

CDP hardware/software development

Andrei Kounine
MIT

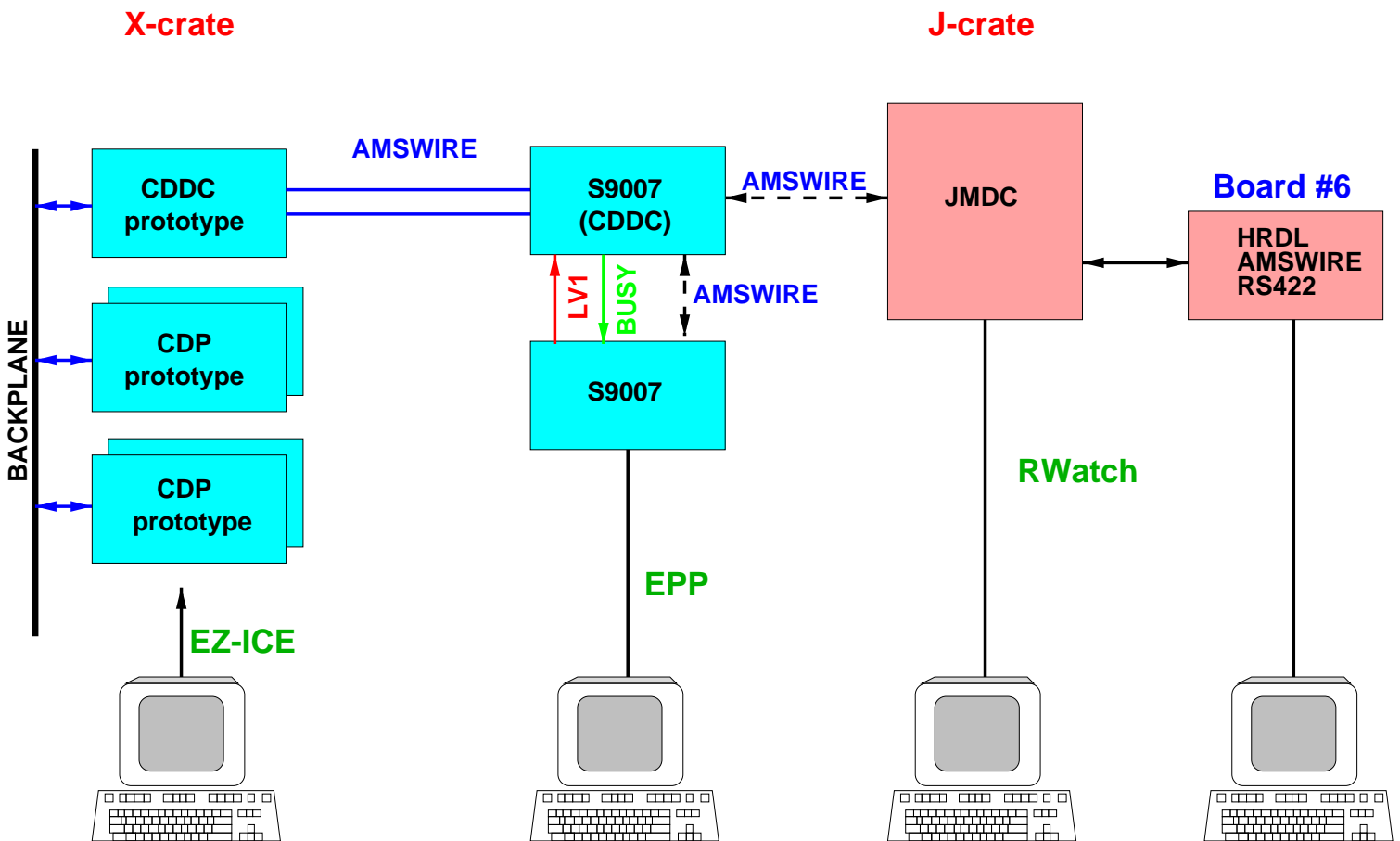
CERN, 9 May 2002

- Test setup - interfaces;
- CDP/CDDC software:
 - DM/PM usage;
 - FLASH programming;
 - Boot procedure;
 - Sequencer and raw event FIFO;
 - Interface to Data Processing;
 - AMSWIRE operations;
 - Data protection
 - Test procedures

Development setup

Setup features

- 4CDP + CDDC + backplane prototypes
- SW/HW development bench
- Interfaces: JTAG, ICE, AMSWIRE



CDP/CDDC software

Ingredients:

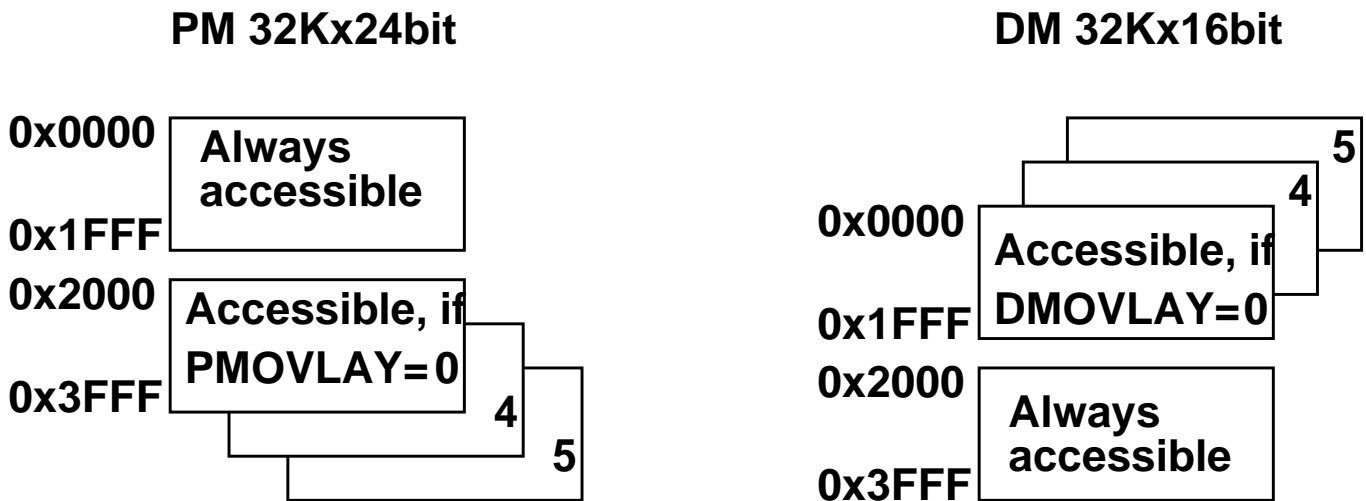
1. Boot procedure;
2. FLASH operations;
3. Data protection;
4. AMSWIRE operations;
5. Event building;
6. In-situ test procedures;
7. Data reduction procedures;
8. Calibration procedures.

Status:

1. Framework:
 - CDP/CDDC/Backplane prototypes;
 - AMSWIRE–EPP, AMSWIRE–PCI interfaces;
 - Working software version exists.
2. Software/Firmware debugging;
3. Ongoing performance studies;
4. SubDs contribution tests;
5. LV1/BUSY emulation to implement.

On-chip memory

ADSP-2187L: 160KB on-chip memory

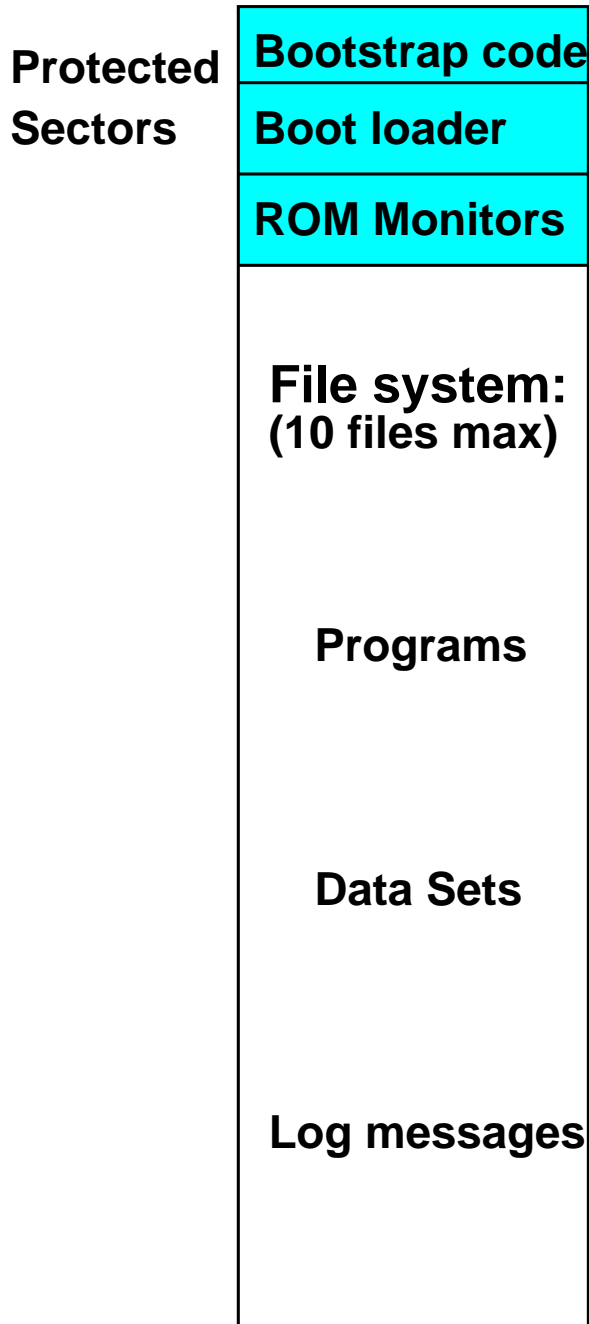


- PM AAC: Common code + constants
- PM OV0: Buffer for FLASH ops
- PM OV4: Data reduction code
- PM OV5: Calibration code

- DM AAC: Program variables
- DM OV0: Temporary buffers
- DM OV4: –
- DM OV5: –

CDP Boot procedure

FLASH MEMORY 512 KByte



- bootstrap code (first 96 bytes at the very beginning of the FLASH memory) loads boot loader to Program Memory.
- boot loader loops over 3 identical “master version” of ROM Monitor; checks CRC by copying it to BM; loads the first one that passes CRC check starting at PM address 0x0000 using BDMA with context reset at the end of BDMA transfer.
- ROM Monitor reads boot parameters from the GateArray (i.e. AMSWIRE) and proceeds accordingly – loads other programs from FLASH memory (Monitor Program with context reset or Event builder and Calibration programs without context reset) to their respective fixed locations in the PM. CRC checks are always performed before loading. In the case when no boot parameters are specified, default program is loaded.

FLASH memory operations

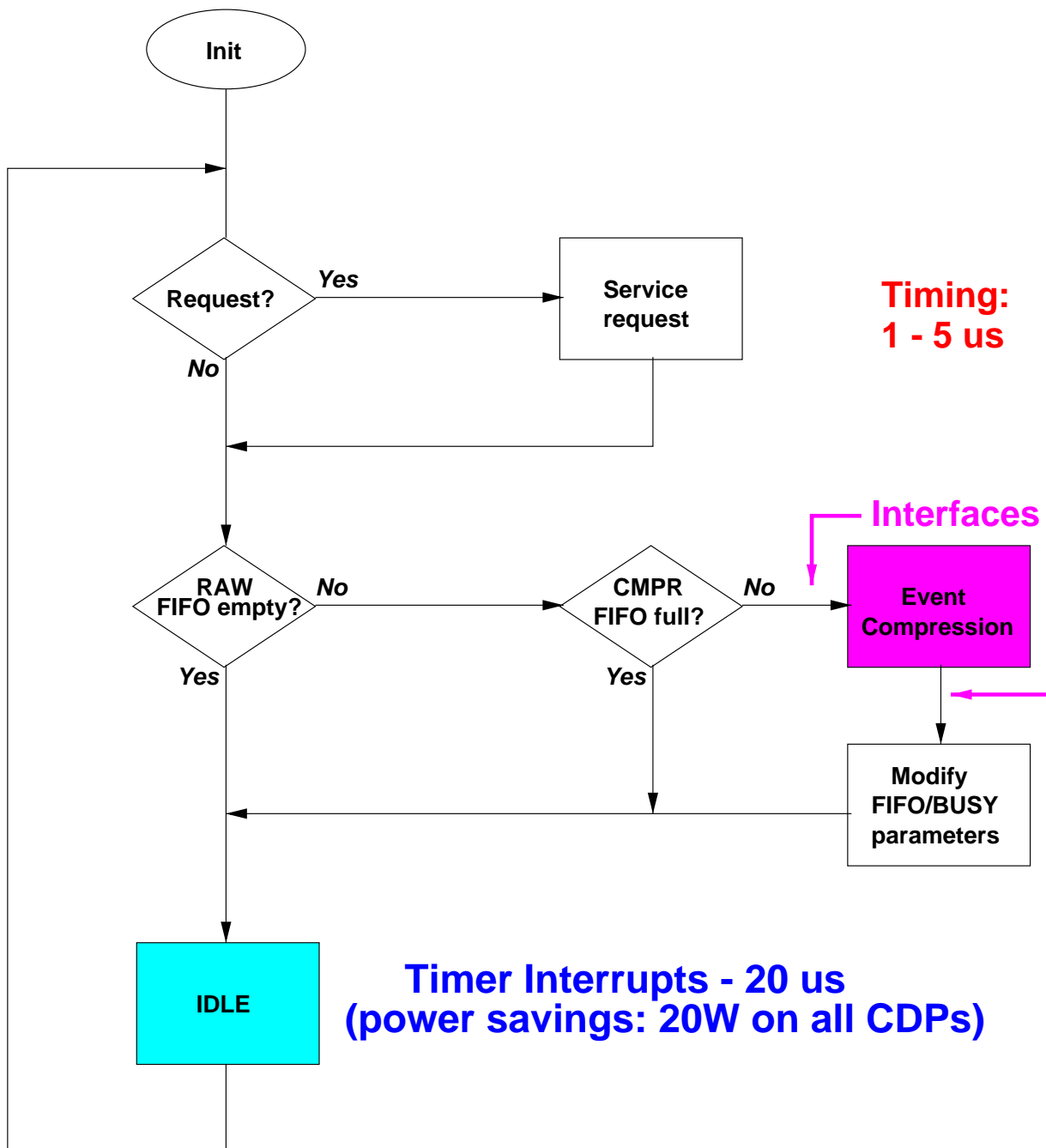
First time programming:

- Monitor Program is loaded to PM AAC; bootstrap code, boot loader and a dedicated FLASH Program utility are loaded to PM OV0 using EZ-ICE emulator;
- Write parameters are hard-wired to the FLASH utility;
- Execution is passed to the FLASH Program utility by setting the PC either by hands (testing phase) or via TCL script (production phase):
 - CRC check is performed by the FLASH Program utility to determine FCS value,
 - this value appended to the program as the last “instruction”.

In situ programming:

- Programs are transmitted over AMSWIRE;
- Upon receiving a program FLASH Program utility decodes it and puts it in the FLASH memory;
- CRC check is performed to validate integrity before the closure of the file;

CDP Program Flow



- *BUSY* is set by LV1
- *BUSY* removed when sequencer is finished and room in RAW FIFO permits

Subdetector interfaces

People:

- TDR – E.Cortina, D.Haas;
- UDR – F.Hauler;
- RDR – J.Marin, G.Martinez;
- EDR – F.Spinella;
- TOF – F.Cindolo;
- LV1 – C.H.Lin;

Programming:

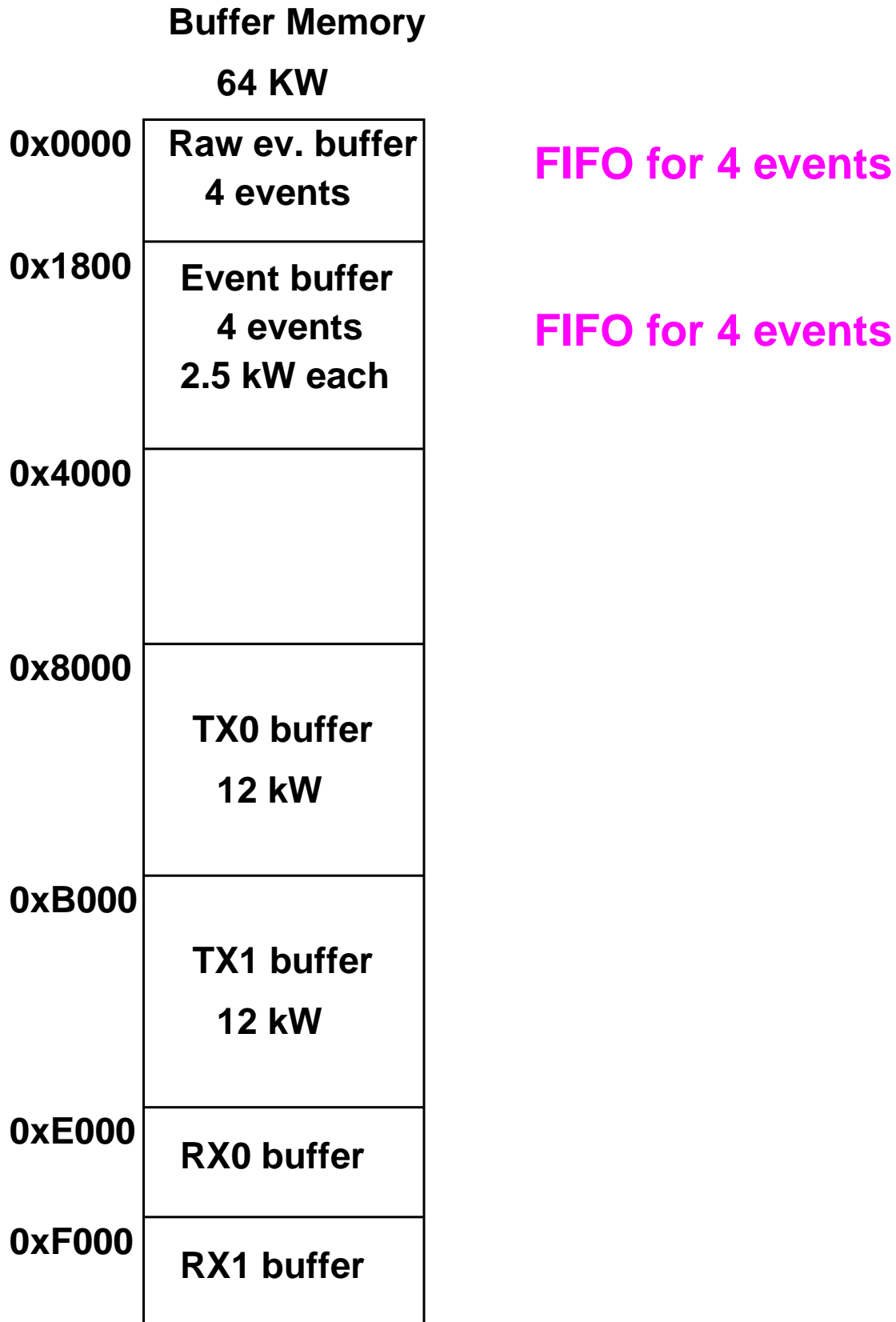
- Input: N_{raw}
- Event in: $N_{\text{raw}} \times (\text{amp} = \text{io}(\text{BM_READ}));$
- Processing: $(\text{amp} \rightarrow \text{val})$
- Event out: $N_{\text{build}} \times (\text{io}(\text{BM_WRITE}) = \text{val});$
- Output: N_{build}

High level communication protocol

AMSWIRE protocol:

1. Optimized for event building procedure;
2. Allows buffering of events at all intermediate DAQ levels;
3. AMSWIRE block consists of a header which is modified as the block travels across DAQ tree, and a block body which stays unchanged;
4. Routing map exists only in one MDC, other nodes know only paths to their immediate “slaves”;
5. Allows a “master” MDC node addressing any path which is permitted by hardware connections;
6. MDC may use both individual and group addressing;
7. No increase of DAQ dead time due to the protocol.
8. Performance:
 - link throughput - 10 MByte/sec without data protection,
 - link throughput - 7.6 MByte/sec with data protection,
 - error per bit rate – $< 10^{-14}$ for 4 KWord packets.

CDP Buffer Memory usage



CDP/CDDC event building

Event building:

1. Event buffering at all levels of DAQ;
2. Data compression and buffering in CDP;
3. Data concentration and buffering in CDDC;
 - During data taking CDDC requests and assembles events from its slaves whenever there is room in its event buffer.
4. Early data protection with 16-bit CRC algorithm:
 - non-empty events are protected in CDP right after data compression;
 - assembled events are protected in CDDC;
5. Maximal event size (CDP – 2.5kW, CDDC – 12kW).

Optimal settings/requirements (no performance penalties at 2 kHz):

1. Readout time – $< 83\mu\text{s}$;
2. Event size – $< 3\text{KByte}$ (on average);
3. CDP raw event buffer size – $\geq 4\text{events}$;
4. CDP compressed event buffer size – $\geq 2\text{events}$;
5. CDP event compression time – $< 320\mu\text{s}$;
6. CDDC event buffer size – $\geq 2\text{events}$;

Data integrity protection

Data corruption for normal data taking:

1. ~ 1 raw event will be upset during the flight.
2. ~ 0.01 built event will be upset in CDP/CDDC BM.
3. $\sim 10^4$ events will be upset in JBU.
4. $\sim 10^2$ upsets in the PM of CDP/CDDC.

CRC protection:

1. AMSWIRE blocks (programs and events for sure);
2. 16 bit Frame Check Sequence (FCS) - protects up to 8 KByte of data against all single- and almost all two-bit errors;
3. Calculated by moving data block to/from BM (DAQ);
4. Early event protection (after event compression).

System synchronisation protection:

1. Periodic checks of the last event number assigned in FE DAQ Modules;
2. It must be the same for all FE modules.

In-situ testing

Hardware tests:

1. Buffer memory test;
2. Data memory test;
3. FLASH read test;
4. CDDC – slave test (MASK);
5. Program is tested before loading from FLASH.

Program integrity tests (software):

1. Every program segment in PM has a FCS as its last “instruction”;
2. Common code contains a routine to calculate FCS by copying it to BM;
3. This routine is called at INIT and sets a program status flag.

Any test shall be performed on a corresponding AMSWire request

In-situ testing

PC interfaces (EPP):

1. Universal AMS interface;
2. Includes 4 AMSWIRE links;
3. AMSWIRE software for PC:
 - Commands – replies;
 - Data retrieval from CDP/CDDC;
 - Handling of LV1/BUSY;
4.;

PC interfaces (PCI):

1. Includes 8 AMSWIRE links;
2. Software provided by X.Cai;