修士論文

ASTRO-H 衛星搭載 軟X線分光器 SXS の 波形処理システム PSP の開発

阿部 祐輝

指導教官:石崎 欣尚

首都大学東京大学院 理工学研究科 物理学専攻

2010年1月

目 次

第1章	はじめに	7
1.1	X 線天文学	7
第2章	ASTRO-H 衛星	11
2.1	衛星の概要・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	11
2.2	搭載検出器	12
2.3	期待される成果	13
笂り咅	\mathbf{Y} 组艺人力口力口以大一名 SYS	17
カリ早 91		17
ა.1 ეე	$ \begin{bmatrix} \mathbf{x} \mathbf{y} \mathbf{y} \mathbf{y} \mathbf{y} \mathbf{y} \mathbf{y} \mathbf{y} y$	17
ე.∠ ეე	$4 \text{ CTDO II SYS} (A-Ray Spectrometer) \dots \dots$	10
5.5	ASTRO-H SAS (Soft A-ray Spectrometer)	18
	3.3.1 SAS W安水性能	19
2.4		20
3.4		24
	3.4.1 SpaceWire の特徴	24
	3.4.2 Remote Memory Access Protocol (RMAP)	25
3.5	本修士論文の目的....................................	27
第4章	デジタル波形処理システム PSP	29
4.1	PSP の構成	29
4.2	PSP の性能要求	30
4.3	10100 (1000) (31
4 4	Mission I/O (MIO) $\vec{\pi} - \vec{F}$	32
1.1	4 4 1 FPGA	32
	4.4.2 MIOボードの構成	34
	4.4.3 XBoy - MIO 間のインターフェース	34
	A A A A A A > 1/I	25
4.5		30
4.0	Science Main Module	40
4.0	461 Science デー タの召信	40
	$4.0.1 \text{Science} \mathbf{J} = \mathbf{J} \\ 0 \\ $	40
	4.0.2 Science リークの 展開	41
	$4.0.3 \mathbf{y} \neq \mathbf{y} \neq$	41
	4.6.4 クリッノ・チャンネルエラー・Science テータエラーのチェック	42
		42
	4.6.6 時間微分の計算	43
	4.6.7 生波形・処埋テータの保存	44
4.7	Science Sub-module	45
	4.7.1 ピクセルパルスのトリガ (PXP)	45

		1		
	1	1		
1	-	1	ŀ	
		-	•	

	4.7.2 ピクセルパルス EDB (Event Dual Buffer)	. 48
	4.7.3 ピクセルノイズのトリガ (PXN)	. 49
	4.7.4 Anti-co パルスのトリガ (ACP)	. 50
4.8	その他のモジュール	. 51
	4.8.1 XBox TC Module	. 51
	4.8.2 XBox HK Module	. 52
	4.8.3 Clock Module	. 52
4.9	SpaceCard $\vec{\pi} - \vec{F}$. 54
	- 4.9.1 SpaceCard ボードの構造	. 54
	4.9.2 TOPPERS/JSP $\mathbf{D} - \mathbf{\lambda} \mathbf{\mu}$. 55
4.10	SpaceCard でのデジタル波形処理	. 56
	4.10.1 セカンダリーパルス 検出	. 56
	4.10.2 グレード付け	. 58
	4.10.3 最適フィルタ処理	. 59
	4.10.4 テンプレート計算	. 61
第5章	MIO - SpaceCard 性能評価試験	63
5.1	MIO – SpaceCard 間通信試験	. 64
5.2	XDS – PSP 間のインターフェース試験	. 65
	5.2.1 BASE_CLK, SMP_CLK, TLM_CLK $\ldots \ldots \ldots$. 66
	5.2.2 CMD_ENA & CMD_DAT / SCI_ENA & SCI_DAT / HK_ENA & HK_DAT $% \mathcal{A} = \mathcal{A} =$. 67
	5.2.3 各信号間のの Delay	. 68
5.3	MIO ScienceModule での Science データ取得試験	. 69
5.4	SpaceCard 処理速度測定	. 70
5.5	データ転送速度測定	. 71
	5.5.1 1 MIO + 1 SpaceCard 構成の場合	. 72
	5.5.2 1 MIO + 2 SpaceCard 構成の場合	. 72
第6章	Xbox-BBM — PSP-BBM 噛み合わせ試験	75
6.1	XBox-BBM	. 75
6.2	Detector Simulator	. 76
6.3	セットアップ	. 76
	6.3.1 LVDS 信号線の電圧チェック	. 76
	6.3.2 Merge XBox – PSP	. 76
6.4	XBox – PSP 間インターフェース試験	. 78
	6.4.1 BASE_CLK & TLM_CLK	. 78
	6.4.2 SMP_CLK & SCI_ENA	. 78
	6.4.3 CMD_ENA & HK_ENA	. 79
6.5	Science データ取得試験	. 80
	6.5.1 Signals from the Detector-sim	. 80
	6.5.2 Enable channels	. 80
	6.5.3 WFRB 内の Signal 波形の検証	. 81
	6.5.4 Detector-sim – XBox-BBM 間の接続変更	. 84
	6.5.5 $\mathbf{\mathfrak{g}}$ pixel $\boldsymbol{\mathfrak{o}}$ Science Data $\boldsymbol{\mathfrak{Fry}}\boldsymbol{\mathfrak{o}}$. 85
	6.5.6 波高値の比較 (ch 0)	. 87
	6.5.7 波高値の比較 (ch0 : Short pulse height)	. 87

6.6	XBox – PSP 間コマンド/レスポンス試験	88
	6.6.1 HK Status データのチェック	88
	6.6.2 バイアス電圧の HK Analog Statas 確認	88
第7章	PSP シミュレーションプログラムを用いた機能評価	93
7.1	PSP シミュレーションプログラム	93
7.2	使用データ	93
7.3	Science Module 処理後の Science データ	94
	7.3.1 パラメータ設定値	94
	7.3.2 トリガ情報のプロット	94
	7.3.3 PXP_PEAKFIND 範囲の変更	102
7.4	CPU 処理のシミュレーション	104
7.5	平均パルス、テンプレート作成	104
7.6	グレード付け	105
	7.6.1 loResPH vs derivMax	106
7.7	最適フィルタによる波高値計算	107
7.8	エネルギー分解能計算	109
	7.8.1 ベースライン分解能	109
	7.8.2 エネルギー分解能 (at Mn-K α : 5.9 keV)	109
	7.8.3 リニアリティー補正	110
	7.8.4 考察	112
第8章	まとめと今後	113
付録A	XDS データ書き換え手順	115
付録B	pxWFRB[0]-[17] 内の波形データ	117

第1章 はじめに

1.1 X 線天文学

紀元前から行われてきた天体観測は、近年まで可視光によるものがほとんどであった。しかし観測技術の発展と共に、電波や赤外、X線、 γ 線といった可視光以外の電磁波を捉えることが可能となり、宇宙物理学は目覚しい発展を遂げた。例えば図 1.1 はかに星雲を異なる波長域で見たイメージで、このように同じ天体を異なる波長で観測することにより、天体で起きている物理現象を詳細に研究することができるようになったのである。中でもX線は宇宙における高エネルギー現象を解明するのに役立つ。



図 1.1: 超新星残骸である「かに星雲」を可視光 (Hubble 衛星) と X 線 (Chandra 衛星) で見たイメージ。 波長によって全く異なるイメージが得られることが分かる。X 線イメージでは、中心に X 線源であるパ ルサーがいることが分かる。



図 1.2: 宇宙からの様々な波長の電磁波が、どこまで地上に届くかを表した図。可視光と赤外線の一部、 電波の領域に谷があり、これらの電磁波は地上に届く。それ以外は、大気の吸収を受けて、地上にはあ まり届かない。 X線は1885年 Wilhelm Röntgen によって発見された約0.1 から100 keVのエネルギー帯の電磁波の一種である。X線が太陽から出ていることは1950年代には知られていたが、図1.2 に示すとおり、X線は地球の厚い大気によって吸収されてしまうので、本格的なX線観測は衛星やロケット技術が発達した、1960年代に開始された。はじめて太陽以外の天体からのX線を検出したのは、1962年 Riccardo Giacconi および Bruno Rossi らによって打ち上げられたロケットによるものであった。月面で反射された太陽 X線を観測する予定が、偶然にもそれまで知られていなかった天体からの X線を検出したのである。その天体は後にさそり座 X1 と名付けられ、全天で最も明るい X線源であることが分かっている。この発見以降、多くの気球や X線天文衛星が打ちげられ、現在までに様々な天体が X線を出していることが分かった。図1.3 にこれまでの衛星およびロケット打ち上げのまとめを示す。Giacconi はこの発見により X線天文学の発展に貢献したことが認められ、2002年にノーベル賞を受賞している。

日本のX線観測は小田稔が考案した「すだれコリメータ」によって、さそり座X1の位置を同定する ことに成功したことから始まった。このコリメータは、2層のすだれ状のコリメータを検出器の上に置 くことで、入射X線の角度による強度変化を感知し、方向を知るという仕組みであり、X線撮像の難し い>10keV以上の太陽X線観測でもこの原理に基づく装置が今も使われている。すだれコリメータは 1979年に打ち上げられた日本の第一機目のX線天文衛星「はくちょう」に世界で初めて観測衛星に搭載 され、次々と新しいX線源の位置を決定することに貢献した。「はくちょう」以降も、最新の技術によ る新しい発見を求めて次々と天文衛星が打ち上げられ、最近では2005年に第5番目の「すざく」の打ち 上げが成功した。



図 1.3: 世界の宇宙 X 線観測の歴史。

X線による代表的な観測対象はプラックホールや中性子星、白色矮星などの高密度天体や銀河団といった巨大な天体、数千万度から数億度という超高温プラズマ、巨大なエネルギーを放出する超新星爆発など、地上ではほとんど再現不可能な極限状態の物理現象を持つ天体である。また宇宙から等方的でほぼ一様なX線も観測され、宇宙背景X線放射と呼ばれている。これはROSAT衛星による月のX線イメージ(図1.4 左)を見れば分かるように、太陽X線を反射していない月面より背景の宇宙の方がX線で明るく光っていることからも見て取れる。これは広がった天体によるものか、遠い点源の重ね合わせによるものかという議論があったが、日本の「あすか」衛星やアメリカのChandra衛星などによって、遠方にある活動銀河核(中心に巨大ブラックホールを持つ銀河)からの放射の重ね合わせでほぼ説明できることが明らかとなっている。さらに、近年ではおおよそ高エネルギー現象とはほど遠い存在に思われがちな太陽系内の惑星や彗星、衛星からのX線も検出されるようになってきた(図1.4 右)。今では金星や地球、火星、木星、土星といった系内のほとんどの惑星からX線が検出されている。

このように X 線観測は現在まで、飛躍的な成長を遂げてきた。これは過去の衛星から、現在稼働中の Chandra 衛星 (アメリカ) や XMM-Newton 衛星 (ヨーロッパ)、そして日本のすざく衛星に至る中で、 検出器や望遠鏡の性能が格段に向上してきたためである。X 線天文学のさらなる進歩のためには技術開 発を行い、新発見が可能な装置を搭載した衛星を打ち上げる必要がある。そこで日本では「すざく」に 続く新たな宇宙 X 線天文衛星 ASTRO-H の開発が進行中である。



図 1.4: 左は月の X 線イメージ。右側の光っている面が太陽 X 線が月面で反射している部分で、反対側 の面は周りよりも暗くなっている。右は Chandra 衛星による木星の X 線イメージ。極付近はオーロラ で、赤道付近は太陽 X 線の散乱で光る。

第2章 ASTRO-H衛星

2.1 衛星の概要

ASTRO-H 衛星は 2014 年打ち上げ予定の日本の第6番目と なる X 線天文衛星である。2005年に日本から打ち上げられた 「すざく」衛星 (ASTRO-EII) の後に、これに続く X 線天文衛 星として New exploration X-ray Telescope (NeXT)の検討が 進められていた。そして2008年に名を改め、正式に科学衛星計 画として発足したのが ASTRO-H 衛星である。現在 JAXA を 中心とした日本の研究機関や大学、NASA、ESA などの国際協 力のもと開発が進んでいる。「すざく」衛星に比べてASTRO-H には、従来の半導体検出器を1桁しのぐ軟X線の超高精度分 光性能と、硬X線望遠鏡による集光撮像能力が追加される。こ れによって、銀河団成長の直接観測や超新星残骸の膨張速度の 直接観測、硬X線背景放射の元となる巨大ブラックホールの発 見、さらにはブラックホール重力の効果の理解など、様々な宇 宙の物理状態が解明できると大きく期待されている。ASTRO-Hには4種類の観測システムが搭載され、0.3 ~ 600 keV と いう広いエネルギー帯域を観測することが可能である。また、 ASTRO-H 衛星内の通信ネットワークには SpaceWire という共 通規格が実装される (3.4 参照)。以下に ASTRO-H の概要と搭 載される各検出器の性能を示す。



図 2.1: ASTRO-H 衛星の完成予想図

望遠鏡焦点距離	軟 X 線系:5.6 m 硬 X 線系:12 m
観測軌道	高度 550 km、軌道傾斜角 31 度の略円軌道
外形寸法 (観測時)	$7.7 \text{ m (W)} \times 2.9 \text{ m (D)} \times 13.8 \text{ m (H)}$
	軌道上で伸展式光学台を約6m伸展
衛星質量	2,600 kg (推薬、液体ヘリウム含む)
発生電力	3,500 W 以上
バス電圧	非安定 50V 系 (リチウムイオンバッテリー使用)
打ち上げ	2013 年度冬期 (H-IIA デュアルロンチを想定)
ミッション期間	3年以上(目標5年)

表 2.1: ASTRO-H の概要



図 2.2: ASTRO-H の全体図

2.2 搭載検出器



図 2.3: 各検出器の配置

観測システム名		基本仕様
軟 X 線分光システム	エネルギー帯域	0.3 - 10 keV
	エネルギー分解能	< 7 eV (@ 6 keV)
SXS (X 線マイクロカロリメータ)	有効面積	$210 \text{ cm}^2 (@ 6 \text{ keV})$
+	画素数	6×6 pixel
SXT-S (軟 X 線望遠鏡)	視野	~3×3 分角
	角度分解能	< 1.7 分角
	動作温度	47 mK
軟 X 線撮像システム	エネルギー帯域	0.3 - 12 keV
	エネルギー分解能	< 150 eV (@ 6 keV)
$SXI (X \ \& CCD)$	有効面積	$360 \text{ cm}^2 (@ 6 \text{ keV})$
+	画素数	$6 \times 1024 \times 1024$ pixel
SXT-I (軟 X 線望遠鏡)	視野	~ 35×35 分角
	角度分解能	< 1.7 分角
	動作温度	-120 °C
硬 X 線撮像システム	エネルギー帯域	5 - 80 keV
	エネルギー分解能	1 - 2 keV (@ 60 keV)
HXI (硬 X 線撮像検出器)	有効面積	$300 \text{ cm}^2 (@ 6 \text{ keV})$
HXT (硬 X 線望遠鏡)	視野	$\sim 9 imes 9$ 分角
	角度分解能	< 1.7 分角
	動作温度	-20 ± 5 °C
軟ガンマ線分光システム	エネルギー帯域	10 - 600 keV
	エネルギー分解能	2 keV (@ 40 keV)
SGD (軟ガンマ 線検出器)	有効面積	$> 200 \text{ cm}^2$ (Photo absorption mode @ 30 keV)
		$>30~{\rm cm}^2$ (Compton mode @ 100 keV)
	動作温度	-20 °C

表 2.2: ASTRO-H の観測システム

2.3 期待される成果

ASTRO-Hの期待されている科学的成果の一つに銀河団成長の直接観測がある。ダークマターを含む 銀河団の総質量を測定し、距離(赤方偏位)の関数として調べることで銀河団成長の理解を深めることが できる。そのためには「すざく」衛星などの従来のX線観測で行われてきた高温プラズマの熱的エネル ギーの測定のみならず、プラズマの乱流や衝突といったバルクな運動エネルギー、加速された粒子のも つ非熱的エネルギーを測定することで銀河団の総エネルギーを正確に知る必要がある。そこから銀河団 を取り巻くガスの圧力を導くことができるため、静水圧平衡の観点から、ガスを閉じ込めるのに必要な 銀河団の総質量を計算することができると考えられている。

銀河団プラズマの典型的な熱運動の速度は $100 \sim 1000 \text{ km/s}$ であると考えられており、エネルギーのドップラーシフトや輝線幅からこの高温プラズマのバルクな運動エネルギーを測定することができる。例として 6.7 keV の輝線に対して $\sim 20 \text{ eV}$ のエネルギー分解能があれば 1000 km/s 程度の速度を知ることができ、 $\sim 2 \text{ eV}$ あれば 100 km/s 相当の速度が分離できるため、より精密な分光能力が要求される。さ

らに、優れたエネルギー分解能が実現できれば、エネルギースペクトル上で各輝線の微細構造を分離す ることが可能となる。光学的に薄いプラズマから出る異なる電離状態の鉄輝線などを分解できれば、輝 線強度比からプラズマの温度や密度といった物理状態を精度良く決定することができるようになる。図 2.4 に 2 eV のエネルギー分解能で得られるスペクトルのシミュレーションを示す。



図 2.4: Warm Hot Intergalactic Medium と呼ばれる、銀河団同士を結ぶ宇宙の大規模構造に沿って存 在すると思われる希薄なプラズマからの放射シミュレーション。2 eV のエネルギー分解能を持つ X 線マ イクロカロリメータ (実線)と典型的な半導体検出器で見た場合 (点線)を比べた。

従来の X 線天文衛星で広く使われているのが半導体検出器 (X 線 CCD など) である。X 線の光電吸収 によって生じた電子ホール対を電気的に読み出すことで、X 線のエネルギーを決定することができ、さ らにピクセル化することで優れた撮像能力を持つ。しかし電子数の揺らぎによって、エネルギー分解能 は理論的に 5.9 keV の入射 X 線に対し約 120 eV となり、図 2.4 のように輝線の微細構造を充分に分離す ることができない。

さらにエネルギー分解能がより良い分光法としては、回折格子があり、これは米 Chandra 衛星や欧 XMM-Newton 衛星に搭載されている。X 線のエネルギーを回折によって、位置情報に焼き直すことで、 エネルギー分解能は1 keV 以下の入射 X 線に対し、約2 eV を達成できる。しかし、広がった天体には 使用が困難であり、また位置情報を読むことから、波長の短い、すなわちエネルギーの高い X 線に対し てはエネルギー分解能が悪くなる。さらに、光を分散させるため、検出効率が低いことも欠点である。 以上のことから、回折格子による分光観測は実質上、エネルギーの低い軟 X 線の明るい点源に限られて しまい広がった天体には適していない。

ASTRO-H 衛星には、上記2つの方法とは異なる分光装置、X 線マイクロカロリメータ (SXS) が搭載 され、半導体検出器を一桁しのぐエネルギー分解能が備わる。この SXS 装置を用いれば、銀河団のよう な広がった天体でも十分なエネルギー分解能で分光観測することができ、これによって、銀河団プラズ マの速度を知ることで運動エネルギーが測定可能となる。図 2.5 に検出器の性能比較 (分光性能と有効 面積)を示す。SXS は2 keV 以上のエネルギー帯域で、エネルギー分解能と有効面積共に他の検出器に 比べて高い性能を持っていることが分かる。

また、HXI や SGD のもたらす硬 X 線領域のスペクトルから銀河団内で加速された相対論的な電子の 非熱的放射を検出することで非熱的エネルギーを測ることができる。このように ASTRO-H 衛星の観測

によって、銀河団の質量測定の精度が大幅に上がると期待されている。 次章では、我々の開発チームである X 線マイクロカロリメータ SXS の詳細について述べる。



図 2.5: 左:入射エネルギーに対する分光能力 (エネルギーを分解能で割ったもの)の比較。右:入射エネルギーによる有効面積の比較。有効面積はX線望遠鏡の面積、冷凍機の入射窓の透過率、および検出器の検出効率を考慮したもの。

第3章 X線マイクロカロリメータ SXS

3.1 X線マイクロカロリメータ

観測天体や X 線のエネルギーによらず、高いエネルギー分解能を持つ分光器が X 線マイクロカロリ メータである。X 線マイクロカロリメータとは、入射エネルギーを素子の微小な温度上昇として読み 取る検出器で、X 線を吸収するための吸収体、温度を読み取るための温度計、吸収した熱を逃がす熱浴 から成る。図 3.1 に構造の概略図を示す。エネルギー E の X 線が吸収体に入射すると、素子の温度は $\Delta T = E/C$ だけ上昇する。ここで C は吸収体と温度計の熱容量の合計値である。吸収された熱はC/G(G は熱浴との熱伝導度)の時定数で指数関数的に熱浴へと逃げて、素子は元の温度に戻る。

現在、最も一般的な X 線マイクロカロリメータは、温度変化を電気抵抗値の変化として読み出す抵抗 温度計方式である。抵抗温度計での一般的なエネルギー分解能は

$$\Delta E_{\rm FWHM} = 2.35 \sqrt{\frac{4k_{\rm B}T^2C}{|\alpha|}} \sqrt{\frac{n}{2}}$$
(3.1)

という式で表される。ここで α は温度計の感度を表す無次元量であり、 $\alpha = d \log R/d \log T$ と定義される。半導体温度計では、 α は負の値で典型的に $-7 \sim 0$ である。また、n は熱伝導度の温度依存性 $(G = KnT^{n-1}$ 、ただし K は定数) による項で、 $3 \sim 5$ の値を取る。熱容量の温度依存性も考慮すると $(C \propto T^3)$ 、式 (3.1) からエネルギー分解能は温度に強く依存することが分かる。そのため、極低温 (~ 100 mK) で動作させることが高いエネルギー分解能を得るためには必須となる。



図 3.1: カロリメータの概略図

3.2 「すざく」XRS (X-Ray Spectrometer)

X 線マイクロカロリメータは 2005 年 7 月に打ち上げられた日本の「すざく」に世界で初めて衛星搭載された。図 3.2 左が「すざく」に搭載された X 線マイクロカロリメータ XRS (X-Ray Spectrometer) であり、6×6 ピクセル並んでいる。XRS の温度計はリンとホウ素をドープしたシリコンで作られている。図 3.2 右に示す。さらに温度計の端には SU8 というポリマーの円柱を配置し、その上に1 ピクセル 当たり 624×624 µm サイズの吸収体を付けて、図 3.2 中央のような形になる。吸収体は X 線の吸収効率 が良く、吸収した X 線をより速く熱に変えて素早く拡散させることができ、なおかつ比熱が小さい方が 良い。そこで XRS では水銀テルルを採用した。その結果、軌道上で 7 eV のエネルギー分解能を達成し た (図 3.3)。XRS の諸元を表 3.2 にまとめる。しかしながら、XRS は軌道上でのヘリウム消失という不 慮の事故のため、打ち上げ後 1ヶ月あまりで機能停止となり、宇宙観測には至らなかった。



図 3.2: 左:XRS の写真。中央:左図の拡大写真。中央がX線受光面になっており、周りに配線が伸び ている。右:温度計1ピクセルの拡大図。



図 3.3: 軌道上で XRS が取得した、校正用の線源 55 Fe からの Mn K $_{\alpha}$ のスペクトル。

3.3 ASTRO-H SXS (Soft X-ray Spectrometer)

そこで私の所属する日本のX線マイクロカロリメータグループは、米欧と共同で、2014年打ち上げ予定のX線天文衛星ASTRO-Hに新しいX線マイクロカロリメータSXS (Soft X-ray Spectrometer)を搭載し、世界初の宇宙観測を目指している。SXS は現在、日米と欧州の国際協力のもと開発が進められている。それぞれ米国がセンサー部、断熱消磁冷凍機 (ADR)、開口部とフィルタ、検出器のアナログ部、ADR 制御回路、日本が室温から He 温度までの冷却系、機械式冷凍機、冷凍機駆動回路、デジタル信号処理回路、SpaceWire ルーターなどの通信ネットワーク周り、欧州 (オランダ、スイス) がフィルターホイールとX線発生装置の部分を担当している。図3.4 に SXS の構成のブロック図を示す。



図 3.4: SXS ブロック図

3.3.1 SXS の要求性能

SXSのサイエンスを満たすために求められている SXSの要求性能 (Scientific Requirements: SRs) は以下の通りである。

#	要求事項	要求値	目標値
SR01	Energy resolution in FWHM (eV)	7	4
SR02	Energy range (keV)	0.3 - 12.0	0.1 - 16.0
SR03	Residual background $(s^{-1} \text{ keV}^{-1})$	1.5×10^{-3}	1.5×10^{-3}
SR04	Field of view $(\operatorname{arcmin}^2)$	2.9×2.9	2.9×2.9
SR05	Detector array	6×6	6×6
SR06	Angular resolution in HPD (arcmin)	1.7	1.3
SR07	Effective area at 1 keV (cm^2)	160	160
	Effective area at 6 keV (cm^2)	210	210
SR08	Lifetime (yr)	3	5
SR09	Energy scale accuracy (eV)	2	1
	Energy resolution accuracy (eV)	2	1
SR10	Count rate per array $(s^{-1})^a$	150	2500^{b}
	Count rate per pixel $(s^{-1})^a$ 20		20
SR11	Absolute time-assignment accuracy (ms)	10	0.08
	Time-assignment resolution (ms)	1	0.08

表 3.1: ミッションサイエンスからの要求

3.3.2 SXS の構成機器

ここでは主に検出器から信号処理系の機器について説明する。

センサー部 (CDA: Calorimeter Detector Assembly)

SXS のセンサー部は基本的に XRS と同じ半導体温度計型カロリメータを継承しているが、いくつか の点で性能をより強化している。開発は NASA/GSFC が担当している。半導体型カロリメータは低い 温度で動作させればさせるほど、比熱が小さくなり、エネルギー分解能が良くなる (式 3.1)。そのため、 SXS では XRS の時よりも動作温度を 10 mK 下げて 50 mK とする。さらに、XRS の吸収体として用い ていた水銀テルルにカドミウムを注入することで、比熱を水銀テルルの ~ 1/4 に抑えることで、地上実 験で 6 keV の入射 X 線に対し、4.2 eV という XRS を凌ぐエネルギー分解能が得られている。また、比 熱を抑えることで素子の大きさも巨大化が可能となるため、有効面積も XRS に比べて大きくなる。表 3.2 に SXS の性能をまとめた。



図 3.5: NASA で制作された検出器の Engineering Model (フライト品と同等の性能をもつプロトタイプ)。図では一部吸収体が外されている。

表 3.2: XRS および SXS の設計値。				
	「すざく」XRS	ASTRO-H SXS		
エネルギー分解能 (FWHM)	$7~{ m eV}$	要求 7 eV/目標4 eV		
有効面積 @ 6.7 keV	$130 \ \mathrm{cm}^2$	$230 \ \mathrm{cm}^2$		
視野	$2.9' \times 2.9'$	$2.8' \times 2.8'$		
吸収体サイズ	624 \times 624 \times t8 $\mu {\rm m}$	814 \times 814 \times t8 $\mu \rm{m}$		
アレイサイズ	6×6	6×6		
動作温度	$60 \mathrm{mK}$	$50 \mathrm{~mK}$		
寿命	2.5~3 年	要求3年/目標5年		

Anti-coincidence 検出器

SXS には宇宙線の入射などといった X 線以外のバックグラウンド除去のために ani-coincidence 検出 器を持っている。anti-co 検出器の応答はカロリメータの応答よりもかなり早く、時定数は < 0.5 msec である。したがってカロリメータ自身がパイルアップによってサチってしまうような場合でも診断を続けることができる。anti-co 検出器はシリコンの PIN 型検出器で、抵抗率 13 ~ 21 k Ω /cm (室温) の n 型半導体の片面にボロンを、片面にリンを打ち込んで、それぞれ p+ 層と n+ 層としている。大きさは 10 mm × 10 mm、厚さは 500 μ m で、カロリメータアレイの 0.63 mm 下に置かれる (図 4.23)。anti-co は 1 台搭載され、読み出しは次節で述べる XBox で行なう。



図 3.6: anti-co 検出器の位置

Cold JFET (Junction Field-Effect Transistor)

X 線カロリメータで検出した信号は、JFET (接合型電界効果トランジスタ)で読み出しを行う。カロリ メータの信号は典型的には 100 μ V 程度で、インピーダンスは 50 MΩ にも達する。このカロリメータか らの高インピーダンス信号は、初段の JFET ソースフォロワでインピーダンス変換され (出力インピー ダンス ~ 1 kΩ at 1 kHz)、差動出力のまま XBox に送られる。ただし JFET はヘリウム温度では動作 せず、100 K 程度に暖める必要がある。SXS ではヒータによって JFET を 130 K の熱浴で動作させる ようにしており、9 nV/ \sqrt{Hz} (目標: 4 nV/ \sqrt{Hz}) 程度の低い雑音レベルを実現している。そして JFET の熱が検出器に入り込まないように、二重のボックスに収めて内側の JFET ボックスは熱伝導度の低い Kevlar ワイヤーで吊り下げる構造になっており、ヒータの熱のほとんどは断熱消磁冷凍機 (ADR)の外 部に位置する予冷用デュワーシールド (28 K) に排熱される (図 3.9)。JFET は各系毎に、JFET パッ ケージと呼ばれる 1 枚のボードにまとめられている (図 3.8)。表と裏にそれぞれ 10 チャンネルの JFET (9pixel + 1 anti-co) を搭載することができる (片側は anti-co が無いため 9 チャンネル)。JFET パッ ケージ には温度計と温度制御用ヒータが 1 つずつ付いている。JFET の駆動、ならびに JFET ボード の温度制御と温度計測は次節で述べる XBox で行う。



図 3.7: JFET ボックス



図 3.8: JFET パッケージ (片側): 左から温度計、ヒーター、10 JFETs



図 3.9: 熱フローマップ

XBox (X-ray Box)

カロリメータで検出した信号信号は、次に XBox (X-ray Box) と呼ばれる SXS のアナログ信号処理部 へと送られる (図 3.10)。XBox は XBox-A, XBox-B と 2 系統搭載され、それぞれ独立してカロリメータ アレイ 18 ピクセル + anti-co 1 チャンネルの信号を扱う。開発は NASA/GSFC が担当している。XBox は以下の様な機能を果たす。

- 検出器からの信号のサンプリング、多重化
- 信号の増幅
- A/D 変換
- カロリメータや JFET への電圧供給、モニタリング
- 検出器の温度制御、モニタリング

XBox 内で A/D 変換されたデータは電磁波ノイズの少ない LVDS (Low voltage differential signaling) で差動出力のまま PSP (Pulse Shape Processor) と呼ばれる装置へと送信される。



図 3.10: XBox-BBM (Bread Board Model: 試験用モデル) 外観

PSP ((Pulse Shape Processor))

PSPはJAXA/ISASを中心に日本側が開発担当しているSXSのデジタル波形処理装置である。XBox でデジタル変換された波形データは、PSP内でさらにデジタル波形処理が施され、パルスのトリガや詳 細な波高値計算が行われる(次章)。

DIST

DIST は、衛星のバス電源を SXS の各機器へと変圧して電源供給を行う機器である。衛星からは、32.5 - 52.0 V 非安定化電源が供給されるので、それを変圧し供給する。また、current limit を設けて電流値を監視しており、電源供給の ON/OFF を行うことができる。

SWR

SXS-SWR は、SXS の各 SpaceWire コンポーネントと衛星内ネットワークとを接続するルータである。冗長系のため電気的に同等な SXS-SWR A/B の 2 系統が存在し、それぞれ各機器と同様に接続されている。開発は NEC が担当している。

3.4 次世代衛星通信規格 SpaceWire

ASTRO-H 衛星では、各衛星機器間の通信を SpaceWire と呼ばれる規格で通信を行う。従来の衛星で は、衛星搭載機器間での通信インターフェイスは衛星ごと、機器ごとに開発されており、各々独自の規 格が使用されていた。そのため、開発期間の長期化、信頼性確保が困難になる、技術が継承されないと いった問題が生じており、科学衛星開発における研究者の負担を大きくしていた。SpaceWire は、この ような衛星搭載のデータ収集システムの問題点を解決するためのネットワーク規格の一つであり、世界 の科学衛星の共通信格を目指したネットワークプロトコルである。SpaceWire は IEEE1355 を元に宇宙 用に改良されたもので、ESA を始めとして、ISAS/JAXA、NASA など世界の主要な研究機関や大学、 企業などが中心となって開発を進めている。SpaceWire により機器間 通信インターフェイスが統一され ることで、装置開発と通信インターフェイス開発を完全 に分離することができる。そのため、装置開発 と通信インターフェイス開発の相互干渉がなくなり、高い信頼性を実現することができる。さらに、全 ての装置が共通の通信インターフェイスを持つことで、機器同士の接続の自由度が高くなり、より柔軟 なシステムの構築が可能になる。

3.4.1 SpaceWire の特徴

SpaceWire は IEEE1355 をベースに開発された高速シリアル通信規格である。信号伝送には Low Voltage Differential Signaling (LVDS) が用いられ、低消費電力で高速通信が可能である。また、+ と – の信号をクロスさせて伝送する差動インタフェースであるため、ノイズに強く、信号線からのノイズ 輻射も少ないという特徴がある。SpaceWireのリンク速度は2 Mbps ~ 400 Mbpsと可変であり、様々な 規模の装置にも柔軟に対応することができる。さらに、ルーター機能を使うことによって様々なネット ワークを組むことができるため、メッシュ状のネットワークを構成することができる。これによりデータ 転送の際に回線の途絶や混雑が生じても迂回ルートを確保することができ、信頼性の高いシステムを実 現できる。SpaceWire に用いるケーブル線はツイステッドペアになっており、ノイズに強い構造となっ ている (図 3.12)。ケーブルは最大で 10m 以上伸ばすことができるため、衛星上での自由な機器配置が 可能である。また、使用されるコネクタは9ピンのD-subコネクタを使用している。SpaceWireによる データ転送は、全てパケット単位で行われる。SpaceWire で規定されるパケットには2種類あり、アド レス (宛先)を指定して任意のデータを送るパケットとあらかじめ内容が決められている最優先コードに 分けられる。前者は図 3.11 のような構造をもったパケット構造になっており、Destination Address は パケットの送り先を示すアドレス部である。SpaceWireの規定ではパケットの先頭を示すフラグがない ため、1 パケットの終わりを示す End of packet marker の次に来るデータをパケットの先頭と解釈する。 後者の最優先コードには Time-Code が含まれる。Time-Code とは衛星搭載機器全てにブロードキャス トされる衛星全体の共通時刻情報である。ASTRO-H 衛星では、GPS 衛星から取得した絶対時刻のタイ ミングから Time-Code を生成し、衛星機器全体へと配信する。各衛星機器はこの Time-Code を元に時 刻の同期を行う。

Destination address			
Cargo			
End of packet marker			

図 3.11: SpaceWire パケットのフォーマット



図 3.12: SpaceWire ケーブルの構造

3.4.2 Remote Memory Access Protocol (RMAP)

ネットワークの末端機器のメモリー空間を操作するためには、その末端機器に Central Processing Unit (CPU)を搭載するのが簡単な方法である。しかし、衛星内のスペース、電力、重量等の制限のなかで末 端機器まで CPU を搭載することが難しい。そこで ASTRO-H 衛星の通信規格 SpaceWire では、CPU 非搭載の機器のメモリー空間まで操作できるプロトコルとして RMAP が規格化されている。Remote Memory Access Protocol (RMAP) は、上位の CPU 搭載ノードから、あたかも自分のメモリ空間の一 部として、ネットワーク末端機器の搭載メモリにアクセスできる機能を提供する。SpaceWire の上位の レイヤーに相当する RMAP はプロトコルレベルで規定されているので、末端機器に CPU を搭載する 等の特別な実装を施す必要がない。



図 3.14: SpaceWire レイヤと RMAP レイヤの関係

図 3.14 に簡単な RMAP パケットのやりとりのイメージを示す。RMAP は Command パケットと Reply パケットの組 (Transaction) をひとつの単位として動作する。Command を出す側を Initiator、 受け取る側を Target と呼ぶ。RMAP には主に、RMAP Write (メモリ書き込み) と RMAP Read (メ モリ読み出し) の 2 つの命令が用いられる。RMAP Read の場合、Initiator がコマンドを送信すると、 Target は要求されたアドレスに対応する値が含まれる Reply パケットを Initiator に返す。Initiator が RMAP Write コマンドを送信すると、Target は要求されたアドレスのメモリを書き換えて、書き換え たことを示す Reply パケットを Initiator に返すようになっている。

ASTRO-H では、データ収集を司る SMU (Satellite Management Unit) を頂点とする tree 状のネットワーク構成をとり、SpaceWire RMAP を用いて、末端の搭載機器にコマンド指令を送信したり、観測装置などから発生する観測データ (テレメトリー) を収集などを行う。テレメトリー・コマンドの送受信の方式としては、RMAP Write で送りつける PUSH 方式と、RMAPread で読みに行く PULL 方式とが考えられるが、ASTRO-H では、コマンド送信は SMU から末端機器に PUSH し、テレメトリー受

信は SMU が末端機器から PULL する方式をとる。これは、中央コンピュータ SMU の設計を簡略化し、 衛星内データのやり取りを SMU で一括管理するためである。

3.5 本修士論文の目的

前述のとおり、XRS では宇宙観測に至らなかったため、依然として X 線マイクロカロリメータでの宇 宙観測は実現していない。SXS での冷却系は、より優れた性能を高い信頼性で実現するための最適化が 行われている。また、信号処理系のアルゴリズムの基本部分は XRS のものを継承しているが、ハード ウェア・ソフトウェアは共に新設計となる。さらに新たな通信規格 SpaceWire を用いることで信頼性を 高めている。現在、我々のグループは三菱重工業 (MHI)の協力のもと、SXS のデジタル波形処理シス テムを司る PSP (Pulse Shape Processor)の開発を行っている。本修士論文では、性能実証モデル (EM) の製作に向けて試験用モデル (BBM)を用いて実証試験を行った。PSP の機能の挙動を確認し、実験デー タを元に、FPGA にボードに実装されるロジックの最適化することで最終的な仕様を決定した。また、 PSP に搭載するアルゴリズムと同等のシミュレーションプログラムを構築し、ソフトウェア上で実デー タを処理させることでトリガされるイベントに対して考察を行った。

以下では、第4章でPSPの機能と各種モジュールの説明を行う。第5章ではXBoxシミュレーターを 用いた接続試験、さらに第6章ではDetectorシミュレーター·XBox-BBM·PSP-BBMの全信号処理 系を接続した噛み合わせ試験について述べる。続いて第7章ではPSPシミュレーションプログラムを用 いたパルス処理の考察について報告する。

第4章 デジタル波形処理システム PSP

PSP (Pulse Shape Processor) は ASTRO-H 衛星に搭載される SXS デジタル信号処理部を担う装置で ある。設計・開発は、首都大、埼玉大、JAXA/ISAS が担当し、設計・製造は三菱重工業 (MHI) が担当 している。PSP を構成する各種ボードの詳細は節 4.4, 節 4.9 で説明する。

4.1 PSPの構成

PSP 周りのブロック図を図 4.1 に示す。PSP は PSP-A, PSP-B の 2 系統が搭載され、それぞれは独立 して機能する。それぞれの XBox から送られてきた Science データ (18 ピクセル + anti-co 1 チャンネ ル) は接続された PSP に LVDS で送信される。PSP に送られた波形データは各チャンネル毎に用意され たバッファ内に格納され、イベントパルスのトリガ、時刻付け、詳細な波高値解析が行われる。それぞ れの PSP は次の 4 つのボードから構成されている。

- Mission I/O ボード (MIO) × 1 (4.4 節)
- SpaceCard ボード \times 2 (4.9 節)
- Power Supply Unit (PSU) $\vec{\pi} \vec{F} \times 1$

それぞれの PSP ユニットのサイズは 188.0×263.0×103.0 mm³ で、重さが 5.773 kg (BBM の場合)。 電源は DIST (distributer) から PSU を介して供給され、入力電圧は 32 - 52V、個々の消費電力は 14.2W (最大 18.9W) である。



図 4.1: PSP とその周辺機器のブロック図

4.2 PSP の性能要求

SXS の設計とサイエンスの要求を満たすための PSP の性能要求 (Performance Requirements: PRs) はいかのとおりである。なお、括弧内は小節 3.3.1 に述べた Scientific Requirements との対応を示している。

[処理全般に対する要求]

- **PR1-01** 観測モード時は全 36 カロリメータピクセルチャンネルと 2 anti-co チャンネルを連続的に処理 を続けこと (SR04, 05)。
- **PR1-02** 2 系統の PSP (PSP-A, B) の内、片方の PSP が失われた場合でも、もう片方の PSP で 18 カ ロリメータピクセルチャンネルと 1 anti-co チャンネルの処理は継続できること。
- PR1-03 PSP はすべてのパルスを観測時間の2%以下の dead time で検出すること。

[カロリメータパルスに対する要求]

- **PR2-01** PSP は 0.3 12.0 keV のイベントを検出すること (SR02)。
- PR2-02 PSP は 12.0 keV を検出できるダイナミックレンジをもつこと (SR02)。
- **PR2-03** 5×(rms noise) より大きいピークをもつパルスを検出すること (SR02)。
- **PR2-04** 重なったパルスで 5 ms 以上離れているものを PR2-03 を満たしつつ検出すること (SR01, 07, 09)。
- PR2-05 重なったパルスで 2 ms 以上離れているものはコントラストが 30 よりも小さければ検出する こと (SR01, 07, 09)。
- **PR2-06** 全てのパルスに対して high-resolution (HR), medium-resolution (MR), low-resolution (LR) かつ、primary パルスか secondary パルスのグレード付けを行うこと (SR01)。
- **PR2-07** HR イベントの処理でのスペクトルの分解能劣化を 0.5 eV 以下に抑えること (SR01, 09)。
- **PR2-08** MR イベントの処理では、HR イベントの分解能 + 1 eV 以下に抑えること (SR01, 09)。
- PR2-09 LR イベントの処理では、HR イベントの分解能の 10 倍以下に抑えること。
- **PR2-10** PSP は 150 cts s⁻¹array⁻¹ のカウントレートを処理可能であること (SR10)。
- **PR2-11** PSP は最も明るいカロリメータピクセルで 20 s^{-1} pixel⁻¹ のイベントを処理すること (SR10)。

PR2-12 パルスの検出時間精度を1%以下とすること。

[anti-co パルスに対する要求]

PR3-01 anti-co パルス検出の dead time を 5 ms/event 以下とすること (SR03)。

[時刻付けに対する要求]

PR4-01 全パルスイベントに対して絶対時刻¹の精度 1 ms 以下で時刻付けすること (SR11)。 ¹絶対時刻とは衛星全体にブロードキャストされた TimeCode によって定義された時刻のこと。 **PR4-02** 全パルスイベントに対して relative 時刻²の精度が 1 ms 以下 (目標 80 µs) で時刻付けをすること (SR03, 11)。

PR4-03 XBox-A, XBox-B のサンプリングクロックのズレを 1µs 以下とすること (SR01, 11)。

4.3 各ボードの機能

MIO ボードには FPGA が実装され、SpaceCard ボードには CPU が搭載されている。PSP-A,B の 2 つの FPGA ボードはそれぞれ独立しており、4 つの CPU もハードウェア・ソフトウェア的に独立して 機能している。PSP の主な機能は大きく分けて 4 つある。以下に PSP の主な機能を挙げる。

- 1. Science データ処理
 - XBox からの Sicence データの逆多重化 [FPGA]
 - カロリメータのピクセル波形データからのパルス検出 [FPGA, CPU]
 - anti-co 検出器の波形データからのパルス検出 [FPGA]
 - カロリメータのイベントパルスのグレード付け (パルスの質によってフラグ付け) [CPU]
 - 最適フィルタ処理 (4.10.3 節)を用いたカロリメータイベントパルスの物理情報の抽出 (波高 値、トリガ時刻) [FPGA, CPU]
 - anti-co イベントパルスからの物理情報の抽出(波高値)[FPGA]
 - Science データのパケット化と送信 [CPU]
 - 最適フィルタ処理用の波形テンプレートの計算 [CPU]
- 2. テレコマンド (TC) の処理小節 4.8.1
 - XBox へのテレコマンドの受け渡し [FPGA, CPU]
 - テレコマンドの解釈と実行 [FPGA, CPU]
- 3. HK (House-Keeping: 状態監視) データの処理 (小節 4.8.2)
 - XBox からの HK データの受け渡し [FPGA, CPU]
 - PSP の HK データの収集 [CPU]
 - HK データのパケット化と送信 [CPU]
- 4. 時刻の同期 (小節 4.8.3)
 - XBox への Base clock, sampling clock の供給 [FPGA]
 - 全 MIO ボード間でのクロック同期 [FPGA]
 - カロリメータ、anti-coの波形データへの時刻付け [FPGA]
 - Time HK データの生成、パケット化と送信 [CPU]

PSP で処理された Science データ及び XBox, PSP の HK データは PSP 内でパケット化され、衛星の 上流システムである SMU (Satellite Management Unit) の要求によって、DR (Data Recorder) へと保 存される。PSP と SMU, DR との通信は SpaceWire 規格で、SWR (SpaceWire ルータ) を介して行わ れる。

²relative 時刻とは異なるピクセル間の絶対時刻のズレのこと。

4.4 Mission I/O (MIO) ボード

XBox から出力された Science データは MIO ボードへと入力される。MIO では主にクロックの生成や イベントパルス・ノイズのトリガ、波形データの保存、各種物理情報の抽出の処理などを行う。これら の処理を行うデジタル処理回路は、MIO ボード上の FPGA と呼ばれるユーザーが自由に内部論理回路 を書き換えることのできるデバイス内に実装され、各種モジュール (節 4.5) として機能する。

4.4.1 FPGA

通常デジタル信号を処理するための回路は、標準ロジック IC などと呼ばれる IC を多数並べて作る。 しかし、実装面積が大きくなり、高速な動作が難しいといった問題がある。そこで、特定用途向けの集 積回路である ASIC(Application Specific Integrated Circuit) が使われる。ASIC を作ることで、回路を 一つの集積回路に収めてしまい、省面積化、高速化を実現することができる。しかし、このような ASIC は半導体工場で製造されるため、開発に時間がかかり、工場の設備を使用するために莫大なコストもか かる。そのため、ASIC 開発段階での試作や、小量の ASIC しか使用しない場合、その度に半導体工場 で LSI を生産するのは非現実的である。また、製造された LSI の中身は書き換えることが出来ないた め、後から回路を修正したりすることは不可能である。そこで、ユーザが自由に回路を書き換えること のできるデバイスとして開発されたのが FPGA (Field Programmable Gate Array) である。同じよう に回路を書き換えることのできるデバイスとして CPLD(Complex Programmable Logic Device) と呼 ばれるものもあるが、両者では内部構造や書き込める回路規模に違いがあり、一般に FPGA の方がよ り大規模な回路を書き込むことができる。初期の頃の FPGA は性能が悪く、非常に高価であったため ASIC 開発時の試作や研究にのみ用いられていたが、現在では性能も上がり安価になったため、実際の 製品に組み込まれて使用されることも多く、人工衛星でも既に使われている。ここでは PSP の試験用モ デル (BBM) で用いられている Xilinx 社の FPGA (Spartan-3 シリーズ) を例に説明するが、基本的な 構造はどこの製品でも同じである。FPGA には

- 1. ロジックセル及びコンフィギャブルロジックブロック (CLB)
- 2. 入出力ブロック (IOB)
- 3. ブロック RAM
- 4. 乗算ブロック 18 × 18 bit
- 5. デジタルクロックマネージャ(DCM)

が基板上に設けられており、それらの間を内部配線が結合することにより目的とするロジックを実現し ている。ロジックセルは4入力のルックアップテーブル(LUT)とDフリップフロップから成り、ルッ クアップテーブルを変更することにより様々なロジックを実現している。このロジックセルを9個集め たものがロジックブロックである。Spartan-3 XC3S2000 には全部で5120個のロジックブロックが存 在している。入出力ブロックはボードの外縁部に位置し、外部との信号のやりとりをを行う。Spartan-3 ではLVCMOS、LVDS、LVTTLなど多数の信号入力に対応している。プロックRAMはデータ保持 機能を有し(720 KB)、乗算器は18 bitの乗算が可能である。回路全体及び局所的なクロックはクロッ クマネージャにより供給される。一般的なFPGAの場合、書き込まれた回路の情報はSRAM (Static Randam Access Memory)のアーキテクチャにより保持される。SRAMは電源を切ると情報を失って しまうという特徴があるため、FPGAは電源を入れ直すたびに回路情報を書き込む必要がある。そこ で、外部に回路情報を記憶した ROM(コンフィグレーション ROM)を実装し、電源投入時に自動的に回 路情報を読み込むようにして使われることが多い。MIOボード上の FPGA も同様のである。そのため FPGA を書き換えるためには、コンフィグレーション ROM を書き換える必要がある。Spartan シリーズに用いられるコンフィグレーション ROM は EEPROM(Electronically Erasable and Programmable Read Only Memory) の一種である Flash Rom である。



図 4.2: FPGA の内部構造



図 4.3: ロジックセルの内部構造

4.4.2 MIO ボードの構成

FPGA: MIOボードは2個の FPGA が搭載されており、一方 (通称 SpaceWireFPGA) には SpaceWire の機能が実装されている。もう一方の FPGA (通称 UserFPGA) は、ユーザが独自の回路を書き込む ことができ、処理内容に合わせてロジックを構築することにより汎用的な運用を行なうことができる。 PSP ではこの UserFPGA に MIO の各種機能を実装することになる。試験用モデル (BBM) では Xilinx 社製 Spartan-3 シリーズが使用されるが、性能実証モデル (EM) では Actel 社製の RTAX 2000 シリー ズを実装し、宇宙環境に適した書き換え不可の焼き切り型の FPGA に変更される。

SDRAM: MIO ボードはデータ保存領域として SDRAM を持つ。BBM では UserFPGA, SpWFPGA 共に SDRAM が直結しているが、EM では UserFPGA 側の FPGA は外される。SDRAM の容量は 64MB で EDAC 機能 (Error Detection and Correction) を持つ。アクセスのバス幅は 32 bit である。

Clock: MIO ボードは水晶振動子を持ち、内部で時刻生成を行う。20 MHz のクロック (Base clock) を 生成し、これを基準にして各コンポーネントの処理タイミングが決まる。2 つの FPGA 間には 30 MHz のバスが組み込まれており、SpWFPGA と SpCard 間の SpaceWire 通信はリンクレート 20 MHz で行 われる。



図 4.4: MIO ボード外観

4.4.3 XBox - MIO 間のインターフェース

PSP-A, PSP-B にはそれぞれ MIO-A, MIO-B が搭載されている。各々の MIO は XBox-A, XBox-B と 9 対の LVDS 線で接続されている (図 4.5)。それぞれツイストペアケーブルになっており、2 つのケーブル間の電圧が異なる差動出力として変換されている。また、MIO 同士で時刻の同期を取る必要がある

FPGA	Xilinx Spartan-3 XC3S2000 4FGG456C (BBM の場合)
	Actel RTAX2000S-CQFP352 (EM の場合)
Interface [接続先]	LVDS (non SpW) [XBox] ×2 (51 ピン)
	LVDS (non SpW)[MIO] ×2 (25 ピン)
	LVDS (SpW) [SpaceCard] ×3 (9 ピン)
	\cdots [PSU] ×1 (15 ピン)
サイズ	$240 \times 220 \text{ mm}^2$



ため MIO-A, MIO-B は 2 対の LVDS 線 (SYNC_BASE, SYNC_SMP) で接続され、同期が取られ ている。各 LVDS 線の名称と用途を表 4.2 にまとめた。



図 4.5: XBox と MIO 間の LVDS 接続

4.4.4 タイミング

MIO ボード上の水晶振動子で生成された BASE_CLK (20 MHz) は MIO から XBox への全ての 通信 (CMD_ENA, CMD_DAT, SMP_CLK) に用いられる。XBox に送信された BASE_CLK は TLM_CLK として反射され、XBox から MIO への通信 (HK_ENA, HK_DAT, SCI_ENA, SCI_DAT) の際の base clock として使われる。各信号のタイミングのフローチャートを図 4.6 に示す。MIO-A, MIO-B はマスター/スレーブの関係に置かれ、マスター MIO で生成した BASE_CLK, SMP_CLK (sampling clock)をスレーブ MIO に配信することで同期を取っている。SMP_CLK は、UserFPGA 内部パラメー タ baseCntRoundup で指定された baseCnt (BASE_CLK で生成したカウンタ)のカウント数毎に立 ち上がる。これにより SMP_CLK (sample clock) の周期が決められ、12.5 kHz もしくは 15.624 kHz (400 or 320 clocks) の 2 通りを選択できるようになっている。

XBox は SMP_CLK の立ち上がりで ADC を開始し、およそ 32.4 µs 後に SCI_ENA (science enable) を有効にすることで MIO へ Science データの出力を始める。SCI_ENA は SCI_DAT (science data)

LVDS 線名	Pin	周期	電流値	説明	
	番号	(Hz)	(mA)		
(from PSP to XBox)					
BASE_CLK_A+	4	5 M	3.4	MIO からの Base clock 送信	
$BASE_CLK_A-$	22	5 M	3.4		
SMP_CLK_A+	24	5 M	3.4	XBox 内で使用する Sampling clock の送信	
$SMP_CLK_A -$	7	5 M	3.4		
CMD_ENA_A+	9	5 M	3.4	$\mathbf{CMD_DAT} \ \boldsymbol{\textit{O}} \ \mathrm{enable}/\mathrm{disable}$	
CMD_ENA_A-	27	5 M	3.4		
$CMD_DAT_A +$	29	5 M	3.4	XBox へのコマンド	
CMD_DAT_A-	12	5 M	3.4		
(from XBox to PSP)					
$\mathbf{TLM}_{\mathbf{-}}\mathbf{CLK}_{\mathbf{-}}\mathbf{A}+$	39	5 M	3.4	XBox からの Base clock を telemetry clock	
TLM_CLK_A-	40	5 M	3.4	として反射	
SCI_ENA_A+	42	5 M	3.4	$\mathbf{SCI}_\mathbf{DAT} \mathbf{O} $ enable/disable	
SCI_ENA_A-	41	5 M	3.4		
SCI_DAT_A+	43	5 M	3.4	XBox からの Science データ	
SCI_DAT_A-	44	5 M	3.4		
$HK_ENA_A +$	46	5 M	3.4	$\mathbf{HK}_{\mathbf{DAT}} \mathcal{O} $ enable/disable	
HK_ENA_A-	45	5 M	3.4		
HK_DAT_A+	47	5 M	3.4	XBox からの HK データ	
HK_DAT_A-	48	5 M	3.4		
(Shield lines)					
BASE_CLK_SHLD_A	5	N/A	N/A	BASE_CLK_A±のシールド線	
SMP_CLK_SHLD_A	25	N/A	N/A	SMP_CLK_A±のシールド線	
CMD_DAT_SHLD_A	30	N/A	N/A	CMD_DAT_A±のシールド線	
CMD_ENA_SHLD_A	10	N/A	N/A	CMD_ENA_A±のシールド線	
(from MIO-A to MIO-B)		1	1		
$SYNC_BASE_A2B+$	9	5 M	3.4	MIO-A からの BASE_CLK	
SYNC_BASE_A2B-	5	5 M	3.4		
SYNC_SMP_A2B+	8	1	3.4	MIO-A からの SMP_CLK	
SYNC_SMP_A2B-	4	1	3.4		
SYNC_A2B_SHLD	3	N/A	N/A	SYNC_(BASE SMP)_A2B± 共通のシールド線	
(from MIO-B to MIO-A)					
SYNC_BASE_B2A+	9	5 M	3.4	MIO-BからのBASE_CLK	
SYNC_BASE_B2A-	5	5 M	3.4		
SYNC_SMP_B2A+	8	1	3.4	MIO-BからのSMP_CLK	
SYNC_SMP_B2A-	4	1	3.4		
SYNC_B2A_SHLD	3	N/A	N/A	SYNC_(BASE SMP)_B2A± 共通のシールド線	

表 4.2: XBox - MIO ボード間の LVDS 線
の有効部分を示す。SCI_ENA の開始は FPGA パラメータである sciEnaDelayLower, sciEnaDelayUpper で指定された time window の間で検出する (式 4.1)。SCI_ENA が有効の間に XBox はピク セルデータ 18ch と anti-co データ 1ch、カウンタ情報が順に並んだ SCI_DAT を MIO へと送信する。 その際、SCI_DAT は TLM_CLK に同期する。

同様にして、CMD_ENA (command enable) の有効部分で CMD_DAT (command data)、HK_ENA (HK enable) で HK_DAT (HK data) をそれぞれ出力する。クロックの動作については Clock Module (小節 4.8.3) で説明する。



図 4.6: XBox とのタイミングチャート

4.5 モジュール

PSP にて行われるデジタル処理の行程は、(1) Science データ処理、(2) テレコマンド処理、(3) HK データ処理、(4) 時刻同期 の4つ存在し、それぞれの作業は CPU と FPGA 間で共有して行われている。 この節では FPGA、つまり MIO ボード内の UserFPGA で行われる処理について述べる。UserFPGA で は上記の処理を次の 4 つのモジュールで行う。

- 1. Science module: XBox から送信されてきた Science データを処理し、バッファへと格納する。 保存されたデータは SpaceCard に回収され、更なる波形解析が行われる (節 4.9)。さらに Science モジュールは以下のもので構成される。
 - (a) Main Module(節 4.6): XBox から出力された Science データの展開、エラーやクリップの チェック、微分値計算を行う
 - (b) Sub-modules(節 4.7):3 種類の独立したアルゴリズムの処理が存在
 - PXP: ピクセルパルスイベントのトリガ・保存
 - PXN: ピクセルノイズのトリガ・保存
 - ACP: anti-co パルスのトリガ・保存
 - **XBox TC module**: SMU から SpaceCard を経由して送られてきたテレコマンドの受け取りと、 それの XBox への受け渡しを行う。

XBox HK module: XBox の HK データを受け取り、SpaceCard へ受け渡す処理を行う。

Clock module: クロックの生成、XBox への輸送を行う。また、衛星本体で生成している GPS time と FPGA クロックのマッチングの記録を行う。

UserFPGA の行う処理のブロック図を図 4.7 に表記する。



図 4.7: UserFPGA プロセスのブロック図

4.6 Science Main Module

このモジュールで行う処理は以下の5つに挙げられる。

- 1. サンプルエラーのチェック、サイエンスデータの展開
- 2. チャンネルエラーとクリップのチェック
- 3. Delay バッファへのデータ保存
- 4. ピクセルデータの時間微分計算
- 5. 生波形データと処理後のデータを WFRB バッファに保存

4.6.1 Science データの受信

XBox-A, XBox-Bからそれぞれ MIO-A, MIO-BへLVDS 線を介して SCI_DAT, SCI_ENA, TLM_CLK が送信されてくる (図 4.5)。Science データの 1 サンプルは sampling clock の立ち下がりに同期して周期

12.5 kHz (or 15.625 kHz) で到着する。SCI_ENA の立ち下がりは SMP_CLK の立ち上がりから以下 の式を満たす BASE_CLK カウント n の間で検索される。

$$sciEnaDelayLower \le n < sciEnaDelayUpper$$
 (4.1)

4.6.2 Science データの展開

XBox から出力されてくる Science データの 1 サンプルのサイズは 312 bit であり、多重化した 18 ピ クセル + 1 anti-co チャンネル (1 チャンネル当たり 16 bit) と sciRecvCnt カウンタ (8 bit) で構成さ れている (図 4.8)。



図 4.8: XBox からの 1 サンプル分の Science データ内訳

各チャンネルは図 4.9 のようなフォーマットであり、14 bit で ADC サンプルの値 (-8192 ~ 8192) を 示し、スペアビットが 1 bit (常時 0)、パリティビット (odd parity) が 1 bit からなる。

パリティビット (parity bit):パリティビットとは、与えられた2進数に対して全体の奇偶性を保つ ために与えられる一桁の2進数(0 or 1)のことで、最も単純なエラー検出符号である。この場合の odd parity は、データの1の数が奇数になるよう設定される。パリティビットはパリティチェックに用いら れ、パリティビットが正しくない場合には転送中に誤りが発生したことを示し、parityErrCnt がカウ ントされる。このようにパリティビットはエラー検出符号であるが、どのビットが変化したかを指摘で きない。しかし、1ビットだけ追加するだけで済む点やいくつかの XOR ゲートだけで生成できる点など の利点がある。

	MSB LSB
Pixel	
chan.	pxAdcSample (14b) Spare (1b, value=0) Parity (1b, odd parity)
Anti-co	
chan.	acAdcSample (14b) Spare (1b, value=0) Parity (1b, odd parity)

図 4.9: Science データのビット割り当て

4.6.3 サンプルエラーのチェック

出力されてきた Science データはモジュールによって以下のサンプルエラーのチェックが行われる。エ ラーが検出された場合はそれぞれのカウンタに付加され、そのサンプルに関して sampleErr に 1 が設 定される。

- サンプルデータの長さが 312 bit とは異なる場合 \rightarrow sciLenErrCnt にカウント
- Time window の間にサンプルデータが来ない場合 → sciLenErrCnt にカウント
- sciRecvCnt カウンタが +1 変化しなかった場合 → sciCntErrCnt にカウント

4.6.4 クリップ・チャンネルエラー・Science データエラーのチェック

クリップ

展開した 19 チャンネル (18 ピクセル + 1 anti-co) の adcSample の値のクリップの有無を確認する。 ADC サンプル値は 14bit (つまり 2^{14}) で表記され、 $-8192 \sim 8191$ の範囲を取り得る。-8192 もしくは 8191 の値であった場合、ADC サンプル値は表示しきれずクリップしていると見なされ、全ピクセルチャ ンネル内のクリップデータの数を pxClipCnt にカウントする。anti-co の場合は acClipCnt にカウン トされる。

チャンネルエラー

19 チャンネルの内、いずれかのチャンネルにパリティエラー (parity error) が生じている場合、その サンプルに関して channelErr に 1 を設定する。

Science データエラー

上記で挙げた sciDataErr と channelErr が合成され、sciDataErr として残される。

$$sciDataErr = (sciDataErr||channelErr)$$
 (4.2)

4.6.5 Delay バッファへの保存

受信した 18 チャンネルのピクセルデータは、それぞれチャンネル毎に設けられた Delay バッファ (pxDelayBuf[ch]) に保存されていく。Delay バッファに格納されたデータは次の微分値計算に用いられる (4.6.6 節)。各 Delay バッファのサイズは、1536 bit = 96 サンプル (pxDelayBufSize×16 bit (1 サンプル当 たりのデータサイズ)) である。DelayBufSize は以下のような内訳で設定される。

DelayBufSize = maxPxOffsetAvgLen + maxPxOffsetAvgGap + derivHalfLenMax

Delay バッファ内は図 4.10 のようにビットが割り当てられており、sciDataErr, pxAdcSample[ch] 値が記録される。anti-coの Delay バッファに関しても同様である。

	MSB LSB
	EEDCBA9876543210FEDCBA9876543210FEDCBA9876543210FEDCBA9876543210FEDCBA9876543210FEDCBA9876543210
pxDelay	
Buf	pxAdcSample (14b) Spare (1b, value=0) sciDataErr (1b)
acDelay	
Buf	🗖 acAdcSample (14b) 🔲 Spare (1b, value=0) 🔳 sciDataErr (1b)
sciRecvCnt	
DelayBuf	sciRecvCnt (8b)

図 4.10: Delay バッファ内のビット割り当て



図 4.11: Boxcar 微分 (derivHalfLen = 8 の場合)

4.6.6 時間微分の計算

PSP では、イベントパルスの検出は送られてきた波形の時間微分した波形を使って行う (4.7.1 参照)。 全チャンネル (anti-co は除く)の全てのサンプルに対して時間の一階微分の値を計算する。微分値の計 算は、個々のサンプルデータに boxcar derivative 関数を畳み込む事によって求める (図 4.11)。あるサン プルデータから自身も含め、向こう derivHalfLen で指定されたサンプル数分と、過去 derivHalfLen 分のサンプルを用いて、以下の式のように微分値を導出する。derivHalfLen は最大 32 まで設定可能 であるため、前後 64 サンプル使用することができる。

$$derivLong[ch](i) = \begin{pmatrix} derivHalfLen[ch]^{-1} \\ \sum_{i'=0} pxAdcSample[ch](i+i') \end{pmatrix} \\ - \begin{pmatrix} \sum_{i'=-derivHalfLen[ch]}^{-1} pxAdcSample[ch](i+i') \end{pmatrix} \\ = derivLong[ch](i-1) + pxAdcSample[ch](i+derivHalfLen[ch]-1) \\ - 2pxAdcSample[ch](i-1) + pxAdcSample[ch](i-derivHalfLen[ch]-1). \end{cases}$$

$$derivative[ch](i) = (derivLong[ch](i) + 2^{derivShift[ch]^{-1}})/2^{derivShift[ch]} \qquad (4.4)$$

derivHalfLen[ch]: $0 \sim 31$ で変更可能 (0 は 32 として扱う: デフォルト 8) derivLong[ch]: 微分値計算後の 20 bit サイズの値 derivShift[ch]: 16 bit にリサイズする際の bit shift に使用するパラメータ (表 4.3) derivative[ch]: 16 bit にリサイズされた微分値

モジュールによって初めに計算された微分値 derivLong[ch] は 20 bit 表記であるが、WFRB (次節) に保存される際は、符号付き 16 bit 表記で記録される。したがって、16 bit にリサイズ ($-32766 \leq$ derivative ≤ 32766) する必要がある。式 4.4 において、2^{derivShift[ch]} での割り算によって小数点以下 を $-\infty$ 方向へ切り捨てている (bit shift と同義)。この時、derivShift[ch] は derivative の値が符号付 き 16 bit に収まるように自動的に選択される (表 4.3 参照)。また、割り算の前に 2^{derivShift[ch]-1} を足し ているのは四捨五入のためである。

$\operatorname{derivShift}$	derivHalfLen
0	$1 \leq \operatorname{derivHalfLen} \leq 2$
1	$3 \leq \mathbf{derivHalfLen} \leq 4$
2	$5 \leq \mathbf{derivHalfLen} \leq 8$
3	$9 \leq \mathbf{derivHalfLen} \leq 16$
4	$17 \leq \mathbf{derivHalfLen} \leq 32$

表 4.3: derivShift と derivHalfLenの対応

4.6.7 生波形 処理データの保存

上記の処理工程を終えたサンプルデータは、SDRAM内に設けられた wave-form ring buffer (WFRB) と 呼ばれるバッファ内に保存される。WFRB は各チャンネル毎に用意されている (\mathbf{pxWFRB} [ch], \mathbf{acWFRB})。 全 19 個の WFRB の容量はそれぞれ 1 MB 用意されており、1 サンプルは 32 bit で記録されていく。つ まり、1 周当たり 262,144 サンプル (= 1 MB/32 bit =2¹⁸)保存することが可能である。1 MB 分全て書 き込まれた時には、書き込み位置 ($\mathbf{writePtr}$)を0 に戻し、初めのサンプルを上書きしていくようになっ ている。

WFRB 内のビット割り当てを図 4.12 に示す。1 サンプルは、ADC サンプル値 (pxAdcSample[ch]) は 14 bit, 微分値 (derivative[ch]) は 16 bit, Science データエラーフラグ (sciDataErr) と writeLap の LSB 値はそれぞれ 1 bit の全 32 bit から成る。



図 4.12: pxWFRB, acWFRB 内のビット割り当て

全 19 個の WFRB には共通のバッファinfo が存在する。バッファinfo では次のサンプルの WFRB 内で の書き込み位置を示す writePtr (18 bit) と、WFRB の周回数を示す writeLap (6 bit) が記録されてい る (図 4.13, 4.14)。さらにバッファinfo には WFRB の初めに書かれたサンプルデータの smpelCnt を sampleCntWFRB として記録する。sampleCnt は Clock Module (節 4.8.3) が run になり、sample clock (小節 4.4.4)の通信が始まってからのサンプル数のことを指す。WFRB に詰められる Science デー タは微分値計算がされたサンプルから保存されていくので、初めの sampleCntWFRB は微分値計算 に使われた過去 32 サンプル分差し引いた値から記録されていく。



図 4.13: writePtr と writeLap の概念



図 4.14: WFRBinfo のビット割り当て

4.7 Science Sub-module

Science sub-module では、Science Main Module での処理を終えたデータを元にイベントのトリガや 物理値の抽出を行い、その結果をイベントバッファに保存する作業を行う。イベントバッファ内のデー タは後に SpaceCard 上の CPU 処理の際に利用される (節 4.10)。

Science Sub-module には、 ピクセルパルスのトリガ (PXP) ピクセルノイズのトリガ (PXN) anti-co パルスのトリガ (ACP) の 3 タイプのタスクが存在する。

4.7.1 ピクセルパルスのトリガ (PXP)

イベントパルスの検出は、ADC サンプル値から計算した時間微分波形を用いて行う。微分波形を用い るのは、接近したダブルパルスの検出が可能となるだけでなく、長い周期をもったノイズがベースライ ンに与える影響を少なくするスムージング効果も期待できるためである (図 4.15)。

ピクセルパルスの取得データの長さは FPGA パラメータで定義されており (1024 or 2048 サンプル)、 次式の ADC サンプルの微分値が設定されたスレッショルドレベルをを越えた瞬間からレコードされ始 める。



図 4.15: ピクセルパルスの例

それぞれレコードされたデータからは以下の4つの State から各種物理量を抽出する。その際のパラメー タの様子を図 4.16 にまとめる。

<u>PXP_READY</u>: ピクセルパルストリガ sub-module の初期化が終わると、ピクセルパルストリガ sub-module は PXP_READY state に移行する。この state でトリガされたパルスから、loResPH (low-resolution pulse height: CPU による最適フィルタ処理を施す前の波高値)を求める際のベースラ イン loResBase[ch] を求める。導出は以下の様な式で行われる。

$$\begin{aligned} \mathbf{loResBase[ch]}(i) &= \frac{1}{2^{\mathbf{pxOffsetAvgLenPow}}} \\ &\times \left(\sum_{i'=\mathbf{pxOffsetAvgGap+1}}^{2^{\mathbf{pxOffsetAvgLenPow}} + \mathbf{pxOffsetAvgGap}} \mathbf{adcSample[ch]}(i-i') + 2^{\mathbf{pxOffsetAvgLenPow-1}} \right) \end{aligned}$$

(4.7)

pxOffsetAvgLenPow と pxOffsetAvgGap は、図 4.16 に示されているとおり、パルスがトリガ されたサンプルより以前の部分のサンプルを指し、小節 4.6.5 で保存された maxPxOffsetAvgLen, maxPxOffsetAvgGap からそれぞれ指定されたサンプル分だけ使用する。式 4.7 では、四捨五入のた め 2^{pxOffsetAvgLenPow-1} を加算し、2^{pxOffsetAvgLenPow} で割り算することで小数点以下を -∞ 方向へ 切り捨てている。

さらに、この state ではトリガされたサンプルの WFRB 内での書き終わり位置 (trigPtr) と周回数 (trigLap) を記録する。

<u>PXP_ARMED</u>: この state では微分波形をモニターして、微分値が減少へと転じる箇所を探し出す。 derivative[ch] < derivPre[ch] そして、ADC サンプル値と微分値のピーク値をそれぞれ pxAdcSam-



図 4.16: ピクセルパルスのトリガ時のパラメータ [最小:最大:デフォルト]

pleMax[ch], derivMax[ch] として保存する。

<u>PXP_FALL</u>: PXP_ARMED がピークを検出後、PXP_FALL state へと移行し、微分値が設定したス レッショルドを下回るまで続く (derivative[ch] < pxFallEndThres[ch])。この state では、引き続き pxAdcSampleMax[ch], derivMax[ch] の検出を続ける。通常、微分値は単調減少していくが、ごく稀 にこの state 中に 2 つ目の小さなイベントパルスが入り込むことがある (これを quick double イベントと 呼ぶ)。ここでは、微分値が一つ前のサンプルの微分値を上回った際 (derivative[ch] – derivPre[ch] ≥ pxQuickDoubleThres[ch])、quick double フラグ (quickDouble[ch]) を立てる。

PXP_PEAKFIND: 微分値が (pxFallEndThres[ch]) を下回ると次に PXP_PEAKFIND state へと 移行する。ここでは微分値は見ずに、pxAdcSampleMax[ch] のみが引き続き検索され、ファースト パルスのピークの ADC サンプル値を上回る場合、pxAdcSampleMax[ch] の値が更新される。ピー ク値を検索する PXP_PEAKFIND の範囲は FPGA パラメータ pxPFStateCntMax で指定されてお り、もしもこの間に pxAdcSampleMax[ch] が更新されれば、さらにそこから pxPFQuitCntMax 分ピーク値の検索を開始する。(pxPFStateCnt ≥ pxPFStateCntMax) もしくは (pxPFQuitCnt ≥ pxPFQuitCntMax) となった場合、PXP_PEAKFIND は終了し、先ほど式 4.7 で導出した loRes-Base[ch] を用いて loResPH[ch] の計算を行う。ADC サンプル値のピークが loResBase を下回る場 合、loResPH は 0 とする (式 4.8)。

$$\mathbf{loResPH[ch]} = \begin{cases} \mathbf{pxAdcSampleMax[ch]} - \mathbf{loResBase[ch]} & (\mathbf{pxAdcSampleMax[ch]} \ge \mathbf{loResBase[ch]}) \\ 0 & (\texttt{L}記以外) \end{cases}$$

$$(4.8)$$

4.7.2 ピクセルパルス EDB (Event Dual Buffer)

トリガされたピクセルパルスのイベント情報は、ピクセルパルス Event Dual Buffer (pxPulseEDB) へと記録される。図 4.17 に示すように、pxPulseEDB は 2 つの bank を持っている。2 つの bank 0, bank 1 はそれぞれ9 チャンネルずつデータが書き込まれる。さらに 1 つの bank の中には 2 つの side が 設けられ、デュアルバッファを構築している。FPGA パラメータによって指定された side に従い、片方 の side に FPGA がデータの書き込み (RMAPwrite) を行い、もう片方は SpaceCard 上の CPU がデー タの読み出し (RMAPread) を行う。各 pxPulseEDB の side の終わりにはそれぞれフッター情報が付 与されている。このフッター情報は CPU の RMAPread アクセスの際にデータと同時に読み込まれる。 RMAPread が完了するとフッター情報内の cpuAck フラグが立てられ、CPU による読み込みが完了し て FPGA による書き込みが可能であることを宣言する。フッター情報にはこの他に、と同様に次の書き 込み位置 (wp : write pointer) と周回数 (lap) が含まれる。また、RMAPread アクセスや cpuAck の 度に付加されるシーケンスカウンタ (seqCnt) も記録する (図 4.19)。

記録されるデータ本体の内容は図 4.18 のようなフォーマットになっている。一つのレコードのデータ長は 64 bit で、パルスレコードの最初のサンプルの trigPtr (18 bit) と trigLap (6 bit), さらに derivMax[ch] (16 bit), loResPH[ch] (14 bit), quickDouble[ch] (1 bit), チャンネル番号 (5 bit) から成る。

pxPulseEDBの各 sideのサイズは1 KB であり、125 レコード (= 64 bit record⁻¹×125 records = 0.98 KB) 記録することができる。



図 4.17: Bank の切り換え

	MSB LSB
	Image: Constraint of the
pxPulse	
EDB	chan(5b) 🗆 spare(4b) 🗖 qckDbl(1b) 🗖 trigLap(6b) 🗖 trigPtr(18b) 🗖 loResPH(14b) 🗖 derivMax(16b)

図 4.18: pxPulseEDB のビット割り当て



図 4.19: EDBinfo のビット割り当て

4.7.3 ピクセルノイズのトリガ (PXN)

後述の SpaceCard 上 CPU で行う最適フィルタ処理(節 4.10.3)で使用するため、Science sub-module ではピクセルノイズの収集も行う。ピクセルノイズのレコード長は FPGA パラメータ pxNoiseRecord-CleanLen[ch] で決められており、ADC サンプルの時間微分の絶対値が設定したピクセルノイズのスレッ ショルドレベルを下回ったところ(|derivative[ch]| ≤ pxNoiseThres[ch])でトリガを行い、そこから レコード長分ノイズデータを取得する(図 6.17)。もしトリガのかかったサンプルからレコード長内にパル スなどが入射することなく、|derivative[ch]| ≤ pxNoiseThres[ch] が続いた場合(pxNoiseCnt[ch] ≥ pxNoiseRecordCleanLen[ch])、ノイズのレコードは完了し、ノイズデータはピクセルノイズ EDB (pxNoiseEDB) へと保存される。トリガがかかってからレコード中に微分の絶対値がスレッショルドを上回ってしまった場合 (|derivative[ch]| $\geq pxNoiseThres[ch]$)、ノイズの収集は破棄され、再度トリガがかかったサンプルから収集をやり直す。

保存先である pxNoiseEDB は、ピクセルパルス EDB と同様、2 つの bank それぞれに 2 つの side が 存在する。振る舞いもピクセルパルス EDB と同じで、片方が FPGA によって RMAPwrite されている 間、もう片方が CPU によって RMAPread が実行される (図 4.17)。また、取得したサンプル内に Science データエラー (sciDataErr) が一つでも含まれていた場合、event エラーフラグを立てる。

ノイズデータの1レコードのデータ長は32 bit で、event エラーフラグ (1 bit)、ノイズレコードの最初 のサンプルの trigPtr (18 bit) と trigLap (6 bit)、チャンネル番号 (5 bit) から構成される (図 4.21)。 pxNoiseEDB の各 side のサイズは1 KB であり、250 レコード (= 32 bit record⁻¹×250 records = 0.98 KB) 記録することができる。



図 4.20: ピクセルノイズの例:例ではpxNoiseThres[ch]=100, pxNoiseRecordCleanLen[ch]=1280, TMPL_LEN_SH=1024 (節 4.10.2)



図 4.21: pxNoiseEDB のビット割り当て

4.7.4 Anti-co パルスのトリガ (ACP)

ACP sub-module では anti-co パルスのトリガを行う。ピクセルパルスとは異なり、anti-co パルスは微 分波形は用いず、ADC サンプル値 (acAdcSample – acAdcSamplePedestal) に基づいてパルスを検 出する (図 4.23)。設定したスレッショルドを越えたサンプルからレコードを開始する (acAdcSample – acAdcSamplePedestal \geq acPulseThres)。レコード長は指定されておらず、(1) カウント数が設定値を上回る (acPulseCnt \geq acPeakfindLen)、もしくは (2) ADC サンプル値がスレッショルドを下回る (acAdcSample – acAdcSamplePedestal \leq acPulseThres) までレコード続けられる。それぞれの レコードからはパルスの計測時間 (duration) と ADC サンプル値の最大値が抽出され、図 4.22 のよう なフォーマットで acPulseEDB に記録される。pxPulseEDB, pxNoiseEDB と同様、acPulseEDB も 2 つの bank それぞれに 2 つの side が存在し、同様の機能を果たす (図 4.17)。各 side の容量は 1 KB で、125 レコード (= 64 bit record⁻¹×125 records = 0.98 KB) 記録することが可能となる。



Time (ms) 0.5 1.5 2 0 1 2500 Duration 2000 ADC sample - pedestal 1500 heidh 1000 Pulse 500 Pulse threshold 0 Pedestal Arrival time -500 5 0 10 15 20 25 30 Time (sample)

図 4.22: acPulseEDBのビット割り当て

図 4.23: Anti-co パルスの例

4.8 その他のモジュール

4.8.1 XBox TC Module

XBox TC (tele-command) Module は、SMU から SpaceCard を介して送られてきたコマンドを XBox へと受け渡す機能をもつ。TC Module の state は、INIT と STDBY で初期化を実行し、RUN でタスク を実行する。XBox へのコマンドは、XBox に HK 情報を要求する HK リクエストと XBox の状態やセッ

ティングを変更する実行コマンドの2種類があり、どちらも同じフォーマットで送信される。SpaceCard からの全てのテレコマンドはxBoxTCBufと呼ばれるバッファにRMAP Write で書き込まれる。XBox TC データのフォーマットは、全8 Byte で決められており、初めの6 Byte にテレコマンドデータ、後ろ の2 Byte に TC Module で計算した CRC (MIO TC-CRC)³が割り当てられる (図 4.24)。TC Module の state の変更やパラメータの変更は SpaceCard の CPU から行う。もし、TC Module が XBox にコマ ンドを送っている最中に CPU から新しいコマンドが送信されてきた場合、それらは無視され、invalid RMAP access エラーとして認識される。



図 4.24: XBox TC データのビット割り当て

4.8.2 XBox HK Module

XBox HK (house-keeping) Module は、XBox からの HK データを SpaceCard を介して SMU へと 受け渡す機能をもつ。HK Module の state は、INIT と STDBY で初期化を実行し、RUN でタスクを 実行する。XBox から送信されてくる HK reply は 3 種類あり、MIO からの HK リクエストに対する reply、MIO からの実行コマンドに対する reply、HK リクエストや実行コマンドに対するレスポンス内 の際のエラー reply である。XBox からの HK データは xBoxHKBuf と呼ばれるバッファに格納され る。xBoxHKBuf は 64 個のブロック (xBoxHKBuf[0]-xBoxHKBuf[63]) からなり、それぞれのブ ロックのサイズは 32 bit である。xBoxHKBuf[0] には、受信した HK やエラーのカウント数などの追 加情報が格納される。SpaceCard からこれらのバッファを RMAP Read することで HK を取得する。



図 4.25: xBoxHKBuf[0] のビット割り当て

4.8.3 Clock Module

Clock Module は、MIO ボード上の水晶振動子にてクロックを生成し、クロックを XBox へと届ける 機能をもつ。また、GPS から受信した GPS time と FPGA カウンタとを保存し、照合を行う。Clock Module の state は、INIT と STDBY は初期化のための状態、RUN では BASE_CLK と SMP_CLK の 生成と XBox への送信、そして FPGA カウンタと GPS time の保存とラッチを行う。BASE では基本的

³巡回冗長検査。エラー検出符号の一種。連続するエラーを検出でき、パリティビットを用いたチェックサム方式よりも検 出精度が高い。

に RUN と同様の挙動を示すが、SMP_CLK の生成と送信は行わない。FPGA カウンタには BaseCnt と 呼ばれるカウンタがある。これは、BASE_CLK の 5 MHz の立ち上がりに同期したカウンタで、0 - 399 (baseCntRoundup = 320)、もしくは 0 - 399 (baseCntRoundup = 400)⁴ の値をとる。BaseCnt が 一周すると、sampleCntL (16 bit) が 1 つ増加する。これは、4.19 s (5.24 s) の長さを持つカウンタで ある。さらに、sampleCntL が一周すると、sampleCntH (16 bit) の値が 1 つ増加する。これは、3.18 day (3.97 day) の長さを持つカウンタである。これらの値と、GPS time の上位 32 bit (upper32bitTI) を、Time-Code [0] でラッチして timeHKBuf と呼ばれるバッファに格納し、絶対時刻を付ける際に利 用される。timeHKBuf の内訳を図 4.26 に示す。



図 4.26: timeHKBufのビット割り当て

4.9 SpaceCard ボード

MIO ボードで処理された波形データは、次に SpaceCard ボード上の CPU によって回収され、更なる 波形解析処理が施される。SpaceCard では主に、イベントの時刻付けやグレード付け、最適フィルタ処理 を用いた詳細な波高値解析、そして衛星の上流システムである SMU (Satellite Management Unit) の要 求によって回収されるデータパケットの生成を行う。SpaceCard でのタスク管理は TOPPERS/JSP カー ネル (Toyohashi OPen Platform for Embedded Real-time Systems / Just Standard Profile Kernel) と 呼ばれるリアルタイム OS によって実行される。

4.9.1 SpaceCard ボードの構造

CPU: SpaceCard ボードには、三菱重工製 SH4 準拠の CPU「SOI-SOC」が搭載される。SOI-SOC と は、JAXA 宇宙科学研究所が民生用最先端 SOI⁵技術をベースに構築した、耐放射線の論理回路設計・製 造技術を用いて、三菱重工業(株)が製品化した宇宙仕様の SOC⁶である。宇宙空間での宇宙線などにも耐 えられるよう極めて高い放射能耐性を持っている。CPUのクロック数は 60 MHz で、OS には TOPPERS (小節 4.9.2)を採用している。

Memory:様々な用途に応じて複数のメモリが搭載されている。表 4.4 に各メモリのサイズと用途をまとめる。

種類	Size/piece	# of	# of	Total size	用途
	(b)	pieces	sets	(MB)	
EEPROM	$512K \times 8$	4+1	1	2	メインプログラムの保存領域 (EDAC 機能無し)
SRAM	$512K \times 8$	4 + 1	2	4	メインプログラム用の作業メモリ (EDAC 機能無し)
SDRAM	$16M \times 16$	2 + 1	1	64	データ保存領域

表 4.4: SpaceCard ボードの各メモリ

Interface: SpaceCard ボードには、MIO ボード、SMU、他の SpaceCard ボードとの通信用のインターフェースを持っている。表 4.9.1 に各コネクタの説明をまとめた。

	-11	1.0. Spaceco		· · · · · ·			
Other end	Protocol	# of	Connector	Gender of	# of	# of in	# of out
		connectors	shape	connectors	pins	channels	channels
MIO boards	LVDS (SpW)	2	micro D-sub	female	9	2	2
SpaceCard	LVDS (SpW)	0	micro D-sub	female	9	0	0
SMU	LVDS (SpW)	2	micro D-sub	female	9	2	2
PSU		1	D-sub	female	15		

表 4.5: SpaceCard ボードのインターフェース

⁵Silicon On Insulator の略。絶縁膜上に形成した単結晶シリコンを基板とした半導体、および半導体技術。超高集積回路の高速化・低消費電力化を実現するためのキーテクノロジーとして期待されている。

⁶System-on-a-Chip の略。マイクロプロセッサーやメモリーなど、ある装置やシステムの動作に必要な機能のすべてを一つの半導体チップに実装する方式。配線の省略による高速化や、部品点数の削減による消費電力節減、装置の小型化や製造コスト低減などのメリットが期待できる。



図 4.27: SpaceCard ボード外観

4.9.2 TOPPERS/JSP カーネル

TOPPERS (Toyohashi OPen Platform for Embedded Real-time Systems) とは、組込みシステム用 リアルタイム OS の業界標準である ITRON 仕様をベースとしたオープンソースなソフトウェア群、も しくはその開発プロジェクトの事を指す。その中でも TOPPERS/JSP (Just Standard Profile) カーネ ルはµITRON4.0 仕様に準拠したリアルタイム OS で、JSP の名前が示す通り、µITRON4.0 仕様のスタ ンダードプロファイル規定に従って実装されている⁷。SpaceCard CPU はこの TOPPERS/JSP カーネ ルを採用し、タスク管理を行っている。タスクのスケジューリングはタスクに与えられた優先度に基づ くプリエンプティブな優先度ベーススケジューリング方式⁸で行われる。タスクは表 4.9.2 のような分割 で優先度が割り振られている。CPU 負荷測定のタスクは一番優先度の低い位置付けであり、CPU が作 業をしていないアイドル時間を使って CPU 負荷を測定する。これらのタスクの内、PSP のユーザータ スクは以下のものがあり、ここで各種 HK の処理や波形データの処理を行う。

7. ユーザ周期タスク(周期起動)

- XBox HK (~ 4 s に 1 回)
- PSP HK (~ 4 s に 1 回)
- Time HK (~ 8 s に 1 回)
- pxPulseEDB, acPulseEDB からのトリガ情報収集 (16 Hz)
- 8. ユーザ非周期タスク(常時起動)
 - 波高値計算タスク
 - 平均波形・ノイズ収集、テンプレート計算タスク

	タスク名	優先度 (1 - 16)
0.	周期ハンドラ・割り込みハンドラ(Not task)	0
1.	I/O タスク (DMA, RMAP など)	1 - 4
2.	スケジュールタスク (TimeCode に同期して周期起動を制御)	5
3.	コマンド取得タスク (周期起動)	5
4.	コマンド実行 (or 受付) タスク (コマンドが到着時)	6
5.	メモリパトロールタスク (周期起動)	7
6.	共通テレメトリタスク (周期起動)	8 - 9
7.	ユーザー周期タスク (周期起動)	10 - 12
8.	ユーザー非周期タスク (常時起動)	13 - 15
9.	CPU 負荷測定タスク (アイドル時)	16

表 4.6: タスクの優先度 (2010年現在)

次節では CPU で行う波形データ処理に関して述べる。

4.10 SpaceCard でのデジタル波形処理

SpaceCard 上で行われるイベント関連のタスクに関して説明する。SpaceCard ではピクセルパルスに 対して詳細な波形解析が施される (anti-co パルスに対しては行わない)。イベントパルスの計算タスクは 各ピクセル毎に用意され、それぞれの CPU にてスケジューリングで処理する順序を決めている。各ピ クセルのタスクでは、まず SpaceCard が MIO ボード上の Event Dual Buffer (pxPulseEDB[ch]) 内の トリガ情報を読み込む。このトリガ情報を元に Wave-Form Ring Buffer (pxWFRB[ch]) から波形デー タを転送し、CPU の SRAM に設けられた pxEvent1stFIFO⁹に詰められる。pxEvent1stFIFO に詰 められる前に全てのパルスに対してセカンダリーパルスのサーチ (小節 4.10.1) が行われ、全パルスに対 してグレード付けがなされる (小節 4.10.2)。その後、生成したテンプレート (小節 4.10.4) を用いた最適 フィルタ処理によって詳細な波高値解析を行う (小節 4.10.3)。これらの処理によって得られた各種デー タは pxEventTelemeFIFO へと詰められ、テレメトリーとしてパケットデータ化され、SMU の要求 によって DR に転送される。以下では、上記の波高値解析までの処理内容を述べる。

4.10.1 セカンダリーパルス検出

pxEvent1stFIFO に転送されたピクセルパルスデータは全てに対してセカンダリーパルスのサーチ が行われる。セカンダリーパルスとは1stパルスがトリガされてから874サンプル (or 1748サンプル) 以 内に入射してきたパルスのことを指す。このセカンダリーパルスとの時間間隔に依存して各パルスがグ レード付けされる (小節4.10.2)。セカンダリーパルスのサーチは、CPU によって準備された平均パルス (小節4.10.4) の微分波形を対象パルスの微分波形から差し引くことで行う。これはMIO ボード上での 処理でトリガしきれなかった小さなセカンダリーパルス (EDB に無いトリガ情報)を検出するためでも ある。サーチは pxEvent1stFIFO 内のパルスがトリガされたサンプルから後ろ874 サンプル (or 1748 サンプル) に対して行われる。この際、WFRB から derivative[ch] 値を読み出し、この微分値を用い

⁷詳細は http://www.toppers.jp 参照

⁸タスクが実行中であっても、より高い優先度のタスクが実行可能となった場合、即座にタスク切り替えを行う方式

⁹FIFO (First In, First Out): キューに格納されたデータの処理方法を意味する。処理はキューに格納された順番で行われ、最初に入ってきたものを最初に処理し、次に入ってきたものは最初の処理が終わるまで待たせる、ということを意味している。

て平均パルスと対象パルスの差し引きを行う。得られた差分が設定したセカンダリーパルスに対するス レッショルドレベル secondThres を超えていればセカンダリーパルスとしてトリガを行う (図 4.28)。 トリガされたセカンダリーパルスは pxEvent2ndFIFO に詰められる。pxEvent1stFIFO に詰められ たイベントパルスから 874 サンプル (or 1748 サンプル) はセカンドパルスのサーチのみを行い、それ以 外は行わない。



図 4.28: セカンダリーパルス検出の概要 (TMPL_LEN_SH=1024 の場合)

4.10.2 グレード付け

トリガされた全ピクセルパルスイベントはそれぞれの隣接したイベントとの時間間隔によって、グレード付けが行われる。anti-co パルスイベントに関してはグレード付けは行わない。グレード付けの 定義を図 4.29 に示す。最も質の高い High-Resolution (HR) イベントは、パルスがトリガされた時間か ら (TMPL_LEN_[S|L]H – preTrigPnts[S|L]H)の間に他のパルスイベントが入射していないものに 定義される。TMPL_LEN_[S|L]H, preTrigPnts[S|L]H は、それぞれ HR テンプレートの全体の長さ と pre-trigger の長さで、表 4.7 の通り Short か Long かで長さが異なる。デフォルトは Short で取得し、 Long は予備のテンプレート長さである。Medium-Resolution (MR) イベントは、パルスがトリガされ た時間から (TMPL_LEN_[S|L]M – preTrigPnts[S|L]M)の間に他のパルスイベントが無いものを指 す。それ以外のイベントパルスは Low-Resolution (LR) イベントとしてグレード付けされる。

グレードは全部で 5 段階 (Hp, Mp, Ms, Lp, Ls; 図 4.29) に分けられる。全ての HR イベントはプラ イマリーイベントである。よりグレードの高いイベントパルス、さらにはセカンダリーパルスよりもプ ライマリーパルスの方が、波高値に他のパルスの影響を受けづらいため、優れたエネルギー分解能が得 られる。Hp, Mp (, Ms) のイベントは、次にそれぞれのテンプレート長さで最適フィルタ処理が施さ れ、波高値解析が行われる。(小節 4.10.3)。TMPL_LEN_[S|L][H|M], preTrigPnts[S|L][H|M] は衛 星ミッションを通じて固定値 (Short or Long) である。

表	4.7:	テ	ン	プ	レー	ト	の	長さ
---	------	---	---	---	----	---	---	----

長さ	TMPL_LEN_XH	$\mathbf{TMPL_LEN}_X\mathbf{M}$	$preTrigPnts_XH$	$preTrigPnts_XM$
Short $(X = \mathbf{S})$	1024	256	150	37
Long $(X = \mathbf{L})$	2048	512	300	75



図 4.29: グレード付けの定義 (Short template の場合)

4.10.3 最適フィルタ処理

MIOボードでは、Science Module でパルスの波高値を loResPH として計測している。しかしなが ら、実際にはパルスに混入したノイズがそのまま加算されてしまうので、単純にパルスのピーク値をとっ ただけでは、理想とするエネルギー分解能を得ることはできない。そこで、その性能を最大限に引き出 すために、X線パルスの大きさを S/N 比が最大になるように定める最適フィルタ処理を行う。複数の サンプルを用いて、平均化によってノイズを減らしてパルスハイトを求めるというこの手法によって、 ファクター数倍の S/N の改善が行うことができる。ASTRO-H 衛星の SXS ではこの手法を使って詳細 な信号処理を行う。

はじめに最適フィルタ処理について簡単に説明する。まず、取得した X 線パルス D(t) の平均パルス を作成し、周波数空間にフーリエ変換を行う。次に周波数空間において、パルススペクトルにノイズス ペクトル N(f) で重みをつけテンプレート T(t) を作成する (小節 4.10.4)。このテンプレートを個々の パルスとクロスコリレーションをとり、最大になる時の値をパルスハイトとする。これを入射 X 線のエ ネルギーに相当するように規格化を行うとエネルギースペクトルが作成できる。しかし、この最適フィ ルタ処理はX線パルス波形が常に同じであり、パルスとノイズが完全に独立であるという仮定を含んで いることを念頭においておく必要がある。

測定から得られたパルスを D(f) とする。これは周波数空間では、規格化されたモデルパルス M(f) に振幅 A をかけたものにノイズ成分 N(f) が含まれたものであり、

$$D(f) = A \times M(f) + N(f) \tag{4.9}$$

のようにかける。これをモデルパルスと呼ぶ。

振幅 A はノイズを含む生データとパルス波形のモデルとの差を最小にするものである。つまり、実際 に得られたパルスとモデルの残差 χ を最小にするような A を最小二乗法で求めてやれば良い。

$$\chi^{2} \equiv \int_{-\infty}^{\infty} \frac{|D(f) - A \times M(f)|^{2}}{|N(f)|^{2}} \mathrm{d}f$$
(4.10)

と書けるので、 χ^2 の微分が0になるようなAは、

$$A = \frac{\int \frac{DM^* + D^*M}{2|N|^2} \mathrm{d}f}{\int \frac{|M|^2}{|N|^2} \mathrm{d}f}$$
(4.11)

で与えられる。D(f)、M(f)は実関数のフーリエ成分なので $D(-f) = D(f)^*$ 、 $M(-f) = M(f)^*$ となるので、

$$\int_{-\infty}^{\infty} \frac{D(f)M(f)^*}{2|N(f)|^2} df = -\int_{\infty}^{-\infty} \frac{D(-f)M(-f)^*}{2|N(f)|^2} df = \int_{-\infty}^{\infty} \frac{D(f)^*M(f)}{2|N(f)|^2} df$$
(4.12)

が成立することから、

$$A = \frac{\int_{-\infty}^{\infty} \frac{DM^*}{|N(f)|^2} df}{\int_{-\infty}^{\infty} \frac{|M|^2}{|N|^2} df}$$
(4.13)

$$= \frac{\int_{-\infty}^{\infty} \frac{D}{M} \left|\frac{M}{N}\right|^2 \mathrm{d}f}{\int_{-\infty}^{\infty} \left|\frac{M}{N}\right|^2 \mathrm{d}f}$$
(4.14)

と記述できる。この式から A は $|M(f)/N(f)|^2$ を重みとした場合の周波数空間での S/N 比 D(f)/M(f)の平均値を表していることがわかる。また、式 (4.14) は \mathcal{F}^{-1} を逆フーリエ変換として、

$$T(t) \equiv \mathcal{F}^{-1}\left(\frac{M(f)}{|N(f)|^2}\right)$$
(4.15)

$$A = \frac{\int_{-\infty}^{\infty} D(t) \mathcal{F}^{-1}\left(\frac{M(f)}{|N(f)|^2}\right) dt}{\int_{-\infty}^{\infty} \left|\frac{M(f)}{N(f)}\right|^2 df} = \frac{\int_{-\infty}^{\infty} D(t) T(t) dt}{\int_{-\infty}^{\infty} \left|\frac{M(f)}{N(f)}\right|^2 df}$$
(4.16)

と変形できる。ここで用いた T(t)を最適フィルタのテンプレートと呼ぶ。

従って、例えば X 線の入射によってあるパルス D(t) が得られたとすると、テンプレートを用いることで、パルスハイト H は、

$$H = S \int D(t)T(t)dt \tag{4.17}$$

あるいは、離散的なデータに対して、

$$H = S \sum_{\mathrm{I}} D_{\mathrm{I}}(t) T_{\mathrm{I}}(t) \tag{4.18}$$

となる。ここで S は最適な規格化定数、 $D_{I}(t)$ 、 $T_{I}(t)$ はデジタル化されたパルスデータとテンプレート である。従って、最適フィルタ処理を行うためには、平均パルス波形 M(f) とノイズパワースペクトル $N^{2}(f)$ を用意すれば良いことになる。ノイズが完全に白色である時、すなわち周波数空間でフラットな 場合は、テンプレートは元の平均パルスと一致する。

4.10.4 テンプレート計算

SpaceCard では最適フィルタ処理で用いるテンプレートの計算を行う。テンプレート計算の例を図 4.30, 4.31, 4.32 に示す。テンプレート計算には平均パルスのパワースペクトル M(f) とノイズスペクト $\mathcal{N}(f)$ が必要となる。

Pulse:テンプレートのモデルパルスには、他のパルスの混入していないクリーンなイベントパルスだけを多数選び、それらを平均することで得られた平均パルスを用いる。CPU ではグレードの高いイベントパルス (Hp) を平均することで平均パルスを生成する¹⁰。得られた平均パルスは高速フーリエ変換され、パルスのパワースペクトル *M*(*f*) が導出される (図 4.30)。

Noise:また、節4.7.3のアルゴリズムで収集されたノイズレコードの内、新しい数百個のノイズレコードを用いてノイズのパワースペクトルを生成する。パルス同様、ノイズレコードは高速フーリエ変換がなされ、フーリエ変換後の全データを平均することで、ノイズのパワースペクトルを生成する(図4.31)。

ー度、パルスとノイズのパワースペクトル M(f), N(f) の生成が完了すれば、式 4.15 に従ってテ ンプレートのパワースペクトル T(t) が導出できる (図 4.32)。テンプレートは表 4.7 の 2 通りの長さ TMPL_LEN_XH, TMPL_LEN_XM で用意される。最適フィルタ処理の際、HR はTMPL_LEN_XH、 MR は TMPL_L-EN_XM の長さのテンプレートを用いて波高値解析が行われる。

決められたテンプレート長さという有限区間であるため、フーリエ変換を行う際に有限区間外は0と みなされてしまう。他のパルスの無い十分に長い HR テンプレートであれば、レコードの端が連続的に 減衰して0に収束すると考えられるので変換の際の影響は少ない。しかし、長さの短い MR テンプレー トでは区間外のイベントからの漏れ込みが考えられ、不連続が生じてしまう。そこで両側にいくにつれ て減衰するよう、ハニング窓関数 (Hanning window)¹¹を掛けてから高速フーリエ変換を行う方法があ る。しかし、一般に窓関数を導入するとエネルギー分解能が劣化することが知られており、現段階で導 入するかは未決定である (2011 年 1 月現在)。

¹⁰平均パルスでセカンダリーパルスのサーチを行わなければグレード付けが行えないため、初観測時には衛星にはあらかじめ保存しておいた平均パルスを使用する。

¹¹中央値が1のRaised Cosineの波形になっている窓関数。サイドローブが比較的小さいため、小さい電力のスペクトルを 検出するのに向いている。



図 4.30: 平均パルスの例: (左) 平均パルスの形状 (右) 平均パルスのパワースペクトル



図 4.31: ノイズの例: (左) ノイズレコード (右) 平均ノイズのパワースペクトル



図 4.32: テンプレートの例: (左) テンプレートのパワースペクトル (右) 最適フィルタテンプレート

第5章 MIO – SpaceCard 性能評価試験

本章では、PSP-BBMの性能を検証するため、MIOボード – SpaceCardボードを接続し、PSPで実現 する各機能が正常に動作しているかを検証する。なお、以下の検証試験内容は、試験によってUserFPGA versionが更新されているため、挙動が異なる場合がある。そのため以下では、試験時に扱う FPGA ver. を記載した上で進めていく。主に行った性能評価試験は以下の通りである。

- 1. PSP 搭載機器の SpaceWire 規格適合試験¹
- 2. Time-Code 配信機能試験¹
- 3. MIO SpaceCard 間通信試験
- 4. MIO SpaceCard 間通信速度測定試験
- 5. XDS PSP 間インターフェース試験
- 6. MIO ScienceModule での Science データ取得試験
- 7. MIO ScienceModule の WFRB, EDB の動作確認試験
- 9. MIO ClockModule の動作確認試験¹
- 10. MIO Master Slave 接続試験¹
- 11. SpaceCard 上 CPU の処理速度測定試験

Link Analyzer

PSP 内の 2 ボード間の通信やその他の PSP 周りの関連機器との接続は SpaceWire 規格 (節 3.4) で行われている。試験において、SpaceWire に規格に沿って PSP が正常に通信を行っていることを確認する ために、SpaceWire パケットデータの中身を「SpaceWire Link Analyzer (LA)」と呼ばれる機器を接続 の間に挟むことで直接確認する。LA は、star dundee 社によって開発された SpaceWire のリンクを監 視する装置で、SpaceWire リンクの間に LA を挟むことで、Time-Code や RMAP パケットなど任意 の条件でトリガをかけて、接続状態を監視することができる。

XBox Digital Simulator (XDS)

PSP の機能試験を行うためには、Science データや HK の受信、クロックの送受信、そしてコマンド 送信といった XBox とのやりとりが必要となる。XBox の実機は NASA/GSFC で開発中であるため、日 本側での PSP 試験用に XBox の入出力を模擬した XDS (XBox Digital Simulator) が GSFC 側から提供 されている (図 5.1)。XDS は、計8 チャンネル分のデータセットを記録しておくことができ、必要に応 じてダイヤルを切り換えて出力データを変更することができる。通常は第7章で使用している波形デー タと同等のもので試験を行う。XDS 内のデータの書き換え手順は付録 A に記載する。

¹埼玉大 下田修論参照





図 5.1: XBox Digital Simulator 外観とブロック図

5.1 MIO – SpaceCard 間通信試験

[FPGAver.=6]

MIO と SpaceCard 間の SpaceWire を用いた通信の試験を行った。まず、MIO の SDRAM のある領 域に SpaceCard からインクリメントデータを RMAP Write で書き込む。次に、同じ領域を SpaceCard から RMAP Read で読み出し、インクリメントデータが保存されていることを確認する。 実験のセッ トアップを図 5.2 に示す。



図 5.2: セットアップ

MIO では SDRAM に 1 KB 以上のアクセスをするにはレジスタの値の変更が必要である。さらに、 SDRAM の動作タイミングの微調整の必要もある。表 5.1 に SDRAM 使用時のレジスタの変更箇所をま とめる。

表 5.1: レジスタの変更値

レジスタ	アドレス	値
1 KB 以上の read/write	$0x0000\ 0008$	0x0000 0001
SDRAM のタイミング調整	$0 \mathrm{xFEC0} 002 \mathrm{C}$	0x0000 0007

RMAP パケットの中身を確認するために、Link Analyzer (LA) を接続している。はじめ 2KB のイ ンクイメントデータを RMAP Write し、同じ領域の 2 KB のデータを RMAP Read した。その結果、 RMAPwrite/Sent パケットと RMAP Read/Reply パケットのデータ部分が一致しており、書き込ん だデータと読み込んだデータは全て一致していることを確認した。以上から、MIO と SpaceCard 間の SpaceWire を用いた通信は正常であることが確認できた。

5.2 XDS – PSP 間のインターフェース試験

[FPGAver.=8.0]

XBox と PSP 間の 9 つの信号を確認するため、XDS を図 5.3 のように接続し、XDS と MIO ボード 間に BOB を挟むことで信号波形をオシロスコープで確認した。まず、双方から信号を流さない状態で 各信号ペアの電圧値を測定した。結果を表 6.2 にまとめる。いずれも Active 側に入っていないことを確 認した。



図 5.3: 各機器の接続

LVDS 端子	電圧値 (V)	High/Low (Active)	LVDS 端子	電圧値 (V)	High/Low (Active)
TLM_CLK(+)	1.09	Low (High)	$SMP_CLK(+)$	1.08	Low (High)
$\mathrm{TLM}_{-}\mathrm{CLK}(-)$	1.48		$SMP_CLK(-)$	1.44	
$SCI_ENA(+)$	1.48	High (Low)	$CMD_DAT(+)$	1.11	Low (High)
$SCI_ENA(-)$	1.08		$CMD_DAT(-)$	1.48	
$SCI_DAT(+)$	1.09	Low (High)	$CMD_ENA(+)$	1.48	High (Low)
$SCI_DAT(-)$	1.48		$CMD_ENA(-)$	1.12	
$HK_ENA(+)$	1.48	High (Low)	$BASE_CLK(+)$	1.07	Low (High)
$HK_{-}ENA(-)$	1.07		$BASE_CLK(-)$	1.44	
$HK_DAT(+)$	1.07	Low (High)			
$HK_DAT(-)$	1.45				

表 5.2: 各信号線の電圧値

5.2.1 BASE_CLK, SMP_CLK, TLM_CLK

PSP の MIO ボードで生成されたクロック BASE_CLK を XBox へ送信し、その波形を確認する。SpaceCard から、MIO 上 FPGA のレジスタに RMAP Write し、Clock Module の state を CLK_BASE へと変更することで BASE_CLK を発 振させた。CLK_BASE state では、SMP_CLK は停止され、 BASE_CLK のみが発振される。レジスタの変更箇所は表 5.3 である。BASE_CLK 発振後にオシロスコープで波形の確認を した (図 5.4)。LVDS はノイズ対策により差動出力であるため 各信号の +- の差分波形を確認する。周波数 5 MHz が 800 mV の電位差で再現された。



図 5.4: BASE_CLK の信号波形。 マゼンダ線は BASE_CLK(+)(-) の 差分を表す。

表 5.3: レジスタの変更箇所

レジスタ名	アドレス	書き込み値	備考
XBoxClkOutInhibitFlag	0x100020D0	0x00000000	enable=0, disable=1
ClockModuleState	0x100020C0	0x00000002	0=CLK_INIT, 1=CLK_STDBY
			$2 = CLK_BASE, 3 = CLK_RUN$

同様にして、XBox から PSP へと返すクロック SMP_CLK の波形を確認する。レジスタ変更を行い、 Clock Module の state を CLK_RUN に変更して SMP_CLK を発振し、波形を確認した (図 5.5)。Sampling rate である 12.5 kHz の周期で信号が送られていることを確認した。

また、XBox が BASE_CLK を受けて TLM_CLK を返しているかを確認した (図 5.10)。こちらも 5 MHz でクロックを返してきていることが確認できた。

表 5.4:	レジスタの変更箇所
--------	-----------

レジスタ名	アドレス	書き込み値	備考
${\it clockModuleState}$	0x100020C0	0x00000003	3=CLK_RUN



図 5.5: SMP_CLK の信号波形。マゼンダ線は SMP_CLK(+)(-) の差分を表す。



図 5.6: TLM_CLK の信号波形。マゼンダ線は TLM_CLK(+)(-) の差分を表す。

5.2.2 CMD_ENA & CMD_DAT / SCI_ENA & SCI_DAT / HK_ENA & HK_DAT

次に、XBox へ SendHKAll コマンド (0x0040 0F34 0FC8 6576) を送信し、CMD_ENA と CMD_DAT の波形の確認を行った。SendHKAll コマンドとは、XBox へすべての HK 情報 (196 byte) の送信を要 求するコマンドである。XBox ヘコマンドを送信するために、XBoxTCModule と XBoxHKModule の state を RUN へと変更した。RMAP Write したレジスタは表 5.5 にまとめる。

表 5.5: レジスタの変更箇所

レジスタ名	アドレス	書き込み値	備考
XBoxTCModuleState	0x10002008	0x00000002	0=XTC_INIT, 1=XTC_STDBY
			2=XTC_RUN
XBoxHKModuleState	0x1000200C	0x00000002	0=XHK_INIT, 1=XHK_STDBY
			2=XHK_RUN

SendHKAll コマンドを送信した際の CMD_ENA の範囲内で有効になっている CMD_DAT の波形から データ内容を読み取った (表 5.8)。送信した SendHKAll コマンドと同じ値を送信できていることが確認 できた。また、図 5.9 の波形から、HK_ENA の有効範囲は HK データ長と同等の時間長さであった ($314\mu s = 196$ Byte(HK のレスポンスサイズ)×8 bit×0.2 μs (転送速度:5 MHz))。

今度は、XDS からの Science データを受信するために Science Module の state を RUN に変更し、 SCI_ENA と SCI_DAT の波形を確認した。変更レジスタは表 5.6 である。

表 5.6: レジスタの変更箇所

レジスタ名	アドレス	書き込み値	備考
SciModuleState	0x10002000	0x00000002	$0 = $ SCI_INIT, $1 = $ SCI_STDBY
			2=SCLSTDUP, 3=SCLRUN

図 5.10 の SCI_ENA に関しても、HK_ENA の有効範囲は HK データ長と同等の時間長さであった (62.4 μ s × 312 bit (SCI_DAT のサイズ) × 0.2 μ s (転送速度:5 MHz))。

Agilent Technologies THU JUL 29 02:23:11 2010						
5000/ 2 :	5000/ 🚦 1	.00V/ 🚦 '	1.00V/ 🔆	5.960% 2	.000%/ Stop	ł 🚺 1.26V
TO MANY THE PARTY					R	
•	-turno	m	America A	·····		
e	-	and the second	American	anna	manin	
		n din meneration.	ar fan de skere en jally	are all a second	an that the state of the state	1
	فالتعقل والطرا					
5 5						
AV - 12 900000		1/1/	- 70 125kUz		AV(2) = -1.995	75V
- 12.800000	Source	17 <u>4</u> X -	Y 1) X1	1.000 X2	€ v1 v2
Manual	2			-800.000ns	12.0000us	- 11 X2

図 5.7: CMD_ENA (黄,緑) と CMD_DAT (青,赤) の信号波形。マゼンダ線は CMD_ENA(+)(-) の 差分を表す。



図 5.9: HK_ENA (黄,緑)とHK_DAT (青,赤)の 信号波形。マゼンダ線はHK_ENA(+)(-)の差分 を表す。

Bit (2 進数)	0000	0000	0100	0000
Hex (16 進数)	0	0	4	0
	0000	1111	0011	0100
	0	F	3	4
	0000	1111	1100	1000
	0	F	С	8
	0110	0101	0111	0110
	6	5	7	6

図 5.8: オシロスコープの波形から読み取った CMD_DAT の中身。



図 5.10: SCLENA (黄,緑) と SCLDAT (青,赤) の信号波形。マゼンダ線は SCLENA(+)(-)の差 分を表す。

5.2.3 各信号間のの Delay

BASE_CLK & TLM_CLK

PSP から BASE_CLK 信号を送信してから、XBox から TLM_CLK 信号が帰ってくるまでの Delay を 測定した。オシロスコープで確認した波形を図 5.11 に示す。波形から BASE_CLK と TLM_CLK の Delay は、14 ns であることが分かった。

SMP_CLK & SCI_ENA

PSP が SMP_CLK 信号を送信してから、SCLENA 信号が active low になるまでの時間 (Time window) を測定した (図 5.12)。波形より、Delay は 20 µs であることが分かる。つまり、XDS との接続試験時 は、sciEnaDelayLower と sciEnaDelayUpper で決まる Time window の範囲を 20 µs が収まるよう に設定すればよい。

CMD_ENA & HK_ENA

続いて、PSP から XBox へ HK を要求するコマンドを送信してから、PSP へ HK が返ってくるま での時間を測定した。オシロスコープで確認した波形を図 5.13 に示す。PSP のコマンド送信が終わり、 CMD_ENA が OFF になってから XBox の HK 送信が終わり、HK_ENA が OFF になるまでの時間は、 41.8 ms であった。



図 5.11: BASE_CLK (黄, 緑), 図 5.12: SMP_CLK (黄, 緑), 図 5.13: CMD_ENA (黄, 緑), TLM_CLK (青, 赤) SCLENA (青, 赤) HK_ENA (青, 赤)

以上により、XDS – PSP-BBM 間の 9 本の信号全てが正常に出力されていることを確認された。

5.3 MIO ScienceModule での Science データ取得試験

XDS 内部に保持しているデータを用いて、PSP-BBM によるサイエンスデータ取得試験を行った。こ の時のセットアップを図 5.14 に示す。はじめに、MIO の Science Module の state を RUN にして XDS ヘクロック配信を行う。クロックを受け取ると、XDS はサイエンスデータの送信を開始する。MIO で は、節 4.7 で説明したモジュールのトリガ機能でこのデータを処理し、WFRB と EDB に保存する。最 終的に WFRB を SpaceCard で RMAP Read して、PSP で受けとったデータと XDS に保存されている データの比較確認を行う。また、EDB の中身も RMAP Read で読み出し、trigPtr の確認を行った。



図 5.14: セットアップ

XDS のダイアル 1 のチャンネル 0 のデータを 10 秒間取得し (ScienceModule を 10 秒間 RUN にした)、WFRB の中身を確認した。WFRB のデータの先頭の SampleCnt は DelayBuffer のサイズから始まっており (ver.2 では 65)、XDS に保存されているデータの 65 番目以降に一致した。次に、EDB の 先頭の trigPtr と、Linux 上でのシミュレーションソフトウェアである sxspspsim (章 7) に同様のデー タを入力したときの trigPTr の比較を行い、一致することを確認した。以上のように、WFRB と EDB ともに正常なデータが保存されていることが確認できた。

5.4 SpaceCard 処理速度測定

PSP-BBMの SpaceCard ボード (#10)単体を用いて、SpaceCard の処理速度を測定した。この測定 では、CPU上でパルスのトリガと最適フィルタ処理を行い、CPU の計算時間をカウントする。パルス のトリガは、CPU で MIO と同じ波形処理のアルゴリズムを組んで行った。最適フィルタ処理で使用す るテンプレートは、ノイズ無しの平均パルスを用いた。通常、最適フィルタ処理において波形とテンプ レートの相関をとる際、テンプレートを比較する箇所を対象波形のピークとその前後にシフトさせた場 合の複数回計算を行い、 χ^2 が最小となる所で波高値 (PHA)を決定するが、今回はシフト量を常に 0 と して 3 回計算を行うこととした。使用したデータは図 5.15 で示した、テンプレートと同じノイズ無し 平均パルスが連続的に並んだデータを使った。各グレード毎の処理速度を比較するために、それぞれパ ルスの到来する時間間隔を変化させ、パルスが全て同じグレード (HR, MR, LR) となるデータを準備 した。



図 5.15: CPU で処理させる波形データ

まず、コンパイラの最適化オプション無しで、CPUを100%使用した場合の処理速度の結果を表5.7 に 示す。節3.3.1 にも挙げられているように、SXSの要求値では150 cts/s/array であるため、PSP-A, B の2系統合わせてSpaceCard4ボード構成の場合、1ボード当たりの要求値は37.5 cts/sとなる。表5.7 の結果より、short templateであれば4ボード構成で十分に要求値を満たせる。しかし、long template で計算する場合、HRとMRが混ざったイベントであれば4ボード構成でも満たせそうだが、十分とは 言えない。なお、LRの最適フィルタ処理時間が短いのは、対象パルスと平均パルスの微分波形の差分 でレコード長分セカンダリーパルスを探すため、あまり時間がかからないためである。

- Long template						
グレード	イベント数	処理時間	(最適フィルタ+トリガ)	カウントレート		
LH	200	$5213 \mathrm{\ ms}$	(1483 ms + 3689 ms)	38.4 cts/s		
LM	622	$11707~\mathrm{ms}$	(1007 ms + 10554 ms)	53.1 cts/s		
LL	2487	$38011 \mathrm{~ms}$	(80 ms + 37594 ms)	65.4 cts/s		
- Short template						
SH	218^{1}	$2455~\mathrm{ms}$	(600 ms + 1818 ms)	88.8 cts/s		
SM	622	$5900 \mathrm{ms}$	(526 ms + 5256 ms)	105.4 cts/s		
SL	2487	$17911 \mathrm{ms}$	(60 ms + 17569 ms)	138.9 cts/s		

表 5.7: 最適化オプション無しの CPU 処理速度

¹イベント数にバグ有り(本来は200)。

次に、上記と同じプログラムを最適化をかけてコンパイルを行った。最適化オプションは処理速度優先のものをかけて実行している。今回も CPU を 100%使用して処理を行った。表 5.8 が処理速度の結果である。最適化オプション無しの場合に比べて、2 倍以上の処理速度が実現できた。この結果より、SpaceCard 4 ボード構成であれば、HR のイベントだけであっても、1 ボード当たりの CPU ロードは short template の場合は 17%, long template の場合は 38%程度で処理が可能であることが分かった。これにより、SpaceCard 4 ボード構成で十分に SXS の要求値を満たせると言える。

- Long template						
グレード	イベント数	処理時間	(最適フィルタ+トリガ)	カウントレート		
LH	218^{1}	$2199~\mathrm{ms}$	(ログ無し)	99.1 cts/s		
LM	622	$5177 \mathrm{\ ms}$	(//)	120.1 cts/s		
LL	2487	$16272~\mathrm{ms}$	(//)	152.8 cts/s		
- Short template						
SH	200	$927 \mathrm{\ ms}$	(//)	215.7 cts/s		
SM	622	$2496 \mathrm{\ ms}$	(//)	249.2 cts/s		
SL	2487	$7033 \mathrm{\ ms}$	(//)	353.6 cts/s		

表 5.8: 最適化オプション (スピード優先) を付けた CPU 処理速度

¹イベント数にバグ有り(本来は200)。

5.5 データ転送速度測定

MIO ボード – SpaceCard 間のデータ転送速度の測定を行った。この試験では、MIO ボードの SDRAM 内のバッファに収められたデータを SpaceCard 側から要求 (RMAP Read) し、MIO から SpaceCard に データを転送する際にかかった時間を測定することで、データ転送速度を求める。これを MIO ボードと SpaceCard ボードの枚数を変えて行うことで、要求速度を満たす最適なボード構成を調べた。MIO ボー ドからの波形データの読み出しの際に要求される速度は、サンプリングレートに応じて

=
$$12.5 \text{ kHz} (15.625 \text{ kHz}) \times 36 \text{ bit} \times 18 ピクセル$$

$$=$$
 7.2 Mbps (9 Mbps)

と求められる。サンプリングレート 15.625 kHz の場合でも要求を満たせるよう、最低 9 Mbps は確保で きるかを検証する。節 3.4 で述べたとおり、SpaceWire のデータリンクを流れるデータは全てパケット の集合で通信されている。そのため、RMAP Read のパケットサイズを 0.5, 1, 2, 4, 8, 16 KByte と場 合分けして測定を行った。なお、SDRAM 内から読み出すデータサイズは 4 MB とし、各パケットサイ ズで転送が完了するまでの CPU 空き時間も同時に測定した。

5.5.1 1 MIO + 1 SpaceCard 構成の場合

まず、1つの SpaceCard から 1 つの MIO 上のデータを RMAP Read した際のデータ転送速度を計測 した。測定結果を図 5.16 に示す。PSP は 2 系統 (PSP-A, PSP-B) 搭載されるが、1 系統当たりのボー ド構成が (1 MIO + 1 SpaceCard) の 2 ボード構成の場合と、(2 MIO + 2 SpaceCard) の 4 ボード構 成の場合の 2 種類を想定して評価を行った。典型的なデータのアクセスサイズは 4 KB²が想定されてい る。パケットサイズ 4 KB 付近で検証すると、1+1 の 2 ボード構成の場合では、要求速度 (9 Mbps) に 達しておらず、パケットサイズを大きくしても不足していることが分かる。2+2 の 4 ボード構成にした 場合、単純に処理するデータ量は半分になるので要求速度も半分 (4.5 Mbps) となるため、要求速度を 満たせている。また、CPU のアイドル時間の割合を見てみると、パケットサイズ 4 KB で 30 %程度と 低く、データの RMAP Read だけで CPU の半分以上を占有してしまっていることが分かった。



図 5.16: 1 MIO – 1 SpaceCard 間の転送速度。赤い点線はそれぞれ 1+1 (9 Mbps), 2+2 (4.5 Mbps)の ボード構成の場合の要求速度。

5.5.2 1 MIO + 2 SpaceCard 構成の場合

次に、2つの SpaceCard から1つの MIO 上のデータを RMAP Read した際のデータ転送速度を計測 した。測定結果を図 5.17 に示す。2台の SpaceCard から同時に MIO の SDRAM へ RMAP Read を行 い、それぞれの転送速度足し合わせた転送速度を求めた。プロット (橙色)を見てみると、パケットサイ ズが2 KB 以上で要求速度 (9 MB)を満していることが分かる。これより、2つの SpaceCard でデータ の読み込みを分担することで十分に要求速度を達成できることが分かった。また、CPU 使用率を見てみ ると、アイドル時間の割合がパケットサイズ4 KB 時に 40 %程度と低く、依然として CPU の負荷が大

²1 パケット内に 1 record length (32 bit×1024 sample) 分が収まる計算。
きいことが問題となっている。



図 5.17:1 MIO – 2 SpaceCard 間の転送速度。赤い点線は要求速度 (9 MB)。

以上の結果より、要求されるデータ転送速度を満たすことのできる PSP 1 系統当たりのボード構成は、 (2 MIO ボード + 2 SpaceCard ボード)、もしくは (1 MIO ボード + 2 SpaceCard ボード) であること が分かった。しかし、いずれの場合でも CPU 使用率が想定よりも高いことが分かった。CPU で処理を 行うタスク (小節 4.9.2 参照) は複数存在し、データ読み出しに半分以上使用することはパフォーマンス 的に望ましくない。本測定で使用した PSP-BBM 仕様の SpaceCard ボードでは、EM (FM) に実装予定 である SpaceCard 上の CPU I/F FPGA に接続される高速ポートおよび SpaceWire ルータが実装され ておらず、MIO ボードは CPU 上の SpaceWire ルータのポートと直結している。

これらの測定結果から、4 ボード構成 (2 MIO ボード + 2 SpaceCard) では各ボードの接続が増え、 機能試験が複雑化するため、コストと時間の面を考慮して性能実証モデル (EM) では、3 ボード構成 (1 MIO ボード + 2 SpaceCard) で製作を進めることが決定した (2010 年 1 月現在)。また、PSP-EM では 高速ポートが実装されるため、通信速度 · CPU 負荷率が共に改善することが見込まれている (見積もり 値: 8.3 Mbps, 52 %)。

第6章 Xbox-BBM – PSP-BBM 噛み合わせ試験

本章では、2010 年 7 月 28 日 ~ 8 月 6 日に NASA/GSFC (Goddard Space Flight Center) にて行った XBox-BBM (Bread Board Model: 試験用モデル) と PSP-BBM の噛み合わせ試験について述べる。本 試験の目的は、NASA 開発担当の XBox-BBM と、日本開発担当の PSP-BBM とを接続して双方の間の 電気インタフェースの確認を行い、検出器からの信号を正常に PSP 側で取得することを目的としてい る。実際の検出器は開発段階であるため、今回は検出器を模擬した Detector simulator を用いる。試験 内容は以下のとおりである。

- 1. XBox-BBM PSP-BBM 間インターフェース試験
- 2. Detector simulator からの Science データ取得試験
- 3. XBox-BBM PSP-BBM 間のコマンド / レスポンス試験
- 4. ADRC-BBM の RMAP 通信試験 (埼玉大 下田修論参照)

6.1 XBox-BBM

NASA 側が準備した XBox-BBM は、XBox-A, XBox-Bの2系統のボードを一つの筐体に収めることのできる ものが提供された。今回は XBox-A に当たるボードのみ が搭載されており、半分の18 ピクセルが処理可能であ る。XBox-BBM のもつ各端子の配置と対応は以下の通り である。

図 6.2 の左上から

J203: XBox-Aのpx[0]-[8] (本実験ではボード無し)
J202: XBox-Aのpx[9]-[17] (本実験ではボード無し)
J232: PSP-A
J222: XBox-Aの電源
J210: XBox-A& XBox-B共用のGND
J234: PSP-B
J224: XBox-Bの電源
J204: XBox-Bのpx[0]-[8]
J205: XBox-Bのpx[9]-[17]

XBox のシャーシ GND は Ditector-sim を通じて Factory GND へ接続



 \boxtimes 6.1: NASA/GSFC σ XBox-BBM



6.2 Detector Simulator

Detector simulator (以下 Detector-sim) は、実際の Detector の信号を模擬したダミーパルスのアナログ信号を送出する装置 である。今回、GSFC 側が用意した Detector-sim は現在運用中 の X 線天文衛星「すざく」XRS の地上試験でも用いられたも のである。Detector-sim から XBox-BBM 内における波形信号 の処理回路を図 6.4 に記す。

Detector-sim で初めに生成されるパルスは、波高値が 10 V の 下に凸の波形であり、周期 1 Hz (変更可能)の単発パルスが連 続的に出力される。10 V と高い電圧のパルスしか出せないた め、抵抗を通すことで出力電圧を mV 単位に下げている。



図 6.4: Detector-sim と XBox の回路図



図 6.3: Detector-simulator 外観

6.3 セットアップ

6.3.1 LVDS 信号線の電圧チェック

まず、Detector-sim と XBox-BBM の接続して GND の確認を 行った。XBox の各端子の接続先を表 6.1 にまとめる。Detectorsim と XBox の GND レベルが同じになるよう GND 同士を接 続し、Detector-sim 側を Factory-GND へ落としている。電源 は AC 電源 119.5 V から変圧して使用する。Detector-sim から の 18 ピクセル分のアナログ波形信号は、分岐したケーブルを 用いて 9 ピクセルずつ XBox に出力する。

XBox の D-sub 端子 J234 に BOB を接続し、XBox – PSP 間の全 9 つの信号線の電圧チェックを行った。なお、電圧測定

XBox	Detector-sim (Cable ID)
J204	J402 (P204)
J205	J402 (P205)
J210	J420
	電源 (DC)
J224	\pm 25V, \pm 25V, \pm 8V

表 6.1: 各端子の接続先

はそれぞれの信号線のシールドが導通していることを確認した上で TLM_CLK のシールドを GND に落 としてから行っている。表 6.2 より、TLM_CLK は仕様上、Low で - 端子側の電圧値が低いはずである が、これは XBox に PSP を接続していないためであると考えられる (図 6.8 参照)。

6.3.2 Merge XBox – PSP

XBox 側と PSP 側の抵抗チェックと電圧チェックを済ませたところで、Detector-sim – XBox と PSP との 接続を行った。接続状況をを図 6.6 に示す。BOB 上で BASE_CLK, SMP_CLK, CMD_ENA, CMD_DAT を対応する端子をショートさせ、GND は BOB の 19 番 (TLM_CLK shield の XBox 側) に落としている。





図 6.5: BOB, Detector-sim – XBox 間接続の様子

LVDS 端子	電圧値 (V)	High/Low (Active)	LVDS 端子	電圧値 (V)	High/Low (Active)
$TLM_CLK(+)$	1.39	High (High)	$SMP_CLK(+)$	1.08	Low (High)
$\mathrm{TLM}_{-}\mathrm{CLK}(-)$	1.01	Low が 正常	$SMP_CLK(-)$	1.44	
$SCI_ENA(+)$	1.49	High (Low)	$CMD_DAT(+)$	1.11	Low (High)
$SCI_ENA(-)$	1.01		$CMD_DAT(-)$	1.48	
$SCI_DAT(+)$	1.02	Low (High)	$CMD_ENA(+)$	1.49	High (Low)
$SCI_DAT(-)$	1.40		$CMD_ENA(-)$	1.12	
$HK_ENA(+)$	1.42	High (Low)	$BASE_CLK(+)$	1.07	Low (High)
$HK_ENA(-)$	1.01		$BASE_CLK(-)$	1.44	
$HK_DAT(+)$	1.01	Low (High)			
$HK_DAT(-)$	1.40				

表 6.2: 各信号線の電圧値



図 6.6: 各機器の接続

6.4 XBox – PSP 間インターフェース試験

オシロスコープのプローバーを BOB の各端子に繋ぎ、各 LVDS 信号の波形の確認を行った。引き続き BOB の 19 番で GND を取っている (図 6.7)。XBox – PSP 間の信号はノイズ対 策のため LVDS を使用しており、差動インタフェースで通信を 行っている。そのため、各信号線の +- の差分を確認する必要 がある。以下に挙げるオシロスコープのキャプチャー画像のマ ゼンダ線は、プローバーの ch 1 と ch 2 の値の差分を表してい る。各信号線 (±) を 2 組ずつオシロスコープで波形を表示し、 2 信号間の ON/OFF のタイミング、また Delay が正常値であ るかを確認した (図 4.6 参照)。



図 6.7: prober の接続

6.4.1 BASE_CLK & TLM_CLK

BASE_CLK は PSP から XBox へと出力される周波数 5 MHz のクロック信号である。TLM_CLK とは、 BASE_CLK に同期して XBox から PSP へと出力されるクロック信号である。図 6.8 では、BASE_CLK が Low になってから TLM_CLK が Low に入るのに 16 ns の Delay が生じている。XDS での試験 (小節 5.2.3) ではこの Delay は 14 ns であったが、誤差の範囲と言える。

6.4.2 SMP_CLK & SCI_ENA

SMP_CLK は、BASE_CLK を元に周波数 12.5 kHz (もしくは 15.625 kHz) で PSP から XBox へと出 力されている Sampling rate を規定するためのクロック信号である。図 6.9 では、SMP_CLK が Active High に入ってから SCLENA が Active Low になるまで 20 μ s であることが分かる。つまり、sciEnaDelayLower と sciEnaDelayUpper で決まる Time window の範囲を 20 μ s が収まるように設定すればよ い。また、SCLENA は SCLDAT の有効範囲を示しており、Active Low の継続時間は 62.4 μ s と SCLDAT の長さと正常に一致している (62.4 μ s = 312 bit (SCLDAT のサイズ) × 0.2 μ s (転送速度:5 MHz))。



図 6.8: BASE_CLK と TLM_CLK の信号波形。 マゼンダ線は TLM_CLK(+)(-) の差分を表す。

	信号名	i	Cha	nne	l (緆	(色)	A	ctive
SCI_EI	NA(+))		ch1(黄	i)			Low
	(-)			ch2(緑	ŧ)			
SMP_C	LK(+))		ch3(行)]	High
	(-)			ch4(赤	;)			
Agilent Tec	hnologies						FRI JU	L 30 22:38	3:38 2010
1 1.00V/ 2 1	.00V/ 🕄 1.00\	// 🖪 1	.00V/	¢- 60.001	20	.00%	/ Stop	t 1	1.23V
т				-					
	andungi.			all abstract in					nisi Kensi-teli
22									· · · · · · · ·
-	an frances i sugar	an and a second	n decenary.		rihana-a	romanda	er hopomerouger	a ang ang ang ang ang ang ang ang ang an	
									-
ŝţ									
		1. (1) (50.0001			ED.Y			
∆x = 20.000000	A Source	17ΔX =	50.000k	TZ X1			NI) = 0.0V X2		1.1/2
Manual	Math	~		-20.4000	Jus		100.000ns	X	T X2

図 6.9: SMP_CLK と SCI_ENA の信号波形。 マゼンダ線は SCI_ENA(+)(-) の差分を表す。

6.4.3 CMD_ENA & HK_ENA

次に、PSP 側から Send_HKall コマンドを送信 して XBox から HK データを取得することで、送信 したコマンドデータの波形と帰ってきた HK デー タの波形を確認した。CMD_ENA、HK_ENA は、 それぞれ CMD_DAT, HK_DAT の有効範囲を示す ものである。図 6.10 より、CMD_ENA が Active Low に入ってから 42 ms 後に HK_ENA のレスポ ンスが来ている。Comand Response Time は最大 で 43 ms なので正常である。また、図 6.11 では CMD_ENA の Active Low 継続時間が CMD_DAT の長さ 12.8µs と正常に一致している (12.8µs = 8 Byte(CMD_DAT のサイズ)×8 bit×0.2µs (転送速 度:5 MHz))。同様に、図 6.4.3HK_ENA の Active Low 継続時間が HK_DAT の長さが 314µs と正常 に一致していることが確認できる ($314\mu s = 196$ Byte(HK のレスポンスサイズ)×8 bit× 0.2μ s (転送 速度:5 MHz))。



図 6.11: CMD_ENA の拡大画像。



図 6.13: HK_ENA と HK_DAT の拡大図。 マゼンダ線は HK_ENA(+)(-) の差分を表す。



図 6.10: CMD_ENA と HK_ENA の信号波形。 マゼンダ線は HK_ENA(+)(-) の差分を表す。



図 6.12: HK_ENA の拡大画像。



図 6.14: SCLENA と SCLDAT の拡大図。 マゼンダ線は SCLENA(+)(-) の差分を表す。

6.5 Science データ取得試験

XBox - PSP 間の9つの信号が正常に通信できていることが確認できたため、次に Science Dataの確認を行う。Detector-simulatorからダミーパルスを周期的に送出し、XBox を介して PSP で正常に取得できているかを検証した。

6.5.1 Signals from the Detector-sim

本実験では実際の検出器ではなく、代わりに Detector-sim から出力されたダミーパルスの処理を行う。 Detector-sim から送出された下に凸のダミーパルスは、XBox-BBM 内で信号の増幅と反転の処理が行われる (図 6.4)。今回 NASA/GSFC 側が用意した XBox-BBM では、開発の関係上、反転処理は チャンネル 0 のみに加えられている。その他のチャンネルに関しては、下に凸のまま波高値が負の値に Clip している状態で実験を行った。なお、Detector-sim から発せられるパルスは全チャンネル共に同じ時刻で送出されている。XBox の ADC のダイナミックレンジは 14 bit で $-3V \sim 3V$ まで表示可能である。パルスのピークとアンダーシュートが同じスケールで上限値 / 下限値内に収まるように、-1.5 V (-4096 adu: ADC サンプル値) のオフセットがかかっている。今回は初めに、周期が 1 Hz, パルスのピーク電 圧が 1 V (+2730 adu), つまり電位差 2.5 V 程度の波高値を持ったダミーパルスを出力して波形データの取得を行った。



図 6.15: ダミーパルスの例

6.5.2 Enable channels

まず、Detector-sim から送出されるチャンネル 0 の波形データを PSP で取得するために, Enable Amplifier コマンドを XBox に送信し、チャンネル 0 が enable に変更されているかを HK Digital Status を RMAPread することで確認する。PSP から送信したコマンド、XBox から帰ってきた HK Digital Status の値を表 6.3 にまとめる。

コマンド名		コマンド packet words	HK Digital Status 16-17 word	
		(16-bit hex)	(include CAL_AMP_ENA)	
EnableAmplifiers	MSB	x0288, [P_1], [P_2], [CRC_CMD]	LSB	x0000, x0100

表 6.3: 送信コマンドと calorimeter amplifiers enable の値

1 word = 16 bit

[P_1] = x0000: LSB から数えて Bit 0 ~ Bit 8 が ch9 ~ ch17 の enable(1)/disable(0) に対応
 [P_2] = x0001: LSB から数えて Bit 0 ~ Bit 8 が ch0 ~ ch8 の enable(1)/disable(0) に対応

[P_2] の ch0 に対応する部分を enable にしたコマンドを送信後, SendHKall コマンドを送ることで返って きた HK Digital Status の CAL_AMP_ENA(calorimeter amplifiers enable state) の値を読んだ。HK status data のコマンドレスポンスサイズは全部で 98 words (196 Bytes), その内 20 words が HK Digital Status を表記している。CAL_AMP_ENA は 16 word 目から 17 word 目にかけて設けられている (表 6.4)。CAL_AMP_ENA の値から、channel 0 が enable に切り替わっていることが確認できた。

Word	Field Bit	Field Name	Field Description
16	63	JFET_VDD2_EN	JFET_VDD2 σ enable 状態 (enable:1, disable:0)
16	62	JFET_VDD1_EN	JFET_VDD1 σ enable 状態 (enable:1, disable:0)
16	61:59	TEMP_CTRL_MODE	temperature controller $\Xi - F$
16	58	ANTICO_AMP_EN	anti-coincident amplifier σ enable σ 状態 (enable:1, disable:0)
16-17	57:40	CAL_AMP_EN	カロリメータ amplifier の enable 状態 (enable:1, disable:0)
17	39:32	CAL_BIAS_REG	カロリメータバイアスの設定電圧値 (Bit 数)

表 6.4: HK Digital Status の 16 - 17 word の bit 対応

(詳細は「Astro-H SXS Xbox Command and Command Response List」参照)

6.5.3 WFRB 内の Signal 波形の検証

チャンネル 0 が enable に切り替わったところで、Detectorsim からカウントレート 1 Hz のダミーパルスを連続的に送信 する。XBox 内で反転・増幅された後の ADC 手前で、オシロ スコープのプローバーで直接アナログ波形を確認する。また、 MIO ボードに搭載されている SDRAM 内に設けられた Wave-Form Ring Buffers (WFRB)の中身を確認し、XBox が送信し ている波形と同等の波形データが PSP 内に保存されているか を検証した。

まず、Detector-sim からダミーパルスを流さない状態で PSP に Science データを取得し、データの中身を確認する。図 6.17 が WFRB の channel 0 に保存されてたデータを dump した adcsample のプロットであり、横軸が時間 (sample 番目)、縦軸が adcsample(adu) である。波形が $-4500 \sim 2500$ adu の peak-topeak を持っており、電圧値に換算するとおよそ ~ 0.7 V のノ イズが乗っていることが分かる。また、周期がおよそ 60 Hz で あり、これは使用している実験室の商用電源と同等である。



図 6.16: XBox へのプローバーの接続



図 6.17: pxWFRB[0] の dump(パルス無し)



次に、ダミーパルスを送信し、channel 0 の WFRB 内の波形データを確認した (図 6.5.3)。ある周期 でノイズが含まれており、オシロスコープで見られる波形も正常でないことが確認された (図 6.21)。そ こで、オシロスコープの prober を channel 1 に付け替えて波形を確認したところ、こちらはきれいな形 状のパルスが見られた (図 6.22)。







図 6.21: ch 0 の波形 (反転無し)



図 6.20: ノイズの拡大図



図 6.22: ch 1 の波形

チャンネル 0 以外の他チャンネルの波形も確認するため、全チャンネルの EnableAmplifiers を ON にして再度 Science データの取得を行った。図 6.23 に ch 0, ch 1 の波形プロットを示す。ch 0 ではノ イズの出力がパルスに対して大きい。ch 1 に関しては、比較的きれいな波形が見られるが、想定してい ないパイルアップイベントが混在していた。又、ch 0 以外は前述の通りパルスに反転処理を加えていな いため、負の値で Clip している。



図 6.23: pxWFRB[0], [1] のデータプロット

その後、XBox 側の端子 |J204|, |J205|をショートさせ、anti-co amplifier を ON にした状態で ch 0 の 波形を確認したが、ノイズの改善は見られなかった。その他のチャンネルも含めた XBox-BBM の問題 点を以下にまとめる。

- px[0] と px[2] はかなり noisy
- px[1] ~ px[17]のV_{in}は反転無し
- Anti-coの信号はADCの手前まではnoisy
 - VD(Volts, digital) は反転処理有り
 - VD = 2.4 V (XBox 側)VD > 3.0 V (PSP 側) enable/disable 共に

6.5.4 Detector-sim – XBox-BBM 間の接続変更

Detector-sim 側の端子 J402 から pixel 9 チャンネル分ず つ 2 つに分岐して出力しているケーブル P204, P205 の接 続先を入れ替えて再試験を行った。各コネクタの接続先は以下 の通りである (表 6.5)。前述同様、接続変更後に ch 0, ch 1 の 波形をオシロスコープ、WFRB の中身それぞれを確認した (図 6.24, 6.25)。ch 0 の波形が接続変更前に比べてきれいな形状に なっている。これにより、Detector-sim と XBox-BBM とを繋 ぐケーブルがノイズに影響していることが分かった。

Detector-sim	(Cable ID)	XBox
J402	(P204)	$J205 \rightarrow J204$
	(P205)	$J204 \rightarrow J205$

表 6.5: 接続変更先



図 6.24: ch 0 (上), ch 1 (下)の波形 の同時表示



図 6.25: WFRB[0] の波形 (上), WFRB[1] の波形 (下)

6.5.5 全 pixel の Science Data チェック

Detector-sim, XBox, PSP を全て接続し、オシロスコープで 波形が送信されていることを確認した後に、Sampling rate を 12.5 kHz (baseCntRoundup = 400), 13.9 kHz (baseCntRoundup = 360), 15.625 kHz (baseCntRoundup = 320) にして、それぞれ全てのチャンネルで正しく 20 秒分のデータ が取得できるか確認した。Detector-sim — XBox 間を繋ぐケー ブルの片側 P204] に問題があったため、P205] のみを接続し て行った。そのためパルス信号が入力されるのは ch 0 ~ ch 8 の 9 チャンネルだけとなる。接続は以下の通りである (表 6.6)。

Detector-sim	(Cable ID)	XBox
J402	(P205)	J204
		J205 (ショート)

表 6.6: ケーブルの接続



図 6.26: ch 0 の波形 (黄), ch 1 の波 形 (緑)

それぞれの Sampling rate でデータを取得した際の結果を表 6.7 にまとめる。Sampling rate が 12.5 kHz の場合、channel 0 では正常にパルスが取得出来ていることが確認できた。ch 5 に関して、データが 取得出来なかったのは Detector-sim を操作している PC 側の問題であることが判明した。又、Sampling rate を 12.5 kHz よりも早めると XBox 内の Buffer が間に合わず、後半のチャンネルのデータが次のデー タで上書きされてしまい、データが破損するということが生じた。13.9 kHz で Science データの取得を 行った場合、ch 17 が上書きされて破損データとなっており、15.625 kHz では全チャンネルデータが初 期値のままであった。12.5 kHz と 13.9 kHz で取得した波形データを付録 B に示す。

表	6.7:	各	Pixel	の取得デ	ータの	結果
---	------	---	-------	------	-----	----

Channel	Sampling rate : 12.5 kHz (baseCntRoundup = 400)
ch 0	正常
ch 1	誤った周波数
$ch 2 \sim ch 4$	正常 $(V_{in}$ なし, 負に $Clip)$
ch 5	データなし
$\mathrm{ch}\; 6 \sim \mathrm{ch}\; 7$	正常 (V _{in} なし, 負に Clip)
$\mathrm{ch}\;8\sim\mathrm{ch}\;17$	データなし

Channel	Sampling rate : 13.9 kHz (baseCntRoundup = 360)
$ch \ 0 \sim ch \ 16$	上記 12.5 kHz 時と同様
ch 17	破損データ

Channel	Sampling rate :15.625 kHz (baseCntRoundup = 320)
ch $0\sim$ ch 17	データなし

上記の接続では正しいデータを取得できなかったため、接続を XBox sim – BOB – PSP BBM と変更 した。データが取得できない原因を追求するため、UserFPGA の version の変更や、sciEnaDelay の値 を変更することで time window を変化させて実験を繰り返した。結果を、表 6.8 にまとめる。

baseCnt-	Rate (kHz)	sciEna	aDelay	UerFP	GA ver.	Rec (s)	Result
Roundup		Lower	Upper	XBox	MIO		
320	15.625	75	150	5	8	20	no data
400	12.5	75	150	5	7	20	correct data
320	15.625	75	150	5	7	20	no data
320	15.625	0	312	5	7	20	wrong data
320	15.625	0	312	5	8	20	wrong data
320	15.625	75	200	5	8	20	no data
320	15.625	0	125	5	8	20	wrong data
400	12.5	0	125	5	8	20	correct data
320	15.625	20	125	5	8	20	no data
320	15.625	1	125	5	8	20	no data
320	15.625	150	200	3	8	20	no data
400	12.5	150	200	3	8	20	correct data
380	13.158	150	200	3	8	20	wrong data
380	13.158	75	125	5	8	10	correct data (except ac)
400	12.5	75	125	5	8	10	correct data
320	15.625	75	125	5	8	10	no data
340	14.705	75	125	5	8	10	no data
320	15.625	0	125	5	8	10	correct data (half-way)
360	13.9	75	125	5	8	10	correct data (only 0-16 ch)

表 6.8: サンプリングレート、FPGA のバージョンを変更し WFRB の中身を確認した結果。

これらの結果より、以下の様な事項が確認できた。

- 20 s分のデータを取得するとWFRBが1周してしまうので、wrong data が入っているようにみ えてしまう。
- SMP_CLK が 12.5 kHz (sciENADelay=400) よりも速くなると、XBox のバッファが書き潰されてしまい、後ろのチャンネルのデータがおかしくなってしまう。
- 破損データなどにより、parity error が3つ以上のチャンネルでつくと、MIO が誤作動をおこし、 正しいデータが取得できなくなる。

本実験の時点で XBox-BBM は、12.5 kHz よりも早い Sampling rate でのデータ取得は対応できていないと言うことが判明した。現在 NASA/GSFC にこの点の改善を要請している段階である。

6.5.6 波高値の比較 (ch 0)

PSP-BBM で Sampling rate = 12.5 kHz で取得した ch 0 の波形データの波高値と、オシロスコープ で測定した波高値を比較し、正常に波形が再現されているか検証した。

図 6.26 の ch 0 より、オシロスコープのカーソルを利用して測ったパルスの波高値はおよそ 1.89 V で あった。XBox-BBM の ADC は縦軸 (電圧値) は $-3V \sim +3V$ を 14 bit で表しており、1 bit を 1 adu (adcsample unit) とすると $-8192 \sim +8192$ adu を取り得る。ADC の縦軸の分解能より、オシロスコー プで測った波高値の adu 換算での見積もりは、

$$1.89 (V) \times \frac{8192 (adu)}{3 (V)} = 5160 (adu) \cdots (推測値)$$
(6.1)

図 B の pxWFRB[0] のプロットより、保存されていたパルスのピーク値は 1130 adu であった。この 時のベースラインは -4050 adu だったため、波高値は

これより、送信された波形とWFRBに格納された波形データの波高値は大よそで一致していたため、 PSPで正常にデータ取得できたと言える。

6.5.7 波高値の比較 (ch0 : Short pulse height)

Detector-simから送信するパルスの波高値を前節のパルスよりも低いものに変更し、同様にPSPで取得出来るかを検証した。取得時の条件は以下のとおりである。

Sampling rate	$12.5 \mathrm{~kHz}$
Debug mode	ON
波高値	$\sim 1.05~{\rm V}$

表 6.9: 波形データ取得時の条件



図 6.27: ch 0 の波形 (黄), ch 1 の波形 (緑)

前節同様、オシロスコープの波高値と ADC の adcsample の値とを比較すると、

$$1.05 (V) \times \frac{8192 (adu)}{3 (V)} = 2867 (adu) \cdots (推測値)$$
(6.3)

$$-1150 (ピーク値) + 4020 (ベースライン) = 2870 (adu) \cdots (実測値)$$
(6.4)

こちらも波高値は大よそ一致していると言える。

6.6 XBox – PSP 間コマンド/レスポンス試験

6.6.1 HK Status データのチェック

「Astro-H SXS Xbox Command and Command Response List」に記載されている全ての XBox コマ ンドを PSP から送信して XBox へ要求を出し、SendHKAll コマンドで HK Status データの値を RMAP Read する事で、XBox が要求値を返しているかを確認した。SendHKAll コマンドは XBox に全ての HKStatus データを要求するコマンドであり、レスポンスサイズは全部で 98 words (196 Byte) で表 6.10 のようなグループに分かれている。今回は XBox コマンドで要求した値が、HK Status データの該当箇 所で変更されているかを1つずつ確認する作業を行った。詳細な結果を表 6.12, 6.13 に示す。Receive カ ウント、Sent カウントが正常にカウントされていることが確認できた。またレスポンスサイズも問題な く、エラーはカウントされず正常であった。

表 6.10: 全 HK パケットデータ (98 words)

0xC400, 0x0040, 0x0F34, 0x0FC8, 0x6576,
[16 HK Analog Group0 status data words],
[20 HK Analog Group1 status data words],
[20 HK Analog Group2 status data words],
[16 HK Analog Group3 status data words],
[20 HK Digital Group status data words],
[CRC_RESP]

6.6.2 バイアス電圧の HK Analog Statas 確認

上記で確認した HK データの内、供給電圧に関するコマンドの確認試験を行った。カロリメータや anti-co にかけるバイアス値、JFET に与えるバイアス値: Vdd, Vss を変更するために、PSP から XBox のレジスタヘコマンドを送信し、送ったコマンドの設定電圧値と HK に入っている電圧値を比較すると いう形で行った。XBox では、図 6.28 のように検出器から読み取った信号電圧を -3 V ~ 3 V に収まる よう変圧してからデジタル値として HK データに載せている。



図 6.28: XBox 内回路の概略図

そのため、HK に書かれている ADC code と電圧値の関係は、以下の式のようになる。

$$(\mathbf{\overline{a}E}\mathbf{\underline{m}}) = \frac{(\text{ADC code})}{4} \times \frac{6}{2^{14}} \times 3 \tag{6.5}$$

ADC code のデータ値は 16 bit 中 14 bit であり、PSP 側での HK データでは 16 bit 表記であるため 4 で 割ることで 2 bit シフトさせている。なお、各コンポーネントに流すことのできる最大バイアス値はそ れぞれ異なっており、XBox コマンド値: 0xFF で指定する最大電圧値は共通ではないことに注意する。 また、XBox の各レジスタに対応したピンの電圧値が初期状態で0Vになっていることを確認するため、 レジスタの変更前にテスターによる確認を行っている(図 6.29)。



図 6.29: Detecter-sim 側の様子

実際にコマンドを送信して得られた HK データの結果を表 6.11 にまとめた。PSP から指定した電圧 値と HK データで返された値から計算した電圧値がほぼ一致していることが確かめられた。この時、テ スターで直接測った電圧値も同値程度に変更されていることを確認している。SetAnticoBias において AnticoBiasout が2段書いてあるのは、時定数 6.2秒の RC 回路が入っているため、電圧が0ボルトに戻 るまでにタイムラグがあったためである。

コマンド名	値	HK Analog Status	電圧換算		
SetCalorimeterBias	-	CALBIAS1=0x0004	0 V	CALBIAS2=0x0004	0 V
$5\mathrm{V}$	$0 \mathrm{xFF}$	CALBIAS1=0x46C0	$4.97~\mathrm{V}$	CALBIAS2=0x46C8	$4.98~\mathrm{V}$
$2.5 \mathrm{~V}$	$0 \mathrm{x7F}$	CALBIAS1=0x2348	$2.48~\mathrm{V}$	CALBIAS2=0x2334	$2.48~\mathrm{V}$
0 V	0x00	CALBIAS1=0x000C	$0.00 \mathrm{V}$	CALBIAS2=0x0008	$0.00 \mathrm{V}$
SetColdJFETVdd	-	ColdJFETVdd1=0x000C	0.00 V	ColdJFETVdd2=0x0010	0.00 V
$5\mathrm{V}$	$0 \mathrm{xFF}$	ColdJFETVdd1=0x46B8	$4.99~\mathrm{V}$	ColdJFETVdd2=0x46BC	$4.99~\mathrm{V}$
$2.5 \mathrm{~V}$	$0 \mathrm{x7F}$	ColdJFETVdd1=0x2324	$2.49~\mathrm{V}$	ColdJFETVdd2=0x2334	$2.49~\mathrm{V}$
0 V	0x00	ColdJFETVdd1=0x000C	0.00 V	ColdJFETVdd2=0x0008	$0.00 \mathrm{V}$
SetColdJFETVss	-	ColdJFETVss=0x0010	0.00 V	-	-
-8.756 V	$0 \mathrm{xFF}$	ColdJFETVss=0x8000		-	-
-4.378 V	$0 \mathrm{x7F}$	ColdJFETVss=0xC258	$-4.34 \mathrm{~V}$	-	-
0 V	0x00	ColdJFETVss=0x000C		-	-
SetAnticoBias	-	AnticoBiasin=0x0008	0.00 V	AnticoBiasout=0x0024	0.00 V
$8.5 \mathrm{V}$	$0 \mathrm{xFF}$	AnticoBiasin=0x7854	$8.46~\mathrm{V}$	AnticoBiasout=0x605C	$8.00 \mathrm{V}$
$4.25 \mathrm{~V}$	$0 \mathrm{x7F}$	AnticoBiasin=0x3C10	$4.22 \mathrm{~V}$	AnticoBiasout=0x4944	$3.98~\mathrm{V}$
0 V	0x00	AnticoBiasin=0x000C	$0.00 \mathrm{V}$	AnticoBiasout=0x0A80	$0.01~\mathrm{V}$
数分後 $ ightarrow$				AnticoBiasout=0x0018	$0.00 \mathrm{V}$

表 6.11: HK Analog Status の確認

以上のようにして、すべてのステータスが正常に機能していることを確認できた。

Command name	Command Code	MIOLen	MIOErr	ErrCnt	RcvCnt	SentCnt	XboxLen	XboxErr	RespTime	Comment
ResetAll	0240 0F34 0FC8	12	0	0	1	1	12	0		0word
SendHKAll	0040 0F34 0FC8	196	0	0	2	2	196	0		POR=1, ALL_RST=1, 92wird
ResetFPGARegisters	0244 0F34 0FC8	12	0	0	3	3	12	0		0word
SendHKAll	0040 0F34 0FC8	196	0	0	4	4	196	0		REG_RST=1, POR=0, ALL_RST=0, 92word
ResetDACRegisters	0248 0F34 0FC8	12	0	0	5	ũ	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	9	9	196	0		DACRst=1, REG_RST=0
EnableCalorimeterBias2	0258 0F34 0FC8	12	0	0	7	7	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	~	8	196	0		CAL_BIAS2_EN=1, 92word
DisableCalorimeterBias2	0254 0F34 0FC8	12	0	0	6	6	12	0	0.0144 ms	0word, scope-68.png
SendHKAll	0040 0F34 0FC8	196	0	0	10	10	196	0		CAL_BIAS2_EN=0, 92word
SetTempControlMode0 00ff	02E0 0F34 0FC8	12	0	0	11	11	12	0	0.0144 ms	0word, scope_69.png
SendHKAll	0040 0F34 0FC8	196	0	0	12	12	196	0		TEMP_CTRL_SETPT=FF, TEMP_CTRL_MODE=0
SetTempControlMode1 0001	0F34 0FC8	12	0	0	13	13	12	0	0.0144 ms	0word, scope_70.png
SendHKAll	0040 0F34 0FC8	196	0	0	14	14	196	0		TEMP_CTRL_SETPT=01, TEMP_CTRL_MODE=1
SetTempControlMode2 0002	0F34 0FC8	12	0	0	15	15	12	0	0.0144 ms	0word, scope_71.png
SendHKAll	0040 0F34 0FC8	196	0	0	16	16	196	0		TEMP_CTRL_SETPT=02, TEMP_CTRL_MODE=2
SetTempControlMode3 0003	0F34 0FC8	12	0	0	17	17	12	0	0.0144 ms	scop.72.png
SendHKAll	0040 0F34 0FC8	196	0	0	18	18	196	0		TEMP_CTRL_SETPT=03, TEMP_CTRL_MODE=3
SetTempControlMode4 0004	0F34 0FC8	12	0	0	19	19	12	0	0.0144 ms	scope_73.png
SendHKAll	0040 0F34 0FC8	196	0	0	20	20	196	0		TEMP_CTRL_SETPT=04, TEMP_CTRL_MODE=4
SetTempControlMode5 0005	0F34 0FC8	12	0	0	21	21	12	0	0.0144 ms	scope_74.png
SendHKAll	0040 0F34 0FC8	196	0	0	22	22	196	0		TEMP_CTRL_SETPT=05, TEMP_CTRL_MODE=5
SetTempControlMode6 0006	0F34 0FC8	12	0	0	23	23	12	0	$0.0144~\mathrm{ms}$	Scope_75.png
SendHKAll	0040 0F34 0FC8	196	0	0	24	24	196	0		TEMP_CTRL_SETPT=06, TEMP_CTRL_MODE=6
SetTempControlMode7 00ff	0F34 0FC8	12	0	0	25	25	12	0	$0.0144~\mathrm{ms}$	Scope_76.png
SendHKAll	0040 0F34 0FC8	196	0	0	26	26	196	0		TEMP_CTRL_SETPT=FF, TEMP_CTRL_MODE=7
SetTempControlMode0 0000	0F34 0FC8	12	0	0	27	27	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	28	28	196	0		TEMP_CTRL_SETPT=00, TEMP_CTRL_MODE=0
EnableCalorimeterBias1	0254 0F34 0FC8	12	0	0	29	29	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	30	30	196	0		CAL_BIAS1_EN=1, 92word
EnableCalorimeterBias2	0258 0F34 0FC8	12	0	0	31	31	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	32	32	196	0		CAL_BIAS2_EN=1, 92word
EnableColdJFETTVdd1	0264 0F34 0FC8	12	0	0	33	8	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	34	34	196	0		JFET_VDD1_EN=1, 92word
EnableColdJFETTVdd2	0268 0F34 0FC8	12	0	0	35	35	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	36	36	196	0		JFET_VDD2_EN=1, 92word
EnableAC	0270 0F34 0FC8	12	0	0	37	37	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	38	38	196	0		ANTICO_AMP_EN=1
EnableAmprifiers 01ff 01ff	$0288 \ 01FF \ 01FF$	12	0	0	39	39	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	40	40	196	0		CAL_AMP_EN=1 (0-17ch), 92word
SetCalorimeterBias 0000	02C0 0000 0FC8	12	0	0	41	41	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	42	42	196	0		CAL_BIAS_REG=FF, 92word, ClorimeterBias1=46CC, ClorimeterBias2=46C4

表 6.12: 送信した XBox コマンドと digital status の値

の値
status
digital
J
<u>۳</u>
ン
\triangleright
Π
XBox
た
د
送信
6.13:
表

Command name	Command Code	MIOLen	MIOErr	ErrCnt	RcvCnt	SentCnt	XboxLen	XboxErr	RespTime	Comment
EnableCalorimeterBias1	0254 0F34 0FC8	12	0	0	1	1	12	0	0.0144 ms	scope_77.png
SendHKAll	0040 0F34 0FC8	196	0	0	2	2	196	0		
EnableCalorimeterBias2	0258 0F34 0FC8	12	0	0	3	3	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	4	4	196	0		
EnableColdJFETTVdd1	0264 0F34 0FC8	12	0	0	5	5	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	9	9	196	0		
EnableColdJFETTVdd2	0268 0F34 0FC8	12	0	0	7	7	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	×	×	196	0		
EnableAC	0270 0F34 0FC8	12	0	0	6	6	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	10	10	196	0		
EnableAmprifiers 01ff 01ff	0288 01FF 01FF	12	0	0	11	11	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	12	12	196	0		CAL.BIAS1=0x04, CAL_BIAS2=0x04
SetCalorimeterBias 00ff	02C0 00FF 0FC8	12	0	0	13	13	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	14	14	196	0		CAL.BIAS1=0x46C0, CAL_BIAS2=0x46C8, CAL_BIAS_LEN=0xFF
SetCalorimeterBias 007f	02C0 007F 0FC8	12	0	0	15	15	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	16	16	196	0		CAL.BIAS1=0x2348, CAL.BIAS2=0x2334, CAL.BIAS.LEN=0x7F
SetCalorimeterBias 0000	02C0 0000 0FC8	12	0	0	17	17	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	18	18	196	0		CAL_BIAS1=0x000C, CAL_BIAS2=0x0008, CAL_BIAS_LEN=0x00,
										CAL_JFET_VDD1=0x000C, CAL_JFET_VDD2=0x0010
SetColdJFETTVdd 00ff	02C4 00FF 0FC8	12	0	0	19	19	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	20	20	196	0		COLD_JFET_VDD1=0x46B8, COLD_JFET_VDD2=0x46BC, JFET_VDD_REG=0xFF
SetColdJFETTVdd 007f	02C4 007F 0FC8	12	0	0	21	21	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	22	22	196	0		COLD_JFET_VDD1=0x2324, COLD_JFET_VDD2=0x2334, JFET_VDD_REG=0x7F
SetColdJFETTVdd 0000	02C4 0000 0FC8	12	0	0	23	23	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	24	24	196	0		COLD_JFET_VDD1=0x000C, COLD_JFET_VDD2=0x0008, JFET_VDD_REG=0x00
SetColdJFETTVss 00ff	02C8 00FF 0FC8	12	0	0	25	25	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	26	26	196	0		JEFET_VSS_REG=0xFF
SetColdJFETTVss 007f	02C8 007F 0FC8	12	0	0	27	27	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	28	28	196	0		JEFET_VSS_REG=0x7F
SetColdJFETTVss 007f	02C8 0000 0FC8	12	0	0	29	29	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	30	30	196	0		JEFET_VSS_REG=0x00
SetAnticoBias 00ff	02CC 00FF 0FC8	12	0	0	31	31	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	32	32	196	0		ANTICO_BIAS_REG=0xFF
SetAnticoBias 007f	02CC 007F 0FC8	12	0	0	33	33	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	34	34	196	0		ANTICO_BIAS_REG=0x7F
SetAnticoBias 0000	02CC 0000 0FC8	12	0	0	35	35	12	0		
SendHKAll	0040 0F34 0FC8	196	0	0	36	36	196	0		ANTICO_BIAS_REG=0x00
SendHKAll	0040 0F34 0FC8	196	0	0	37	37	196	0		

第7章 PSPシミュレーションプログラムを用いた機 能評価

2010年1月現在、PSP は試験用モデル (BBM)を用いた検証結果から、FPGA 部の仕様を決定し、フラ イト品とほぼ同等なプロトタイプである PSP の性能実証モデル (EM)の製作を開始している。PSP-EM の性能評価として、PSP のデジタル波形処理のアルゴリズムが、実際のカロリメータ検出器の生波形 データに対して正常に機能を果たすかの検証を行うことになる。その際、デジタル波形処理を行った場 合に予想される出力と実際の出力結果との比較が必要となる。本章では、PSP の機能をソフトウェア上 で模擬したシミュレーションプログラムの環境を用意し、PSP の予想される出力に対して評価を行った。

7.1 PSP シミュレーションプログラム

PSP シミュレーションプログラムで行う処理は以下の通りである。

sxspspsim	ANL/FTOOLS
-----------	------------

/	
– pspsimBinRead	:波形データの Binary ファイル読み込み
- pspsimSciMain	: delayBuffer への格納、微分値計算、 WFRB への保存
– pspsimAcPulseNoise	:antico パルス・ノイズの検出(トリガ)
– pspsimPxPulseNoise	:ピクセルパルス・ノイズの検出 (トリガ)
– pspsimPxPuNoRecWrite	: FITS ファイルへの書き出し (節 7.4 で導入)
– pspsimPxPulseCalc	:グレード付け · 波高値解析 (節 7.4 で導入)

なお、時系列の関係上、節 7.3 と節 7.4 で扱う PSP シミュレーションプログラムは version が異なる ため注意する。

7.2 使用データ

PSP シミュレーションプログラム (以下 PSP-sim) で走らせる波形データは、NASA/GSFC で開発を 行っている SXS のカロリメータ検出器の試験用モデル (BBM) から実際に取得した生波形データを元に したものである。データ形式はすでに XBox での処理を終えたバイナリデータとなっており、データ自 体は章 5 で使用した XBox シミュレーター内に保存されているデータと同じ物である。この波形データ の詳細を表 7.1 に記す。このデータは符号無し 16 bit であるため、PSP-sim で扱うためには 2 bit シフ トさせて (4 で割って) 符号付き 14 bit 表記にする必要がある。また、XBox 内でのオフセット電圧を再 現するために 0x4000 を差し引かなければならない。このオフセット電圧はパルスのピークとアンダー シュートが ADC の上限値と下限値内に収まるよう設定されている。つまり、X を元データの ADC サ ンプル値 (16 bit) とすると、PSP-sim に出力すべき ADC サンプル値は

ADC サンプル値 (14bit)
$$\equiv \frac{X - 0x8000 - 0x4000}{4}$$

 $\equiv \frac{X}{4} - 12288$ (7.1)

となり、上記のように変換後 PSP-sim の処理に使用した。しかし、このフォーマットでは上限値が 4095 で頭打ちとなってしまい、4096 ~ 8191 の値を取り得ない。そのため ADC サンプル値は 4095 で clip してしまうことに注意する。

表	7.1:	波形デー	-タの詳細
---	------	------	-------

ファイル名	R73_2009_10_16_01.bin
内容	XBox での A/D 変換後の全 752,347,136 サンプルの ADC サンプル値
	含んでいる特性 X 線:Al, K, Mn の K 線 (NASA/GSFC 調べ)
	カウントレート: 0.6 count/s (cross-talk 含まず)
	30 Hz のハイパスフィルターで処理済み (AC カップリング)
フォーマット	1列のデータ行列
	符号無し 16 bit データ
	ベースライン = $0x7FFF(16$ 進)
	Net order (Big Endian) ¹

¹1バイトごとに分割された数値データを最上位のバイトから順番に記録/送信する方式。

7.3 Science Module 処理後の Science データ

7.3.1 パラメータ設定値

準備した波形データに MIO ボードの Science Module を模擬した処理を行い、その出力データに対し て検証を行った。各種パラメータの設定値は表 7.2 の通りである。なお、Sampling rate は 12.5 kHz を 仮定している。

この波形データは、約 60 ks のデータで 148,030 パルスがトリガされており、cross talk などの非正 規イベントも含めた場合のカウントレートは、およそ 2.5 count/s であった。

7.3.2 トリガ情報のプロット

シミュレーション処理後に出力された、pxPulseEDB に入る値のダンプファイルを用いて、以下の ように各種プロットの検証を行った。

loResPH のスペクトル

トリガされた全てのイベントパルスの loResPH のヒストグ ラムをプロットした (図 7.1)。使用データの情報から波形デー タの中には、Al, K, Mn の特性 X 線を含んでいることが分かっ ている。各輝線のエネルギー (表 7.3.2) と図 7.1 の輝線と見ら れる loResPH の値を比較すると、loResPH 値が 5380 (chan) 付近の輝線が Mn-Kα線であることが推測できる。これを基準

特性 X 線	エネルギー (eV)
Al-K α	1486.6
Al-K β	1557.4
$\text{K-K}\alpha$	3312.9
$\text{K-K}\beta$	3589.6
Mn-K α	5894.2
Mn-K β	6490.4

表 7.3: 特性 X 線のエネルギー

Enspepeini (ereion 2000	12 00	
パラメータ名	設定値	説明
derivFilter	"-16,-16,-16,-16,-16,-16,-16,-16,	"Derivative filter weights"
	16, 16, 16, 16, 16, 16, 16, 16, 16 "	
pxPulseInhibitMask	= 262142	"Pixel pulse inhibit mask"
pxPulseThres	= 200	"Pixel pulse threshold"
pxOffsetAvgGap	= 5	"Pixel offset average gap for loRes"
pxOffsetAvgLenPow	= 4	"Pixel offset average power of two for loRes"
pxFallEndThres	= 0	"Pixel FALL state end threshold"
pxPFInhibitFlag	= 0	"Pixel PEAKFIND state inhibit flag"
pxPFStateCntMax	= 5	"Pixel PEAKFIND state count max"
pxPFQuitCntMax	= 0	"Pixel PEAKFIND state quit count max"
pxQuickDoubleThres	= 1	"Pixel quick double threshold"
pxNoiseInhibitMask	= 262142	"Pixel noise inhibit mask"
pxNoiseThres	= 100	"Pixel noise threshold"
pxNoiseRecordCleanLen	= 2048	"Pixel noise record clean length"

sxspspsim version = 2009-12-08

表 7.2: Science Module 機能に関連するパラメータの設定値

としてスペクトル内に見られる各輝線に対して、簡単に Gaussian でフィットを行った。少なくとも、Al, K, Mn のそれぞれの輝線が確認できているが、その他にも 800 ~ 900 (chan) 付近にも不明なラインが 見られた (図 7.3)。また、Mn-K α のラインフィットより半値全幅 (FWHM) を求めると、 σ を標準偏差 として

FWHM =
$$2\sqrt{\ln 2} \sigma \times \frac{\text{Mn-K}\alpha \, \mathcal{O} \, \Xi \, \lambda \, \mathcal{I} \, \mathcal{I} - [\text{eV}]}{\text{Mn-K}\alpha \, \mathcal{O} \, \text{loResPH} \, \mathbf{i} \, [\text{chan}]}$$
 (7.2)
~ $2.35 \sigma \times \frac{5894}{5382} \sim 18.66 \, [\text{eV}]$

となり、loResPH値の段階で $Mn-K\alpha$ 5.9 keV の入射 X 線に対して約 18 eV のエネルギー分解能が出ていることが見積もれる。



図 7.2: (上) Mn-K α , K β 線 (下) K-K α , K β 線のフィット



図 7.3: (上) Al-Ka, K\beta線 (下) 不明なライン のフィット

図 7.4 は各輝線のエネルギーと loResPH 値の相関を示したものである。理論上、カロリメータのパル スハイトは入射 X 線エネルギーに対して線形の関係をもっている。そこで Al, K, Mn の Ka, Kβ 線の データ点に対して

 $E = aPH + bPH^2 + cPH^3$

という loResPH 値の 3 次関数でフィットし、そこからの誤差を比較した。図 7.4 より、Al-K α 線が -125 eV、Al-K β 線は -40 eV ほどフィット関数からずれていることが分かった。何らかのオフセットが乗っているものと推測されるが、この評価は loResPH 値で行っているのでパルスハイトが正確ではない。そのため最適フィルタ処理を施してから評価する必要がある (節 7.7)。



eV/chan Al-K α 1486.55/1355.21 1.09 $Al-K\beta$ 1557.44/1507.58 1.03 $K-K\alpha$ 3312.89/3222.75 1.02 $K-K\beta$ 3589.60/3469.37 1.03 $Mn-K\alpha$ 5895.04/5382.07 1.09 $Mn-K\beta$ 6490.45/5834.04 1.11

表 7.4: loResPH 値 1 chan 当たりのエネルギー

Time vs loResPH , derivMax

loResPH 値の時系列でのプロットを図 7.5 に示す。Al, K, Mn のラインの他に低エネルギー側でいく つか不明なラインが見えていることが分かる。ここで同様にして微分値のピーク derivMax 値を時系列 でプロットし、loResPH 値の場合との比較をした。loResPH 値と derivMax 値の場合では、Al, K, Mn のラインはほぼ一致しているように見えるが、Al のラインよりも下のエネルギー領域では両者の値の 振るまいが異なっていた。また、derivMax 値で見ると Mn-K α 線はおよそ 24200 (chan) であり、今回 のパルスのトリガをかけるスレッショルドレベル (pxPulseThres) は 200 としているので、低エネル ギー側のしきい値は 50 eV 程度ということになる。



 \boxtimes 7.7: Time vs loResPH (Mn-K α / K-K β , K α)

図 7.8: Time vs loResPH (K-K β , K α / 不明)

 \boxtimes 7.9: Time vs loResPH (loResPH < 600 chan)

loResPH vs derivMax

loResPH 値と derivMax 値の相関を調べた (図 7.10)。loResPH 値に対して derivMax 値が線形増加す る正規イベントの他に、いくつかのブランチや、直線から大きく外れたイベントパルスが見られた。こ れらのイベントを実データから抜き出し、波形の検証を行った。



 \boxtimes 7.10: loResPH vs derivMax

- loResPH = 16000 chan 付近

図 7.10 の loResPH = 16000 chan 周辺に見られるイ ベントパルスの波形を確認したところ図 7.11 のようなパ ルスが得られた。SEQ_NO (シーケンス番号) とは使用 データの頭から数えたサンプル数である。このパルスの 場合、ファーストパルスの 6.56 ms (82 サンプル) 後に セカンドパルスが入射している。PSP-sim が loResPH 値 を計算する際、ADC サンプル値のピーク値からトリガ 直前の計算した loResBase 値を差し引いて求めるが (式 4.7)、直前にファーストパルスがあるためピーク値より も loResBase 値が上回ってしまい、結果として loResPH が負の値になってしまっている。loResPH は 14 bit の符 合無し整数として定義しているので、負の値はアンダー フローして (2^{14} -[元の値]) として表現されていた。実際 の Science Module では負の loResPH 値は 0 とされる仕 様になっている。



図 7.11: SEQ_NO :18571040 (x = 21040)青:derivative 緑:adcSample

- loResPH = 8200 chan 付近

図 7.10 の loResPH = 8200 chan 付近に見られる急激 に satiation してしまっているイベントパルスの波形を 確認したところ図 7.12 のようなパルスが得られた。ADC サンプル値の波形に見られるように上方で波形が clip し ているのが分かる。これは節 7.2 で述べた通り、使用デー タのフォーマットの問題で ADC サンプル値が 4095 以上 を取れないため、この付近でパルスが clip していること になる。実際の PSP では ADC サンプル値が 8192 (2¹⁴ : 14 bit) が上限値のため、loResPH はこれより高い値で clip する。

– Cross-talk イベント

図7.10において、loResPH(0-1000)の範囲で正規イベ ントよりもやや大きい傾きをもって伸びているイベント が存在する。パルス波形が、正規イベントよりも時定数 の短い形状をしているため、隣接する他のピクセルから の cross-talk イベントだと考えられる。このような crosstalk イベントは正規のイベントパルスと波形が異なるた め、PSP でトリガされた場合でも最適フィルタ処理が施 されると波高値が負になったりと正常に処理されない。 cross-talk イベントは複数のピクセルに同時刻に来るも のが多いため、cross-talk イベントの除去は衛星から地上 にデータを降ろした段階で、オフラインで同時刻のイベ ントを除外する方法が考えられている。



図 7.12: SEQ_NO :13749684 (x = 684)青:derivative 緑:adcSample



青:derivative 緑:adcSample

- Double pulse

図 7.10 に見られる、正規イベントの左側から左下に伸 びる 2本のブランチを成すイベントを抜き出したところ、 図 7.14 のようなダブルパルスイベントが確認できた。実 際には、CPU で平均パルスで差し引いてセカンダリーパ ルスの検出を行うため、ダブルパルスのパルスハイトは より正確な波高値が計算される (節 7.7)。

- QuickDouble

この波形データにおける全 148030 イベントの内、 61089 イベント (41%) が quickDouble のフラグが付い ており、高い割合で存在した (図 7.15)。そこで、quick-Double フラグのあるイベントパルスをいくつか確認した ところ、正規の quickDouble イベントの他に図 7.16 のよ



図 7.14: SEQ_NO :258049904 (x = 904)青:derivative 緑:adcSample

うなパルスが含まれていた。ある程度波高値の大きいパルスは AC カップリング¹により、一度マイナ スに沈みこんでから再び波高値が正となり、pxPulseThres 値を越えてしまうため、 1 つのパルスでト リガを 2 回かけてしまっていた。この様なトリガされたパルスのテールではゆっくりとベースレベルに 戻るので、ベースラインに含まれているノイズを引っかけてしまい (derivative[ch] – derivPre[ch] \geq pxQuickDoubleThres[ch])、 quickDouble と判断されていたと考えられる。実際には、CPU でグレー ド付けをする際に、平均パルスを差し引いてセカンダリーパルスの検出を行うため、この様なパルスは 除外できると考えられる (節 7.6)。





青:derivative 緑:adcSample

図 7.15: QuickDouble フラグ付きのイベントパル スのプロット

¹XBox の ADC にはハイパスフィルターを入れることで AC カップリングで信号を読み出している。これは温度安定度な どによって DC レベルが振らつくとパルスのトリガがしにくくなる上、必ずしも DC 成分は必要としないため、AC 成分のみ を取りだしている。ハイパスフィルターの RC 回路は微分回路でもあるため、副作用としてパルスが沈み込んでしまう。また、 XBox にはハイカットフィルター回路も組まれており、サンプリングレートよりも高い周波数をもったノイズをカットしてい る。

7.3.3 PXP_PEAKFIND 範囲の変更

パルスのアンダーシュート後のテールがスレッショルドレベルを超えてしまう非正規の quickDouble イベントを除外するために、パラメータを変更することで再度イベントのトリガ処理を試みた。変更パラ メータは表 7.5 のとおりである。小節 4.7.1 で述べたようにパルスがトリガされてから PXP_PEAKFIND state への移行後、(pxPFStateCnt \geq pxPFStateCntMax) となるまで PXP_PEAKFIND state が 続けられる。しかし、前節では pxPFQuitCntMax = 0 で処理を行っていたため、パルスのピークを検 出後すぐに PXP_READY state へと戻ってしまい、PXP_PEAKFINDstate が機能していなかった。そこ で、微分波形が十分に 0 を下回ってから十分にベースラインへと収束するであろうサンプル数を推測し、 PXP_PEAKFIND state 範囲を広げることでトリガ処理を行った。トリガされたイベントの loResPH 値 と derivMax 値の関係を図 7.17、さらにそこから quickDouble フラグの付いたイベントパルスのみをプ ロットしたものを図 7.18 に示す。

表 7.5: 変更したパラメータ

パラメータ名	設定値
pxPFStateCntMax	$5 \rightarrow 400^1$
pxPFQuitCntMax	$0 \rightarrow 400^1$

¹ 実際の FPGA では 8 bit 表記であるた め 255 までしか設定できない。



図 7.17: loResPH vs derivMax (PEAKFIND 範 囲を広げた場合)

図 7.18: QuickDouble フラグの付いたイベントパ ルス (PEAKFIND 範囲を広げた場合)

前節の図 7.10 と図 7.17 を比較すると、正規イベントから左側に伸びるダブルパルスイベントを含んだブラ ンチは除外されている。図 7.17 の derivMax = 14000 chan 付近に横に伸びるイベントの並びが見られた。 波形データを抜き出して確認したところ図 7.19 のようなダブルパルスが見つかった。PEAKFIND state 中にファーストパルスのピーク値よりも波高値の大きいパルスが入射したため、pxAdcSampleMax 値のみが更新されている。







図 7.20: SEQ_NO :7134285 (x = 1285)青:derivative 緑:adcSample

QuickDouble フラグの付いたイベントパルス (図 7.18)を見てみると、こちらも derivMax = 14000 chan 付近にイベントが見られる。波形を確認すると図 7.20 のような正規の quickDouble イベントであることが分かった。PXP_FALL state 中に次のイベントが入射し、pxderivMax 値は更新されなかったが、pxAdcSampleMax 値は上回っていたため値が更新されている。



青:derivative 緑:adcSample

7.22: 図 7.21 のベースライン拡大 青:derivative 緑:adcSample

また、loResPH 値が0~2000 程にかけて低い derivMax 値のイベントパルスは前節同様、図7.21 に 見られるパルスのテール部分をトリガしているものであった。この様に、cross-talk イベントのような波 高値の低いパルスをトリガした場合、すぐに波形がベースラインに戻るものの PEAKFIND state は400 サンプル経過するまで継続される。その間に波高値の高い正規のイベントが入射し、PEAKFIND state が終了した時点がそのパルスのテール部分であったためトリガがかかってしまったと考えられる。

以上の検証結果から、PXP_PEAKFIND state の範囲を広げることで、ある程度の非正規イベントの 除去は行えることが確認できた。しかし、過度に範囲を広げてしまうと PEAKFIND の最中に別のパル スが入射してしまい、pxAdcSampleMax が新たに来たパルスの値につられて更新されてしまう可能 性が高まる事が分かった。これでは正確にイベントを区別できておらず、PEAKFIND の機能としては 望ましくない。そのため、クロストークやパルスのテールといった非正規イベントの処理はFPGA 側で 行うには限界があり、CPU 側の処理で解決させる必要がある。

7.4 CPU処理のシミュレーション

前節までは FPGA 処理までの波形処理を検証したが、SpaceCard 上 CPU のタスクであるイベント のグレード付け、最適フィルタ処理も PSP-sim で再現する。各種パラメータの設定値を表 7.6 を示す。 PSP-sim の version の関係上、節 7.3 とは boxcar 微分の方法が異なり、パラメータが変更されているが、 実質的には同等の処理を行っている。

•	•	0010 10 00	
eveneneim	version	$-2010_{12}08$	
svebebenn	VCISIOII	- 2010-12-00	

パラメータ名	設定値	説明
sampleRateInHz	= 12500	"sample rate in Hz"
pxPulseInhibitMask	=262142	"Pixel pulse inhibit mask"
pxNoiseInhibitMask	=262142	"Pixel noise inhibit mask"
derivHalfLen00	=8	"Derivative filter half length (0-32)"
pxPulseThres	=50	"Pixel pulse threshold"
pxOffsetAvgGap	$=\!5$	"Pixel offset average gap for loRes"
pxOffsetAvgLenPow	=4	"Pixel offset average power of two for loRes"
pxFallEndThres	=0	"Pixel FALL state end threshold"
pxPFInhibitFlag	=0	"Pixel PEAKFIND state inhibit flag"
pxPFStateCntMax	$=\!5$	"Pixel PEAKFIND state count max"
pxPFQuitCntMax	=0	"Pixel PEAKFIND state quit count max"
pxQuickDoubleThres	=1	"Pixel quick double threshold"
pxNoiseThres	=25	"Pixel noise threshold"
pxNoiseRecordCleanLen	=2048	"Pixel noise record clean length"
preTrigPnts	=150	"Number of samples before pulse trigger"
preTrigPntsH	=150	"Number of samples before pulse trigger for H-res"
preTrigPntsM	=37	"Number of samples before pulse trigger for M-res"
secondThres	=100	"Threshold for 2nd pulse"
secondTrigGapLen	=25	"Gap in samples to start 2nd pulse search"
secondThresUseLen	500	"Length to use 2nd threshold"
avgPulseMax	$=\!5386.65$	"Maximum value of the avgpulse in ADU"
$avgPulse_eV$	=5894.2	"X-ray energy of the avgpulse in eV"
$phaScale_eV$	=0.5	"Energy scale of the calculated PHA in eV/chan"
noiseInterval	=12500	"Minimum length between noise records"
clipSkipLen	=874	"Interval in sample to skip when clipped"

表 7.6: Science Module 機能に関連するパラメータの設定値

7.5 平均パルス、テンプレート作成

セカンダリーパルスのサーチをするための平均パルス、そして最適フィルタ処理にもちいるテンプレート波形を準備する。実際の PSP では、機上でリアルタイムに新しいパルスレコードとノイズレコードか

ら、平均パルスとテンプレートの生成を行われる。しかし今回は、使用データから手動でクリーンな波 形データを数百パルス程度選定し、平均することで平均パルスを生成した(図7.23)。また、PSP-sim に よって出力されたノイズレコードを用いることでノイズスペクトルを生成し(図7.24)、平均パルスと合 わせることで(式4.15)、テンプレート波形を作成した(図7.25)。同様にして Medium-Resolution 用の テンプレートも作成した。

平均パルス用のパルスの選定に際して、使用データの頭 10⁸ サンプルを PSP-sim で走らせてトリガさ れたパルスの内、Mn-K α 線のエネルギーに相当する波高値のものを収集した。図 7.2 の Mn-K α を参考 に、ピークの ADC サンプル値が 5365 ~ 5410 の範囲のイベントパルスを抜き出した。さらに、他の入 射イベントによって形状が歪んでいるものを除外するため、波形の最小値、時間積分値も範囲指定する ことでよりクリーンなパルスのみを選定した。表 7.7 に選定基準をまとめた。なお、PSP-sim 処理後の 出力データの波高値は電圧値に変換されているため、XBox の ADC 上限値 3 V = $2^{14}/2$ を元に電圧値 で波高値の範囲指定を行っている。

表 7.7:	平均パル	ス用イベ	ントパ	ルスの)選定基準
--------	------	------	-----	-----	-------

制限項目	指定範囲
最大値	1.9648 - 1.9813 (V)
最小値	-0.522 - -0.516 (V)
時間積分値	0.95 - 1.05
該当イベント数	339 / 958 パルス





図 7.23: 5.9 keV 相当のイベント (339 パルス) か ら生成した平均パルス: (Record length = 1024)

図 7.24: ノイズレコード (36855 レコード) から生 成したノイズスペクトル: (Record length = 1024)

7.6 グレード付け

前節で作成した平均パルスを用いてセカンダリーパルスのサーチを行い、各イベントパルスにグレー ド付けを行った。グレード付けは小節 4.10.2 の定義に従った。出力結果から得られた Hp, Mp, Ms, Lp, Ls それぞれのカウント数を表 7.8 にまとめた。



図 7.25: 平均パルスとノイズスペクトルから生成したテンプレートの波形: (Template length = 1024)

グレード	カウント数	カウントレート	割合
Hp	44704	0.74 count/s	30.6%
Mp	3119	0.05 count/s	4.3%
Ms	3237	0.05 count/s	
Lp	30282	0.50 count/s	65.0%
Ls	64766	1.07 count/s	

表 7.8: グレード毎のカウント数

7.6.1 loResPH vs derivMax

次に、トリガされたイベントの loResPH 値と derivMax 値の相関をグレード毎に調べた (図 7.26, 7.27, 7.28)。最適フィルタ処理を行う前の図 7.10 に比べ、正規イベント以外のブランチなどが消え、loResPH 値に対してリニアに derivMax 値が上がっているのが分かる。このことから平均パルスの差し引きによるセカンダリーパルスのサーチによって行ったグレード付けの処理が非正規イベントに対して有効であることが明らかとなった。しかし、それと同時に、ランダムに生じるとされる cross-talk イベントも Hp, Mp, Ms では見受けられなくなり、全て LR に含まれていた。これは使用したデータの cross-talk イベントが実際にはランダムに到来しておらず、何らかの因果関係により正規イベントに近接した時刻、つまり LR の範囲に到来していると考えられる。これにより、LR イベントの割合が高くなった可能性がある。



図 7.26: loResPH vs derivMax (Hp イベント)

図 7.27: loResPH vs derivMax (Mp イベント)

図 7.28: loResPH vs derivMax (Ms イベント)

7.7 最適フィルタによる波高値計算

作成したテンプレート波形を用いて、上記の Hp, Mp, Ms に対して最適フィルタ処理 (小節 4.10.3) を 行い、各イベントパルスのの波高値 (PHA) を求めた。以下に各グレード毎の PHA, loResPH, derivMax のスペクトルを示す。図 7.38, 7.39, 7.40 は各グレードの Mn-K α 線を PHA, loResPH, derivMax で同時 プロットしたものである。Mp, Ms の 2 つはカウントレートが低いため統計が悪いが、Hp で見てみると PHA, loResPH, derivMax の順にエネルギー分解能が良いことが分かる。PHA に関しては、軌道角運動 量の違いによる Mn-K α 1, Mn-K α 2 が分離できていることが見てとれる。



sxstbools/, abe/20101208/sample1_7e8-master/ishisaki/piot-pulsecalc_Hp.com



図 7.29: Hp の PHA のスペクト ル

図 7.30: Hp の loResPH のスペ クトル

図 7.31: Hp の derivMax のスペ クトル



図 7.32: Mp の PHA のスペク トル



図 7.35: Ms の PHA のスペクト ル



図 7.38: Mn-KaのHpイベント (黒: PHA, 赤: loResPH, 青: derivMax)







図 7.34: Mp の derivMax のスペ クトル



図 7.36: Ms の loResPH のスペ クトル



図 7.37: Ms の derivMax のスペ クトル





図 7.39: Mn-KaのMpイベント (黒: PHA,赤: loResPH,青: derivMax)

図 7.40: Mn-Ka の Ms イベント (黒: PHA, 赤: loResPH, 青: derivMax)
7.8 エネルギー分解能計算

前節で得られた最適フィルタ処理後の波高値 PHA の値からエネルギー分解能の計算を行った。

7.8.1 ベースライン分解能

マイクロカロリメータのエネルギー分解能は、ベースライン分解能によって制限される。ベースライン分解能とは、X線入射パルスが無い時のノイズによるベースラインの揺らぎをエネルギーに換算したものである。ここでのノイズは、カロリメータの半導体温度計の動作点での抵抗Rによって決まる取り除くことのできないカロリメータの固有ノイズや、信号読み出し装置のJFETから来る読み出しノイズなどによって決まる。一方、X線入射時のノイズは、X線パルスの入射によって温度計の抵抗Rや温度計感度 αが動作点とは異なるため、Rやαに大きく依存する固有ノイズが変化し、S/N比が異なる値となる。また、カロリメータ素子へのX線入射位置の依存性によって波高値のばらつきが生じるため、実際のエネルギー分解能はベースライン分解能より劣化する場合が多い。

今回は、PSP-sim のピクセルノイズトリガ (PXN) 機能で収集したノイズレコードに対して、前節の最 適フィルタ処理で適用させたものと同じ最適フィルタテンプレートを適用することでベースラインの揺 らぎを調べた。なお、エネルギーの変換は、PHA のスペクトルの Mn-Ka 線のピークの PHA を 5894.2 eV として算出している。図 7.41 がベースラインのヒストグラムの結果である。図中の赤線はこのヒス トグラムに 1 つのガウシアンをフィットさせたものである。このフィットから、ベースラインのエネル ギー分解能への寄与を求めると、FWHM で 3.94 eV であった。



7.8.2 エネルギー分解能 (at Mn-Ka: 5.9 keV)

次に、X線パルスが入射時のエネルギー分解能を調べる。エネルギー分解能は、使用データ内に含ま れている Mn-K α のラインに対してフィットを行って導出する。Mn-K α 線はさらに、軌道角運動量の違 いによって Mn-K α 1 (5.89875 keV), K α 2 (5.88765 keV) の 2 つのラインをもっている。この 2 つのライ ンは非常に接近したエネルギー範囲に現れるため、優れたエネルギー分解能がなければ分離することが できない。これらのラインは、K α 1 が 2 本、K α 2 が 5 本の計 7 本の自然幅をもつ Lorentzian の重ね 合わせで表すことができる。PHA から得られた各グレードのエネルギースペクトルに対してフィットを 行ったものが図 7.42 である。青線が個々の Lorentzian の寄与を示しており、自然幅をもつ Lorentzian を Gaussian でコンボリューションしてフィットしたのが赤線である。Mp, Ms に関しては統計が悪く フィットが上手くいっていないため Mp よりも Ms の方が良い分解能が出ているが、Hp イベントで見た

ところ、エネルギー分解能は FWHM で 4.8 eV と 2 つのラインがしっかりと分離できていることが確認 できる。同時に、FWHM を K α のもので固定した Mn-K β のエネルギースペクトルのフィットを図 7.43 に示す。こちらも Mp, Ms ともにカウント数が少なく、フィットができていない。



図 7.42: 右から Hp, Mp, Ms のエネルギースペクトル (Mn-K α 1, K α 2)



図 7.43: 右から Hp, Mp, Ms のエネルギースペクトル (Mn-K_β)

7.8.3 リニアリティー補正

カロリメータは理想的には入射 X 線のエネルギーによらないとされているが、実際には入射するエネ ルギーが高くなりパルスハイトが大きくなると、ゲインのばらつきの影響が大きくなり、エネルギーの 低い方が分解能は良くなるという傾向がある。そこで、PHA とエネルギーの関係を $Mn-K\alpha$, $K\beta$ の 2 つ のピークを用いて、この 2 点を通るような 2 次関数

$$PHA = aE + bE^2 \tag{7.3}$$

でフィットし、これを補正関数とすることで各イベントパルスの PHA を補正してエネルギーに変換す る。ここでの、*a*, *b* がフィットパラメータである。図 7.44 に各グレード毎のフィット結果を示す。それ ぞれの縦軸 PHA はすでに Mn-Kα のラインのピーク値が 5.89875 keV になるよう規格化されている。



図 7.44: 右から Hp, Mp, Ms のリニアリティー補正関数 (Mn-K α , K β)

これにより得られたフィットパラメータから、PHA をエネルギーに変換することで、リニアリティー 補正後のエネルギー分解能を調べた。結果を図 7.45 にまとめる。補正前に比べて Hp, Mp, Ms 共にエネ ルギー分解能が劣化しているのが分かる。しかし、依然として 6 eV 前後と 2 つのラインをある程度分 離できる高い分解能を実現できている。また、Hp のフィットパラメータを用いてベースライン分解能を 再度導出してみたところ、こちらは分解能が改善していた (図 7.47)。これは、補正関数の低エネルギー 側での傾きが大きくなっているため、同じノイズの振幅に対してエネルギーに換算した場合の自然幅が 短くなったためである。



図 7.45: 右から Hp, Mp, Ms のリニアリティー補正後のエネルギースペクトル (Mn-Ka1, Ka2)



図 7.46: 右から Hp, Mp, Ms のリニアリティー補正後のエネルギースペクトル (Mn-K)



図 7.47: リニアリティー補正後のベースライン分解能

7.8.4 考察

以上の結果から、PSP による波形処理アルゴリズム (イベント抽出、グレード付け、波高値解析) は、 SXS の要求性能 (< 7 eV) を満たすエネルギー分解能を達成できることが立証できた。本検証では、PSP を模擬したシミュレーションプログラムで行ったため、検出器も含めた実機での評価試験で実際に線源 からの X 線を捉えることで同程度のエネルギー分解能が得られることを期待する。

第8章 まとめと今後

現在、2014年打ち上げを目指し、日米欧の国際協力でX線天文衛星「ASTOR-H」の開発が進められ ている。衛星には軟X線分光システムSXS(Soft X-ray Spectrometer)が搭載され、0.3 ~ 12 keV のエ ネルギー領域を7eV以下という超高精度のエネルギー分解能を実現することができる。検出器には半導 体温度計を用いたX線マイクロカロリメータと呼ばれる装置が使われる。入射X線光子のエネルギー を素子の微小な温度変化で捕らえ、極低温下(50mK)で動作させることで優れたエネルギー分解能を得 ることができる。カロリメータで検出されたX線パルス信号は米開発担当のXBox(X-ray Box)と呼 ばれる装置へと出力される。ここでは、信号のフィルタリング、増幅、A/D変換などが行われ、ノイズ の少ないLVDS(Low Voltage Differential Signaling)で差動出力で日本開発担当のPSP(Pulse Shape Processor)と呼ばれる装置へと送られる。PSPとは、SXSのデジタル波形処理システムのことであり、 私はこのPSPの開発チームに所属している。PSPは首都大、埼玉大、ISAS/JAXAが設計・開発を担当 し、三菱重工業(MHI)が開発・製造を行っている。

PSP は主に FPGA を搭載した Mission I/O (MIO)ボードと、CPU を搭載した SpaceCard ボードと呼 ばれる 2 つの共通ボードで構成されている。XBox から送信されてきた検出器のデジタル波形データは まず MIO ボードに送られ、微分波形の計算とそれを用いたパルスのトリガ(抽出)を行う。トリガ情報 と波形データは、内部メモリや SDRAM といった記憶領域に保存される。CPU はその保存された波形 データからイベントパルスを呼び出し、さらに詳細なパルスのトリガを行う。そして、それぞれのイベ ントパルスの前後のイベントパルスとの時間間隔に応じて全てのイベントにたいしてグレード付けを行 う。グレード付けをされた各イベントは最適フィルタと呼ばれる処理が施されることでより正確な波高 値が求められる。

我々は PSP の試験用モデル (BBM: Bread Board Model) を用いて性能評価試験を行い、その結果を 元に MIO の FPGA ロジックの仕様の決定を完了した。現在、フライト品と同等の性能を持つ性能実証 モデル (EM: Engineering Model) の製作を進めている段階である。

今回私は、試験用モデルを用いた性能評価試験を行い、PSPの各モジュールの機能が正常に動作して いるかを検証した。XBoxと同等の出力をする XBox Digital Simulator (XDS)を用いて、PSP に実際 の波形データを出力し、XDS – MIO 間の電気インタフェースが問題ないことを確認した。また、MIO 内の波形処理のモジュールが正常に機能していることを確認し、各バッファに波形データやトリガ情報 などが正常に取得されていることを確かめた。

また、2010 年 8 月に XBox を開発している NASA/GSFC を訪問し、実際に XBox-BBM との噛み合わせ試験を行った。こちらでも、XBox-BBM – PSP-BBM 間の電気インタフェースに問題がないことを実証した。さらに、検出器を模擬したダミーパルスを送出する Detector-Simulator を接続し、XBox を介して PSP で正常にデジタル波形データの取得、保存が行われていることを確かめた。

並行して、CPUの処理速度測定と、MIO – SpaceCard 間のデータ転送速度を測定を実施した。CPUの処理速度測定では、CPUを100%使い、典型的な平均パルス波形を連続的に処理を行わせた。その結果、PSP に要求される150 cts/s/arrayを十分に満たしていることが判明した。

MIO – SpaceCard 間のデータ転送速度測定では、計測結果から、要求される転送速度を満たす最小の ボード構成は PSP 1 系統当たり 1 MIO + 2 SpaceCard ボードの 3 ボード構成であることが分かった。 但し、CPU 負荷が大きく、空き時間が 40%程度しか確保できていなかったが、CPU の処理速度の結果 から、この程度の空き時間でも許容範囲は超えないことが分かった。転送速度に関しては、次期モデル

で改善されることが見積もられている。

また、3ボード構成で開発を進めるためには MIO ボードで行う処理を1つの FPGA に収めなくてはな らず、FPGA リソースを確保する必要が生じた。FPGA ロジック最適化のため、冗長パラメータの削除 や微分計算方法の変更が提案された。特に微分計算は最もロジックを使用する処理の1つであったが、 PSP に要求される性能に影響を与えない方法をとる必要があった。私は、PSP と同じアルゴリズムで波 形データの微分計算の際の boxcar derivative 関数のデータ長を変えて計算を行い、その各データ長で計 算した際のノイズの RMS を調べることで、PSP の要求性能を満たせる最適なデータ長を調べた。その 結果、±32 の長さを取れば、十分に性能を満たせることが判明し、変更後の微分値計算方法に問題がな いことを確かめた。これにより、EM 品ではこの3ボード構成で開発を進めることが決定している。

また、これらとは別に、ソフトウェア上で PSP の機能を模擬した PSP シミュレーションプログラム を用いて PSP の波形処理のアルゴリズムが実際の検出器のデータに対して正常に機能するかを検証し た。その結果、FPGA の微分波形をもちいたトリガ機能と、CPU によるセカンダリーパルスのサーチ 処理によって、目的の正規イベントパルスを正常に抽出できていることが分かった。また、トリガされ たパルスデータから各グレードに対してエネルギー分解能を算出したところ、期待されるエネルギー分 解能が得られた。これにより、PSP のアルゴリズムの妥当性が証明された。

今後、EM品の完成しだい、実際にNASAの開発しているXBoxと実際のカロリメータ検出器とを繋 いだ end-to-endの試験を実施予定であり、その試験結果に基づいて実際の衛星に搭載されるフライト品 (FM)の設計を確立させていく予定である。

付録A XDSデータ書き換え手順

XDS 内のデータの書き換えは、FPGA の Rom を一時的に書き込み専用にして、データファイルを XBox に流し込む必要がある。

データ書き換え手順は以下のとおり行う。

- 1. sof ファイル (FlashRom の write 専用ファイル) を Qualtus で読み込む (今回は GSFC の XBox 専用 sof ファイル: Flash_wr_sw6_blks0_7.sof を使用)
- 2. FPGA に sof ファイルを読み込む場合は図 A.1 の様に USB Blaster (ALTERA 製) を左側の端子 に接続
- 3. FlashRom に書き込むデータの入った PC と mini USB で接続 (図 A.1)



図 A.1: PC - XDS の接続内部

dit Operate Tools	Browse Window Help		
1	12 length	2 If File stream	
vin Ihandie t	o Device Gror Stream So	urce Choose File Type	
	Create	Ð	Binary (16bit)
KUSB-0 Pound	Quickose moode		
Data	 Stream from File university of the separated her values 	from 19 columns, or if Binary file-#	of 16bit words
per file read # of	16 bit words should be divisible by 19		
		Eat number of 16 bit words t	n rand per file read
4980736 # of 16 b	bit words to send per file read s	hould be divisible by 19	o sena per nie reau
3		Data File Path/name:	
Create Data Type	Data Stream Created by this	Program	
Random values	✓ 2a.Choo	ise Type of Data Stream to Cre	ate
\$1995 # of 16	5 bit words to send per Cycle		
# 1995 # of 16 defau	6 bit words to send per Cycle ited to multiple of 19		
# 1995 # of 10 defau	6 bit words to send per Cycle alted to multiple of 19 yted Data stream continuously 2b. Choose wheth	her to send one block or contir	uous stream
# 1995 # of 16 defau Send creat	6 bit words to send per Cycle afted to multiple of 19 ited Data stream continuously 2b. Choose whetl	her to send one block or contir	uous stream
# 1995 # of 10 defau	6 bit words to send per Cycle afted to multiple of 19 afted Data stream continuously 2b. Choose wheth	her to send one block or contir	uous stream
# 1995 # of 16 defau Send creat	6 bit words to send per Cycle afted to multiple of 15 afted Date stream continuously 2b. Choose wheth Date Stream	her to send one block or contin	uous stream
Elements in Queue	6 bit words to send per Cycle and beto multiple of 19 and Data stream continuously 2b. Choose wheth Data Stream Stopped	her to send one block or contin Data packets send to usb port 32000 –	uous stream
1995 # of 1e defau Send creat Elements in Queue 10000 - D Booo - D Cueue Si	bit words to send per Cycle atted to mulpike of the second second second second second Data Stream Stoppel 3. Press Button to begin	her to send one block or contin Data packets send to usb port 32000 - 9 0 corport	uous stream
Elements in Queue Si Queue Si Dono - D Queue Si	bit work to send per Cycle and the markies of the second second second second med Data Stream Stream 3. Press Button to begin	her to send one block or contin Data packets send to usb port 30000 - 	wous stream
# 1995 # of 16 defau Image: Send creating the send creating th	bit words to send per Cycle and Dina stream continuous) 2b. Choose wheth Dina Stream Torcent 3. Press Button to begin	her to send one block or contin Data packets send to usb port 30000- \$27900 - \$25000 -	uous stream
# 1995 # of 1 a defau defau Send cres Send cres Biements in Queue Queue Si 8000 - 0 Queue Si 00000 \$10000 - 0 \$4000 - 0 \$4000 - 100000 \$4000 - \$410000	bit works to send par Cycle and Data stream continuously 2b. Choose wheth Data Stream Trigopoli as	her to send one block or contin Data packets send to usb port 30000 - 927500 - 925000 - 22500 -	wous stream
1995 # of 11 defad defad 0 Send creat 0000- D 0000- D 0000- 0 0000- 0 0000- 10000 0000- 4.11 Queue St 0000- 4.11 Queue St	b & words to send par Quée de la huste a 't and Duéa tream continuculy' 2b. Choose wheth D 2 Stream 3 Press Button to begin are its filling that is being created than the outh part is sending a.	her to send one block or contin Data packets send to usb port 30000 - \$27500 - \$25500 - 22500 - 20000 -	uuuus stream
1995 + dilideral defaulteral defaulter	bit works to send per Cycle and Data stream continuously 2b. Choose wheth Data Stream 	her to send one block or contin Data packets send to usb port 3000 - \$2000 - \$2500 - \$2500 - \$2500 - 2000 - 17500 - 17500 -	uous stream
1998 4 of 1/2 drau Send ores Send ores 0000- 0 0000- 0 0000- 0 0000- 0 0000- 0 0000- 0 0000- 0 000- 0 000- 0 000- 0	b & work to send per Què adde handles d'a transmostre Med Duès dream continuously 2b. Choose wheth Dubles 3000000 30.Press Munten to begin are les filling then data is being created than the usb port is sending it. Betherem dua to settor or fin read Quès (ndice) Duble to write (Datam)	Bet to send one black or contin Data packets send to usb port 30000 42 20000 42 20000 10000 10000	uous stream
1998 4 of 14 Image: Send one Send one Image: Send one Image: Send one Image: Se	bå vedta streken corkinucali 2 the Chonse whether the number of the constraints of t	her to send one block or contin Data packets send to uib port 30000- 42 27500- 42 2000- 17500- 17500- 15500- 15500- 15500-	uous stream
1995 e of 17 1995 e of 17 1995 Send ores 2000- 0 1000- 0	be a work to send per Oyde de houses of the output of the mod Diva tream continuous) 2h. Choose wheth mod Diva tream continuous arrows arror arrows arror arr	her to send one black or costin Data parkets send to usb part 30000- 4 27000- 4 27000- 20000- 15500- 15	uous stream
1995 • of 11 disa disa disa send creat Bements in Queue Queue Si 8000 - Queue Si 0 4.000- 0 - 0 - 0 - 0 - 0 - 0 - 0 - 0 - 0 - 0 -	b & works to and per Cycle and Data stream continuous) 2b. Choose wheth Data Stream 	her to send one block or contin Data packets send to usb port 30000- 42 27500- 42 2500- 20000- 17500- 1550-	uous stream
	be a work to send per Oyde add bona date ais add bona	her to send one block or contin Data parkets send to usb port 30000- 4 27000- 20000- 1500	uous stream
	b & works to and per Cycle deformation in the second second second second Data Stream Base 3. Press Button to begin mere is filling then data is less proceed that the way per it is working and cycles (ndec) Data to are (Stream) Base (Stream)	And the send one block or control Data packets send to usb port 30000- 4 27500- 4 25000- 15500-	uous stream
(1995 • of it it of the second s	be a work to send per Oyde add to handbe i d' add to ha	her to send one block or contin Data parters send to usb port 30000- 4 25000- 2000- 15500	uuus stream
1995 4 of all of a	b & vada to and par Cycle de to matter al to the Data stream continuous) 2b. Choose whether Data Stream are 5 Million to begin are 5 Million Stream to begin the Million Stream to begin	her to send one block or control Data packets and to usb port 30000- 42 27500- 42 25500- 15500-	uous stream
1995 4 of a of	b & vadio sandpar Què ade l'autorità de la sub benga arres la filiada de la sub benga cented arres la filiada de la sub benga cented arres es filiang then data sub benga cented tan the useb port sandparts arres de la coles (object) la bittere data coles arres de la coles (object) la bittere data coles (object) la bittere data coles (object) la bittere data coles (object) la bittere data coles (object) la coles d	her to send one black or contin Data parties send to usb port 3000- 4 2500- 2000- 15500- 15500- 15500- 15500- 15500- 25500-	uous stream

図 A.2: LabView 操作画面

4. XDS 電源 ON

- 5. LabView ver.7.1 (National instruments 製) を起動 (図 A.2)
- 6. Run をクリック data stream で file を選択 type of file で binary (16bit) を選択
- 7. Data stream from file に word 数 を指定する (4980736 words) 備考参照
- 8. data file path/name で書き込むデータファイルを選択 (1 block 分) → sub-simshift19-0.bin を選択

9. *-0.bin を書き込む時は、XDS のダイヤルを switch1 にする (図 A.3)



*-1.bin の時は switch2, *-2.bin の時は switch3, ... と繰り返す (switch8 まで存在)

 \boxtimes A.3: XDS \mathcal{O} switch

- 10. stopped をクリック → ウィンドウに波形が走り始める
- 11. 波形が停止し、書き込みが終了したら stop をクリック
- 12. さらに続けて file を書き込むには、LabView を再起動 (ウィンドウを閉じる)
- 13. 以降は 8 switch (32 block) 分、8. ~ 12. を繰り返し

(備考)

- XDS のデータの書き換えは、1dataset 当たり 32 block 書き込み可能。
 1 block =9961472 byte (= 0x980000 byte)= 4980736 words
 1 block= 262144 sample ~ 21sec なので、32 block で約 11 分の Science data になる。
- 1 つの sof ファイルで 8 block 書き込み可能。
 → 1 データセットに対して 4 つの sof ファイルが必要
- QuickUSB.dll (bitwise 社製モジュール)を用いるため、1度に転送できるサイズは16MBに制限されている。
- XDS に取り付けられたダイヤルは、読み取り時はデータセットの切り替え (8 channel) 用、書き込み時は1つの sof ファイルに対する block 切り替え (8 block) 用と役割が異なるため要注意。

FlashRomはNand type, Nor typeの2タイプあり、XDSは前者を使用している。Nand typeのFlashRomは工場出荷時点でBad blockと呼ばれる書き込みできないエリアが存在するため、sofファイルにBad blockの位置を認識させる必要がある。そのため、個々のXDSに対応したsofファイルしか使うことができない。

(使用した LabView ファイル)

Astro_H_Data_for_19ch_binary_file_v6.llb

→ 解凍後、起動ファイル (Astro_H_Data_for_19ch_binary_file_v6.vi) を生成



図 B.1: pxWFRB[0]-[2]の波形 (Sampling rate: 12.5 kHz)



図 B.2: pxWFRB[3]-[5]の波形 (Sampling rate: 12.5 kHz)



図 B.3: pxWFRB[6]-[8] の波形 (Sampling rate: 12.5 kHz)



図 B.4: pxWFRB[0]-[2]の波形 (Sampling rate: 13.9 kHz)



図 B.5: pxWFRB[3]-[5] の波形 (Sampling rate: 13.9 kHz)



図 B.6: pxWFRB[6]-[8]の波形 (Sampling rate: 13.9 kHz)



図 B.7: pxWFRB[0]-[2]の波形 (Sampling rate: 12.5 kHz)



図 B.8: pxWFRB[3]-[5]の波形 (Sampling rate: 12.5 kHz)



図 B.9: pxWFRB[6]-[8]の波形 (Sampling rate: 12.5 kHz)

謝辞

本修士論文を進めるにあたり、たくさんの方々のお力添えを頂きました。この場をお借りして、感謝 の気持ちを送らせて頂きたいと思います。

私がこの研究室に入ったのは学部を含めると、もう3年前のことになります。私の研究生活は破門からのスタートでした。卒論も書かずに遊び呆けていた私を、破門だとおっしゃいつつも (bad joke だと思いたいですが...)研究室のメンバーとして受け入れて下さった寛大な大橋先生には大変感謝致しております。お忙しいにも関わらず、日頃から学生部屋に顔を出して下さっては私たち学生のくだらない雑談にお付き合い下さったり、時には学生が立ち直れなくなる一歩手前の毒をお吐きになることもありましたが、これも大橋先生の愛情と受け止め、なんとかここまでやってくることができました。天文ゼミや論文紹介、学会の発表練習の際でも、知識の至らない私にも理解しやすいよう分かりやすく噛み砕いてご教授下さり、自分自身大変勉強させて頂きました。本当にアホな学生ですみませんでした。

指導教官である石崎さんには言葉では表しきれない程の感謝をしております。TESの人員が不足して いるにも関わらず、PSPの実験にお誘い頂いた時のことは今も忘れません。石崎さんは覚えていらっしゃ らないかも知れませんが、ある飲み会の席で「自分のやりたいことをやるのが一番だ」とおっしゃって 下さったことが自分にとっては大変嬉しかったです。ログの取り方に始まり、どんな問題に対しても諦 めることなくあらゆる視点からアプローチすることで解を導き出す姿勢など、沢山のことを学ばせて頂 きました。また、GSFCでのPSP 噛み合わせ実験への参加という大変貴重な経験もさせて頂きまして、 本修論にも掲載することができました。少しでも石崎さんのお力になれたかどうか分かりませんが、石 崎さんと実験できたことを大変光栄に思います。3年間本当にお世話になりました。奥さまとわかばちゃ んにもよろしくお伝え下さい。

江副さんにも大変お世話になりました。聴衆に分かりやすい解説の仕方や資料の作り方、実験プロセスの組み方など、江副さんの仕事に対する姿勢にはいつも学ばせて頂いてばかりでした。低能な雑談にも楽しそうに参加して頂きとても感謝しております。ただ、Apple 純正の充電池を購入されていたのには驚いてしまいました。あれは eneloop と同等品ですよね。。

河原さんとは同じ部屋だったこともあり、色々とご迷惑をお掛けしました。研究の合間(?)の世間話は とても楽しく、あらゆる話題に対して批判的に返してくる河原さんにはいつも感服しておりました。石 津との一生噛み合うことのない会話がツボでした。たまーに、頭の悪い私に宇宙論を噛み砕いてご教授 頂きましてありがとうございました。

私の研究生活は、数々の先輩方や後輩達の支え無しではここまでやってくることができませんでした。 赤松さんには大変お世話になり、ご迷惑をおかけてしまいました。生意気で至らない私を見捨てること なく、丁寧に実験を教えて頂き、いつも進捗を気にかけて下さったりと本当に感謝しています。ありが とうございました。林さんとは実験で一緒になることはありませんでしたが、飲み会などでそれはそれ はお世話になりました。ときには横田さんから守って頂き、ときには酔っぱらってまじめに語らったり、 無礼にも先輩をイジリたおしたりと、アホな学生を構ってくださいましてありがとうございました。お 二人共 D 論大変だとは存じますが、頑張って下さい! かげながら応援しております。

横田さん、石川さん、佐藤さんは歳が近い先輩として大変構って頂きました。特に横田さんとは一つ の布団で寝た仲(?)でとても笑わせてもらいました。横田さんのマシンガントークは尊敬に値します。ホ ントに楽しかったです。石川さんは作業中でも嫌な顔一つせず、くだらない話に厳しい突っ込みを入れ たり、私のきれいなお尻を写真に収めてくださったりと、私たちの姉御として君臨なさっていました。佐 藤さんとは研究室でもあまりおしゃべりする機会は少なかったように思いますが、たまに私が調子づい ている時に、聞こえるか聞こえないかぐらいの声量で厳しい一言を下さり、我に返ることができること がありました。本当に先輩方のおかげで楽しい時間を過ごすことができました。ありがとうございます。

同期である、石津、塩野目、辺見ちゃんは、同じ試練を乗り越えた仲間として感慨深いものがありま す。本当に良い意味でも悪い意味でも居心地の良いゆったりとした仲間でした。修論書き奮闘中に敵将 を討ち取ったのは良い思い出です。辺見ちゃんにはいつも驚かされてばかりでした。自ら谷間世代とか なんとか言っていましたが、本当にこのメンバーが同期で良かったなと思っています。みんなありがと う!また飲みにでも行きましょう。

それから後輩のみんなには先輩にも関わらず、本当にお世話になりました。特に大石さんには沢山の 迷惑をお掛けしたと思います。TES 要員が不足しているにも関わらず、PSP をやりたいと言った私を快 く応援してくれたことには本当に感謝しています。至らない先輩でしたが、皆さん優秀な後輩達だと心 から思っているので私を反面教師にぜひ今後も活躍していってほしいと思います。

そして最後に PSP の研究を行うに当たって、埼玉大の方々には多大なる感謝の気持ちを述べたいで す。突然、どこの馬の骨とも分からぬ学生がひょこっと PSP チームに参加したにも関わらず、チームの メンバーとして快く受け入れて下さいました田代先生や辻本さんには大変感謝しております。瀬田さん は役立たずな私にいろいろ教えて下さったり話しかけて下さったりと、メンバーとして扱って下さった ことが大変嬉しかったです。本当にありがとうございました。そして同期として一緒に戦った下田君。 僕も君と一緒に実験できて本当に良かったです。チームに同期が増えることで自分の役割が少なからず とも分けられて関わらず、嫌な顔一つせず一緒に楽しく議論しながら実験できたことはとても楽しく有 意義な時間でした。「同期にあべちゃんがいてくれて良かった」と言ってくれたときは、照れくさいです がとても嬉しかったです。これから D 論に向けてぜひ僕の分まで活躍していって下さい!

長々と述べさせて頂きましたが、本当に僕はこの研究室の一員であったことを誇りに思います。皆様、 短い間でしたが本当にありがとうございました。