

データ収集用 FPGA ファームウェア (Kalliope-DC-RCNP 版) 機能仕様書



2019年2月19日 1.1.3版
(株)Bee Beans Technologies

改版履歴

版数	日付	VER ^(注1) (注2)		内容	確認 担当
		NIM			
1.0	2016年11月8日	NIM	16110901	初版制定	— 石綿
		2012	16110902		
		2014	16110803		
1.0.1	2016年11月14日	NIM	16111401	Copper Header を抑止した場合にバッファフルによりイベントが止まるバグの修正 本文修正なし	— 石綿
		2012	16111402		
		2014	16111403		
1.1.1	2018年12月21日	NIM	16111401	確認者欄追加 データ廃棄アルゴリズムにおける誤記修正 修正対象 Volume2014 のみ ASIC のシーケンス変更 ASIC_POL レジスタ (0x0E8) 追加 DAC_CLBn を DAC_STRI にピン名変更 DAC_CLBn を別ピンで再定義	岩瀬 石綿
		2012	16111402		
		2014	18121303		
1.1.2	2019年2月18日	NIM	16111401	修正対象 Volume2014 のみ ピン名称・配置は 1.0.1 版以前に戻す ASIC のシーケンス変更 ASIC_POL レジスタ (0x0E8) 変更 図 4-2 を追加	岩瀬 石綿
		2012	16111402		
		2014	19021803		
1.1.3	2019年2月18日	NIM	16111401	修正対象 Volume2014 のみ DAC_CLBn を DAC_CLBn[0] にピン名変更 DAC_DIN[0] を DAC_CLBn[1] にピン名変更 DAC_CLBn[1] の機能追加 図 4-1 の修正 表 4-4 の注記追加、 表 4-4 の DAC_DIN[1] を DAC_DIN に修正 図 4-2 の DAC_DIN[1] を DAC_DIN に修正	岩瀬 石綿
		2012	16111402		
		2014	19021903		

(注1) RBCP レジスタの 0x000~0x003 の値です。

(注2) 改版に伴う FPGA の変更がない場合は、網掛けで表示しています。

【目次】

1. 適用	1
2. 構成	1
2.1. HIT_ENC.....	1
2.2. HIT_DELAY	2
2.3. EVT_TOP_SER	2
2.4. EVT_2ND_SER	2
2.5. EVT_3RD_SER	3
2.6. EVT_4TH_SER	3
2.7. EVT_5TH_SER	3
2.8. EVT_6TH_SER	4
3. TCP データ	5
3.1. 5G イベント	6
3.2. Key Word.....	6
3.3. Trigger Count.....	7
3.4. 01 イベント	7
3.5. 02 イベント	8
3.6. 03,04 イベント.....	8
3.7. Trailer.....	9
4. RBCP メモリマップ	10
4.1. VER レジスタ (0x000~0x003)	11
4.2. FPGA_ID レジスタ (0x004~0x007)	11
4.3. EVENT_NUM レジスタ (0x008~0x00B)	11
4.4. FPGA_CTRL レジスタ (0x00C)	11
4.5. KEY_WORD レジスタ (0x00D~0x00F)	12
4.6. DELAY レジスタ (0x010~0x013)	12
4.7. GATENET_TIME レジスタ (0x0E1~0x0E7)	12
4.8. ASIC_POL レジスタ (0x0E8)	13
5. データ破棄アルゴリズム	15

1. 適用

本仕様書は、Kalliope-DC-RCNP 版として作成したファームウェアの機能を記述するものです。本ファームウェアは「Kalliope-DC 版」の回路に立ち上がり検出の 04 イベントを追加したものです。本回路は Volume2012, Volume2014, NIM 版それぞれに実装しています。それぞれのファームウェアの名称を以下に示します。

表 1-1 ファームウェアの名称

ファームウェア名	説明
yy.mm.dd-DCTOT-VOLUME2012	VOLUME2012 用
yy.mm.dd-DCTOT-VOLUME2014	VOLUME2014 用
yy.mm.dd-DCTOT-NIM	NIM ケース用

yy.mm.dd はファイルバージョンを区別する日付です

2. 構成

Kalliope ファームウェア DC 版の TCP データ処理の構造を図 2-1 に示します。

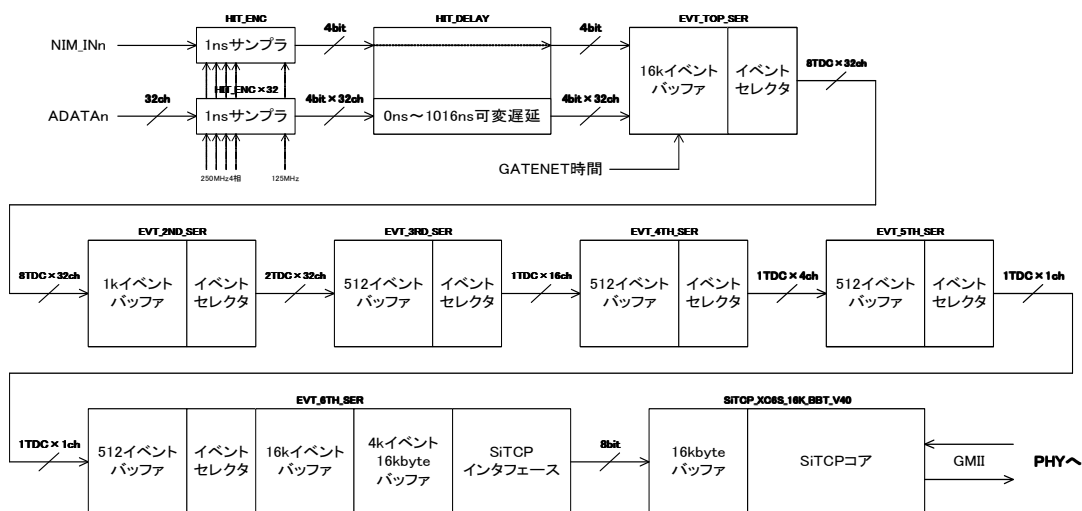


図 2-1 Kalliope ファームウェア DC-RCNP 版の TCP データ処理構造

2.1. HIT_ENC

HIT_ENC は 1ns サンプラです。入力信号を 4 相の 250MHz クロックでサンプリングする事で 1ns の分解能を実現しています。サンプリング精度が±200ps となるように配置を調整しています。(PLL のジッタ含まず) サンプルされた結果からエッジを検出します。エッジの最少間隔が 8ns 以上の場合のみ検出します。検出された結果は 125MHz 毎に 8 ポジションの TDC データとして出力されます。なお、HIT_ENC は入力信号毎に独立に実装しています。

2.2. HIT_DELAY

HIT_DELAY は 0ns~1016ns (127×8ns) の可変遅延を ADATA に付加するためのモジュールです。遅延量は 8ns 単位に設定できます。ADATA で固定的に発生する遅延量を補償するために NIM_IN の信号も遅延させています。なお、遅延量は RBCP レジスタの 0x010~0x013 で設定します。レジスタは 32bit ですが、設定可能範囲は 0~127 です。

2.3. EVT_TOP_SER

EVT_TOP_SER は約 128 μ s 中のすべてのイベントを記録できるバッファです。8ns 単位に記録するため、1 レコードに 1~32 のイベントが格納されます。従って約 16k~512k のイベントを格納できます。本バッファでは、NIM_IN イベントを 1 イベントしか格納できません。また、TCP データ処理過程でデータを破棄するのは、本バッファに空きがない状態でイベントが発生した場合のみです。

本バッファは後段のバッファに空きがあれば 24ns 毎に 1 レコードのデータを処理します。また、この処理で NIM_IN イベントの情報をイベントデータに重畳します。

2.4. EVT_2ND_SER

EVT_2ND_SER でも 1 レコードに 1~32 のイベントを格納します。EVT_TOP_SER では、NIM_IN イベント情報を 1 つまでしか格納できませんでしたが、本バッファでは NIM_IN イベントの情報は、1 レコードの情報として格納されます。

本バッファは後段のバッファに空きがあれば 32ns 毎に 1 レコードのデータを処理します。また、この処理で 8ns のイベントデータを 2ns 単位に並べます。

2.5. EVT_3RD_SER

EVT_3RD_SER でも 1 レコードに 1~32 のイベントを格納します。EVT_2ND_SER では 8ns 単位に同一レコードに記録していましたが、EVT_3RD_SER では 2ns 単位に同一レコードに記録します。

本バッファは後段のバッファに空きがあれば 32ns 毎に 1 レコードのデータを処理します。また、この処理で 2ns のイベントデータを 1ns 単位に並べ、32ch 単位のデータを 16ch 単位に変更します。さらに NIM_IN イベント情報を 3 イベントに分解します。これは、処理の過程でバス幅が狭くなる為、1 イベントのままではすべてのデータを転送できないためです。

2.6. EVT_4TH_SER

EVT_4TH_SER では 1 レコードに 1~16 のイベントを格納します。EVT_3ND_SER では 2ns 単位に 32ch を同一レコードに記録していましたが、EVT_4TH_SER では 1ns 単位に 16ch を同一レコードに記録します。

本バッファは後段のバッファに空きがあれば 32ns 毎に 1 レコードのデータを処理します。また、この処理で 16ch 単位のデータを 4ch 単位に変更します。

2.7. EVT_5TH_SER

EVT_5TH_SER では 1 レコードに 1~4 のイベントを格納します。EVT_4TH_SER では 16ch を同一レコードに記録していましたが、EVT_5TH_SER では 4ch を同一レコードに記録します。

本バッファは後段のバッファに空きがあれば 32ns 毎に 1 レコードのデータを処理します。また、この処理で 4ch 単位のデータを 1ch 単位に変更します。また、NIM_IN イベントが後段の処理に必要なフォーマットに変更可能なように 4 イベントから 7 イベントにします。

2.8. EVT_6TH_SER

EVT_6TH_SER は三つのバッファを持ちます。初めのバッファでは1レコードに1イベントを格納します。

本バッファは後段のバッファに空きがあれば16ns 毎に1イベントのデータを処理します。また、この処理では、TDCのBit16への繰り上がりフラグを分離して02イベントを挿入します。また、NIM_IN イベントを最終出力形式に変更します。

次のバッファは出力形式になった32bit イベントデータを16k 格納します。本バッファがイベントデータのメインバッファとなります。

最後のバッファは、32bit イベントデータをSiTCP インタフェースの適合させるため8bit のバスサイズに変更します。この時エンディアンの変更を可能としています。ただし、エンディアンの変更は32bit 中の8bit データの並び順のみ変更できます。

3. TCP データ

TCP データフォーマットを図 3-1 に示します。NIM_IN の検出毎にこのフォーマットが繰り返されます。データの生成は NIM_IN の検出で 5C イベントからそれに続く Copper Header とその中に埋め込まれている 01 イベントを出力し、続いて 02 イベント(カウント値 0x0000)を出力します。その後、エッジ検出のたびに 03, 04 イベントが生成され、65, 536ns 毎に 02 イベントが生成されます。次の NIM_IN イベントで Copper Trailer が出力された後、次の先頭である 5C イベントが出力されます。

なお、エッジ検出がないと 03, 04 イベントは出力されません。これらのイベントは TCP データストリーム上で時間順序が保存されます。同時の場合は、NIM_IN イベント、02 イベント、03 イベントの順に出力されます。03, 04 イベントではチャンネルの小さい順に出力されます。

また、図中の左側に示した、「5C イベント」「Copper Header」「Copper Trailer」は、RBCP のコントロールレジスタ (0x00C) で出力を独立に抑止できます。

さらに 04 イベントもコントロールレジスタ (0x00C) で出力を抑止できます。

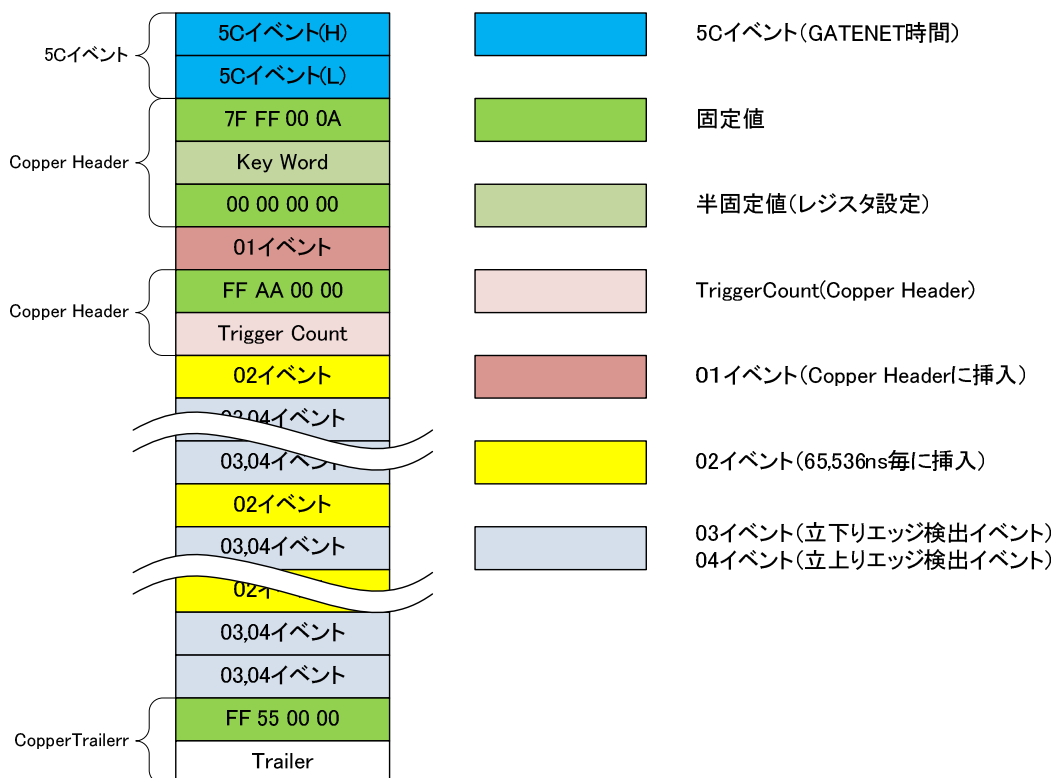


図 3-1 Kalliope ファームウェア DC 版の TCP データフォーマット

3.1. 5C イベント

5C イベントは例外的に 64bit イベントです。5C から始まる 32bit に続く 32bit の値は任意の値をとるので注意が必要です。64bit のイベントフォーマットは、GATENET のイベントフォーマットと互換です。しかしながら、時刻は内部的に生成されているため、事前にレジスタ (0x0E1~0x0E7) に GATENET 時刻を設定する必要があります。5C イベントのフォーマットを図 3-2 に示します。なお、GATENET 時刻は 2008 年 1 月 1 日 0:00.00 を原点として定められています。

0x5c		S[29:0]	SS[14:0]	US[10:0]
0x5C	8bit	5C イベントを示す固定値		
S[29:0]	30bit	秒単位のカウンタ		
SS[14:0]	15bit	1/32,768秒カウンタ(約30.5 μs)		
US[10:0]	11bit	25nsカウンタ		

図 3-2 5C イベントフォーマット

3.2. Key Word

32bit の Key Word は、Copper Header の一部です。先頭の 8bit は 0x00 固定で、残り 24bit はファームウェアによって異なります。ファームウェアとキーワード値の関係を表 3-1 に示します。これらの値は NIM_IN の検出後、約 100ns でサンプルされます。Key Word フィールドフォーマットを図 3-3 に示します。

表 3-1 ファームウェアによるキーワード値

ファームウェア	キーワード値
VOLUME2012 用	8ns 単位でカウントする 24bit フリーラン・カウンタの値
VOLUME2014 用	
NIM ケース用	下位 3bit は IN1~IN3 の値、上位 21bit は常に 0

0x00	Keyword[23:0]
------	---------------

0x00	8bit	固定値
Keyword[23:0]	24bit	キーワード値

図 3-3 Key Word フィールドフォーマット

3.3. Trigger Count

32bitのTrigger Countは、Copper Headerの一部です。先頭の24bitはTrigger Counterで、NIM_INの検出回数を示します。残りの8bitは0x00固定です。

なおTrigger Counterは、TCPセッションを切断すると0に戻ります。また、セッションを開いたのちに1回目に検出したNIM_INで送出されるTrigger Counterの値は0となります。Trigger Countフィールドフォーマットを図3-4に示します。

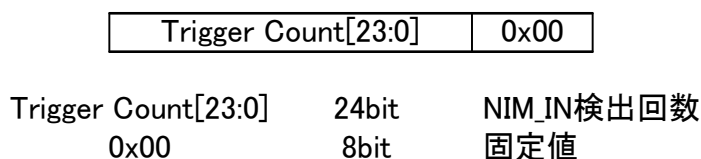


図 3-4 Trigger Count フィールドフォーマット

3.4. 01 イベント

32bitの01イベントはCopper Headerの中に埋め込まれるイベントで、NIM_INの検出回数を出力するためのイベントです。先頭の8bitは01イベントを示す固定値です。残り24bitはTrigger Countで、NIM_INの検出回数を示します。Trigger Countの値はロールオーバーします。01イベントフォーマットを図3-5に示します。

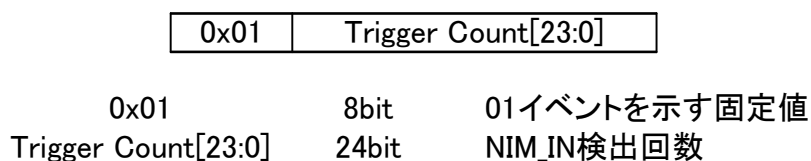


図 3-5 01 イベントフォーマット

3.5. 02 イベント

32bit の 02 イベントは Copper Header に続いて出力された後、65, 536ns 毎に出力される NIM_IN 検出からの時間の上位 16bit を出力するためのイベントです。後述する 03 イベントでは NIM_IN 検出からの時間の下位 16bit のみを出力するため、このイベントを用いて上位 16bit を補います。上位 16bit もロールするので、約 4.2 秒後に 0 に戻ります。しかしながら、01 イベントが無いまま 02 イベントの TDC 値が減少する事でロールを確認できます。

先頭の 8bit は 02 イベントを示す固定値です。続く 8bit は module ID、残り 16bit が NIM_IN の検出からの時刻の上位 16bit を示します。module ID は、IP アドレスの下位 8bit です。IP アドレスの下位 8bit なので、Force Default に設定されていると 0x10 に固定されます。

02 イベントフォーマットを図 3-6 に示します。

0x02	MID[7:0]	TDC[31:16]
0x02	8bit	02 イベントを示す固定値
MID[7:0]	8bit	モジュールID (IPアドレスの下位8bit)
TDC[31:16]	16bit	NIM_IN検出時刻からの時間の上位16bit

図 3-6 02 イベントフォーマット

3.6. 03, 04 イベント

32bit の 03, 04 イベントは ADATA のエッジを検出するたびに出力される NIM_IN 検出からの時間の下位 16bit を出力するためのイベントです。前述の 02 イベントを用いて上位 16bit を補います。03 イベントは立下り（信号の検出開始）を示し、04 イベントは立上り（信号の検出終了）を示します。

先頭の 8bit は 03, 04 イベントを示す ID です。続く 8bit は検出したチャンネル番号を表し、残り 16bit が NIM_IN の検出からの時刻の下位 16bit を示します。チャンネル番号は、0~31 の値をとります。03 イベントフォーマットを図 3-7 に示します。

ID	CH[7:0]	TDC[15:0]
ID	8bit	0x03: 立下りイベント 0x04: 立上りイベント
CH[7:0]	8bit	エッジ検出したチャンネル番号
TDC[31:16]	16bit	NIM_IN検出時刻からの時間の下位16bit

図 3-7 03,04 イベントフォーマット

3.7. Trailer

32bit の Trailer は、Copper Trailer の一部です。互換性維持のため同一のフォーマットとなっていますが、TxBufFull ビットは、直前のフレームの一部または全部を破棄した事を示す LOS ビットとなっています。この Bit が 1 の時は、直前の 5C イベントから Copper Trailer 間に破棄があった事を示しています。破棄のアルゴリズムについては、「5 データ破棄アルゴリズム」を参照してください。

bit18 を除くすべての bit は固定値です。Trailer フィールドフォーマットを図 3-8 に示します。

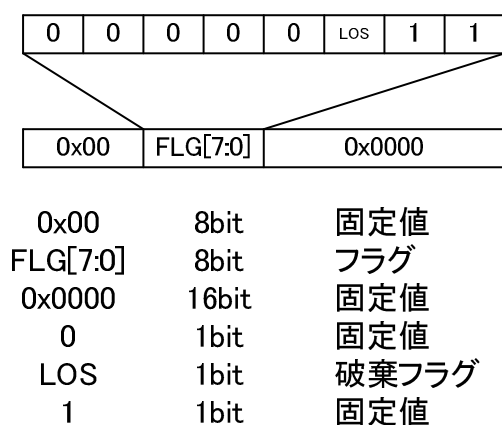


図 3-8 Trailer フィールドフォーマット

4. RBCP メモリマップ

RBCP を用いて各種設定を行います。RBCP に関しては基本的に変更していませんが、一部レジスタの追加を行っています。また、一部レジスタは入力が無くなる事により、未使用となっています。表 4-1 に RBCP のメモリマップを示します。

表 4-1 RBCP レジスタマップ

アドレス	領域 (byte)	レジスタ名	説明
0x000~0x003	4	VER	FPGA のバージョン識別用
0x004~0x007	4	FPGA_ID	FPGA 種別コード
0x008~0x00B	4	EVENT_NUM	Trigger Count 値
0x00C	1	FPGA_CTRL	制御レジスタ
0x00D~0x00F	3	KEY_WORD	Key Word レジスタ
0x010~0x013	4	DELAY	入力付加遅延設定レジスタ
0x014~0x017	4	GAP	未使用
0x018~0x019	2	PARAM	on-chip procedure の引数
0x01A~0x01B	2	CMD	on-chip procedure の制御
0x01C~0x01D	2	DISPLAY1	on-chip procedure の表示 1
0x01E~0x01F	2	DISPLAY2	on-chip procedure の表示 2
0x020~0x07F	3byte × 32ch	DacData1	ASIC コントロール Bank1
0x080~0x0DF	3byte × 32ch	DacData2	ASIC コントロール Bank2
0x0E0	1	—	未使用
0x0E1~0x0E7	7	GATENET_TIME	GATENET 時間
0x0E8	1	ASIC_POL	ASIC コントロール極性制御
0x0E9~0x0EF	7	—	未使用
0x100~0x13F	2byte × 32ch	eCount	未使用
0x140~0x17F	2byte × 32ch	nCount	未使用
0x180~0x1FF	128	—	未使用
0x200~0x27F	4byte × 32ch	eInteg	未使用
0x280~0x2FF	4byte × 32ch	nInteg	未使用

4.1. VER レジスタ (0x000~0x003)

VER レジスタはファームウェアの新旧を識別するためのレジスタです。合成した年月日とその日の中のシリアルを 4byte で表示します。

4.2. FPGA_ID レジスタ (0x004~0x007)

FPGA_ID レジスタはファームウェアの種類を識別するためのレジスタです。本ファームウェアは修正前と同じ「20 02 00 10」としてあります。

4.3. EVENT_NUM レジスタ (0x008~0x00B)

EVENT_NUM レジスタはTCPセッションを開いてからNIM_INを検出した回数を表示します。この値はTrigger Count フィールドや01 イベントのTrigger Counter の値と同じです。

なお、Trigger Counter は本レジスタサイズと同様 32bit ですが、TCP データでは 24bit である為、下位 24bit が使用されます。Byte 単位の排他制御はないため、読み出し中に更新される可能性がある事に注意してください。

4.4. FPGA_CTRL レジスタ (0x00C)

動作モードを指定するレジスタです。詳細は表 4-2 を参照してください。

表 4-2 FPGA_CTRL レジスタのビットマップ

bit	名称	説明	初期値
7	NU	未使用	0
6	BYTE_SWAP_ENABLE	1 を設定すると 32bit データを下位バイトから出力します。	1
5	NU	未使用	0
4	EVT04_MASK	1 を設定すると 04 イベントの出力を抑止します。	0
3	COPHDD_MASK	1 を設定すると Copper Header の出力を抑止します。	0
2	COPTRL_MASK	1 を設定すると Copper Trailer の出力を抑止します。	0
1	GATNET_MASK	1 を設定すると 5C イベントの出力を抑止します。	0
0	TDC_ENBALE	未使用	0

4.5. KEY_WORD レジスタ (0x00D~0x00F)

本ファームウェアでは未使用です。

4.6. DELAY レジスタ (0x010~0x013)

ADATA に付加する遅延値を 8ns 単位で設定します。レジスタフィールドは 32bit ですが、設定できる値は 0x00000000~0x0000007F であり、付加できる遅延量は 0ns~1,016ns です。初期値は 0x00000000 です。

4.7. GATENET_TIME レジスタ (0x0E1~0x0E7)

GATENET 時間を設定します。5C イベントを設定する場合、本レジスタに 2008 年 1 月 1 日 0:00.00 からの経過時間を設定します。フォーマットの詳細は「3.1 5C イベント」を参照してください。本レジスタの書き込みは 0x0E1 から 0x0E7 の順に書き込むと 0x0E7 を書きこんだタイミングで時刻が設定されます。読み出しは、0x0E1 から 0x0E7 の順に読み出すと、0x0E1 を読み出した時刻が読み出されます。初期値は、起動時間を 2008 年 1 月 1 日 0:00.00 とします。

4.8. ASIC_POL レジスタ (0x0E8)

本レジスタは Volume2014 専用レジスタです。ASIC を制御する信号線の機能を設定します。図 4-1 にレジスタ設定時の波形を示します。なお、Ver16111403 以前の DAC_CLBn ピンは機能追加に伴い、DAC_CLBn[0] にピン名変更しています。DAC_CLBn[1] は、Ver16111403 以前では未使用 (DAC_DIN[0]) となっていた J1-B34 (FPGA の AA6, 回路図名称の DAC_SCLK) に割り当てています。

本レジスタで DAC_CLBn[0] ピンの機能や極性、DAC_CLK の極性を変更できます。なお、DAC_CLBn[1] は、レジスタ設定に関わらず負論理のプレクリアとして機能します。

表 4-3 ASIC_POL レジスタのビットマップ

Bit	名称	説明	初期値
7	NU	未使用	0
6	NU	未使用	0
5	NU	未使用	0
4	NU	未使用	0
3	DAC_CLBn_SEL	00: DAC_CLBn[0] は、プレクリア信号	0
2		01: DAC_CLBn[0] は、ポストロード信号 1X: DAC_CLBn[0] は、データシフトイネーブル信号 (SSB)	
1	DAC_SCLK_POL	0: SCLK は立ち上がりエッジ 1: SCLK は立ち下がりエッジ	0
0	DAC_CLBn_POL	0: DAC_CLBn[0] を負論理で使用 1: DAC_CLBn[0] を正論理で使用	0

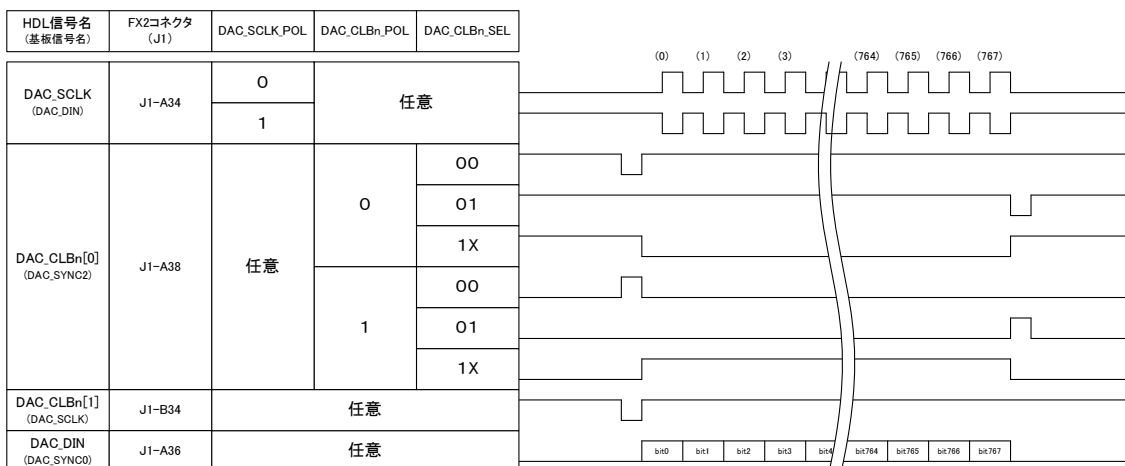


図 4-1 Volume2014 の ASIC 制御信号における ASIC_POL レジスタの効果

表 4-4 Volume2014 の DAC_DIN におけるシリアルデータのビット位置

bit 位置	送出データ	
	dacBank[1:0] = 1	dacBank[1:0] = 2
0	レジスタ 0x22 の bit0	レジスタ 0x82 の bit0
1	レジスタ 0x22 の bit1	レジスタ 0x82 の bit1
2	レジスタ 0x22 の bit2	レジスタ 0x82 の bit2
3	レジスタ 0x22 の bit3	レジスタ 0x82 の bit3
4	レジスタ 0x22 の bit4	レジスタ 0x82 の bit4
24k+8m+n	レジスタ 0x22+3k-m の bit n	レジスタ 0x82+3k-m の bit n
764	レジスタ 0x7D の bit4	レジスタ 0xDD の bit4
765	レジスタ 0x7D の bit5	レジスタ 0xDD の bit5
766	レジスタ 0x7D の bit6	レジスタ 0xDD の bit6
767	レジスタ 0x7D の bit7	レジスタ 0xDD の bit7

※k は 0~31 の整数、m は 0~2 の整数、n は 0~7 の整数です。

※DAC_DIN[0] は、0 固定です。

※bit 位置は、図 4-1 の DAC_DIN の波形を参照してください。

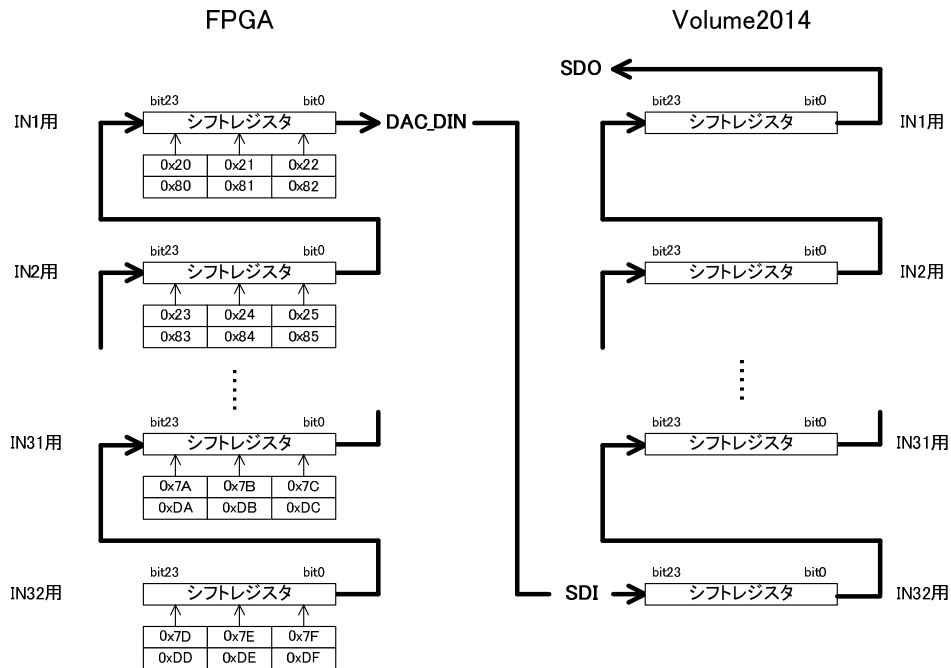


図 4-2 Volume2014 の DAC_DIN におけるシリアルデータの概念図

5. データ破棄アルゴリズム

内部バッファに空きが無くなりデータを破棄した場合の処理について記載します。破棄したデータが NIM_IN であった場合、次の NIM_IN 入力があるまですべてのデータを破棄し続けます。なお破棄を行うバッファでは、トレーラやヘッダのデータは1つのレコードに記録されるため、部分的な破棄は発生しません。また、NIM_IN を破棄した場合でも Trigger Counter は更新されます。

破棄するデータが 02, 03, 04 イベントの場合、次の NIM_IN まで破棄を続けます。

この破棄アルゴリズムによって 02 イベントが途中で抜ける事は発生しませんので、TDC の上位 16bit を誤って認識する事はありません。