

uSD-CONF2A 仕様書(暫定)  
(for Xilinx ユーザ)  
(USDCNF-MD2A-001A)

Rev 0.1 2013.06.13

悟空株式会社

## 改訂履歴

日時	担当者	Rev	改訂内容概要
2013/06/13	大庭	0.1	新規作成 (草案)

## 目次

1. 機能・特徴	4
2. 形状	7
電気特性	8
3.1. 絶対定格	8
3.2. 推奨動作範囲	8
3.3. 消費電流	8
3.4. 電源シーケンス	8
4. Pin 機能表	9
4.1. uSD-CONFIG2A ピン配置図(Top View)	9
4.2. USDCNF-MD3-001A ピン機能表	10
4.3. XMODE 詳細	11
4.4. XAREA 詳細	12
5. 接続例	13
5.1. XILINX	13
5.1.1. uSDCONF2A 接続例	13
6. タイムチャート	19
5.1. 信号出力部ブロック図	19
5.2. タイムチャート	19
6. LED	20
6.1. LED の点灯条件	20
6.2. バージョン情報表示機能	21
7. “CONFIG.TXT”の準備	22
7.1. “CONFIG.TXT”とは	22
7.2. “CONFIG.TXT” サンプル	23
7.3. Commands 速見表	24
7.4. Commands 詳細	25
8. 機能詳細	32
8.1. ワードアライナ機能	32
8.2. bit 連結による Multi FPGA Configuration 機能	33
8.3. “CONFIG.TXT”による Multi FPGA Configuration 機能	34
9. モジュールのソケットからの取り外しについて	37
9.1. モジュール裏面	37
10. 制約事項	39
11. アップデート機能(順次対応予定)	41

## 1. 機能・特徴

- 本モジュールは、microSD カードと 2 個の NandFlash を搭載し、Windows システムによって microSD カードに記憶されたバイナリデータを直接 FPGA に転送してコンフィギュレーションを行う機能、及び、microSD カード上のバイナリデータを一旦 NandFlash に転写し、NandFlash からバイナリデータを FPGA に転送してコンフィギュレーションを行う機能を有します。
- 本モジュールは 4Gbit 16bit 幅 NandFlash を 2 個実装し、1G バイトの NandFlash 空間に 32bit 幅でアクセスします。
- NandFlash は 1G バイト空間のうち 880Mbyte の空間を 16 個の 55Mbyte のエリアに分割し、AREA[3:0]ピンによりエリアを指定することが可能です。残りの NandFlash の空間は、エラーブロックを検出後、Format を実行したときに、エラーブロックを隔離し、不足分を残りのエリアから割り当てます。
- microSD カードには、最大 2GB までの FAT16 でフォーマットされた Card を使用でき、容量の許す限りバイナリデータを格納することができます。  
(FAT32 でフォーマットされた SD カードや、2GByte を超える SDHC には現状対応していません。)
- SD カードからのバイナリデータの読み出しは 50MHz の HighSpeed モードで転送が行なわれ、200Mbps の転送速度を有します。
- microSD カードの電源の供給を制御し、microSD カードが INACT 状態に陥った場合でも、システムの電源を落とすことなく microSD カードの電源を ON/OFF し、INACT 状態から復旧することが可能です。
- microSD カードへのバイナリデータの書き込みは、Windows システム上の Explorer 上で Drag & Copy するだけで簡単に行えます。JTag ケーブルを接続する煩わしさがありません。
- microSD カード上に格納する“CONFIG.TXT”の名称のアスキーキャラクタで記述された制御ファイルにより、FPGA に転送するバイナリデータの選択や、コンフィギュレーションに関する各種パラメータを指定することが可能です。
- microSD カード上に格納されたバイナリデータは、“CONFIG.TXT”上で、バイナリデータ名と AREA[3:0]ピンによって指し示される 0~F の 16 個の 16 進数文字と関連付けを行うことにより、AREA[3:0]ピンに接続されたロータリーSW 設定により、最大 16 個のバイナリデータを瞬時に選択し、FPGA を直接コンフィギュレーションすることが可能です。  
また、NandFlash への転写モードが選択されている場合、AREA[3:0]ピンと“CONFIG.TXT”により選択されたバイナリデータを、同じく AREA[3:0]ピンで指定された NandFlash の分割されたエリアに対して、各種パラメータとともに転写します。
- 大規模 FPGA や、DaisyChain によるコンフィギュレーションを行う場合で、かつエリアサイズを超えたバイナリデータを取り扱う必要がある場合、uSDCONF2A 自身の論理を、エリアサイズを拡張した論理に置き換える必要があります。  
分割数 16 分割をデフォルトとし、8分割、4分割、2分割、分割なしの 5 種類の更新データを準備しています。
- 更新データの書き換えには、uSDAdapter と JTAG ケーブルが別途必要です。
- NandFlash から FPGA へのコンフィギュレーションが選択された場合、NandFlash からの読み出しデータに対して ECC 機能によりエラービットの訂正が行われます。
- ECC は 512 バイト単位で計算され、4bit までのエラーの修正を行います。

このとき、転送終了後に、LED を1回ブリンクさせます。

- ECC エラーが512バイト単位で 5bit 以上発生した場合、データは修復されません。  
このため、コンフィギュレーションは失敗に終わります。  
このとき、転送終了後に、LED を4回ブリンクさせます。
- ECC エラーは1ビットでも検出されると、発生したブロックに対しエラー履歴を残します。  
再フォーマットを行ったとき、エラーを検出したことがあるブロックは隔離されます。
- フォーマットは、修復不可能のエラーを検出した場合や、修復されたエラー検出が頻発するようになったときに行ってください。
- MODE[3:0]ピンにより NandFlash から FPGA へのコンフィギュレーションモードが選択された場合、NandFlash の 32bit バス幅の恩恵より、FPGA への高速コンフィギュレーションが可能になります。  
NandFlash からの読み出しは、50MHz 2 クロックで 1word 読みだすことができるため、  
50MHz/2clk x 32bit = 約 800Mbps のデータ転送速度となります。(オーバヘッドがあるため、この数値より少し落ちる値になります。)
- FPGA への転送バス幅は、MODE[3:0]ピンの設定により 1bit(Slave Serial モード)と 8bit(SelectMAP モード)から選択することが可能です。(MODE ピン参照)
- FPGA への転送クロック CCLK は、“CONFIG.TXT”で指定することにより、下記の速度から選択可能です。  
50MHz(デフォルト),25MHz, 12.5MHz, 8.33MHz, 6.25MHz,5MHz, 3.125MHz,1.56MHz, 0.78MHz, 0.39MHz
- “CONFIG.TXT”の先頭で、メーカー指定コマンドを入力すると、そのメーカーに固有のパラメータ値をデフォルトとして設定することが可能です。
- ワードアライナ機能を“CONFIG.TXT”上で有効にすることにより、“.bit”ファイルの先頭にある管理情報を読み飛ばし、32 個の FF から転送を開始することが可能です。  
これにより、“.bit”ファイルの先頭にある管理情報によってバス幅検出用のパターン位置が不定となることを防ぐことが可能となり、FPGA のバス幅が 1bit,8bit はもとより、16bit,32bit に対してもバイナリデータとして、“.bit”ファイルも指定することが可能になります。(uSD-CONF2A は 1,8bit バス幅のため、設定に左右されません。)
- 0 から 15 回のコンフィギュレーションのリトライ回数が指定できます。
- バイナリデータの先頭にプリアンブルを挿入することや、バイナリデータの最後にポストアンブルを可変長で挿入することが可能です。
- PROGB の立ち上がりから INITB の立ち上がりまでの間隔や、INITB の立ち上がりから CCLK の出力開始までの間隔を指定することが可能です。
- 複数の FPGA をコンフィギュレーションする方法を3通りから選択することが可能です。
  - ① 従来の CSI/CSO のデジチェーンによる Multi-FPGA コンフィギュレーション  
iMPACT によるファイルの連結が必要です。(Slave Serial のみ)
  - ② bit 連結による Multi-FPGA コンフィギュレーション  
バイト単位にビット多重されたバイナリデータを最大 8 個まで Slave Serial 接続するモードです。  
(バイト多重するハードウェアは組み込まれていますが、ソフトを準備していません。  
③の機能で、専用のソフトウェアを使わずに簡単に Multi-FPGA のコンフィギュレーションが可能  
なためです。)
  - ③ “CONFIG.TXT”上のファイルの連結指定による Multi-FPGA コンフィギュレーション

“CONFIG.TXT”上で、0:data1.bin + data2.bin + .... のように記述することにより、モジュールがこれを解釈し、コンフィギュレーション時にモジュールが各 FPGA の CS 信号を制御することにより順次コンフィギュレーションするモードです。

連結のための専用のソフトウェアが必要ありません。

NandFlash への転写時は、連結されたファイルと閾値情報が Nandflash に書き込まれます。

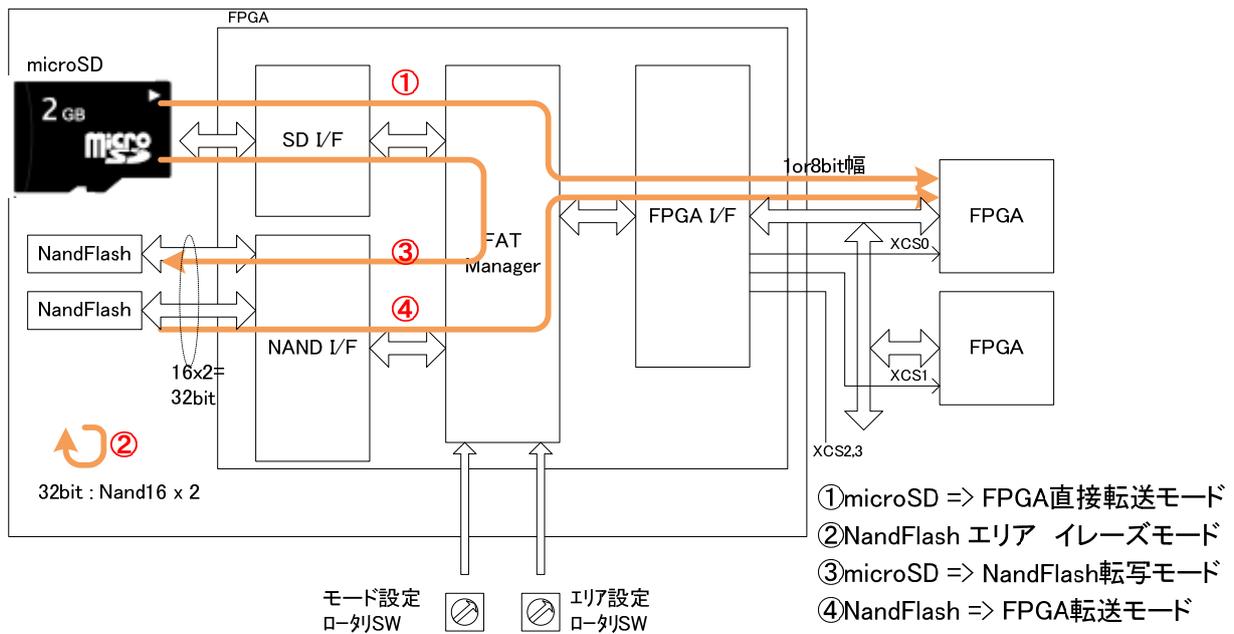


図 1.1 uSD-CONF2A ブロックイメージ図

## 2. 形状

uSD-CONF2A モジュールの形状を掲載します。

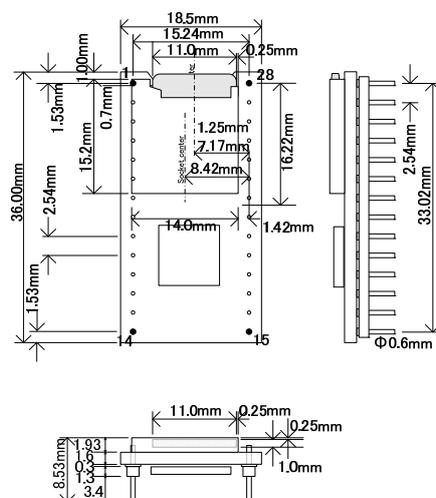


図 2.1 uSD-CONF2A 形状図

- ・ uSD-CONF2A の使用 pin は、丸型 0.6mm 径の MAC8 の HQS-2-5-14P を両側に 600mil の間隔で配置しています。 (<http://www.mac8sdk.co.jp/mac8/pdf/HQS.pdf>)
- ・ ソケット選択時は、0.6mm  $\phi$  に対応していない場合が多いのでご注意ください。

### ソケット例

下記はソケットの一例です。上記 HQS-2-5-14P に適合するソケットをご利用下さい。

- (1) W3128TRC (Winslow) (RS 品番 402-822)

(<http://docs-asia.electrocomponents.com/webdocs/002b/0900766b8002bd23.pdf>)

- (2) その他 調査中。

- ・ SDCard の中心がモジュールの中心と一致します。
- ・ 各 pin には 70g の力が加わるため、挿抜には 1kg 近い力がかかる必要があります。このため、取り外し時に、ドライバなどで力を加えられるプロテクト板をモジュール下前後に配置しますので、そのプロテクト板に力を加えて取り外しを行って下さい。
- ・ 引き抜き工具として、Sunhayato IC Extractor(GX-7)用のヘッドを開発中です。準備ができましたら HomePage にて公開いたします。

## 電気特性

### 3.1. 絶対定格

Supply Voltage VCC3V	-0.2V	to	3.75V
Supply Voltage VCCIO	-0.2V	to	3.75V
Input or I/O Tristate Voltage Applied	-0.2V	to	3.75V
Storage Temperature(Ambient)	-65°C	to	150°C

### 3.2. 推奨動作範囲

Supply Voltage VCC3V	3.135V	to	3.465V
Supply Voltage VCCIO	1.32V	to	3.465V
Ambient Temperature	0°C	to	70°C

### 3.3. 消費電流

品目		Typcal(実測)	Max	備考
Module 本体	VCC(3.3V)		mA	
	VCCIO(3.3V 時)		mA	
SD カード	VDD(3.3V)		100mA	規格上の Max 値
合計			mA	

表 3.3 消費電流表

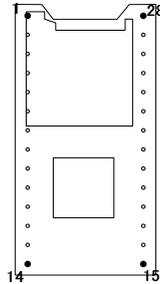
### 3.4. 電源シーケンス

- ・ VCC3V と VCCIO の間には、電源シーケンスの指定はありません。  
両者がともに動作許容範囲に入ったときに動作を開始します。

#### 4. Pin 機能表

uSD-CONF2A モジュールのピン機能表を掲載します。

##### 4.1. uSD-CONF2A ピン配置図(Top View)



## 4.2. USDCNF-MD3-001A ピン機能表

PinNo	信号名	Dir	モジュール内終端	外部終端	Descriptions
1	VCC3V	—			3.3V 電源入力
2	VCCIO	—			入出力信号の VCCIO 電源入力。
3	XAREA0	I	4.7kΩ Pullup	不要	Bitdata 選択用ロータリ SW 入力(LSB)
4	XAREA1	I	4.7kΩ Pullup	不要	Bitdata 選択用ロータリ SW 入力
5	XAREA2	I	4.7kΩ Pullup	不要	Bitdata 選択用ロータリ SW 入力
6	XAREA3	I	4.7kΩ Pullup	不要	Bitdata 選択用ロータリ SW 入力(MSB)
7	GND	—			GND
8	DONE	I	330Ω Pullup	不要	FPGA DONE 信号
9	CCLK	O	無し	100Ω Pullup+ 100Ω Pulldn	FPGA CCLK 信号
10	PROGB	O	4.7kΩ Pullup	不要	FPGA PROGB 信号
11	INITB	O	4.7kΩ Pullup	不要	FPGA INITB 信号
12	XMODE0	I	4.7kΩ Pullup	不要	動作モード選択信号 0
13	XCS0	O			DaisyChain 用 CS 信号(1 番目の FPGA)
14	XCS1	O			DaisyChain 用 CS 信号(2 番目の FPGA)
15	XCS2	O			DaisyChain 用 CS 信号(3 番目の FPGA)
16	XCS3	O			DaisyChain 用 CS 信号(4 番目の FPGA)
17	XMODE1	I	4.7kΩ Pullup	不要	動作モード選択信号 1
18	D0(SOUT)	O			FPGA 8bit Parallel data0(Serial Data)
19	D1	O			FPGA 8bit Parallel data1
20	D2	O			FPGA 8bit Parallel data2
21	D3	O			FPGA 8bit Parallel data3
22	GND	—			GND
23	D4	O			FPGA 8bit Parallel data4
24	D5	O			FPGA 8bit Parallel data5
25	D6	O			FPGA 8bit Parallel data6
26	D7	O			FPGA 8bit Parallel data7
27	XMODE2	I	4.7kΩ Pullup	不要	動作モード選択信号 2
28	XRST	I	4.7kΩ Pullup	不要	パワーオンリセット信号

表 3.1 uSD-CONF2A ピン機能表

## 4.3. XMODE 詳細

Mode [2:0]	Mode 名	詳細
[111]	microSDCard=>FPGA 直接転送モード(シリアル転送)	microSDCard=>FPGA コンフィグレーションモード(Slave Serial)
[110]	microSDCard=>FPGA 直接転送モード(8bit 転送)	microSDCard=>FPGA コンフィグレーションモード(Select Map 8bit)
[101]	Nand Flash=>FPGA 直接転送モード(シリアル転送)	NandFlash=>FPGA コンフィグレーションモード(Slave Serial)
[100]	Nand Flash=>FPGA 直接転送モード(8bit 転送)	NandFlash=>FPGA コンフィグレーションモード(Select Map 8bit)
[011]	microSDCard=>NandFlash 転写モード	AREA 選択 SW で指定された値を基に、microSDCard 上の CONFIG.TXT ファイルの#0~#F コマンドで指定されたコンフィギュレーションデータを、NandFlash 上の所定のエリアに転写します。 転写はイレーズを自動的に実行したあと転写されます。
[010]	N NandFlash エリア イレーズモード	AREA 選択 SW で指定された NandFlash 上のエリアをイレーズします。 NandFlash 上のコンフィギュレーションデータを消去したい場合に使用します。
[001]	NandFlash 初期フォーマットモード	NandFlash を初期化します。 データ格納エリアにバッドブロックが発生してコンフィギュレーションが正常に終了しなくなった場合に実行します。
[000]	No operation	FPGA の制御ピンが全て High-Z になります。

## 4.4. XAREA 詳細

- “CONFIG.TXT”ファイル上でSDカード上に格納されたバイナリデータと AREA[3:0]ピンによって指し示される 0~F の 16 個の 16 進数文字と関連付けがなされます。
- XAREA[3:0]入力は負論理で入力され、正論理に反転された AREA[3:0]信号と“CONFIG.TXT”上の 16 進数文字と比較が行われ、一致した 16 進文字と関連付けられたバイナリデータをコンフィギュレーションデータとして選択します。
- 表 4.4 は負論理で入力された XAREA[3:0]の信号レベルと、それによって指示されるエリア番号の対応表です。

XAREA				詳細
3	2	1	0	
H	H	H	H	エリア 0
H	H	H	L	エリア 1
H	H	L	H	エリア 2
H	H	L	L	エリア 3
H	L	H	H	エリア 4
H	L	H	L	エリア 5
H	L	L	H	エリア 6
H	L	L	L	エリア 7
L	H	H	H	エリア 8
L	H	H	L	エリア 9
L	H	L	H	エリア A
L	H	L	L	エリア B
L	L	H	H	エリア C
L	L	H	L	エリア D
L	L	L	H	エリア E
L	L	L	L	エリア F

表 4.4 XAREA[3:0]とエリア番号表

## 5. 接続例

### 5.1. XILINX

#### 5.1.1. uSDCONF2A 接続例

##### 5.1.1.1. Slave Serial/SelectMap8 接続(1:1 接続)

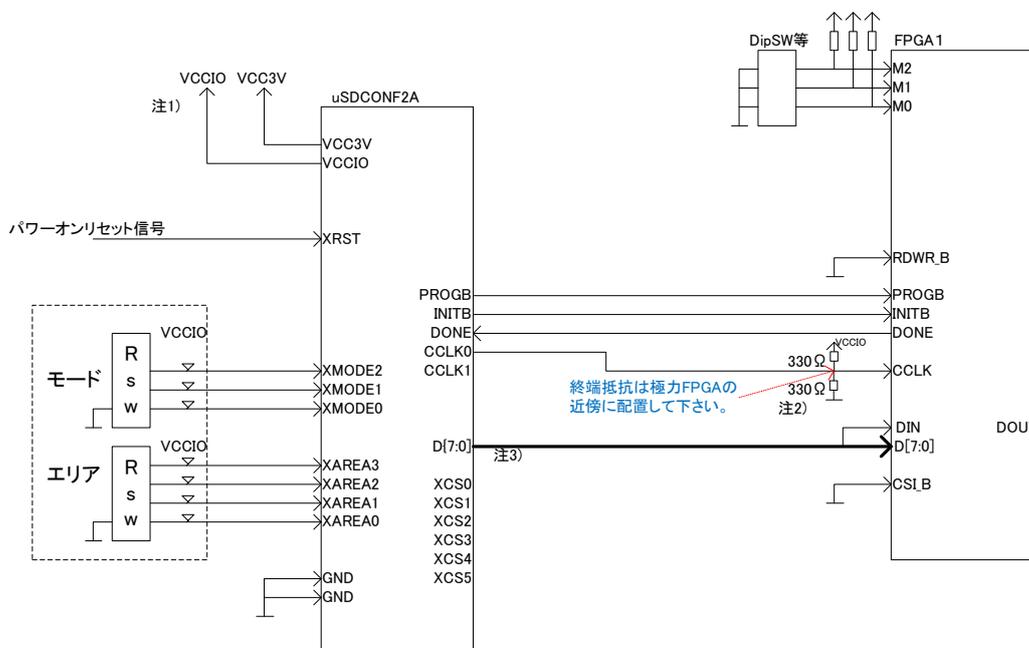


図 5.1.1.1(1) SlaveSerial/SelectMap8 接続例(1:1 接続)

- 注1) モジュールに入出力される全ての信号線レベルは VCCIO に依存します。  
VCCIO 電圧 2.5V から 3.3V 内においては可変に選択可能ですが、1.8V やさらに低い電圧レベルに対しては、モジュール搭載の FPGA データを入れ替える必要があります。
- 注2) FPGA の近傍に配置して下さい。また複数の FPGA と接続される場合は、最遠端の FPGA の近傍に配置して下さい。
- 注3) FPGA に D0 とは別にシリアル用の DIN 信号が存在する場合は、DIN に D[0]信号を接続してください。

5.1.1.2. Slave Serial 接続(1:N 接続)

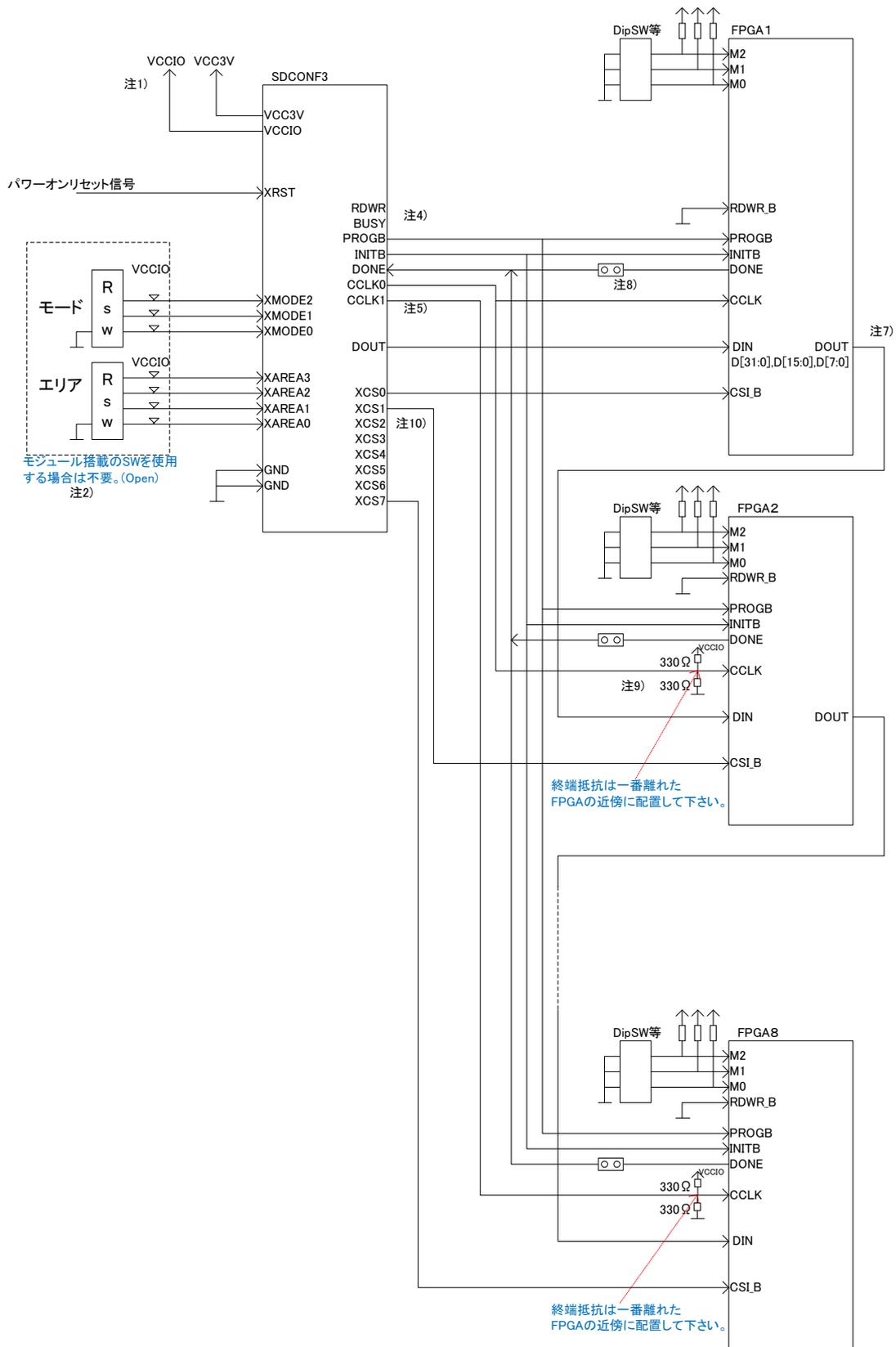


図 5.1.1.2 SlaveSerial 接続例(1:N 接続)

- 注1) 5.1.1.1 参照。
- 注2) 5.1.1.1 参照。
- 注3) 5.1.1.1 参照。
- 注4) 5.1.1.1 参照。
- 注5) CCLK0 と CCLK1 には同一の CLK が出力されています。  
複数の FPGA をコンフィグレーションする際にご利用下さい。
- 注6) 5.1.1.1 参照。
- 注7) Slave Serial モードでは CSI\_B による制御が行えないため、デジチェーンのルートを確保する必要があります。  
後述の bit 多重方式による Multi-FPGA 接続方式のほうがパフォーマンスが出ます。  
DIN 信号が専用ピンの場合は 5.1.1.4 のような接続が可能です。(Slave Serial と SelectMap とではバイナリデータを別々に用意する必要があります。
- 注8) DONE 信号をショートピンで切り離せるようにすることにより、デバッグの初期段階などで限定的な FPGA の立ち上げが可能となります。
- 注9) 最遠端の FPGA の近傍に配置して下さい。

## 5.1.1.3. Slave Serial/SelectMap8,16,32 接続(1:N 接続)+bit 多重

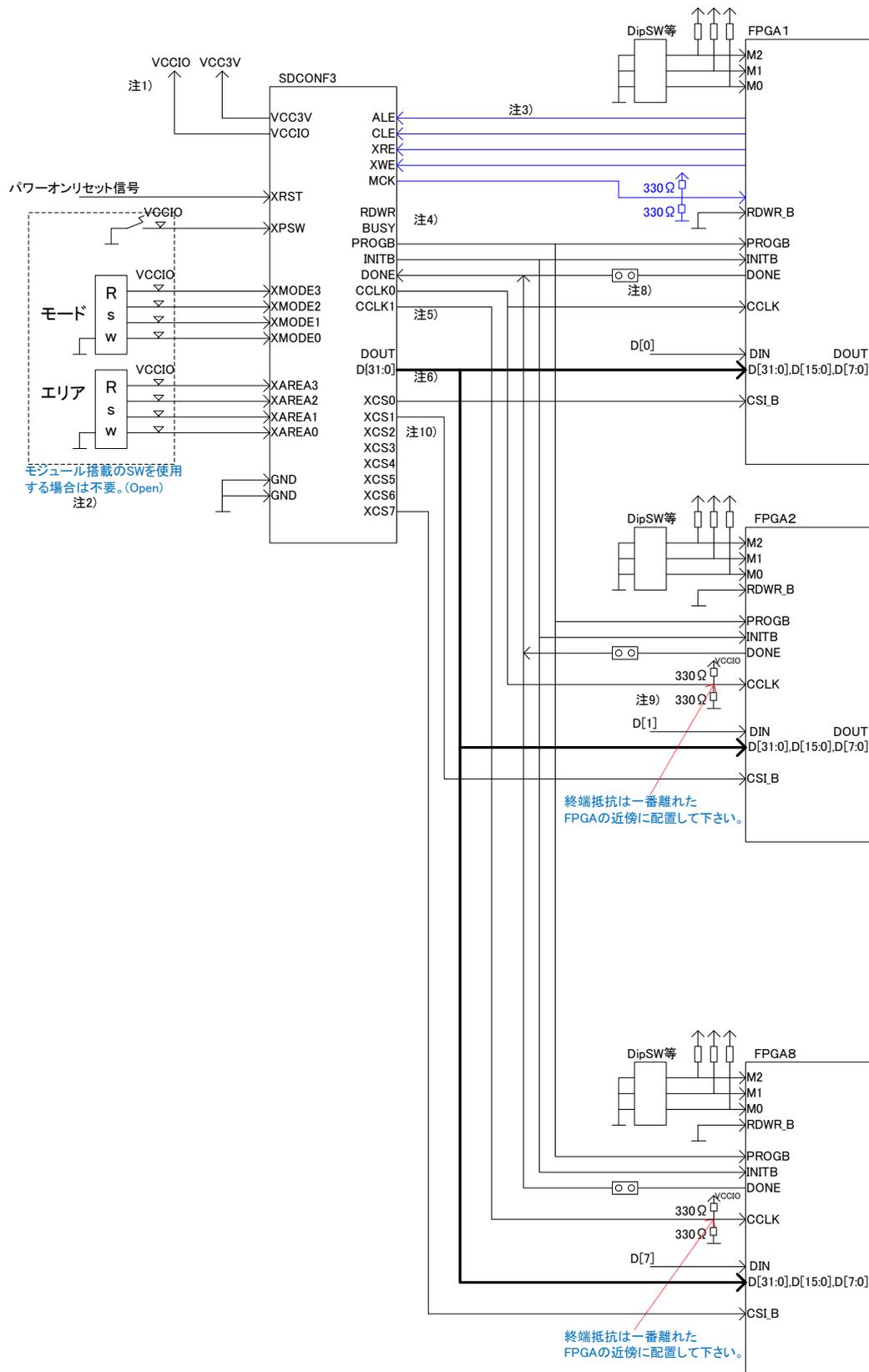


図 5.1.1.3 SlaveSerial/SelectMap8,16,32 接続例(1:N 接続)+bit 多重(差し替え要)

注) 5.1.1.2 の構成に Slave Serial モード時に bit 多重方式を採用するための接続方法です。DIN ピンが D[0]と共用の場合は選択回路を入れる必要があるでしょう。



5.1.1.5. 通常 Slave serial による Multi-FPGA 接続構成

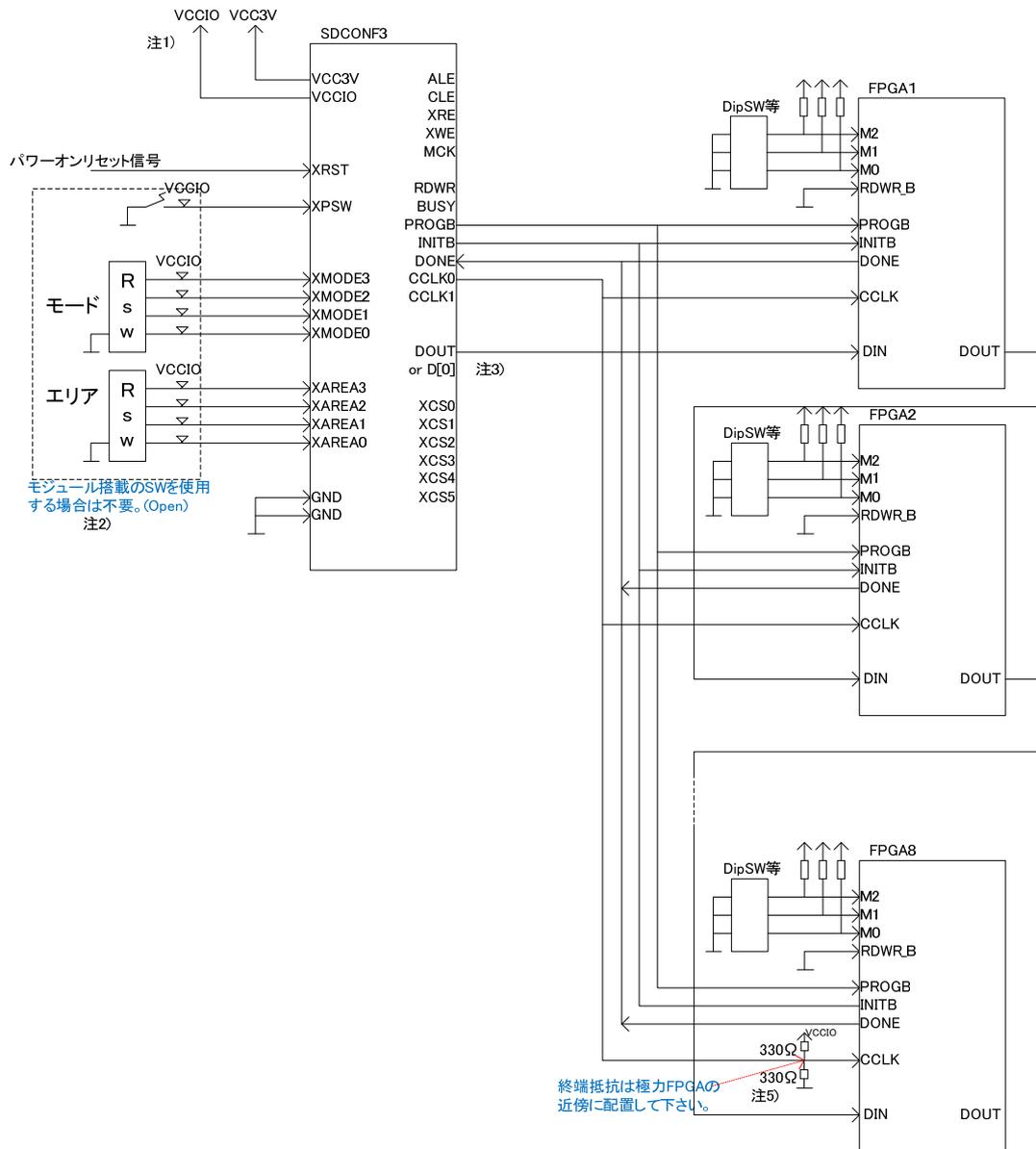
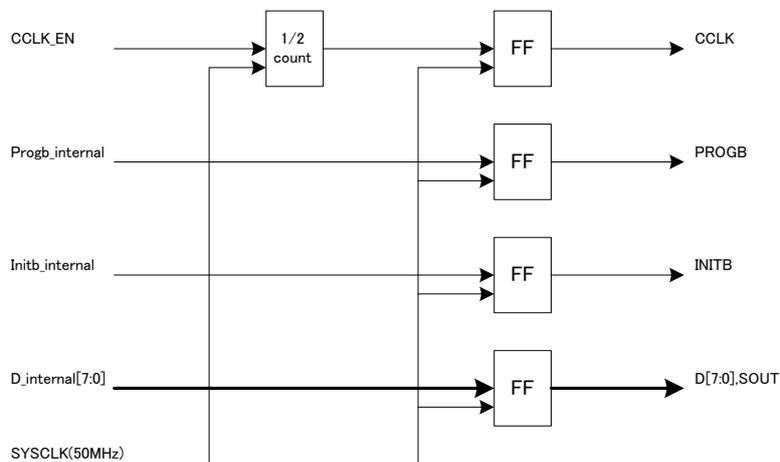


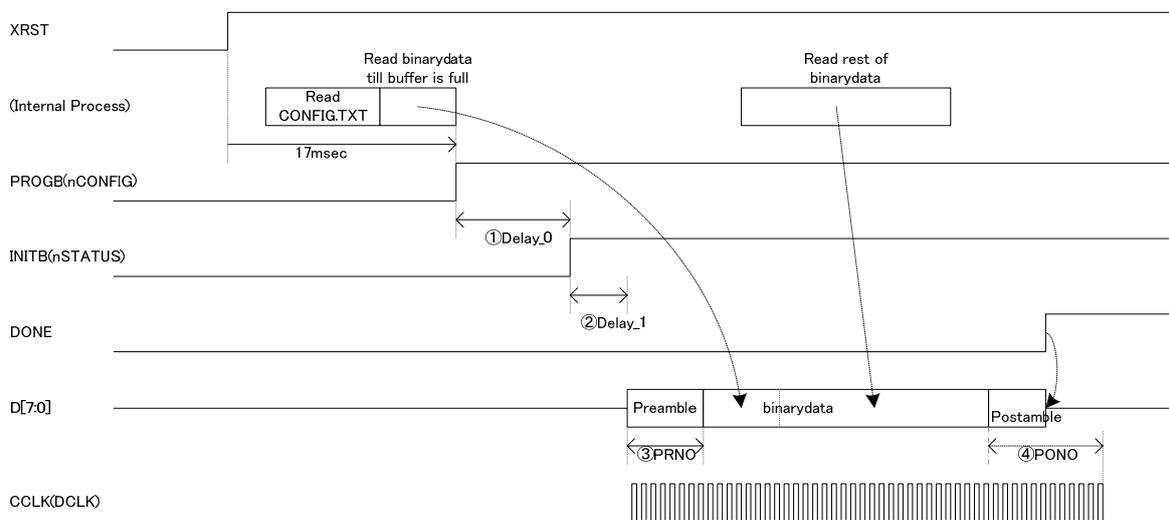
図 5.1.1.4 通常 Slave serial による Multi-FPGA 接続例(差し替え要)

## 6. タイムチャート

### 5.1. 信号出力部ブロック図



### 5.2. タイムチャート



- ①Delay\_0 parameter is set by the command #P : D0 = xxxx\_xxxx      Delay\_0 = D0 \* 20nsec
- ②Delay\_1 parameter is set by the command #P : D1 = xxxx\_xxxx      Delay\_1 = D1 \* 20nsec
- ③PRNO parameter is set by the command #P : PR = xxxx\_xxxx      Insert all "1" PRNO times by using CCLK(DCLK).
- ④PONO parameter is set by the command #P : PO = xxxx\_xxxx      Insert all "1" PONO times by using CCLK(DCLK).

・XRST 信号は 5msec 以上アサートして下さい。また、チャタリングのない信号を入力して下さい。

## 6. LED

- SD-CONF3 にはモジュールの状態を表示するための LED(緑)が実装されています。(写真後日差し替え)

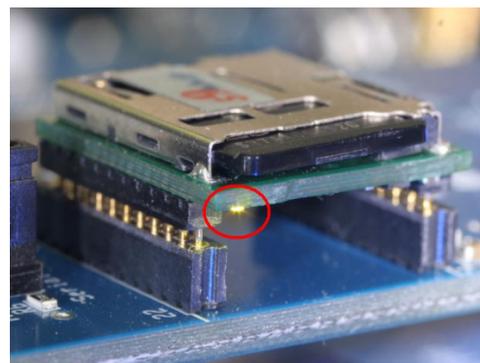


図 6.1 LED の実装位置

### 6.1. LED の点灯条件

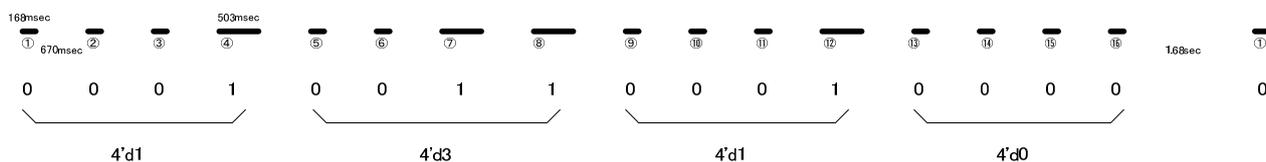
- LED の点灯条件は以下の通りです。

LED 状態	条件
点灯	Slave Serial モードで SD よりデータを転送している間。 Slave SelectMap8 モードで SD よりデータを転送している間。
点滅 167msec 点灯+ 167msec 消灯の 繰り返し。	Slave Serial/SelectMap8 モードで SD のルートディレクトリに“CONFIG.TXT”を認識できなかった場合。 原因: “CONFIG.TXT”ファイルが存在しない。 スペルが間違っている。 “CONFIG.TXT”ファイルが 8 文字以上のキャラクタとして登録されている場合。(SD 上でコピーし、rename した場合など。)
	Slave Serial/SelectMap8 モードで“CONFIG.TXT”によって指定されたバイナリデータがルートディレクトリ上で認識できなかった場合。 原因: バイナリデータが存在しない。 スペルが間違っている。 バイナリデータが 8 文字以上のキャラクタとして登録されている場合。(SD 上で 8 文字以下に rename した場合など。)
無点灯	SD が挿入されていない場合は消灯のままです。
バージョン 情報表示	“CONFIG.TXT”で#P : C9 = 1 が指定されている場合。 上記の SD が挿入されていない場合を除き、モードに関係なく表示されます。

表 6.1 LED 点灯条件

## 6.2. バージョン情報表示機能

- “CONFIG.TXT”で#P: C9 = 1 を指定することにより、本モジュールの裏面に実装されたLED(図 7.1 参照)の点滅により、モジュールのバージョン情報を表示させることができます。
- バージョン情報は、16 ビットからなり、4 ビット毎に分割した 4 桁のヘキサデータとして表現されます。
- モールス信号のように、LED の点灯時間の長短で“0”と“1”の二進情報を表現します。
- 168msec の短い点灯と 670msec の消灯で“0”を表します。
- 503msec の 3 倍長い点灯と 670msec の消灯で“1”を表します。
- 16 回の点灯のあと、1.68sec の消灯区間で一巡したことを意味します。
- 一周目は無意味なデータが入っているため読み飛ばして、二周目から読みだして下さい。
- 図 7.2 は LED の点滅を時間系列上で太線で表した例です。時間は左から右に経過し、太線が点灯中を表し、太線と太線の間空白は消灯区間を表します。
- LED の点滅の長さを見ながら“0”、“1”をメモしていくと、16 個の 2 進の繰り返しとなります。4 個ずつ区切ってバイナリをヘキサに変換すると、“1310”となります。これは現在の最新情報で、バージョンが“1h”、リビジョンが“31h”、サフィックスが“0h”であることを示します。
- バージョン情報は、大規模な変更が行われない限りこの数値が続きます。
- リビジョン情報は、論理の修正があるアップデートが行われるとインクリメントします。
- サフィックス情報は、同一論理において、インタフェース条件の異なるものが存在する場合に割り振られます。



Version 1. 310

図 6.2 LED 点滅例

## 7. “CONFIG.TXT”の準備

### 7.1. “CONFIG.TXT”とは

- SD カードから FPGA へ直接コンフィギュレーションを行う場合、若しくは、SD カードから NandFlash ヘデータを転写する場合に参照されるテキストファイルです。
- ファイル名は 半角アスキーの“CONFIG.TXT”(小文字、大文字小文字混在も可)に固定化されており、SD カードのルートディレクトリに置く必要があります。
- “CONFIG.TXT”ファイルには、ビットファイル名や、各種パラメータを列記します。
- “CONFIG.TXT”ファイルで取り扱えるキャラクタは、半角アスキーコードのみです。
- “CONFIG.TXT”ファイル内のコマンドは、大文字、小文字、両者混在で記載してもかまいません。例えば PO(ピーオー)など、Po と記載して 0(ゼロ)との紛らわしさを避けることができます。漢字やひらがなは使えません。全角の空白の混入には注意して下さい。
- “CONFIG.TXT”ファイルには、下記のコマンドが存在します。

## 7.2. “CONFIG.TXT” サンプル

- SD カードの root ディレクトリに、“CONFIG.TXT”(小文字も可)の名称のファイルを準備する必要があります。

```

// This file is config test           //single' indicates comment out. '/' is familiar to verilog user.
#M : X                               //A: Altera,X:Xilinx, L:Lattice(future option)
#S : 1                               //0:50M(def),1:25M,2:12.5M,3:8.333M,4:6.25M,5:5M
                                       //6:3.125M,7:1.56M,8:0.78M,9:0.39M,F:Active

//Swap parameters
#P : SS = 1                          //Swap bit
##P : SB = 0                         //Swap byte
##P : SW = 0                         //Swap word

//Preamble/Postamble parameters
#P : PR = 0000_0040                 //Preamble insert Number by counting CCLK
#P : Po = 0000_8000                 //Postamble insert Number by counting CCLK

//Delay parameters
#P : D0 = 0000_0100                //delay Number from PROGB to INITB by counting 50MCLK
#P : D1 = 0000_0100                //delay Number from INITB to CCLK by counting 50MCLK.

//Command parameters
#P : C0 = 1                         //WordAligner
#P : C1 = 0                         //nCONFIG,nSTATUS Pump ON
#P : C2 = 0                         //MultiFPGA 2to1 x 4
#P : C3 = 0                         //MultiFPGA 4to1 x 2
##P : C9 = 1                        //Version Information display on LED.

//Binary data area
//TEST_LED.bin                     //If binary file appear without "#n :",
                                       //the binary file will select immediately.
#0 : TESTLED0.bin                   //bin file
#1 : TESTLED1.bin                   //bin file
#2 : TESTLED2.bit + TESTLED3.bit + TESTLED4.bit //+による連結
#3 : TESTLED3.BIT                   //bit file
#4 : TESTLED4.bin
#5 : testled5.bin
...
#F : TESTLEDF.bin
//end

```

図 7.2. “CONFIG.TXT” サンプル

## 7.3. Commands 速見表

第 1op	第 2op	Value	機能詳細
//			コメントアウト
#M:		X	メーカーデフォルト指定
#S:			CCLK 速度選択
#P:	SS=	0/1	シリアル出力 MSB/LSB スワップ
	SB=	0/1	バイト出力 MSB/LSB スワップ
	SW=	0/1	Word,Dword 出力バイトスワップ
	PR=	xxxx_xxxx	プリアンブル長指定
	PO=	xxxx_xxxx	ポストアンブル長指定
	PM=	xxxx_xxxx	File+連結のミドルアンブル長指定
	D0=	xxxx_xxxx	PROGB~INITB までのディレイ間隔
	D1=	xxxx_xxxx	INITB~Data 出力開始ディレイ間隔
	C0=	0/1	ワードアライナ機能
	C1=	0/1	PROGB,INITB Pump ON
	C2=	0/1	P2to1 x 4port on SelectMap8
	C3=	0/1	P4to1 x 2port on SelectMap8
	C4=	0/1	DONE 信号無視
	C5=	0/1	Future use
	C6=	0/1	XCS copy merge detect on
	C7=	0/1	Future use
	C8=	0/1	Future use
	C9=	0/1	Ver 表示
#R:		0~F	リトライ回数
#0:		File1 + file2 +	エリア0ファイル指定
~			エリア 1~E ファイル指定
#F:		File1 + file2 +	エリア F ファイル指定

## 7.4. Commands 詳細

## (a) “/” (“slash”)

- コメントアウトを行います。
- 行中に“/”を検出すると、“CR”までのそれ以降の文字列をコメントとして読み飛ばします。
- サンプルでは、Verlog の慣例に従い、“//”で表記しています。

## (b) #M : A/L/X

- Maker を指定します。デフォルトはありません。
- 本パラメータを指定することにより、7.4 (2)のパラメータが自動で設定されます。
- デフォルトパラメータ値を変更したい場合、本コマンドの後に変更したいパラメータのコマンドを記述して下さい。設定値が上書き修正されます。

## (c) #S : 0/1/2/3/4/5/6/7/8/9/F

- FPGA 側の CCLK(DCLK)の速度を指定します。

0 : 50MHz(デフォルト)	6 : 3.125MHz
1 : 25MHz	7 : 1.56MHz
2 : 12.5MHz	8 : 0.78MHz
3 : 8.333MHz	9 : 0.39MHz
4 : 6.25MHz	A~E : 未定義
5 : 5MHz	

F : Master モードの CCLK は入力となります。(現状未対応)

- これらの値は、ご利用になる SD カードによりばらつきが生じる場合があります。
- また、ファイルの連続性にも起因しますので、ファイルの書き換えを重ねているうちに FAT の連続性が無くなり、オーバーランが発生し、コンフィギュレーションが終了しなくなる場合があります。この場合は、SD カードを FAT16 で再フォーマットして FAT の連続性を再度確保して下さい。
- オーバーランが発生した場合、モジュールの裏に実装されている緑の LED が点滅します。(167msec 点灯+167msec 消灯の繰り返し。)
- Master モードの記述内容は、インプリの段階で変更される場合がありますので、あくまでも参考として下さい。

## (d) #P : SS = 0/1

- ・ シリアルモード転送時(XMODE ピン = High)、D0(SOUT)ピンに最初に出力されるビットの MSB/LSB の出力順番の選択を行います。
  - 0 : MSB ファースト
  - 1 : LSB ファースト(デフォルト)
- ・ 本コマンドはシリアル転送のときのみ有効で、次の#P:SB コマンドの影響は受けません。

## (e) #P : SB = 0/1

- ・ バイト単位内で、MSB と LSB を入れ替えます。  
本モジュールでは、通常、SD カード上のバイナリデータのバイト単位の MSB が D7 ピンに、LSB が D0 ピンに出力されます。  
本パラメータを ON にすると、バイト内で MSB と LSB がスワップされます。
  - 0 : Byte Swap 無し(デフォルト)
  - 1 : Byte Swap 有り
- ・ 本コマンドはパラレル転送のときのみ有効で、前記の #P:SS コマンドには影響を与えません。

## (f) #P : SW = 0/1

- ・ 本モジュールでは、16bit、32bit のバス幅指定ができないため、指定しても機能しません。
- ・ Full バージョンのソリューションでは、MODE[3:0]ピンにより、FPGA のバスを 32bit/16bit/8bit/1bit から選択できますが、32bit バス、16bit バス時に有効な設定で、Word 内を Byte 単位でスワップすることが可能です。
- ・ Xilinx Virtex-4 をコンフィギュレーションする場合に使われるコマンドです。
- ・ FPGA バスが 32bit モードのとき、下記のように選択されます。
  - 0 : Word Swap 無し
  - 1 : [31:24] => [ 7: 0]  
[23:16] => [15: 8]  
[15: 8] => [23:16]  
[ 7: 0] => [31:24]
- ・ FPGA バスが 16bit モードのとき、下記のように選択されます。
  - 0 : Word Swap 無し
  - 1 : [15: 8] => [ 7: 0]  
[ 7: 0] => [15: 8]

## (g) #P : PR = xxxx\_xxxx

- ・ バイナリデータを FPGA に送信する前に、プリアンブルとして Data Bus を All“1”にした状態で、指定回数の CCLK を出力します。
- ・ 指定は 16 進数で行い、8 桁全てを指定する必要があります。  
0000\_0000 から FFFF\_FFFF まで指定できます。
- ・ 16 進数間にアンダーバー“\_”を任意に入れることが可能です。

- ・ 0000\_0000 を指定すると、プリアンブルは出力されません。
- ・ 指定がない場合のデフォルトは 0000\_0000 です。
- ・ Altera の RBF ファイルの先頭にある 32 個の FF についてはデータと認識して出力されますので、本パラメータの指定の有無に係わらず出力されます。

(h) #P : PO = xxxx\_xxxx

- ・ バイナリデータを FPGA に送信した後に、ポストアンブルとして Data Bus を All“1”にした状態で、指定回数の CCLK を出力します。
- ・ 指定は 16 進数で行い、8 桁全てを指定する必要があります。
- ・ 0000\_0000 から FFFF\_FFFF まで指定できます。
- ・ 16 進数間にアンダーバー“\_”を任意に入れることが可能です。
- ・ 0000\_0000 を指定すると、プリアンブルは出力されません。
- ・ FFFF\_FFFF を指定すると、CCLK は停止せずに出力されつづけます。  
この場合、#R コマンドは機能しません。
- ・ 何も指定しないと、デフォルトとして 0000\_1000 が設定されます。
- ・ CCLK は DONE 信号がアクティブになっても出力され続けます。
- ・ Data バスはポストアンブル送信途中でも、DONE 信号がアクティブになると High-Z に開放されません。

(i) #P : D0 = xxxx\_xxxx

- ・ PROG\_B がリリースされて INIT\_B がリリースされるまでのディレイ間隔を指定します。
- ・ 指定は 16 進数で行い、8 桁全てを指定する必要があります。
- ・ 0000\_0000 から FFFF\_FFFF まで指定できます。
- ・ 16 進数間にアンダーバー“\_”を任意に入れることが可能です。
- ・ 設定数値 x 20nsec がディレイ時間となります。
- ・ デフォルトは 0000\_1000 が設定され、約 82usec のディレイ間隔が得られます。

(j) #P : D1 = xxxx\_xxxx

- ・ INIT\_B がリリースされて、送信の許可をモジュールに出すまでのディレイ間隔を指定します。
- ・ 指定は 16 進数で行い、8 桁全てを指定する必要があります。
- ・ 0000\_0000 から FFFF\_FFFF まで指定できます。
- ・ 16 進数間にアンダーバー“\_”を任意に入れることが可能です。
- ・ 設定数値 x 20nsec がディレイ時間となります。
- ・ Pump\_On コマンド( #P:C1 コマンド)が OFF のときの緩やかな立ち上がりを考慮して、デフォルト値として 0000\_0010 が設定されており、20nsec x 16=320nsec 後に CCLK がアクティブになります。
- ・ ワードアライナ機能が有効な場合(#P:C0=1)、有意なデータまで読み飛ばすため、データが出力されるまでにさらに時間を要する場合があります。

(k) #P : C0/1/2/3/4/5/6/7/8/9 = 0/1

- ・ 0 から 9 のコントロールビットが定義されています。

## ① #P: C0 = 0/1 ワードアライナ機能

- ・ 本パラメータに“1”を設定すると、バイナリデータの先頭付近に存在する文字情報を読み飛ばし、文字情報のあとに位置する 32 個の FF によるプリアンブル領域が先頭となるようにワードアライナを行います。
- ・ 先頭の文字情報がなく、32 個の FF によるプリアンブルから始まるデータは、FF の数を損なうことなく出力されます。
- ・ 本パラメータを設定した状態で、32 個の FF(検出は 5 個の FF で実行しています。)が存在しない場合、データは出力されないこととなります。
- ・
- ・ “.bit”ファイルに適用すると、先頭領域の管理情報が削除され、“.bin”ファイルとほぼ同じフォーマットとなります。
- ・ 本機能は、FPGA のデータバスが 16bit,32bit の時には位相補正機能としても働き、Xilinx のように、バス幅を識別させるための識別パターンが存在する場合、先頭に存在する文字情報の長さに関わらず、32bit 単位でしっかりと調整することが可能となります。

## ② #P: C1 = 0/1 PROGB、INITB Pump ON 指定

- ・ 本パラメータに“1”を設定すると、PROGB、INITB 信号を、それぞれ Low からリリースするときに、50MHz 1clk 分(20nsec)High レベルを出力し、その後 High-Z となり、信号の立ち上がりを鋭角にすることが可能です。
- デフォルト(“0”)

## ③ #P: C2 = 0/1

- ・ 本パラメータに“1”を設定すると、SlaveSelectMAP8 モードにおいて、8bit のデータバスを 4 つの区画に分け、それぞれの区画の 2 ビット単位でパラレルシリアル変換して出力されます。

D[1:0] => D[0]

D[3:2] => D[2]

D[5:4] => D[4]

D[7:6] => D[6]

- ・ ビットの出現順番は、#P: SS コマンドに準拠します。(“1”の時 LSB first)
- ・ #P:C2 と#P:C3 コマンドは、C3 コマンドが優先されます。

## ④ #P: C3 = 0/1

- ・ 本パラメータに“1”を設定すると、SlaveSelectMAP8 モードにおいて、8bit のデータバスを 2 つの区画に分け、それぞれの区画の 4 ビット単位でパラレルシリアル変換して出力されます。

D[3:0] => D[0]

D[7:4] => D[4]

- ・ ビットの出現順番は、#P: SS コマンドに準拠します。(“1”の時 LSB first)
- ・ #P:C2 と#P:C3 コマンドは、C3 コマンドが優先されます。

- ⑤ #P : C4 = 0/1
- ・ “1”に設定時、DONE 信号を無視します。
- ⑥ #P : C5 = 0/1
- ・ 未定義(Reserved)
- ⑦ #P : C6 = 0/1
- 0:+コマンドによる連結時の出力。  
1:copy コマンドによる連結時の出力。
- ・ 複数バイナリファイルをコンフィグレーションする場合の XCSI の出力方法の選択を行ないます。
  - ・ DOS プロンプト上で以下のコマンドで連結したバイナリファイルを使用する場合、本パラメータは”1”に設定する必要があります。  
`> copy file1.bin/B+ff4096.bin/B + file2.bin/B (+ ff4096.bin/B + file3.bin/B) mergfile.bin`  
/B はバイナリファイルを意味します。  
ff4096.bin は 4096 バイトの ffh が書き込まれたファイルです。(添付ファイル)  
()内を繰り返すことにより、最大 8 個のバイナリファイルを連結することができます。  
mergfile.bin は連結後のファイル名になります。  
バッチファイルで予め準備しておく、生成を単純化することが可能です。
  - ・ +コマンドによる連結の場合、本パラメータは”0”に設定する必要があります。
- ⑧ #P : C7 = 0/1
- ・ 未定義(Reserved)
- ⑨ #P : C8 = 0/1
- ・ 未定義(Reserved)
- ⑩ #P : C9 = 0/1
- ・ 本コマンドを ON にすると、動作モードに関わらず、モジュールのバージョン情報が LED の点滅により表示されます。
  - ・ 表示は 16 ビット単位で一巡し、長い点灯は“1”を、短い点灯は“0”を意味します。
  - ・ 先頭から 4bit 単位で区切って Hex 変換すると、モジュールのバージョン情報が得られます。
  - ・ 先頭の 4bit がバージョンを、次の 4bit とその次の 4bit の 2 桁でリビジョンを、最後の 4bit でサフィックスを表します。
  - ・ バージョン情報は、大きな変更があった場合に変更されます。
  - ・ リビジョンは、簡易な変更があった場合に変更されます。
  - ・ サフィックスは、同一バージョン、サフィックスにおいて、出力電圧などに違いを持たせた場合に割り付けられます。
  - ・ 表示方法の詳細は 8.3. バージョン情報表示機能の項目をご参照下さい。

## (L) #R : 0~F

- ・ コンフィギュレーション失敗時のリトライ回数を指定します。
- ・ 0 回から 15 回まで指定できます。(デフォルト 0)
- ・ ポストアンプを送信し終わった段階で DONE 信号をモニタし、アクティブになっていない場合コンフィギュレーションが失敗したと判断し、PROGB,INITB を Low に落としてコンフィギュレーションをやり直します。
- ・ INITB によるコンフィギュレーションの失敗の監視は行っていません。

## (m) bitfile-name

- ・ 上記の(a)または(b)に属さないキャラクタで始まる行は、Line-Processor はバイナリデータ名として扱います。(注:Line-Processor はキャラクタを解析するシーケンサ論理)
- ・ バイナリデータは、“.bit”と“.bin”のどちらのサフィックスも指定することができます。
- ・ バイナリデータ名は 8 文字以下である必要があります。(拡張 FAT16 未対応)
- ・ バイナリデータ名の前後にスペースや TAB を含むことができます。(ファイル名の中にスペースや TAB を挿入することはできません。)
- ・ バイナリデータ名を”+”で連結することにより 8 個の FPGA まで CS 制御による Multi-FPGA コンフィギュレーションを行うことが可能です。 bitfile1.bit + bitfile2.bit + bitfile3.bit + bitfile4.bit
- ・ “+”による連結の際、バイナリデータ名と”+”の間には必ず1つ以上の空白を挿入して下さい。
- ・ “+”による連結の際、途中で改行を加えないで下さい。

## (n)#0 ~ #F : bitfile name

- ・ バイナリデータ名を 0 から F の 16 個の 16 進数と関連付けを行います。
- ・ 本コマンドにより関連付けが行われた場合、AREA ロータリーSW の指定エリアと同一の番号の関連付けが存在した場合、関連付けされたバイナリデータをコンフィギュレーションの対象とします。
- ・ #の後には、0~9(30h~39h) , A~F(41h~46h) 若しくは a~f(61h~66h)を置くことができます。
- ・ “.”の後に関連付けをしたいビットファイル名を記述します。
- ・ バイナリデータは、“.bin”と“.bit”のどちらのサフィックスも指定することができます。
- ・ バイナリデータ名は 8 文字以下である必要があります。(拡張 FAT16 未対応)
- ・ バイナリデータ名の前後にスペースや TAB を含むことができます。(ファイル名の中にスペースや TAB を挿入することはできません。)
- ・ バイナリデータ名を”+”で連結することにより 8 個の FPGA まで CS 制御による Multi-FPGA コンフィギュレーションを行うことが可能です。 #0:bitfile1.bit + bitfile2.bit + bitfile3.bit + bitfile4.bit
- ・ “+”による連結の際、バイナリデータ名と”+”の間には必ず1つ以上の空白を挿入して下さい。
- ・ “+”による連結の際、途中で改行を加えないで下さい。

## 7.5. コマンドパラメータ デフォルト値

- ・ “CONFIG.TXT”設定されるパラメータのデフォルト値を示します。

## (1) メーカー指定無しの場合のデフォルト値

Maker Code	Maker Name	デフォルト設定値	備考
指定無	—	#P : SS = 0 (MSB ファースト) #P : SB = 0 (Swap 無し) #P : SW = 0 (Swap 無し) #S : 0 (50MHz Passive モード) #P : PR = 0000_0000 (プリアンブルなし) #P : PO = 0000_1000 (ポストアンブル 4096CLK 挿入) #P : D0 = 0000_1000 (PROGB ~ INITB : 82usec) #P : D1 = 0000_0010 (INITB ~ CCLK : 320nsec) #P : C0 = 0 (ワードアライナ無効) #P : C1 = 0 (Pump ON 無効) #P : C2 ,C3 = 0 (Bus 分割 Multi FPGA mode 無効) #P : C9 = 0 (バージョン表示モード 無効)	

## (2) メーカー指定をした場合のデフォルト値

Maker Code	Maker Name	デフォルト設定値	備考
X	Xilinx	#P : SS = 1 (LSB ファースト) #P : SB = 0 (Swap 無し) #P : SW = 0 (Swap 無し) #S : 0 (50MHz Slave モード) #P : PR = 0000_0100 (プリアンブル 256CLK 挿入) #P : PO = 0001_0000 (ポストアンブル 65536CLK 挿入) #P : D0 = 0008_0000 (PROGB ~ INITB : 10msec) #P : D1 = 0000_0010 (INITB ~ CCLK : 320nsec) #P : C0 = 1 (ワードアライナ有効) #P : C1 = 0 (Pump ON 無効) #P : C2,C3 = 0 (Bus 分割 Multi FPGA mode 無効) #P : C9 = 0 (バージョン表示モード 無効)	

## 8. 機能詳細

### 8.1. ワードアライナ機能

- ・ バイナリデータ上の実質的な送信開始情報である 32 個の FF を検出します。
- ・ FPGA に転送不要な管理情報等を削除します。
- ・ 16bit や 32bit 幅で FPGA に転送するとき、有効データの始まりのバイト位置をダブルワード単位で位相を補正して FPGA に転送します。
- ・ 本機能により、配置配線の結果として出力されるサフィックスが“.bit”のバイナリデータを取り扱うことが可能になります。
- ・ “CONFIG.TXT”の#P : C0 コマンドで“1”を指定することにより有効になります。  
(デフォルト 0)(メーカー指定した場合はデフォルト1)

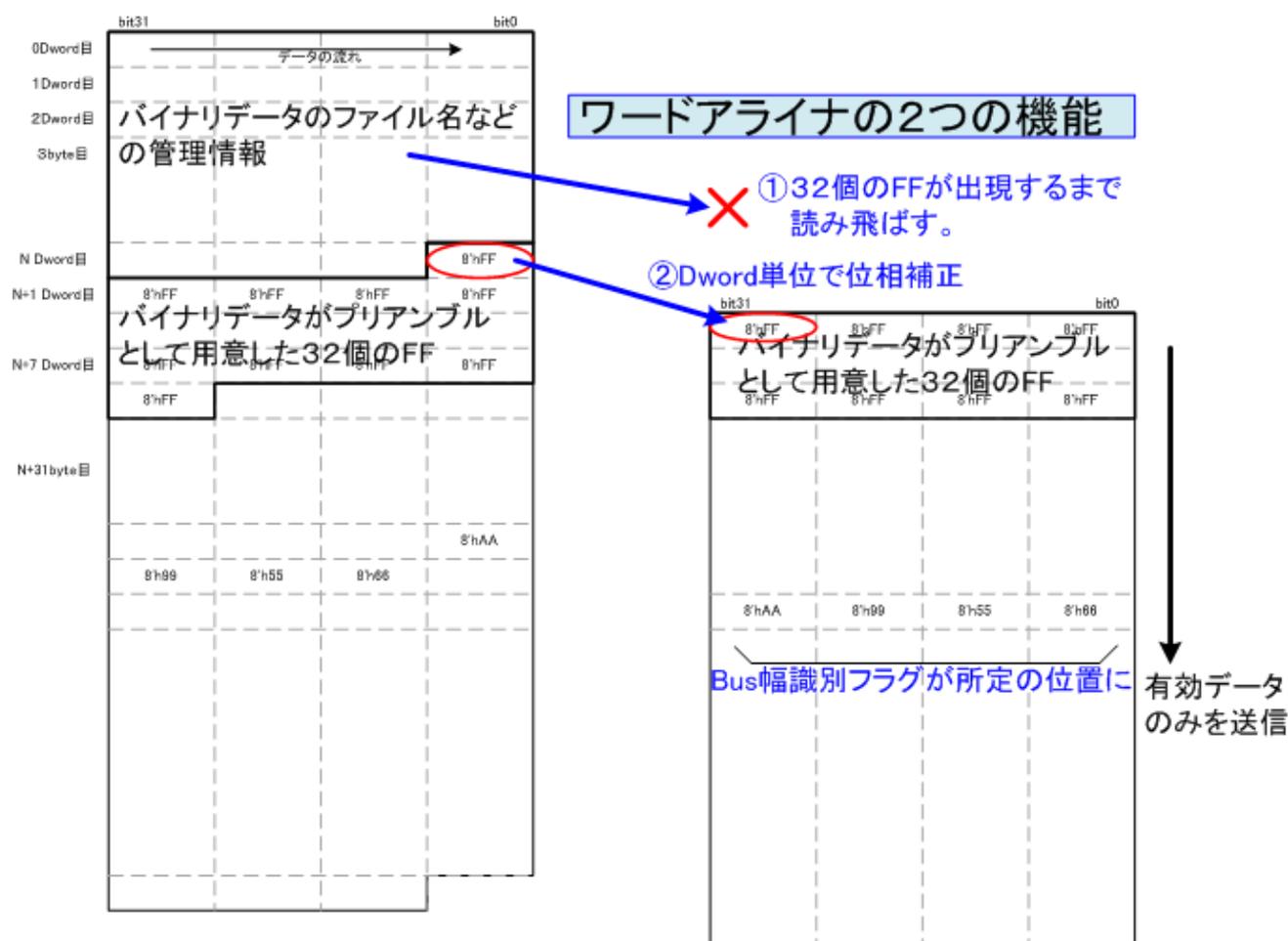


図 8.1. ワードアライナ機能説明図

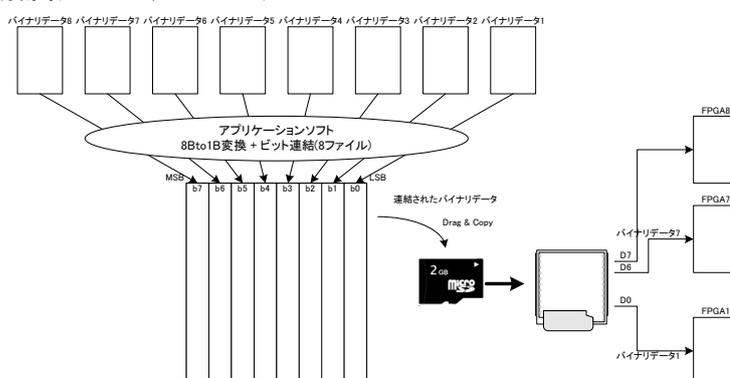
## 8.2. bit 連結による Multi FPGA Configuration 機能

- 添付されるアプリケーションソフトと、コマンドパラメータの設定により、シリアル転送により同時に複数の FPGA をコンフィギュレーションすることが可能です。
- Daisy-Chain 方式による複数の FPGA をコンフィギュレーションする方法とは別の方式になります。
- SD カードの実効読出速度が 200Mbps に対して、50Mbps のシリアル通信を 1 本だけ行うことは非効率で、本方式によりリソースを分割することにより、4 本まではシリアル通信の実効速度を下げることなく転送することが可能になります。

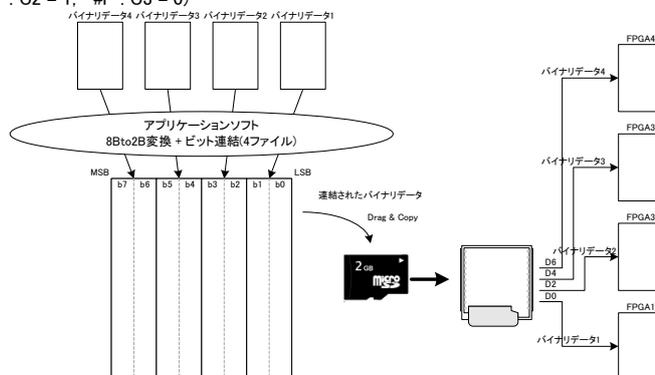
分割個数が 8 個の場合で、FPGA 割り当てに空きがある場合や、それぞれの FPGA のサイズが異なる場合、Daisy-Chain 方式より効率が悪くなる場合があります。

### bit連結によるMulti FPGA Configurationの動作原理

(1) 8分割時(#P : C2 = 0, #P : C3 = 0)



(2) 4分割時(#P : C2 = 1, #P : C3 = 0)



(3) 2分割時(#P : C2 = 0, #P : C3 = 1)

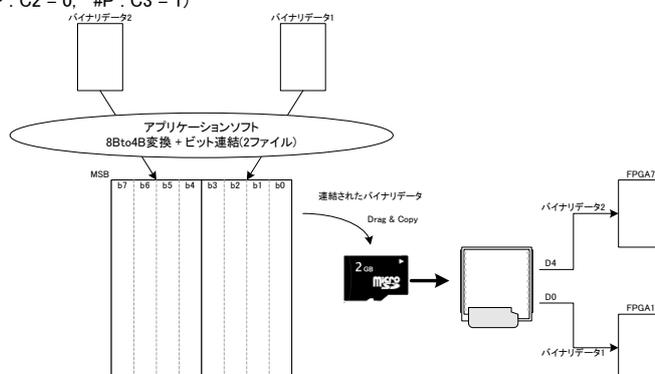


図 8.2. bit 連結による Multi FPGA Configuration の動作原理図

### 8.3. “CONFIG.TXT”による Multi FPGA Configuration 機能

- “CONFIG.TXT”ファイル上で単独バイナリデータを指定する代わりにバイナリデータを順次”+”により連結指定することにより、モジュールは最初のバイナリデータの転送を CS0 をアサートして開始し、転送が終了するとネゲートし、順次 CS 番号をインクリメントしながら該当のバイナリデータを転送することが可能です。

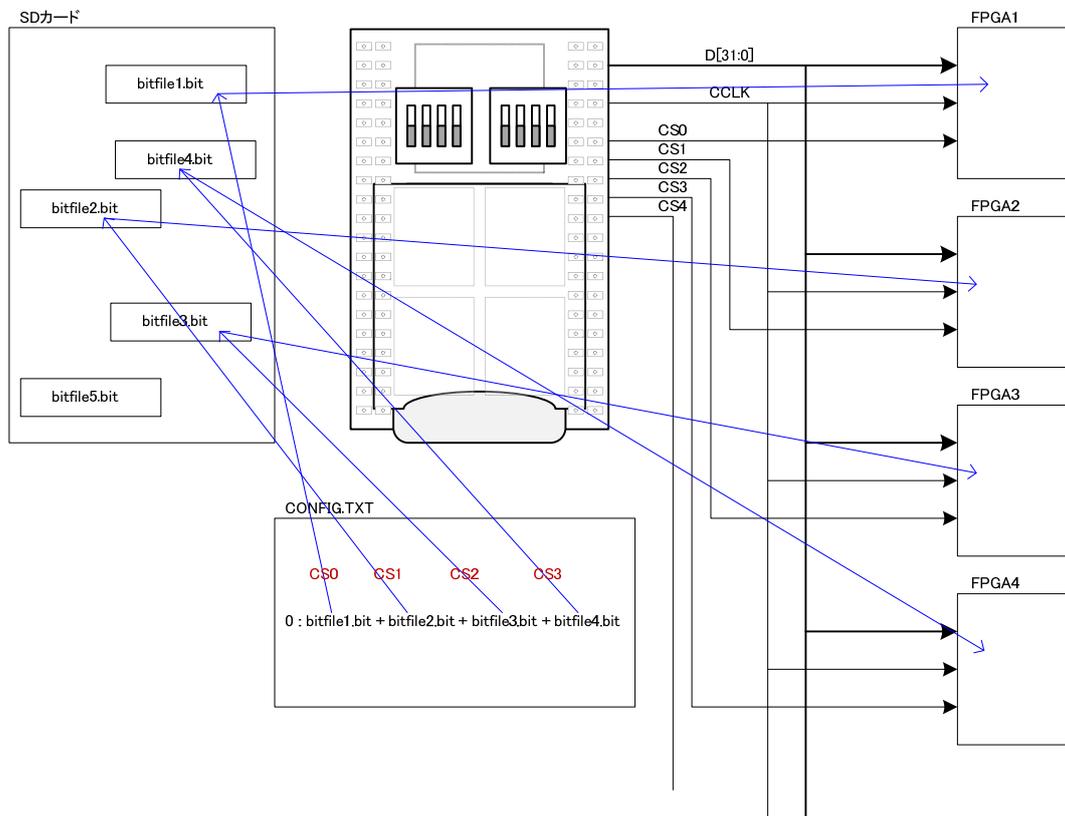
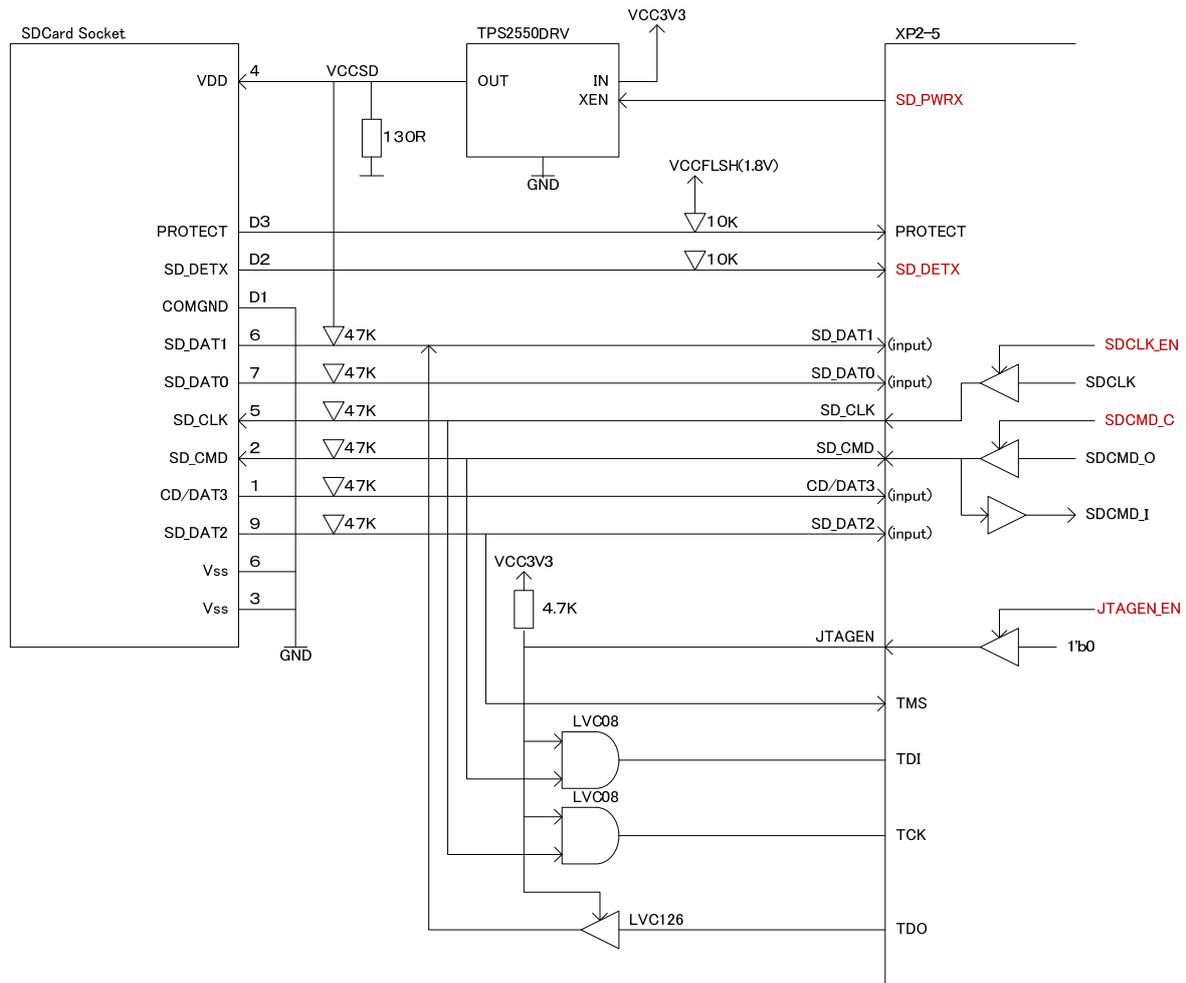


図 8.3. “CONFIG.TXT”による Multi FPGA Configuration の動作原理図(差し替え要)

## SDCard の活線挿抜について



## SDCard の活線挿抜対策

## ① SD カード電源の制御

- XRST 信号、SD\_DET X 信号によって下記の制御を行ないます。

$$\text{assign SD\_PWRX} = \text{XRST} \mid \text{SD\_DET X};$$

- SD\_DET X は、SDCONF モード、TRANS モード時の SDCard へのアクセスの最初の段階で1度だけチェックを行ないます。  
その際、SD\_DET X == 1'b1で未挿入の場合、挿入を検出するまでループします。
- SD\_DET X==1'b0を検出すると、7FFFh x 20nsec = 655usec 経過後、SDCard と通信を開始します。  
SDCard と通信中に SDCard が引き抜かれた場合、SDCONF3は ERROR ステートに遷移します。  
ERROR ステートに遷移した場合、電源を再投入するか XRST をアサートすることにより ERROR ステートから復旧することが可能です。

## ② IO ピンの Tri-State 制御

- SD\_DAT[0:3]については、入力信号のため処理は行なっていません。
- SD\_CMD については、通常入力方向の信号ですが、SDCard の挿抜時に出力に向いている場合を考慮して XRST,SD\_DET X がアサート時、Hi-Z 出力となるよう修正いたします。(0209a は未修正)

- ・ SD\_CLKについては、JTAGポート選択時 Hi-Z 制御となる論理が入っておりますが、XRST,SD\_DET<sub>X</sub>がアサート時も Hi-Z 出力となるよう修正致します。(0209a は未修正)

### ③ JTAGEN 信号の制御

- ・ 電源投入時、リセット立ち上がり時、SD\_DAT0 が Low になっていないかを確認します。  
もし SDCard\_Adapter が挿入されていて、SD\_DAT0 が Low に Pulldown されている場合、本モジュールは SDCard ソケットに JTAG ケーブルが挿入されたと判断し、JTAGEN 信号を Hi-Z とし、外部 4.7K Ω Pullup 抵抗により High に確定します。
- ・ これにより、TDI,TCK が JTAG ポートに入力され、同時に TDO が SD\_DAT1 ラインに出力されます。
- ・ XP2 CPLD に論理が書かれていない状態では、JTAGEN 信号は Hi-Z と認識され、JTAG ポートの接続が確立します。
- ・ XRST 動作時、SDCard が抜かれた状態において、SDCard への電源は停止状態になりますが、JTAG ポートへの SD\_DAT1(TDO)信号が出力されないよう、JTAGEN は Low に固定されます。(0209a 以前は XRST アサート時、JTAG ポート選択となっております。)

## 9. モジュールのソケットからの取り外しについて

- SD-CONF3 はモジュール構成となっているため、ターゲット基板に実装する際にソケット実装することで、システムの評価が終了したあとに取り外して他のシステムに使いまわすことが可能です。
- しかしながら、図 9.1.のように、モジュールの裏側は高密度にチップ部品を実装しており、取扱いに注意して取り外さないと不用意に部品を押しつぶしたり、パターンを剥離してしまいかねません。
- 本章では、モジュールを末永くお使いいただくために、モジュールの裏面の実装の状況を把握していただき、取り外しの注意点等をご紹介します。

### 9.1. モジュール裏面

- 両側の赤枠で囲った部分に、他の部品より背丈のあるブロック片を実装しています。モジュールの取り外しの際には、このブロック片に力が加わるよう取り外して下さい。

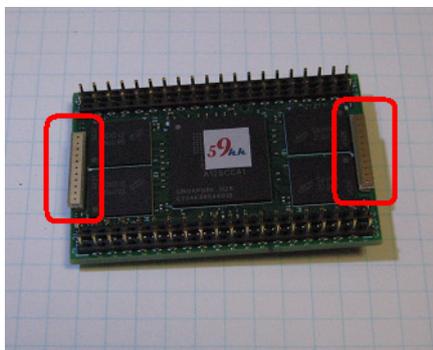


図 9.1. SD-CONF3 Bottom View (差し替え要)

## 9.2. モジュールの取り外し方(差し替え要)

- ・ 付属のコの字型の金具を図 9.2.2 のようにモジュールの下に滑り込ませ、モジュール下の両端のブロック片に交互に力が加わるようテコによって持ち上げるように少しずつ抜き取って下さい。
- ・ 一気に持ち上げると、四隅のうちの一か所だけが持ち上がってしまってピンが曲がる原因になります。

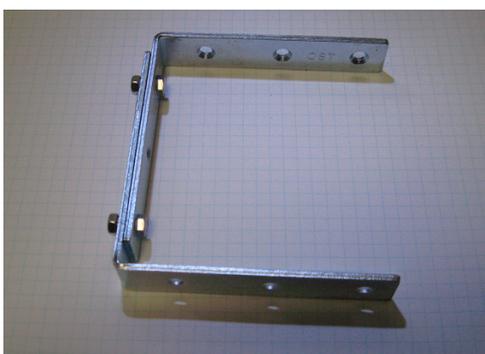


図 9.2.1 取り外し用金具

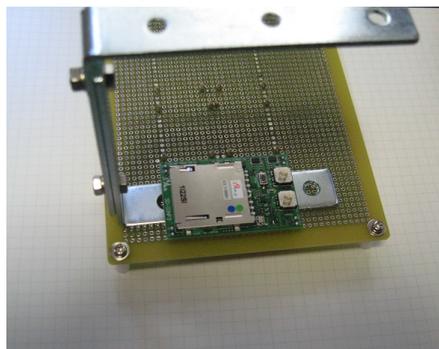


図 9.2.2 金具使用例

## 10. 制約事項

- ① 使用する SD カードは 2G バイト以下の FAT16 でフォーマットされたものをご利用下さい。  
FAT32 でフォーマットされている 2G バイトを超える SDHC 品には対応していません。  
2G バイト以下の SD カードでも、FAT32 で再フォーマットされたものは動作しません。  
再フォーマットを行う場合は FAT16(Microsoft Windows のボリュームのプロパティのファイルシステムでは FAT と表示されます)を選択して下さい。
- ② ビットファイル名は英数字の 8 文字以下にして下さい。  
拡張 FAT16 には対応していません。  
大文字小文字どちらも使用できますが、識別は行いません。  
“\_”(アンダーバー)、“-”(ハイフン)も文字として使用できます。  
尚、8 文字以上のファイルを一旦 SD カードにコピーして、rename によりファイル名を 8 文字以下に修正させた場合、修正後も依然 8 文字以上のファイルとして取り扱われてしまい、先頭の 7 文字しかファイル名の識別要素として機能致しませんので、パソコンでファイル名を修正後、コピーを行って下さい。
- ③ “CONFIG.TXT”ファイルやバイナリデータの SD カードへのコピーは、Windows システムより行って下さい。  
Linux や Unix システム上で SD カードにコピーすると、正しく動作しません。  
(Linux や Unix 上で生成されたバイナリデータを Windows システムを介してコピーする分には問題ありません。)
- ④ “CONFIG.TXT”のリスト中、先頭から 512 バイトまでが認識されます。  
512 バイトを超えたリストは現状認識されません。  
(本制約は解除されました。ファイルサイズに関係なく、自由に設定いただけます。)
- ⑤ uSDCard と通信を行っている最中に XRST のアサートなどによって通信を中断させると、uSDCard は INACT 状態に遷移します。この状態には出口は無く、この状態に遷移すると、uSDCard の電源を再立ち上げしないかぎり、この状態から脱出することができません。  
この状態に陥った場合、システムの電源を再立ち上げするか、uSDCard を挿抜して復旧させて下さい。  
(本制約は解除されました。電源を再立ち上げしなくても、リセットにより復旧が可能です。)
- ⑥ 本モジュールでは、ハードウェアの簡略化のため、uSDCard のパケットの送信単位である 512byte 単位のデータの区切りとしてビットファイルを扱っており、ファイルの最後の 512byte で割り切れないデータに対しては、uSDCard から読みだされたデータをそのまま FPGA に送信しています。  
FPGA 内では、本来 CRC により、受信データの正統性を確認して立ち上がるはずですが、Virtex5 の SlaveSerial 転送において、このままでは正常にコンフィギュレーションが終了しません。  
このため、下記の手順によりファイルの最後に all“1”のデータを 4096byte 付加していただくようお願い致します。  
本制約事項につきましては、近々にバージョンアップにて対処致します。
- Bit(bin)ファイルの存在するディレクトリに ff4096.bin と bitecopy.bat をコピーする。  
Bitecopy.bat をテキストエディタで開き、ソースファイルとディスティネーションファイル名を所定の名前に変更する。  
—————copy source.bin/b + ff4096.bin/b dist.bin  
(bat ファイル内の source.bin と dist.bin のファイル名を変更する。)  
Bitecopy.bat をダブルクリックする。

(本制約事項は解除されました。16bit モード、32bit モードにおいてワードの途中でファイルが終了する場合は、FF を挿入します。)

- ⑦ Virtex4 の SelectMap では、バイト内で MSB/LSB 間で bit スワップを行う必要があります。  
uSDCONF Solution では SEL\_V4 という端子に論理を与えることにより対応していましたが、本モジュールにおきましては、Pin 数が限られていることから、暫定的に“CONFIG.TXT”の未使用のコマンドに SEL\_V4 の論理レベルを割り振っています。
- #W:D
- 本コマンドは本来 FPGA のデータサイズがダブルワード(32bit)である定義ですが、本モジュールでは上記コマンドを記述すると、SEL\_V4 が内部的に ON になります。
- (本制約事項は解除されました。#P:SS,#P:SB,#P:SW を使って指定することが可能です。)
- ⑧ bit 連結による Multi FPGA Configuration 機能はインプリメントされていますが、バイナリデータを連結するためのアプリケーションソフトを準備中です。
- ⑨ CCLK のクロック速度選択機能で、CCLK を受信して動作するマスタモードは定義されていますが、現在インプリメントされていません。

現状、下記の制約事項が存在します。

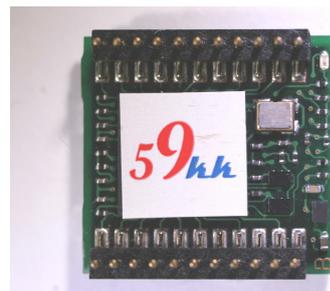
- ⑩ NandFlash の 64 ビット化が組み込まれていません。
- ⑪ “+”連結による NandFlash への転写、NandFlash からのコンフィグレーション機能が実装されていません。  
現状、copy による連結で行なう必要があります。  
SDCard からの“+”連結によるコンフィグレーションは動作しています。
- ⑫ ECC によるフォーマット機能が組み込まれていません。(現状 uSDCONF2 と同じ仕様になっています。)
- ⑬ “+”連結によるコンフィグレーション(現状は SDCard のみ)において、ファイル間の CCLK の挿入個数を指定する MDNO コマンドが機能していません。現状、前のバイナリデータを出力し終わって次のバイナリデータを出力するまでの間、ALL”1”データと CCLK が常に出力されます。

## 11. アップデート機能(順次対応予定)

“59kk”のシールの貼られたモジュールは、128ビットの暗号 Key によって Encryption がかけられています。msd-Adapter と JTag ケーブルをご用意いただくことにより、同じ暗号 Key によって暗号化された bit ファイルを悟空株式会社のホームページよりダウンロードしてアップデートしていただくことが可能です。

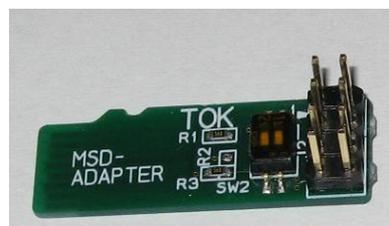
“59kk”のシールの貼られていないモジュールにつきましては、暗号 Key の書き込みが必要です。弊社もしくは担当営業にご相談下さい。

(JTag ケーブルは、悟空株式会社にも販売する予定です。)



今後、下記のようなアップデートを予定しています。

- 機能改版(一部有償)
  - ◇ FAT32 対応
  - ◇ SDHC 対応
  - ◇ コンフィギュレーション終了後の SD カードへのインタフェース機能
  - ◇ SPI(Master/Slave)終端機能
  - ◇ JTag 終端機能
    - JTag からのコンフィギュレーション
    - ROM データの書き換え
    - 別の uSDCONF の書き換え
  - ◇ など
  - ◇ 機能によっては排他的な組み込みになる場合があります。



以上

### お問い合わせ等連絡先

悟空株式会社

担当 : 大庭(オオバ)

〒223-0057

横浜市港北区新横浜 1-19-3 アドホックビル 4F-E

FreeCall : 0800-7775559 (平日 9:00~18:00)

電話 : 045-590-6227 Fax : 050-3156-1404

Email : [info01@59kk.jp](mailto:info01@59kk.jp) URL : <https://www.59kk.jp>