SUBJECT: SPIRE Peak-up Mode Requirements

PREPARED BY: K.J. King

DOCUMENT No: SPIRE-RAL-PRJ-001969

ISSUE: Draft 1 **Date:** 23rd March 2004

APPROVED BY: **Date:**

Distribution

Bruce Swinyard
Sunil Sidher

SergioMolinari

Change Record

ISSUE	DATE	Changes
Draft 1	23rd March	First issue for comments

TABLE OF CONTENTS

FIGURES

TABLES

Glossary

1. INTRODUCTION

1.1 Scope

The feasibility and implementation of a peak up mode for SPIRE has been discussed in RD01. This document describes the operational implementation of such a mode in terms of the instrument commanding and on-board processing required.

1.2 Documents

1.2.1 Applicable Documents

AD01 IID Part A

1.2.2 Reference Documents

RD01 Peakup mode implementation and simulation (SPIRE-RAL-NOT-001968), Issue 1.0, 22nd March 2004

2. PEAK-UP PROCEDURE

RD01 proposes that the position of the peak of a source may be found by obtaining samples of the source signal in positions forming a cross in the CHOP and JIGGLE directions. In order to determine the source signal the BSM will be chopped at each position and detector data taken at both on and off source positions. The OBS has the task of taking the data generated and calculating the offset to be applied to the telescope to bring the source on to the required pixel.

Note: RD01 assumes that he source is 'well behaved' – i.e. the source is significantly larger than any change in background signal across the scanned area and that there are no significant instrument changes (e.g thermal drift) that affect the sensitivity of the central pixel during the observation. If this Is not the case then a revised algorithm may be necessary.

As a general rule the OBS does not look into the science data that it receives from the DRCU and just formats and packages it into telemetry packets to be sent to the ground. In this case, however, it is necessary for the OBS to deal with the science data contents. To do this it must be made aware of how the data is taken and when the data will arrive in the data stream from the DRCU. This will be done in the first case by passing arguments to the Peak-up mode command and in the second case by ensuring that the input data stream is stopped and completely handled before starting the Peak-up procedure, which will consist of:

1. Stop the DCU science data generation (The OBS will not need data from the MCU or SCU so it is not necessary to stop these data streams)
2. Flush the DCU FIFO (This clears all detector data from the DPU buffers)
3. Send the commands to the instrument to perform the Peak-up Observation (This will generate a set of detector data which will accumulate in the DCU FIFO)
4. Send the Peak-up command to the OBS to perform the peak-up calculation (The OBS can read the data from the DCU FIFO and as a result of the calculation it will generate an TM event packet to cause the telescope to move to the correct position.)

Performing the procedure in this way puts constraints on the parameters of the Peak-up Observation:

PKP-PRO-010: The total DCU data generated by the Peak-up Observation shall be less than one half of the DCU FIFO size.
This ensures that the DCU 'FIFO half full' interrupt is never triggered by the observation. Thus the OBS can be sure that the data it requires is available in the FIFO when it receives the Peak-up command.

3. Peak-up Observation

Following RD01, the Peak-up Observation has to take data at positions corresponding to a cross, in the Chop and Jiggle directions, centred on the position corresponding to the central pixel lying on the source position.
For example for a 9 x 9 cross the following pattern will be covered:

J9

J8

J7

J6

C1 C2 C3 C4 C5/J5 C6 C7 C8 C9

J4

J3

J2

J1

The Observation will be implemented in the following way:

1. Set DCU data mode (assumed to be PSW to reduce the amount of DCU data generated)
2. Set initial Jiggle position
3. Take chopped data at this position
4. Repeat steps 2-3 for each Jiggle position
5. Set Jiggle position for central pixel
6. Take chopped data at this position
7. Repeat step 6 for each Chop position

The OBS does not store the observed position in any easily accessible manner so in order to be able to pass the list of jiggle and chop positions to the OBS they will be constrained:

PKP-OBS-010: The list of observed Peak-up mode Chop positions shall start with the minimum chop position, be equally spaced in the chop direction and shall contain an odd number of positions with the Chop position of the central pixel being the middle position in the list

PKP-OBS-020: The list of observed Peak-up mode Jiggle positions shall start with the minimum chop position, be equally spaced in the chop direction and shall contain an odd number of positions with the Jiggle position of the central pixel being the middle position in the list

PKP-OBS-030: The Peak-up positions shall be visited first in increasing Jiggle position at a fixed Chop position followed by increasing Chop position at a fixed Jiggle position.

For example for the 9 x 9 cross above the positions are visited in the following order: (C5,J1), (C5,J2), (C5,J3), (C5,J4), (C5,J5), (C5,J6), (C5,J7), (C5,J8), (C5,J9), (C1,J5), (C2,J5), (C3,J5), (C4,J5), (C5,J5), (C6,J5), (C7,J5), (C8,J5), (C9,J5).

This implementation can be described by 6 parameters:

1. Initial Chop position ($\mathbf{P_{chop}}$)
2. Increment between Chop positions ($\mathbf{D_{chop}}$)
3. Number of Chop positions ($\mathbf{N_{chop}}$)
4. Initial Jiggle position ($\mathbf{P_{jigg}}$)
5. Increment in Jiggle position ($\mathbf{D_{jigg}}$)
6. Number of Jiggle positions ($\mathbf{N_{jigg}}$)

The data taking at each position will be done using a set of CHOP Command List commands, which take the following parameters:

1. Chop on-source position
2. Chop off-source position
3. Number of Chop Cycles ($\mathbf{N_{cycles}}$)
4. Period of Chop cycle
5. No of BSM fames per Chop position
6. Time between BSM frames
7. No of DCU frames per Chop position ($\mathbf{N_{DCU}}$)
8. DCU frame sampling period
9. Delay from chop movement to start of DCU frames

Each execution of the CHOP Command List will generate $\mathbf{N_{DCU}}$ DCU frames on-source followed by $\mathbf{N_{DCU}}$ DCU frames off-source, repeated $\mathbf{N_{cycles}}$ times.

With this information it is possible to define the contents of the DCU FIFO after the execution of the Peak-up Observation:

4. PEAK-UP CALCULATION

The OBS will be commanded to execute the Peak-up mode with a telecommand containing the following arguments:

1. Increment between Chop positions ($\mathbf{D_{chop}}$)
2. Number of Chop positions ($\mathbf{N_{chop}}$)
3. Increment in Jiggle position ($\mathbf{D_{jigg}}$)
4. Number of Jiggle positions ($\mathbf{N_{jigg}}$)
5. Number of Chop Cycles ($\mathbf{N_{cycles}}$)
6. No of DCU frames per Chop position ($\mathbf{N_{DCU}}$)
7. The location of the reference pixel in the DCU Frame (16 bit word count from the start of frame), $\mathbf{N_{pixel}}$

PKP-CAL-010: The OBS is then required to execute the following algorithm to obtain the position of maximum signal in the Chop and Jiggle directions (**ChopPosn** and **JigglePosn)**:

1. Set **JiggleSignal** = 0
2. Set **JigglePosn** = 0
3. **n** = **-** int($\mathbf{N_{jigg}}$ /2)
4. Set **sample** = 0
5. For $\mathbf{N_{cycles}}$
    a. For $\mathbf{N_{DCU}}$ frames
        i. Add the value for $\mathbf{N_{pixel}}$ to **sample**
    b. For $\mathbf{N_{DCU}}$ frames
        i. Subtract the value for $\mathbf{N_{pixel}}$ from **sample**
    c. If **sample** is greater than **JiggleSignal**
        i. **JiggleSignal** = **sample**
        ii. **JigglePosn = n**
6. **n** = **n** + 1
7. If **n** less than $\mathbf{N_{jigg}}$ goto step 4
8. Set **ChopSignal** = 0
9. Set **ChopPosn** = **0**
10. **n** = - int($\mathbf{N_{chop}}$ /2)
11. Set **sample** = 0
12. For $\mathbf{N_{cycles}}$
    a. For $\mathbf{N_{DCU}}$ frames
        i. Add the value for $\mathbf{N_{pixel}}$ to **sample**
    b. For $\mathbf{N_{DCU}}$ frames
        i. Subtract the value for $\mathbf{N_{pixel}}$ from **sample**
    c. If **sample** is greater than **ChopSignal**
        i. **ChopSignal** = **sample**
        ii. **ChopPosn = n**
13. **n** = **n** + 1
14. If **n** less than $\mathbf{N_{chop}}$ goto step 12
15. If abs(**ChopPosn*** $\mathbf{D_{chop}}$) less than **MaxChop** AND abs(**JigglePosn * $\mathbf{D_{jigg}}$**) less than **MaxJiggle**
    a. Issue TM(5,1) event

*Note: **MaxChop** and **MaxJiggle** are the values of the maximum movement allowed in the Chop and Jiggle directions in BMS pointing units (these correspond to10 arcsec movement and will be determined by on-ground calibration)*

PKP-CAL-020: The OBS will issue an TM(5,1) event packet to notify the telescope of the movement required The contents of the event packet are defined in the Appendix. The format of the event packet is:

0 0 0 0 1	APID1
1 1	Count
Length =31	
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1	
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	
TIME	
EVENTID = 0x0504	
SID = 0x5101	
OBSID	
BBID	
EVENTCOUNT	
INSTRID	
THETAY	
THETAZ	
Checksum	

Parameter	Comment
EVENTID	0x0504
SID	0x5101
OBSID	Observation ID
BBID	Building Block ID
EVENTCOUNT	Sequential counter for TM(5,1) events
INSTRID	Instrument ID = 0x0002
THETAY	Rotation angle about Y axis (= **ChopPosn*** $\mathbf{D_{chop}}$)
THETAZ	Rotation angle about Z axis (= **JigglePosn *** $\mathbf{D_{jigg}}$)

5. IMPLEMENTATION

The Peak-up calculation could be implemented either directly in code in the OBS or as a Command List but this would imply further requirements:

PKP-IMP-010: The VM will support the extraction of science data from the DCU FIFO into a memory area accessible from a Command Linst

PKP-IMP-020: The VM will support the generation of a TM(5,1) event packet

APPENDIX

1. FORMAT OF PEAK-UP EVENT PACKET TO BE GENERATED BY SPIRE

This definition comes from IID Part A (section 5.12.5)

SPIRE uses an event packet (service TC(5,2)) to command the spacecraft a new pointing target based on:
- the previous inertial target (inertial pointing only)
- two µ-rotation angles (<10 arcsec) derived from the peak-up packet

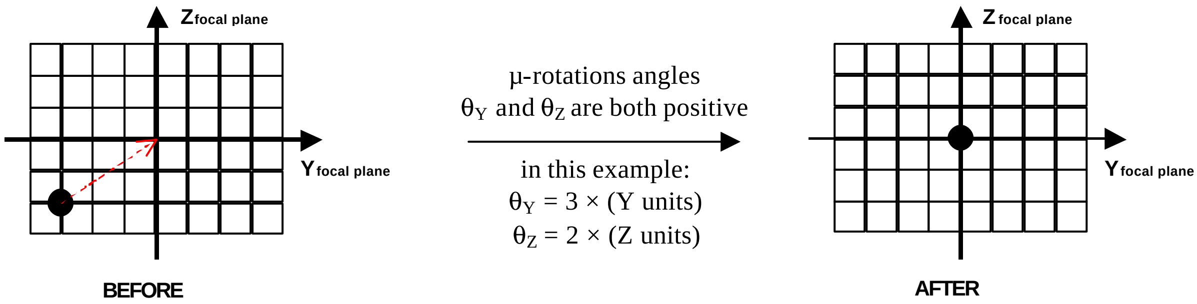
In order to have a consistency between the spacecraft and the instruments, and to have the same interface for both HIFI and SPIRE, the following interface is defined:

The event packet shall contain the following parameters:
- the relevant instrument (HIFI or SPIRE), coded on 16 bits (HIFI: 1, SPIRE: 2)
- the µ-rotation angle θ_Y, coded on 16 bits (see format below)
- the µ-rotation angle θ_Z, coded on 16 bits (see format below)

The normal to the focal plane is X, positively oriented as for the satellite X axis.

Sign convention:
The commanded µ-rotation angles correspond to the desired translation of the image inside the instrument focal plane frame. Note that with this convention, θ_Y (resp. θ_Z) does not represent the rotation around Y (resp. Z).

The signs shall follow this convention:



1.1 Absolute value convention:

The absolute value of both µ-rotation angles shall be given as 16-bit signed integers.
In accordance with the PS-ICD:
- bit 0 shall be the most significant bit
- bit 15 shall be the least significant bit

Bit 0 shall be used as the sign bit (0 for plus, 1 for minus)

1.2 Format of SPIRE packet parameters:

SPIRE identifier																µ-rotation angle Y																µ-rotation angle Z															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	M	.	.	.	.	.	.	.	.	.	.	.	.	.	.	L	M	.	.	.	.	.	.	.	.	.	.	.	.	.	.	L

Additional information required from instruments:
Some data are needed by the spacecraft to transform the peak-up parameters into attitude angles. They may be reprocessed by the ground after in-flight calibration.
The following set shall be provided for each instrument.

- $\mathbf{Y_{FP}}$ = 3 components of $Y_{focal\ plane}$ in satellite frame
- $\mathbf{Z_{FP}}$ = 3 components of $Z_{focal\ plane}$ in satellite frame
- $\mathbf{k_Y}$ = value of low significant bit used for µ-rotation angle Y, to transform integer θ_Y into radians. Angle [in rad] = k × integer.
- $\mathbf{k_Z}$ = value of low significant bit used for µ-rotation angle Z, to transform integer θ_Z into radians. Angle [in rad] = k × integer.