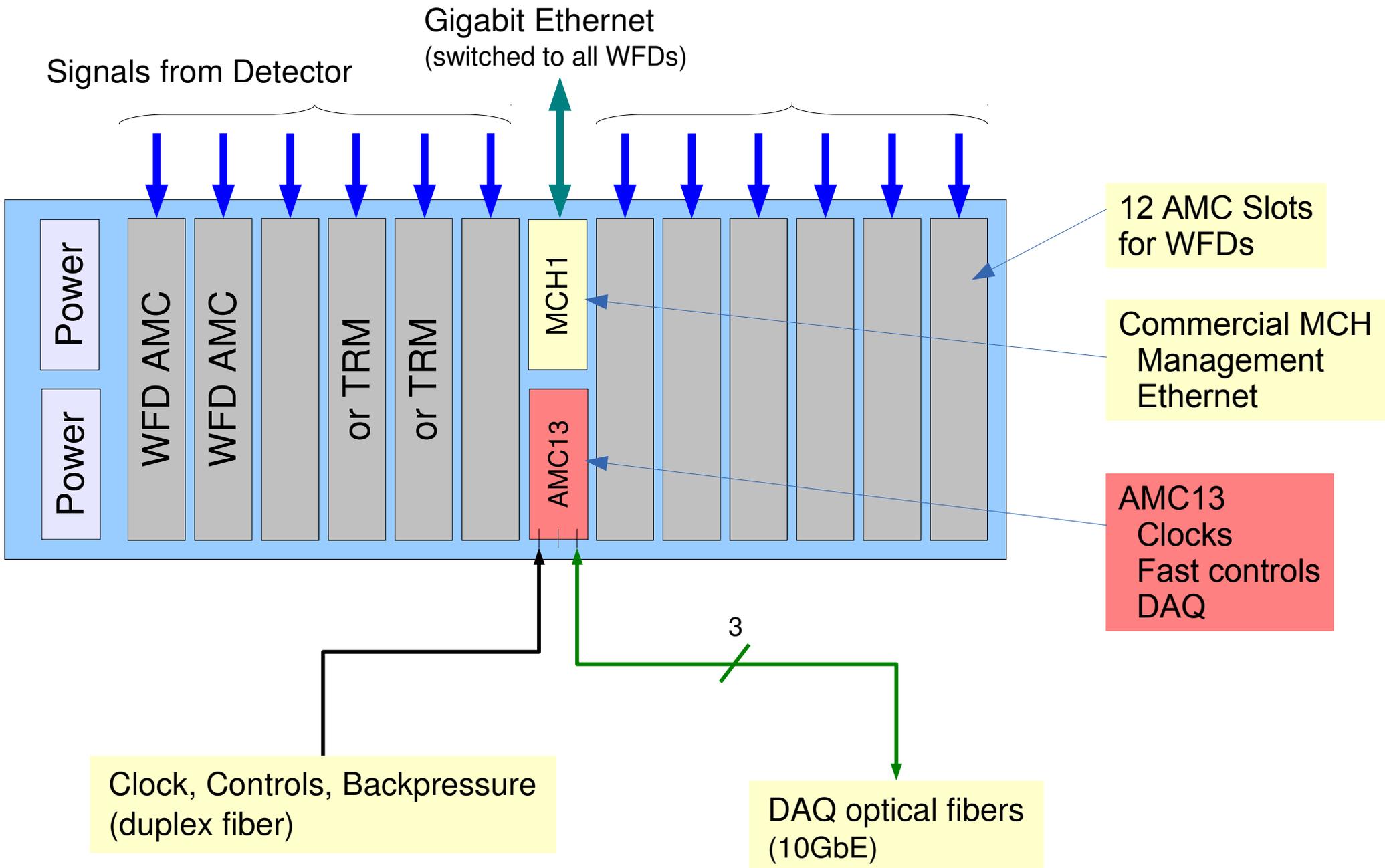


System Overview





g-2 uTCA WFD Crate





CMS TCDS PI Module

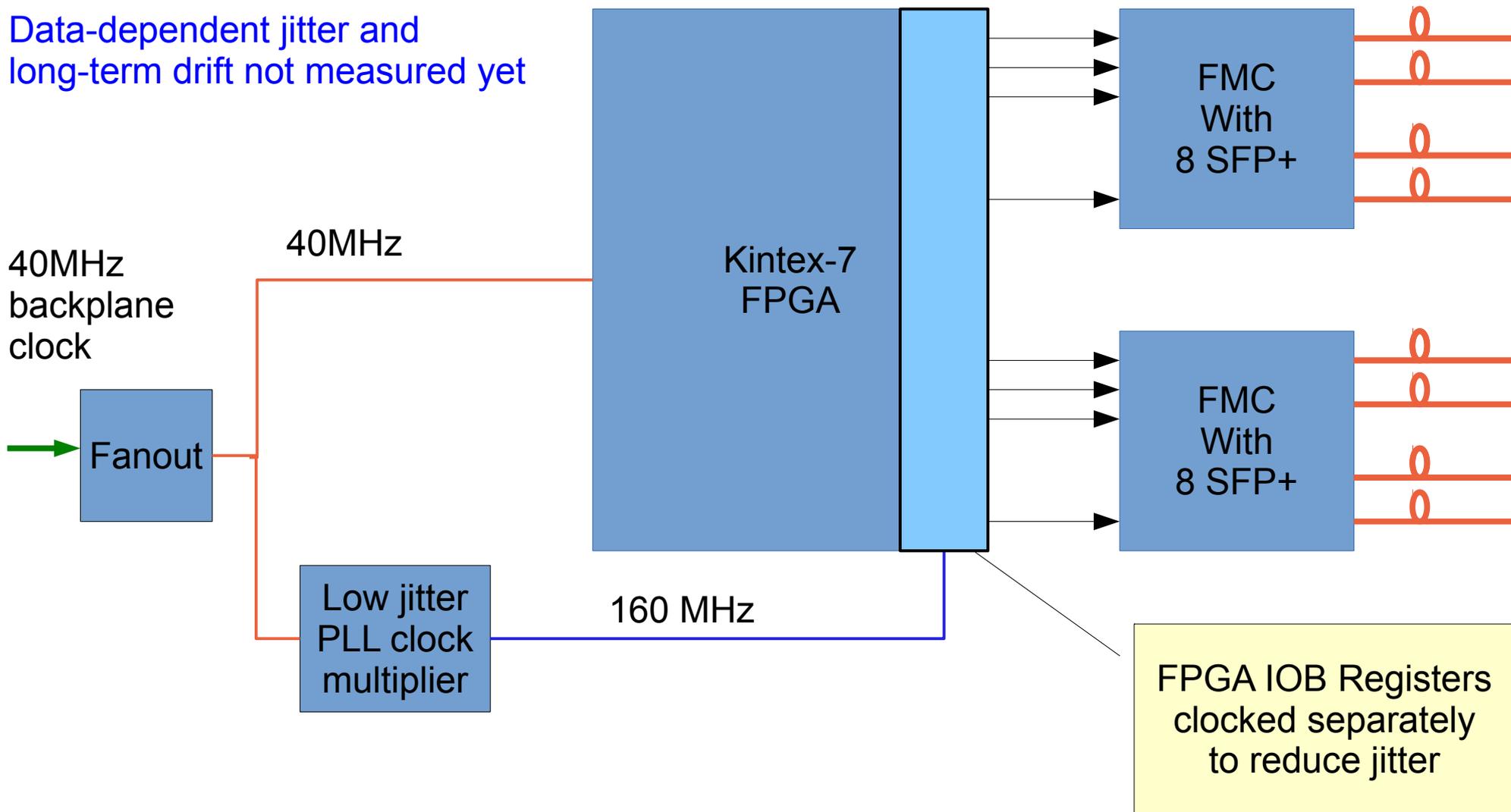


(based on FC-7)

Random jitter measured as
11 ps rms.

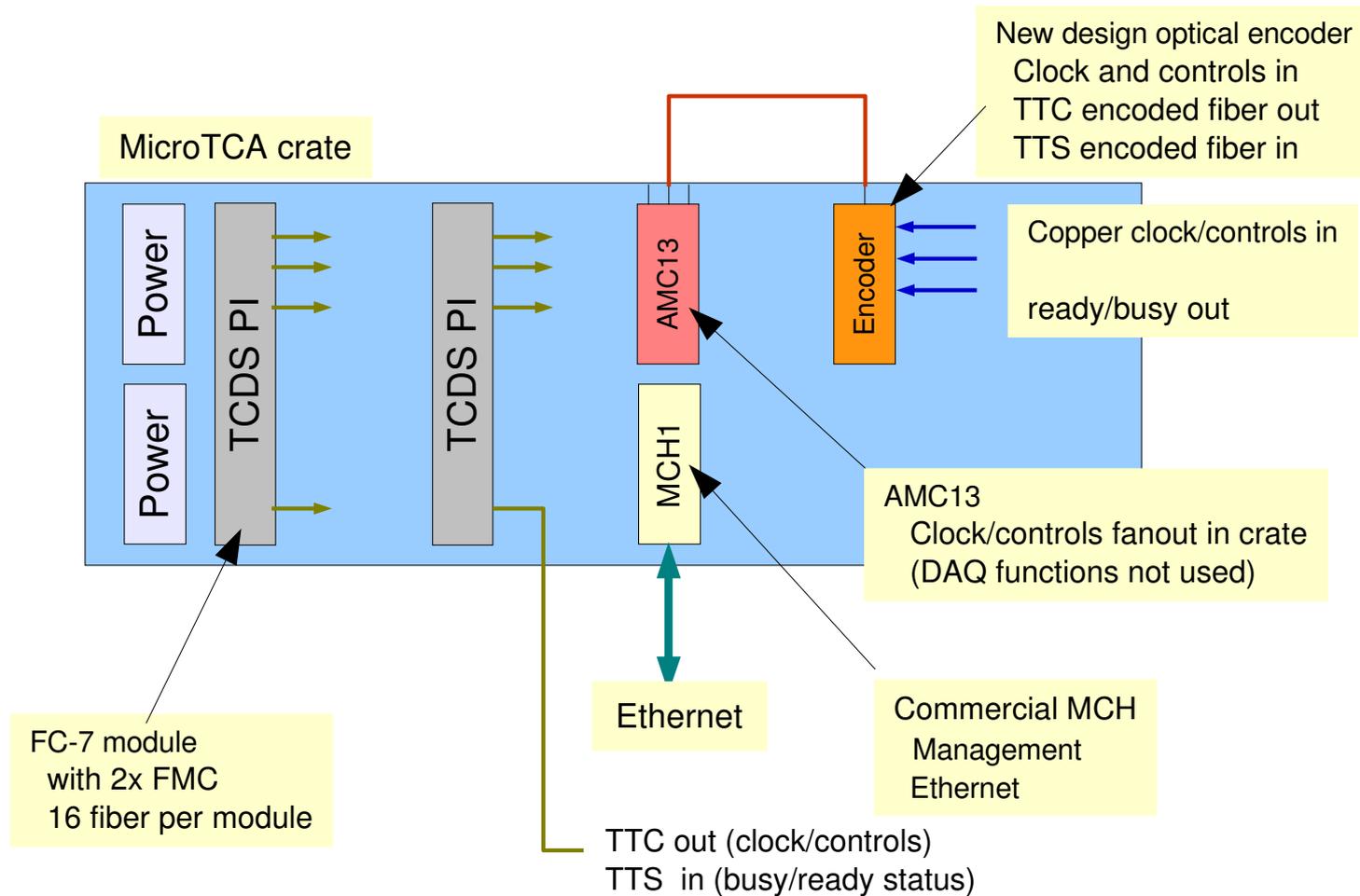
Data-dependent jitter and
long-term drift not measured yet

Differential pairs on PCB
with encoded clock/controls





Proposed g-2 clock distribution





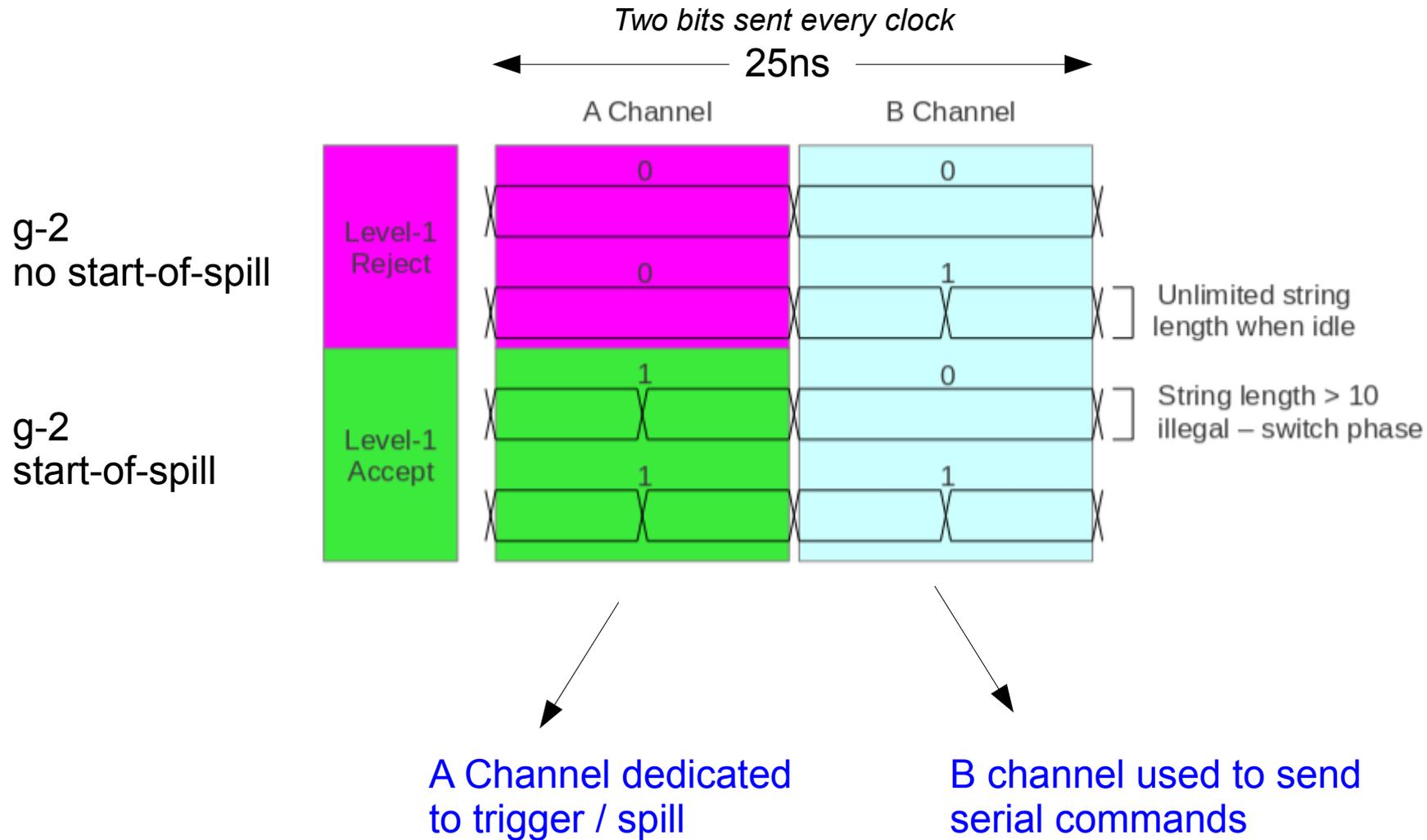
TTC vs TTS ?

- **TTC (Trigger Timing and Control)**
is the downstream fiber protocol
 - Sends real-time trigger and control to detectors
 - 40MHz clock, 80Mb/s data
 - Triggers (e.g. spill) and encoded commands
- **TTS (Trigger Throttling System)**
is the upstream protocol
 - Send ready / overflow warning / busy / error status to central location in real time
 - Can be used e.g. to skip start-of-spill signals if the DAQ gets bogged down for any reason to avoid buffer overflows
 - Can signal if synchronization is lost so that a reset can be performed in a timely fashion to avoid lost beam time



Fiber Protocol - TTC

Downstream: TTC -- 160MHz biphas mark encoding





TTC Broadcast Commands

Pre-defined by CMS legacy

Command	CMS	Meaning	G-2 Notes
-----1	BCR	Bunch count reset	May not use?
-----1-	ECR	Event count reset	Spill number reset to '1'
pppppp00	OCR	Orbit count reset	Send once per run?

Others suggested for g-2

Command	CMS	Meaning	G-2 Notes
pppppp00			Select spill type 1 (muon fill)
pppppp00			Select spill type 2 (laser trigger)
pppppp00			Select spill type 3 (pedestal trigger)

“Event number” (EvN) is incremented every trigger/spill

“Bunch count number” (BcN) is for CMS the clock tick within the machine orbit
It is reset periodically (every 3564 clocks for CMS) by BCR broadcast command.
It is a 12-bit value. For g-2 this could just be the low 12 bits of a 44 bit time stamp.

“Orbit number” (OrN) is incremented every BCR for CMS. May not need BCR for g-2.

All 3 (EvN, BcN, OrN) are stamped in the header of every event by AMC13
Also if present in header send from TRM, WFD AMC13 will check them (*very useful!*)



Fiber Protocol - TTS

Upstream: TTS – 400Mb/s 8b10b encoding

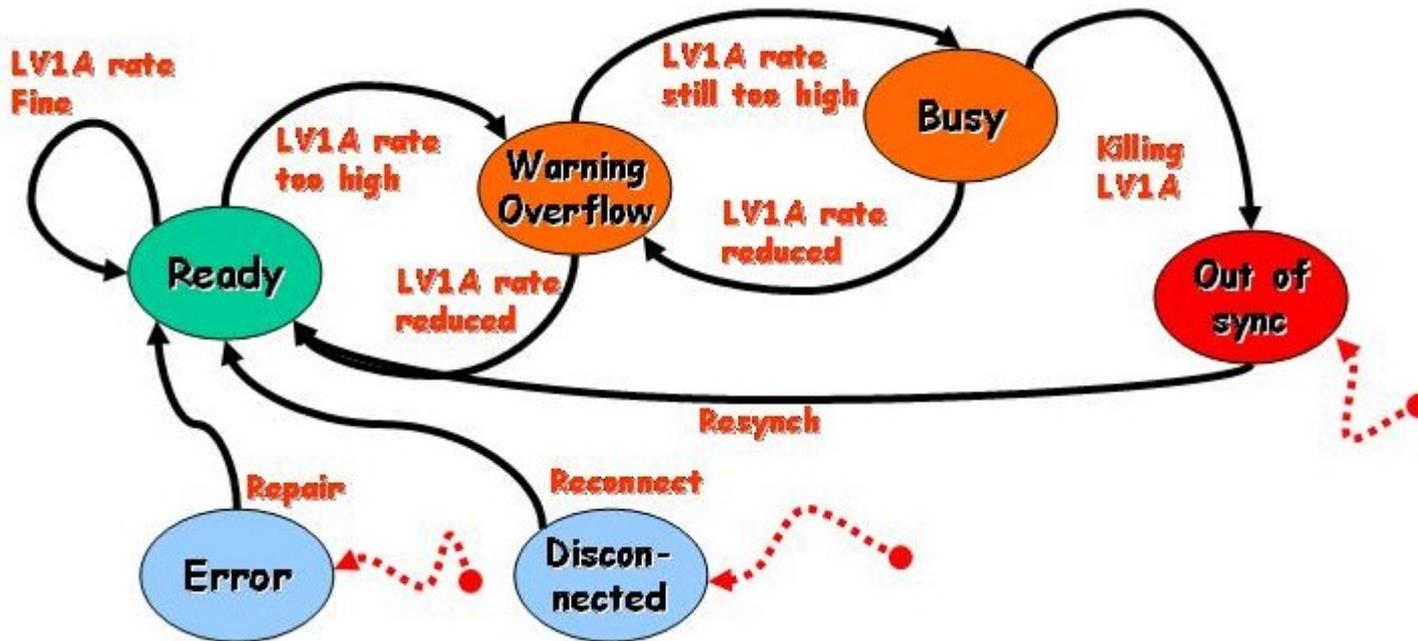
Fiber bit rate is 400Mb/s

Encoding is 8B10b as used in e.g. GbE

One byte takes 10 clocks, so one byte is sent every 25ns or one master clock tick

Low 4 bits encode state as shown below.

If state is unchanged, 3 of 4 bytes are the 8b10b control sequence K28.5



State Encoding

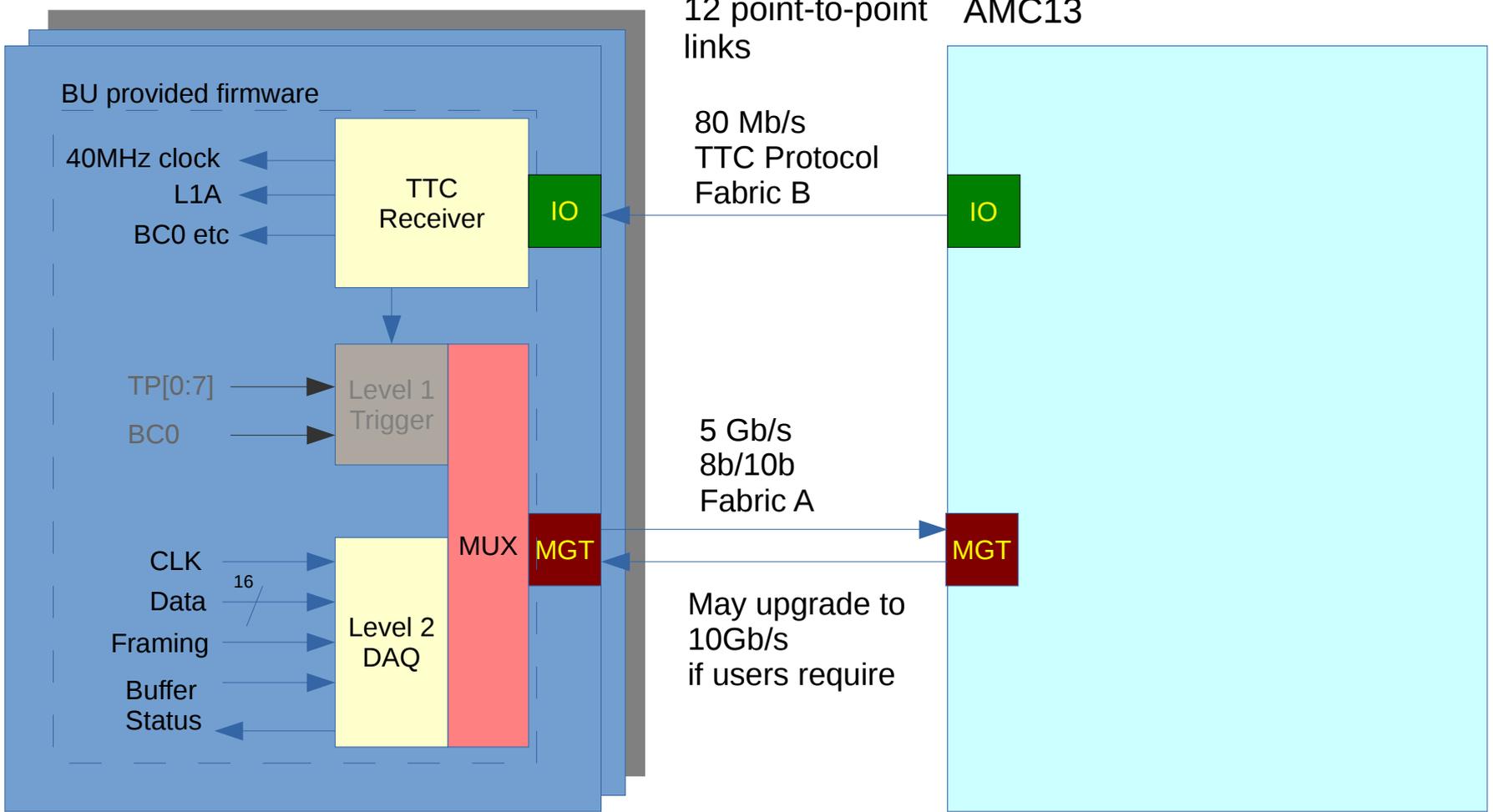
0000	Disconnected
0001	Overflow warning
0010	Out of sync
0100	Busy
1000	Ready
1100	Error
1111	Disconnected



DAQ

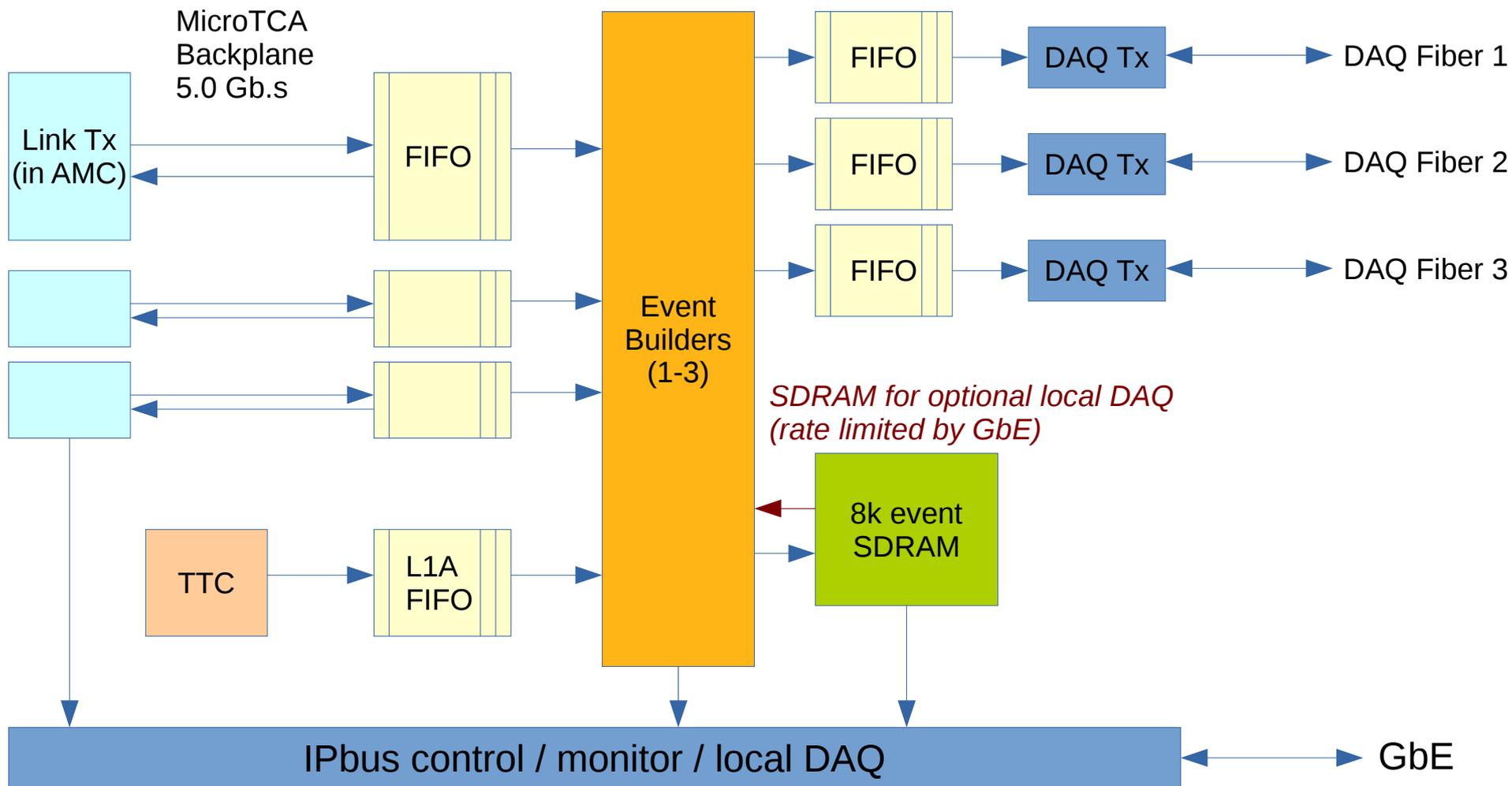


AMC Modules (WFD or TRM)





DAQ in AMC13





DAQ link AMC Firmware

We supply firmware with the following ports (Virtex-6, Kintex-7, Virtex-7 versions)

```
entity DAQ_Link_7S is
  port (
    --
    --
    -- TTS port
    TTSclk          : in  std_logic; -- clock source which clocks TTS signals
    TTS             : in  std_logic_vector (3 downto 0);
    -- Clock
    SYSCLK_IN      : in  std_logic; -- System clock (specify frequency as generic)
    -- Data port
    EventDataClk   : in  std_logic; -- clock for event data (any freq)
    EventData_valid : in  std_logic; -- used as data write enable
    EventData_header : in  std_logic; -- first data word
    EventData_trailer : in  std_logic; -- last data word
    EventData      : in  std_logic_vector (63 downto 0);
    AlmostFull     : out std_logic; -- buffer almost full
    Ready          : out std_logic; -- ready for data
  end DAQ_Link_7S;
```

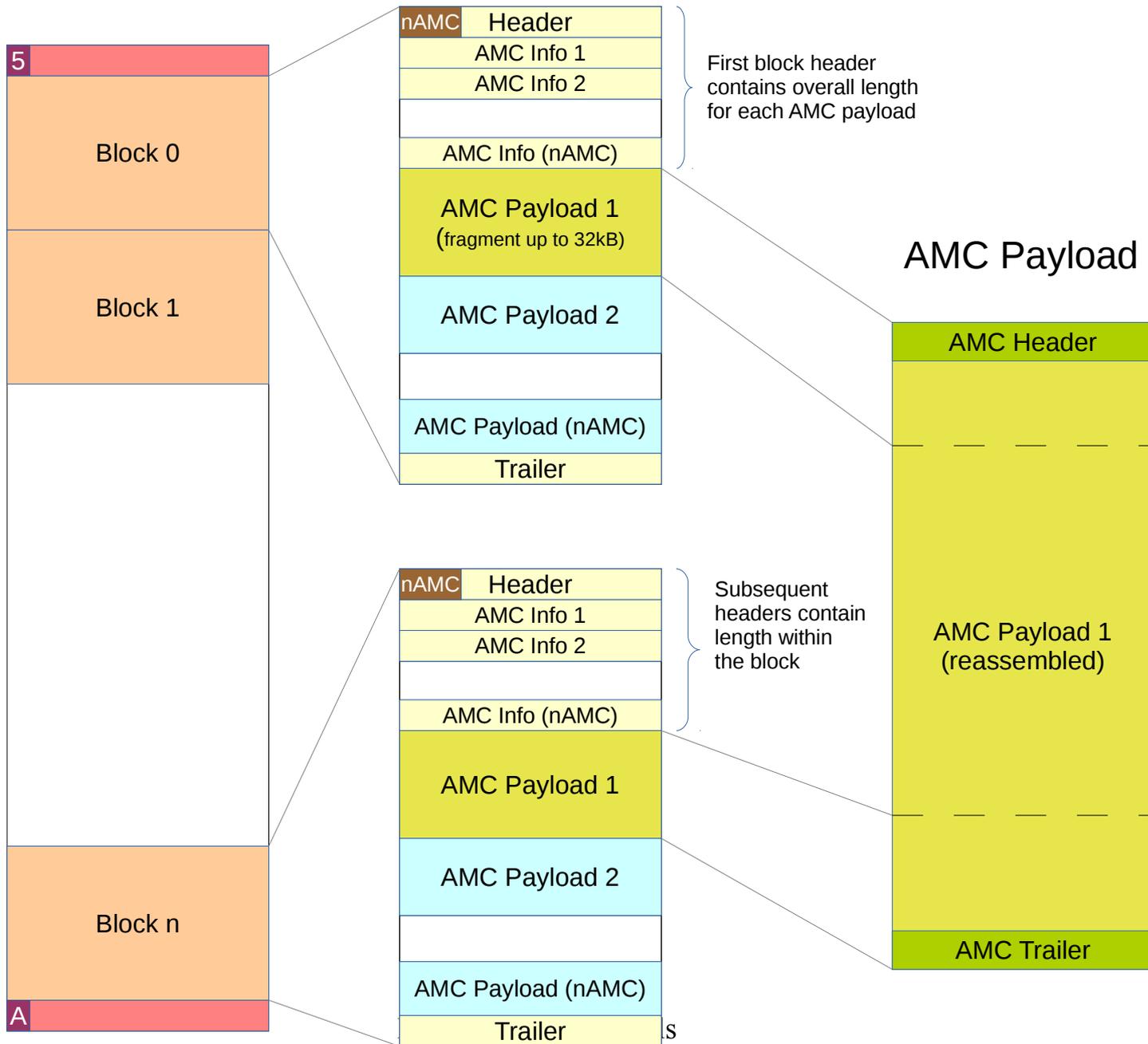


Data Format



AMC13 Event

Block Stream

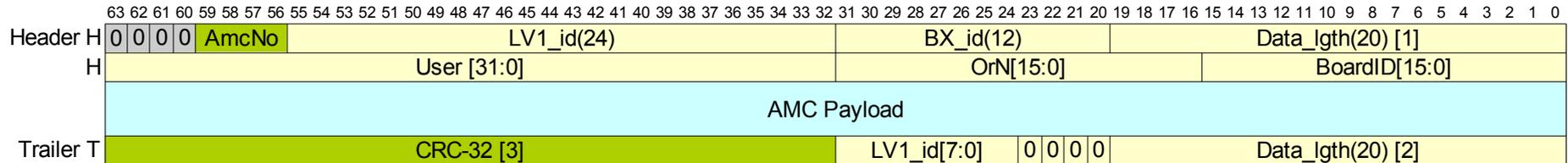




Data format to AMC13



Data format as delivered by user logic to AMC firmware



- LV1_id** reset to 1 on ECR TTC command
 increment every spill (regardless of type)
 note, two places in data
- BX_id** optional, but useful. Suggest to reset to 0 on BCR TTC command,
 increment on 40MHz clock, and put in header as shown
- OrN** optional, but useful. Suggest to reset to 0 on ECR TTC command,
 increment every BCR, and put in header as shown
- Data_lgth** Length of data is 64-bit words (note, two places in data)
- User, BoardID** can be anything from AMC13 perspective
- AmcNo and CRC-32** filled in by AMC13



Data format to DAQ



		63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDF Header	K	0x5					Evt_ty					LV1_id(24)												BX_id(12)						Source_id(12)						FOV		H x		\$ \$																									
		uFOV					Res					nAMC						Reserved (0)						OrN[31:0]								0x0																																	
		L	M	S	E	P	V	C	AMC1_size(24)						0	0	0	0	Blk_No(8)			AmcNo			BoardID																																								
		L	M	S	E	P	V	C	AMC2_size(24)						0	0	0	0	Blk_No(8)			AmcNo			BoardID																																								
Payload Block	D	L M S E P V C						AMC12_size(24)						0	0	0	0	Blk_No(8)			AmcNo			BoardID																																									
		AMC1 Payload																																																															
		AMC2 Payload																																																															
		AMC12 Payload																																																															
		CRC-32						0	0	0	0	Blk_No(8)			LV1_id(8)			BX_id(12)																																															
Subsequent Payload Blocks (if required)																																																																	
CDF Trailer	K	0xA					Evt_lgth(24)												CRC(16)						C F		x x		Evt_stat		TTS		T R		\$ \$																														

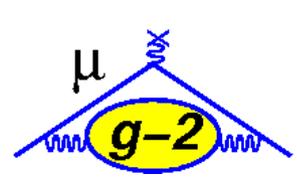
Status bits in AMC headers:

- M "More" bit set to 1 for all but last block
- M "Segmented" bit set to 1 for all but first block
- E "Enabled" bit set to 1 if this AMC input enabled
- P "Present" bit set to 1 if data present for this AMC
- C "CRC" bit set to 1 if CRC is valid (last block only)
- V "Valid" bit set to 1 if EvN, BcN match
- L "Length" set to 1 if length in header is correct



Summary

- Every AMC card should implement a “TTC receiver” (we can supply VHDL) to decode trigger and broadcast commands
- Every AMC card should count fake bunch crossing and orbit numbers and put in data so AMC can check synchronization
- Every AMC should send a TTS state (ready, overflow warning, busy etc)
- Every AMC should provide data for every trigger (start of spill) signal in the correct format



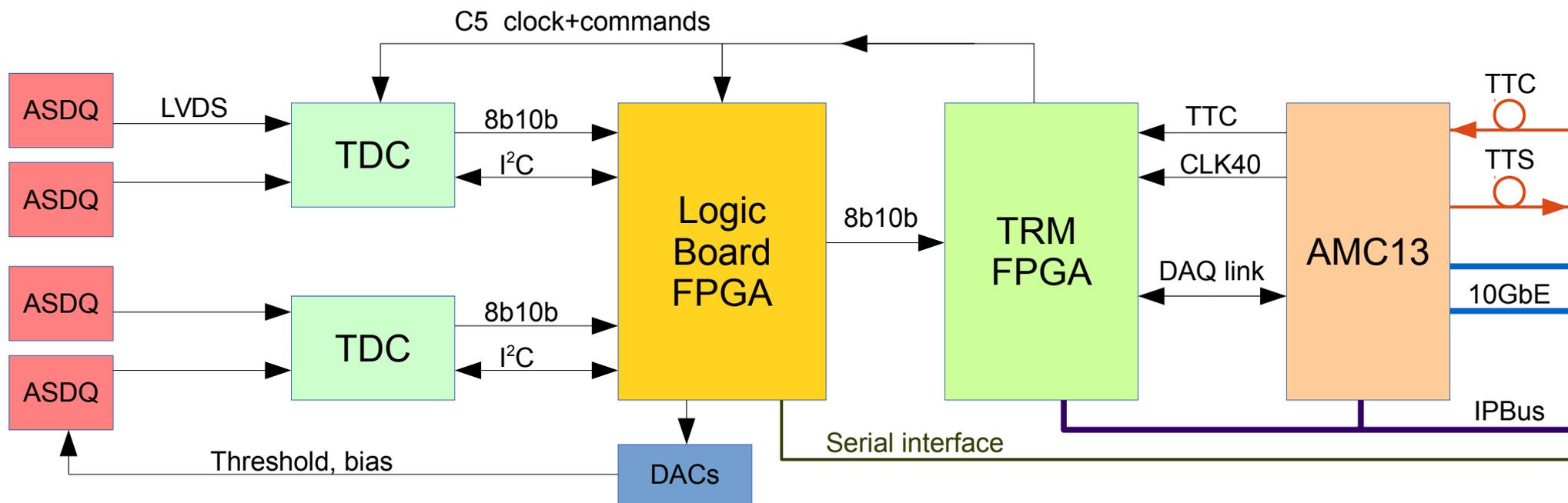
Backup Slides



Tracker Readout



Physical Connections



Logical Connections

For Further Discussion...



Run control
Set run parameters

Slow controls
Temp/press
voltage readout

Fast controls
Start spill / type
Reset