XMM Optical Monitor

DEPARTMENT OF PHYSICS
UNIVERSITY OF CALIFORNIA, SANTA BARBARA

Author: Cheng Ho (LANL)

XMM-OM Fast Mode Data Format

DOCUMENT: XMM-OM/UCSB/TC/0038 25-June-98

Contents

1 General Overview

The XMM-OM DPU acquires data from the MIC detector and perform necessary processing to retain targeted scientific information while reducing data volume to fit the allotted telemetry bandwidth. There are currently two major data processing modes: image mode and fast mode. The image mode is meant to collect long exposure images (corrected for satellite drift) over a large area. the image data data format is straightforward: after compression, it is simply a concactnation of pixel values. The fast mode, on the other hand, is meant to retain high time resolution available from the MIC detector and targeting time variable sources, which make up a large fraction of important X-ray sources. Due to the constraints in TM rate, the fast mode will be restricted to a small area, i.e. in the number of pixels from where fast mode data are collected is limited. But the fast mode data will be three dimensional, with the time resolution making up the third dimension.

This document contains the following sections: Section 2 describes the core of the OM fast mode data acquisition. Section 3 discusses peripheral operations leading up to and resulting from the OM fast mode. Section 4 reviews the steps necessary for recovering the fast mode data. Section 5 discusses the fast-mode related information contained in the downlinked data set DP_WDW. Section 6 discusses the content of the data set DP_BFAST. Section 7 reviews the content of the real fast mode data DD_FST and the detailed reconstruction procedure. Section discusses the basic nature of the reocntructed fast mode data.

2 Fast Mode Data Acquisition Core

The MIC detector offers single photon counting/imaging with good time resolution which opens the possibility of acquiring data for fast time variable sources. While the image mode data are acquired by simply accumulating counts onto DPU on-board RAM, the fast mode data are on the other hand acquired by the DPU in a manner that is at the first sight convoluted. The scheme is designed to be robust, to maximize information content to conform to the telemetry badnwidth constraint, to handle the asynchronocity between MIC data output and DPU timing, and to be consistent with available DPU and other OM resources.

To better explain the non-intuitive OM fast mode data acquisition, we describe the core of the OM blue fast mode acquisition scheme in this section. The overall view of fast mode data acquisition sequence will be discussed in section 3.

2.1. Fast Mode Area Basic Setup: The basic setup unit of the blue fast mode data acquisition and data formatting is the "blue fast mode area." (The blue fast mode area is sometimes referred to as blue fast area, or BFAST [pronounced as 'b-fast'] in the New Mexican dialect.) Up to two fast mode areas can be allocated, each mapping to one of the two independent blue DSPs. Each blue fast area will have a fixed sampling time. Each will be associated with a number of (up to 512) pixels in the ping/pong (small words) memory area. The addresses in the blue fast area must be within valid memory window but need not be contiguous. (See DPU operation manual for discussions on memory, science and detector windows.) For example, a fast area of 16x16 within a 256x256 memory window is allowed. A fast mode area containing two blue fast mode science windows, one 10x10, one 20x20 pixels and both in valid memory windows, making up a total of 500 (<512) pixels, is also allowed. As each blue fast area is assigned to one of the two blue DSPs, the two fast areas can have separate sampling time.

It should be stressed very strongly that the fast mode science windows are closely related to the fast mode areas, but are NOT the same. The former are requested by the observer on the ground and the latter are allocated onboard by the DPU's tracking reference frame and window management algorithm. See 3.2. for more detailed description of the allocation of the blue fast area and association with user requested science windows.

2.2. Fast Mode Sampling: The MIC electronics will embed two unique markers (0x0, excluding parity bit) into the data stream to demark the discrete MIC CCD frame. (Current baseline MIC CCD frame time is 10 ms.) The DSP, triggering on the specified number of MIC CCD frame markers (corresponding to the sampling time), will read the "image" from the corresponding blue fast area. This read is done in batches of up to 64 pixels per millisecond.

2.3. Image Differencing: The read-in image of the fast area is differenced from an existing image of the same area for the previous sampling. The differenced image, called slice, is made up of the new photons which arrived in the past sampling time.

2.4. Photon Address Formulation: The fast mode slice, i.e. the differenced image, is scanned, and those pixels with nonzero value indicate new photon(s). The equivalent of each new photon's address in the linear representation of the three dimensional data is calculated. For example, let's say we have two blue fast mode science windows, 10x10 and 20x20 pixels each, with the same 5x10 msec resolution. They can be combined to form a 500 pixel fast area. A photon arriving in the first 50 msec at (5,5) of the first science window will have a 3D address of 55 (= 5*10 + 5). A photon arriving at the second 50 msec at (10,10) of the second fast mode science window will have a 3D address of 810 (=500 + 10*10 + 20*10 + 10), where the 500 is to indicate the crossing of one sampling time. A third photon arriving at the fourth 50 msec at (3,7) of the second fast science window will have an address of 1667. In this representation, the quotient of the address divided the fast area size gives the time slice number (i.e. timing information), and the residue gives the pixel location within the fast area (i.e. imaging information).

2.5. Photon Address Encoding: Instead of storing the full 3D address as a monotonically increasing series, the blue DSP performs additional differencing of consecutive photon address to reduce bits required to store the same information. In the example above, instead of 55, 810, 1667, the three photons' addresses will be encoded and stored as 55, 755, 857, in the 16-bit memory.

2.6. Tracking Frame Crossing: The blue fast mode acquire data from the accumulating image stored in the OM DPU's small word memory. The fundamentanl working concept of the OM is ping/pong tracking utilizing template raw image data acquired in two equal and alternating memory area. This must also be reflected and handled in the fast mode data acquisition. This is implemented as follows: At each tracking frame crossing, a fast mode slice is forced to flush out any data that comes in the partial slice. A frame crossing separator is attached to signal the tracking frame crossing. The format of the frame separator is as follows:

```
INT16 separator_head (fixed value 0x1ace and 0x1bee alternating);
INT16 marker_remainder (number between 0 and sampling\_time);
INT16 base_crossing;
INT16   separator_tail (fixed value 0x1fff);
```

More details of the tracking frame crossing separator are discussed in Section 7.

3 Operation Procedures of OM Fast Mode

From the user's standpoint, the "end-to-end" fast mode data acquisition procedure is as follows:

3.1. The observer, with the help of ground-based observation tools, requests a number of science windows for the particular exposure. Among them could be one or more fast mode science windows, specified as science window mode 1 (See ICU-DPU interface document and other OM observing mode documents for more details). The observer should also specify the location and size of the window, within the constraints discussed below. The observer also specifies the sampling time as an integer multiple of the MIC CCD frame time. The observer can have more than one fast mode windows and can specify up to two different fast mode sampling times. The following is an example of the science window configuration requested by user:

< Table 1. User Requested Science Windows >

x0	y0	xsize	ysize	mode	par1	par2	par3	par4	base	mmw
1032	1032	496	496	0	1	1	0	0	0	0
1056	528	32	32	0	0	0	0	0	0	1
1200	1200	16	16	1	5	0	0	0	0	0
1400	1300	16	16	1	10	0	0	0	0	0

In this case, the user requests four science window, the first two are image mode (mode=0). The third is a fast mode window of size 16x16 with a sampling time of 5*10 ms. The fourth is a fast mode window of size 16x16 with a sampling time of 100 ms.

3.1.1. It should be noted that under normal circumstances, the DPU utilizes two blue DSPs to capture data from the MIC electronics. The DPU does have provision to utilize only one of the two blue DSPs to capture MIC data. If such case is requested by the observer, then the desired sampling time should be DOUBLED: specifying a value of 10 for the par1 of a fast mode (sampling_time) will yield a real sampling time of 10*10ms/2 = 50 msec. In such case, the observer should be careful to NOT request odd number of sampling time as this will lead to unpredictable and probably unusable data.

3.2. The window configurations are uploaded to the DPU for memory and window management. After the OM acquires a reference frame and selects an appropriate number of guide stars, a memory/window management package is invoked to allocate DPU resources for the data acquisition and processing. After the tracking guide star windows, memory windows, and detector windows have been allocated, the science windows are assigned. For each of the fast mode science windows, the attachment will be assigned following the conditions below:

3.2.1. It must reside in a valid memory window.

3.2.2. If there is no fast mode area allocated, then a new fast area will be allocated and the fast window will be attached to that fast area, provided that it has no more than 512 pixels.

3.2.3. If there are existing allocated fast areas, then

3.2.3.1. if the sampling time of the fast mode science window matches that of the existing fast area, then

3.2.3.1.1. if the resulting fast area has no more than 512 pixels, then the fast mode window is attached to that existing fast area, or

3.2.3.1.2. if the resulting fast area has more than 512 pixels, then the algorithm will try to attach to the next existing fast area (3.2.3.1) or allocate a new fast area (3.2.4);

3.2.3.2. if the sampling time of the fast mode window does not match that of the existing fast area, then the algorithm will try to attach to the next existing fast area (3.2.3.1) or allocate a new fast area (3.2.4);

3.2.4. If a new fast area is called for and there will be no more than two fast areas, then a new fast area will be allocated similar to 3.2.2.

3.2.5. If there are two a new fast area is called for and there will be no more than two fast areas, then a new fast area will be allocated similar to 3.2.2.

The science window configuration will be telemetered down as the data type DP_WDW. In this data set, the attachment of the fast mode science window to the fast area will be specified. In addition, for each allocated blue fast area, a data set of type DP_BFAST will be telemetered. These data sets will contain essential information of the blue fast areas. The format of these data sets will be discussed later.

3.3. The fast mode data acquisition and formatting will be started and terminated contemporaneously as the OM frame-by-frame tracking, in a manner as described in Section 2. The end result of the fast mode acquisition is a series of numbers stored in 16-bit memory.

3.4. At the end of the exposure, the image mode and fast mode data are submitted to a queue for compression. The data will be compressed during the subsequent exposure unless explicitly flushed. The compressed data are then sent down to the ground via the ICU.

4 Recovery of Fast Mode Data

To recover the fast mode data, one needs to reverse the process described above in the following steps:

4.1. Establish all auxilliary information from DP_WDW and DP_BFAST. See the next two sections.

4.2. Removal/Stripping all ICU/TM packet headers. This is covered in another document.

4.3. Removal of headers of DPU-ICU data interface. This is covered in another document.

4.4. Decompress the compressed fast mode data. This is covered in another document.

4.5. Reconstruct the absolute photon 3D addresses based on the decompressed data. This is discussed in Section 7. The fast mode data is now ready for analysis.

5 Blue Fast Mode Window Information (DP_WDW)

The data set DP_WDW is sent down immediately after the tracking reference frame is processed and windows are set up. See ICU-DPU interface protocal document for more complete details of DP_WDW. Here we point out the information contained in DP_WDW that is specific to the fast mode. A dump in of the science window section of an example DP_WDW data set yields the following:

```
< Table 2. Part of Data Dump of DP_WDW >

0001120  0000 0408 0000 0408 0000 01f0 0000 01f0
0001140  0000 0000 0000 0001 0000 0001 0000 0000
0001160  0000 0000 0000 1008 0000 0000 0000 0420
0001200  0000 0210 0000 0020 0000 0020 0000 0000
0001220  0000 0000 0000 0000 0000 0000 0000 f040
0001240  0004 0410 0000 0001 0000 04b0 0000 04b0
0001260  0000 0010 0000 0010 0000 0001 0000 0005
0001300  0000 0001 0000 0000 0000 0000 0001 60b0
0001320  0000 0000 0000 0578 0000 0514 0000 0010
0001340  0000 0010 0000 0001 0000 000a 0000 0002
0001360  0000 0000 0000 0000 0002 f114 0000 0000
0001400  0000 0236 0000 0327 0000 0020 0000 0020
0001420  0000 000f 0000 0000 0000 0000 0000 0000
0001440  0000 0000 0004 1597 0000 0002 0000 077c
0001460  0000 0451 0000 0020 0000 0020 0000 000f
0001500  0000 0001 0000 0000 0000 0000 0000 0000
0001520  0004 2311 0000 0003 0000 02dd 0000 050c
0001540  0000 0020 0000 0020 0000 000f 0000 0002
0001560  0000 0000 0000 0000 0000 0000 0004 334c
0001600  0000 0004 0000 0770 0000 0687 0000 0020
0001620  0000 0020 0000 000f 0000 0003 0000 0000
```

```
0001640  0000 0000 0000 0000 0004 4417 0000 0005
0001660  0000 003c 0000 029c 0000 0020 0000 0020
0001700  0000 000f 0000 0004 0000 0000 0000 0000
0001720  0000 0000 0004 530c 0000 0006 0000 06a9
0001740  0000 00ec 0000 0020 0000 0020 0000 000f
0001760  0000 0005 0000 0000 0000 0000 0000 0000
0002000  0004 624c 0000 0007 0000 0252 0000 0552
....
```

A more readable translation of this dump yields

< Table 3. Translation of DP_WDW in Table 2 >

x0	y0	xsize	ysize	mode	par1	par2	par3	par4	base	mmw
1032	1032	496	496	0	1	1	0	0	0x1008	0
1056	528	32	32	0	0	0	0	0xf040	0x40410	1
1200	1200	16	16	1	5	1(*)	0	0	0x160b0	0
1400	1300	16	16	1	10	2(*)	0	0	0x2f114	0
566	807	32	32	15	0	0	0	0	0x41597	2
....										

The two key information related to the fast mode are marked with asteriks (*). These singifies that the fast mode science window at (1200,1200) of size (16,16) is attached to fast area 1 and fast mode window at (1400,1300) of size (16,16) is attached to fast area 2.

6 Blue Fast Mode Area Information (DP_BFAST)

In the presence of the fast mode science window and allocated fast mode area, data sets of data type DP_BFAST will also be downlinked for later processing. See ICU-DPU protocol for detailed structural definition of this data type. A dump of an example DP_BFAST yields the following:

< Table 4. Part of Data Dump of DP_BFAST >

```
0000000  00f4 001a 0000 0001 0000 0005 0000 0100
0000020  0068 0000 006c 0000 0000 0000 0001 60b0
0000040  0001 60b1 0001 60b2 0001 60b3 0001 60b4
0000060  0001 60b5 0001 60b6 0001 60b7 0001 60b8
0000100  0001 60b9 0001 60ba 0001 60bb 0001 60bc
0000120  0001 60bd 0001 60be 0001 60bf 0001 62b0
0000140  0001 62b1 0001 62b2 0001 62b3 0001 62b4
0000160  0001 62b5 0001 62b6 0001 62b7 0001 62b8
0000200  0001 62b9 0001 62ba 0001 62bb 0001 62bc
....
0002020  ---- ---- ---- ---- ---- ---- 00f4 001a
0002040  0000 0002 0000 000a 0000 0100 006c 0000
0002060  0070 0000 0000 0000 0002 f114 0002 f115
0002100  0002 f116 0002 f117 0002 f118 0002 f119
0002120  0002 f11a 0002 f11b 0002 f11c 0002 f11d
0002140  0002 f11e 0002 f11f 0002 f120 0002 f121
0002160  0002 f122 0002 f123 0002 f314 0002 f315
....
```

Each DP_BFAST contains the following informations in sequence:

```
INT32 exposure_id;
INT32 fast_area_id (1 or 2);
INT32 sampling_time (integer multiple of MIC CCD frame time);
INT32 n_fast_pixels (<=512);
INT32 fast_destination;
INT32 fast_end;
INT32 n_fast_events;
INT32   fast_address[min(area_size,503)];
```

A direct translation of the DP_BFAST above yields the following information for the two blue fast areas:

```
        < Table 5. Translation of Data Dump of DP_BFAST >

BFAST 1 BFAST 2
------- -------

INT32 exposure_id; f4001a f4001a
INT32 fast_area_id (1 or 2); 1 2
INT32 sampling_time; 5 10
INT32 n_fast_pixels (<=512); 256 256
INT32 fast_destination; 0x680000 0x6c0000
INT32 fast_end; 0x6c0000 0x700000
INT32 n_fast_events; 0 0
INT32   fast_address[]; 0x160b0... 0x2f114...
```

A points need to be noted:

6.1. All science window in DP_WDW with mode 1 (fast mode) and non-zero sampling time (par1) much have a fast area number (par2) with a value of 1 or 2. The sampling time of the science window must match the sampling time reported in the appropriate DP_BFAST.

6.2. The sum of pixels from all fast mode windows (mode 1) in DP_WDW attached to the same fast area (specified by par2) must match the area_size of the DP_BFAST. In the above example, only one fast mode science window at (1200,1200) with size 16x16 is attached to BFAST 1. The n_fast_pixels of the BFAST 1 is 256 which equals 16x16. Same can be checked for BFAST 2. This should also apply and be checked against all multiple science window fast areas.

6.3. The numbers fast_destination and fast_end are provided for consistency check. They siginfy the allocated range for the storage of precompression fast mode formulated data, but not the exact number of fast mode records n_fast_events (which can only be ascertained AFTER the data are acquired.) The two blue fast areas SHOULD NOT have overlapping ranges.

7 Blue Fast Mode Data (DD_FST)

In this section, we discuss the reconstruction of the fast mode data to the extent that it can be analyzed as imaging and timing data. We assume that the proper data extraction and decompression have been applied.

The dump of an example DD_FST given below:

```
        < Table 6. Dump of An Example DD_FST Data Set >

0000000  00f4 001a 0000 a304 0000 0001 0000 0000
0000020  0000 03e7 ffff fc50 0000 154f 0000 4e6e
0000040  0000 0fdb 0000 1268 0000 1ace 0000 0005
0000060  0000 bd00 0000 1fff 0000 0507 0000 3de3
0000100  0000 0339 0000 3c4f 0000 18aa 0000 006f
```

```
0000120  0000 06f1 0000 0257 0000 1fa2 0000 0224
0000140  0000 1bee 0000 0005 0000 fd99 0000 1fff
0000160  0000 0b83 0000 1256 0000 3be1 0000 0f26
0000200  0000 1216 0000 095d 0000 02f9 0000 085f
0000220  0000 2203 0000 1ace 0000 0005 0000 e8ae
0000240  0000 1fff 0000 0334 0000 050c 0000 152b
0000260  0000 07e8 0000 3d52 0000 2c5e 0000 020c
0000300  0000 013b 0000 0bed 0000 0aa2 0000 0473
0000320  0000 1bee 0000 0005 0000 e44c 0000 1fff
0000340  0000 2e89 0000 4e7e 0000 0697 0000 0cbb
0000360  0000 0fe5 0000 1ace 0000 0005 0000 d73e
0000400  0000 1fff 0000 32df 0000 2051 0000 0816
0000420  0000 0e57 0000 3e9e 0000 01a4 0000 1bee
0000440  0000 0005 0000 e0df 0000 1fff 0000 3495
0000460  0000 08a5 0000 23e1 0000 17ea 0000 094d
0000500  0000 0404 0000 224d 0000 1ace 0000 0005
0000520  0000 dfa3 0000 1fff 0000 2e53 0000 59a7
```

The fast mode data follow a header for DD_ST of six words. For generality, the decompression algorithm provides 32 bit for every fast mode datum. To provide sufficient dynamic range, the reconstructed fast mode data stream should also be 32 bit integers. The fast mode data in the absolute 3D address format is reconstructed as follows:

7.1. Start with a 32-bit number called base, set it to zero.

7.2. For each fast mode datum which is NOT part of a frame crossing separator (see 7.3), the reconstructed fast mode datum is calculated as (base + fast mode datum). And the reconstructed datum is assigned as the new base. Or in a pseudo-code form:

```
repeat i
{
reocnstructed_datum[i] = base + fast_mode_datum[i];
base = reocnstructed_datum[i];
i = i+1;
}
```

Thus in the example in Table 6, the first four reconstructed data is 0x154f, 0x63bd, 0x7398, 0x8600.

7.3. The frame crossing separator must be carefully identified and processed. The key effect of frame crossing separator is the re-alignment of the base and it DOES NOT contribute entries to the reconstructed fast mode data stream.

7.3.1. The frame separator is jointly identified by the separator head (fixed and alternating between 0x1ace and 0x1bee), and the fixed separator tail of 0x1fff. Care must be taken in properly and simultaneously identifying the head and tail.

7.3.2. The word after the second word in the separator is the marker_remainder. It should have a value between 0 and sampling_time, inclusive. It signifies the number of markers encountered in the last slice forced by the tracking frame crossing.

7.3.2.1. If the value equals the sampling_time, then it signals that the tracking frame exactly coincides with a fast mode slice.

7.3.2.2. If the value does not equal the sampling_time, then the last slice prior to this frame crossing should be combined with the first slice following the frame crossing.

7.3.3. The third number in the separator, base_at_crossing, is the base used at the time of the frame crossing, and key to bridging the fast mode data reconstruction across the tracking frame crossing, with the following procedures.

7.3.3.1. The number base_at_crossing should be treated or converted to a signed value, using the 15-th bit as the sign bit.

7.3.3.2. A new base should be constructed by subtracting the signed value of base_at_crossing from the base calculated up to this point.

7.3.3.3. A consitency check must be performed at this point to make sure that the newly calculated base is an integer multiple of n_fast_pixels for this particular fast area.

7.3.3.4. If the tracking frame crossing coincides with a fast mode slice (i.e. the case 7.3.2.1), then the base should be decremented by n_fast_pixels.

7.4. The fast mode data reconstruction procedure runs through every datum with the algorithm of 7.2., and the processing of the frame crossing separator described in 7.3 which is triggered by the identification of separator head, tail and other characteristics. It should be emphasized again that the frame crossing separators DO NOT contribute entries to the reconstructed fast mode data stream.

8 Basics of Reconstructed Fast Mode Data Stream

The reconstructed fast mode data stream is ready for image and timing analysis. Here is a review of the nature of the reconstructed fast mode data stream. Let's say the array of 32-bit integers, A[], contain the reconstructed fast mode data stream.

8.1. The integer array B[] calculated by

B[] = A[]/n_fast_pixels

gives the photon arrival time in units of the sampling_time.

8.2. The array C[] calculated by

C[] = A[] modulo n_fast_pixels

gives the imaging location of the photons.

8.3. If the fast mode area consists of, say, two fast mode science window of size N1 and N2 (N1+N2=n_fast_pixels), then those photons with 0<=C[]<N1 belongs to the first science window and those with N1<=C[]<n_fast_pixels belongs to the second science window.