

uSDCONF2A 仕様書  
(for XILINX ユーザ)  
(uSDCONF2A8-001-xxV)

Rev0.41 2014.03.03 (ROM Ver1.803 対応)

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)
2014.03.03	大庭	0.41	uSDCONF2A_DS_A_140303aV1803.doc をベースに全面変更。

## 目次

1. 機能・特徴 .....	5
2. 形状 .....	8
電気特性 .....	9
3.1. 絶対定格 .....	9
3.2. 推奨動作範囲 .....	9
3.3. 消費電流 .....	9
3.4. 電源シーケンス .....	9
3.5. リセットに関して .....	9
4. Pin 機能表 .....	10
4.1. uSDCONF2A ピン配置図(Top View) .....	10
4.2. USDCNF-MD3-001A ピン機能表 .....	11
4.3. XMODE 詳細 .....	12
4.4. XAREA 詳細 .....	13
5. 接続例 .....	14
5.1. Slave Serial/SelectMap8 接続(1 : 1 接続) .....	14
5.2. Slave Serial/SelectMap8 接続(1 : N 接続) .....	15
5.2.1. XCS による Multi-FPGA 接続構成 .....	15
5.2.2. CSO_B, CSI_B 伝播による Multi-FPGA 接続構成(selectMAP8) .....	16
5.2.3. DOUT, DIN 伝播による Multi-FPGA 接続構成(SlaveSerial) .....	17
5.2.4. bit 多重方式による Multi-FPGA 接続構成 .....	18
6. タイムチャート .....	21
6.1. 信号出力部ブロック図 .....	21
6.2. タイムチャート .....	22
6.2.1. SD カードから BinaryData を 1 個読み出し Config する場合 .....	22
6.2.2. NandFlash から BinaryData を 1 個読み出し Config する場合 .....	23
6.2.3. SD カードから BinaryData を 3 個読み出し転送の場合 .....	24
7. LED .....	26
7.1. LED の実装位置 .....	26
7.2. LED の点灯条件 .....	26
7.3. バージョン情報表示機能 .....	27
8. “CONFIG.TXT”の準備 .....	28
8.1. “CONFIG.TXT”とは .....	28
8.2. “CONFIG.TXT” サンプル .....	29
8.3. Commands 速見表 .....	30
8.4. Commands 詳細 .....	31

8.5.	D1,D2,PO の設定について .....	39
8.6.	コマンドパラメータ デフォルト値.....	40
9.	機能詳細.....	41
9.1.	ワードアライナ機能.....	41
9.2.	bit 連結による Multi FPGA Configuration 機能.....	42
9.3.	“CONFIG.TXT”による Multi FPGA Configuration 機能 .....	43
9.4.	SDCard の活線挿抜について.....	44
9.4.1.	構成図 .....	44
9.4.2.	SDCard の活線挿抜対策.....	45
10.	モジュールのソケットからの取り外しについて.....	46
10.1.	モジュール裏面 .....	46
10.2.	モジュールの取り外し方 .....	47
11.	添付品.....	48
12.	VCCIO 電圧とシールの色 .....	49
13.	発注型格 .....	49
14.	制約事項 .....	50
15.	アップデート機能(順次対応予定) .....	51

## 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グ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グレーションを行う機能を有します。
- ・ microSD カードには、最大 2GB までの FAT16 でフォーマットされたカードを使用します。  
(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。)
- ・ バイナリデータや制御ファイルの"CONFIG.TXT"は、microSD カードのルートディレクトリに書き込む必要があります。このため、最大で書き込めるファイルの数は512個に制限されます。  
この個数の範囲内であれば、microSD カードの容量を超えない範囲で、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。
- ・ microSD 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ます。これにより、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。
- ・ 本モジュールは、FAT16 を認識するため、microSD カードへのバイナリデータの書き込みは、Windows システムの Explorer 上で Drag & Copy するだけで簡単に行えます。専用のソフトを起動したり、JTag ケーブルを接続する煩わしさもなく、パソコンの基本操作だけで、誰にでも扱うことができます。
- ・ 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、コンフィグレーションに関する各種パラメータを指定します。
- ・ 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 を直接コンフィグレーションすることが可能です。
- ・ 本モジュールは 4Gbit 16bit 幅 NandFlash を 2 個実装し、1G バイト 32bit 幅の NandFlash 空間を有します。
- ・ 本モジュールは、この NandFlash の1Gバイトの空間を、簡易的な FAT で管理し、出荷前の Bad Block や、運用中に発生した ECC エラーを含む Block を管理し、隔離します。
- ・ ECC は512バイト単位に計算され、4bit までのビットエラーを訂正し、FPGA に転送します。  
このとき、訂正可能なエラーがあったことを、転送中の LED の点灯に続いて、1回だけブリンクさせます。  
また、エラーの発生したブロックに対してエラーが発生した履歴を保存します。  
保存されたエラー履歴は、再度フォーマットを行ったときに、隔離の対象として使われます。  
ブリンクが頻発するようになった場合、早めの再フォーマットを推奨します。
- ・ ECC エラーが 512 バイトに対して 5bit 以上であった場合、エラー訂正が行われなため、FPGA に対してエラーのあるバイナリデータを転送します。FPGA ではCRCエラーとして処理されます。  
このとき、訂正可能なエラーがあったことを、転送中の LED の点灯に続いて、4回ブリンクさせることにより通知され、このときは DONE 信号も立ちません。  
また、エラーの発生したブロックに対して 5bit 以上のエラーが発生した履歴を保存します。  
保存されたエラー履歴は、再度フォーマットされたときに、隔離の対象として使われます。

4回ブリンクが発生してコンフィグレーションが正常に完了しなかった場合、直ちに再フォーマットすることを推奨します。

- 本モジュールは、NandFlash の1G バイトの空間のうち、880M バイトを16個の55M バイトのエリアに分割し、AREA[3:0]ピンによりエリア番号を指定し、残りの空間で、ECCエラー検出後の再フォーマット時の隔離ブロックの補完を行います。

尚、単体、+連結で55M バイトを超えるターゲット環境に関しては、分割数を減らしたモジュールやバイナリデータをご提供致します。(論理データの書き換えでの対応になります。)

NandFlash に書き込むデータのサイズについては、予め把握しておく必要があります。

NandFlash への転写モード時、分割エリアサイズを超えるバイナリデータに対しては、次のエリアの上書きが行われてしまい、ECC の大量発生原因となってしまうため、注意が必要です。(次回 Ver1.804 以降の ROM では、エリアを越える転写に対して、プロテクト機能が働き、エリアを跨いでの転写を行いません。また、エリア内にデータが収まりきらなかったことを LED の点滅によって知らせます。

- NandFlash へのバイナリデータの書き込みは、転写モードによって行います。  
転写は、AREA[3:0]ピンで指定されたエリア情報と、“CONFIG.TXT”よりエリア情報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。
- MODE[3:0]ピンにより NandFlash から FPGA へのコンフィグ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 への高速コンフィグ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、

$50\text{MHz}/2\text{clk} \times 32\text{bit} = \text{約 } 800\text{Mbps}$  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。) また、FPGA インタフェース部は、50MHz 8 ビットでインタフェースしているため、実際の転送速度は Max50MByte/sが上限となります。

- 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(#P:C0=1)を“CONFIG.TXT”上で有効にすることにより、被転送ファイルの先頭にある管理情報を読み飛ばし、データバイトが FFh が連続する領域まで読み飛ばして、最初の FFh から転送を開始することが可能です。

XILINX の先頭の管理情報を含む".bit"のサフィックスのバイナリデータに対して有効です。

ワードアライナの目的は、管理情報を読み飛ばして、ワード、ダブルワード単位で先頭の位相を揃えることにあり、バイト単位のアクセスではどちらに設定しても影響はありません。(デフォルトは"1"となっています。)

- 0 から 15 回のコンフィグ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プ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プ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(D2 パラメータ)や、INITB の立ち上がりから

CCLK の出力開始までの間隔(D1 パラメータ)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 “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.bit + data2.bit + ... 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 nCS 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。

外付け回路なしで、最大4個までの FPGA を制御することが可能です。

また、外付け回路を追加することにより、最大8個までの FPGA を制御可能になります。

Xilinx の SlaveSerial では CSI が機能しないため XCS 制御が行えず、動作しません。

② 従来の CS0/CSI のデ이지チェーンによる Multi-FPGA コンフィギュレーション

iMPACT プログラムによるファイルの連結が必要です。

①のSlaveSerialでの問題点は、この構成で対処する必要があります。

③ 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 SlaveSerial 接続するモードです。

(ビッ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。)

①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。)

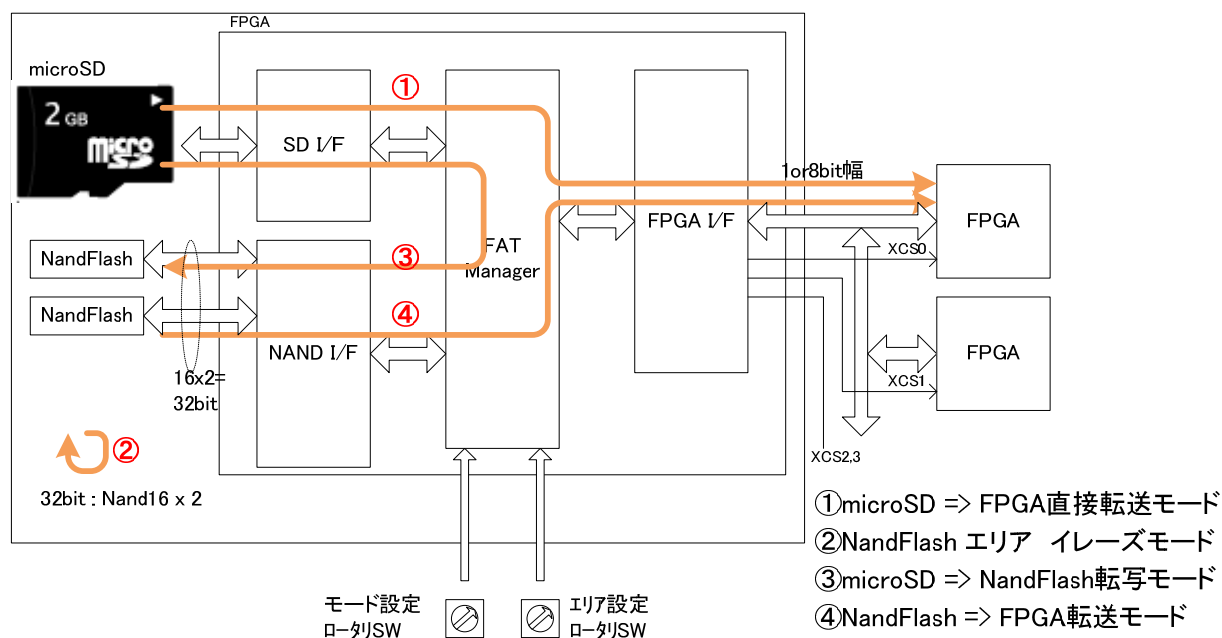


図 1.1 uSDCONF2A ブロックイメージ図

## 2. 形状

uSDCONF2A モジュールの形状を掲載します。

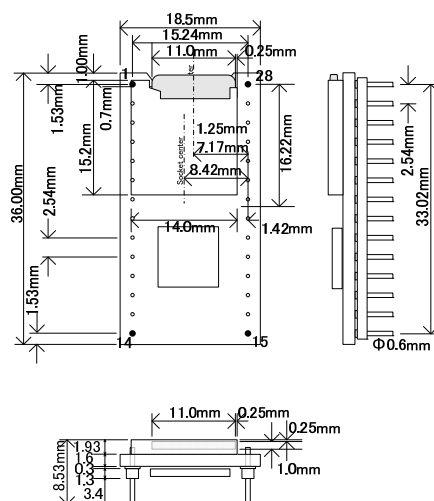


図 2.1 uSDCONF2A 形状図

- ・ uSDCONF2A 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、0.6mmφ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。

多くのソケットが、0.512mm のインチ径までが多いのでご注意ください。

- (1) AR 28 HZL/7-TT (ASSMANN WSW) (RS 品番 674-2498)  
(<http://docs-asia.origin.electrocomponents.com/webdocs/0da4/0900766b80da42ae.pdf>)
- (2) IC26-2806-GG4 (山一電機)  
(<http://www.yamaichi.co.jp/products/picsocket/ic26dip/pdf/ic26dip.pdf>)

- ・ 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 近い力をかける必要があります。  
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 の IC EXTRACTOR(GX-7)用のヘッドを有償でご用意しています。



## 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。

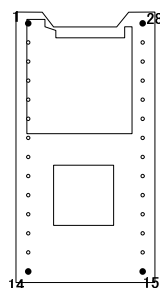
### 3.5. リセットに関して

- ・ リセットは Low でアサートされる負論理入力です。
- ・ FPGA 内部で、電源投入から立ち上がったときに、それを検出して、FPGA 内部の論理にリセットパルスを発行する機能を搭載しているため、リセット信号がなくても動作します。
- ・ しかしながら、モジュールに対して、明示的にリセット信号を入れていただくことを推奨しており、リセット時間は、搭載している FPGA が電源安定から約 8msec で動作可能状態となるため、電源が安定してから 10msec 以上のリセット信号を入力することを推奨しています。

#### 4. Pin 機能表

uSDCONF2A モジュールのピン機能表を掲載します。

##### 4.1. uSDCONF2A 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 選択用ロータリー SW 入力(LSB)
4	XAREA1	I	4.7kΩ Pullup	不要	Bitdata 選択用ロータリー SW 入力
5	XAREA2	I	4.7kΩ Pullup	不要	Bitdata 選択用ロータリー SW 入力
6	XAREA3	I	4.7kΩ Pullup	不要	Bitdata 選択用ロータリー 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	I	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 uSDCONF2A 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グ 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 上のコンフィグ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グレーションが 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。 FPGA の JTAG と接続したい場合に選択します。

#### 4.4. XAREA 詳細

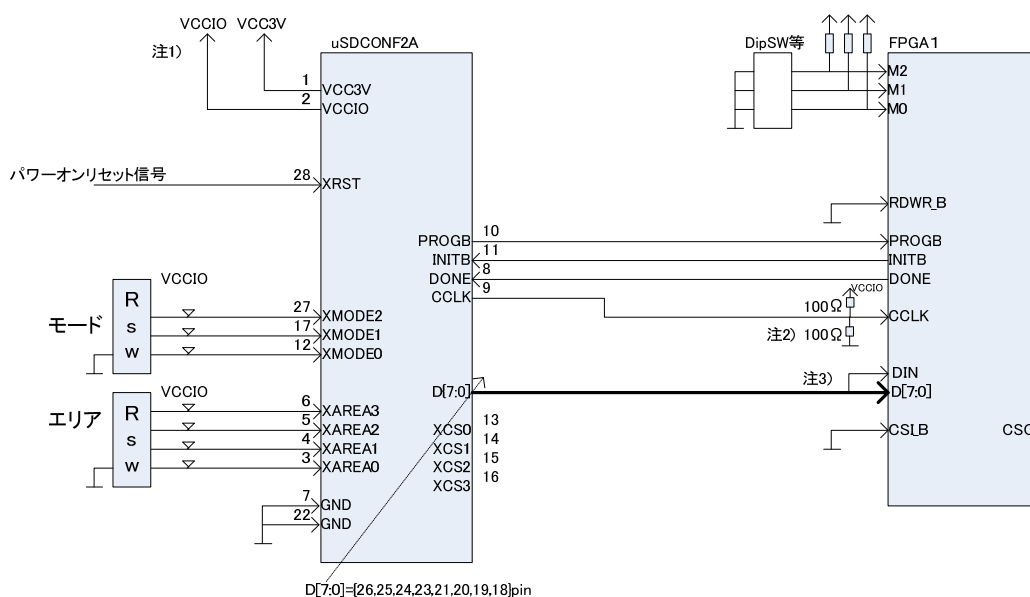
- ・ “CONFIG.TXT”ファイル上でSDカード上に格納されたバイナリデータとAREA[3:0]ピンによって指し示される0～Fの16個の16進数文字と関連付けがなされます。
- ・ XAREA[3:0]入力は負論理で入力され、正論理に反転されたAREA[3:0]信号と“CONFIG.TXT”上の16進数文字と比較が行われ、一致した16進文字と関連付けられたバイナリデータをコンフィグレーションデータとして選択します。
- ・ 表4.4は負論理で入力されたXAREA[3:0]の信号レベルと、それによって指示されるエリア番号の対応表です。
- ・ 外付けのロータリーSWとしてリアルコードタイプを実装すれば、ロータリーSWの選択番号とエリア番号が一致します。
- ・ ロータリーSWの代わりにCPUから制御することも可能です。

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. 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。

FPGAのシリーズによってユーザガイドの終端方法の推奨が異なる場合があります。

その場合は、ユーザガイドの指定を優先して下さい。

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。

## 5.2. Slave Serial/SelectMap8 接続(1 : N 接続)

### 5.2.1. XCS による Multi-FPGA 接続構成

- ・ SelectMAP8 への適用となります。
- ・ SlaveSerial モードでは CSI\_B が機能しないため、本構成は適用できません。

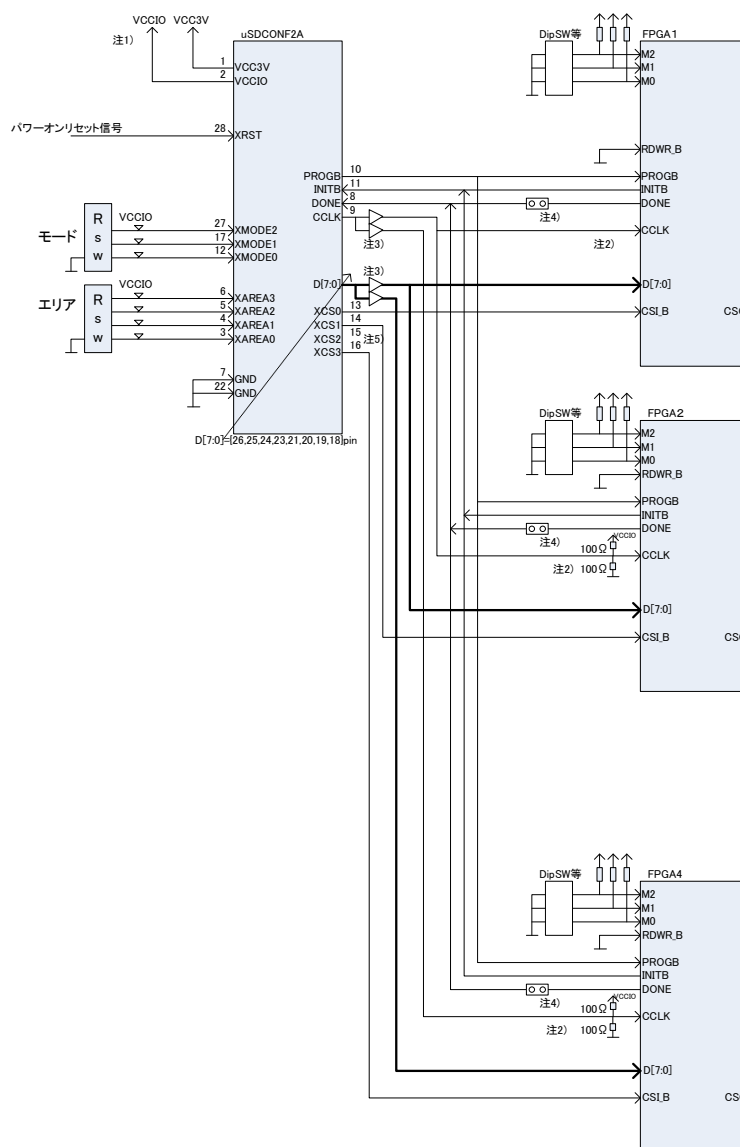


図 5.2.1 SelectMAP8 接続例(1:N 接続)

- 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。FPGA のシリーズによってユーザガイドの終端方法の推奨が異なる場合があります。  
その場合は、ユーザガイドの指定を優先して下さい。
- 注3) CCLK と D[7:0] については、Fanout を 2 に抑えて下さい。
- 注4) デバッグ等で、立ち上げない FPGA が存在する場合、CONF\_DONE 信号を切り離せるようにしておく必要があります。
- 注5) #P : C7 コマンドと、図 8-4 の外付け回路との併用で、XCS を 8 本まで拡張可能です。

### 5.2.2. CSO\_B,CSI\_B 伝播による Multi-FPGA 接続構成(selectMAP8)

- ・ iMPACT でファイルを連結する必要があります。
- ・ SlaveSerial では CSI\_B が無効なため、動作しません。

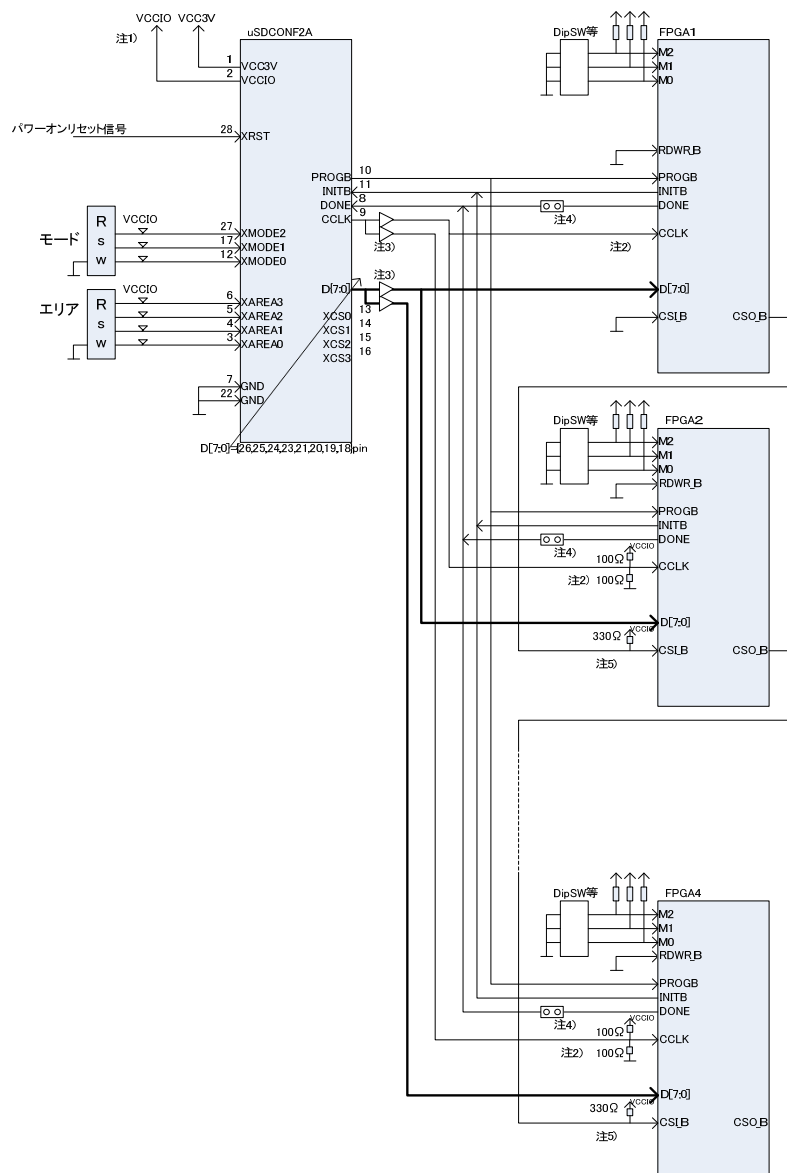


図 5.2.2 CSO\_B,CSI\_B 伝播による接続例(1 : N 接続)

- 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 と接続されるときは、最遠端の FPGA の近傍に配置して下さい。FPGA のシリーズによってユーザガイドの終端方法の推奨が異なる場合があります。  
その場合は、ユーザガイドの指定を優先して下さい。
- 注3) CCLK と D[7:0] については、Fanout を 2 に抑えて下さい。
- 注4) デバッグ等で、立ち上げない FPGA が存在する場合、CONF\_DONE 信号を切り離せるようにしておく必要があります。切り離せるのは最遠端の FPGA からになります。
- 注5) CSO\_B 出力が OpenDrain のため、PullUp 抵抗が必要です。



## 5.2.3. DOUT,DIN 伝播による Multi-FPGA 接続構成(SlaveSerial)

- ・ iMPACT でファイルを連結する必要があります。

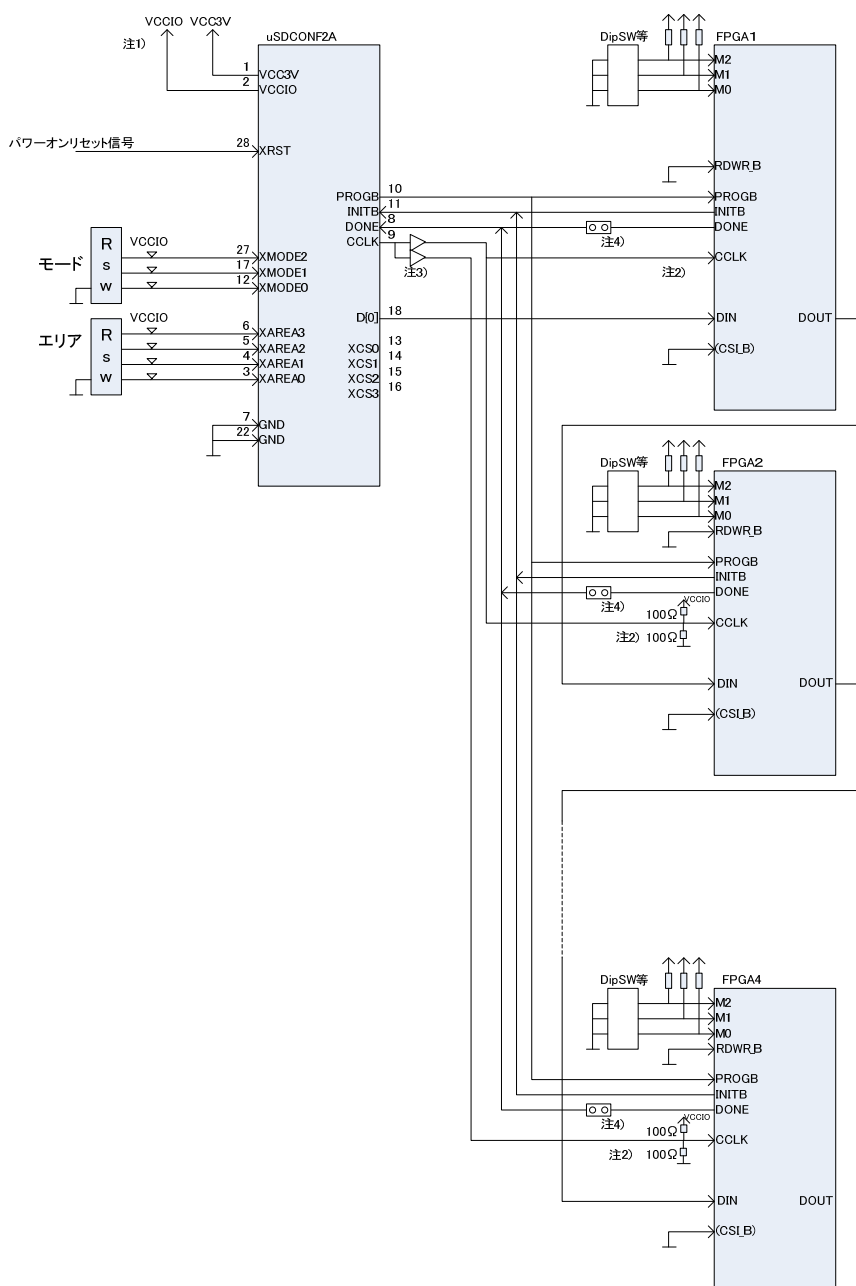


図 5.2.3 DOUT,DIN 伝播による接続例(1 : N 接続)

- 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 と接続されるときは、最遠端の FPGA の近傍に配置して下さい。FPGA のシリーズによってユーザガイドの終端方法の推奨が異なる場合があります。  
その場合は、ユーザガイドの指定を優先して下さい。
- 注3) CCLK と D[7:0]については、Fanout を2に抑えて下さい。
- 注4) デバッグ等で、立ち上げない FPGA が存在する場合、CONF\_DONE 信号を切り離せるようにしておく必要があります。切り離せるのは最遠端の FPGA からになります。

## 5.2.4. bit 多重方式による Multi-FPGA 接続構成

### 5.2.4.1. 8 多重の場合

- ・ バイナリデータを、ビット列で 8 本を多重した合成バイナリデータを準備することにより、下記の構成を組むことができます。
- ・ #P:C2,#P:C3 コマンドは共に"0"になります。

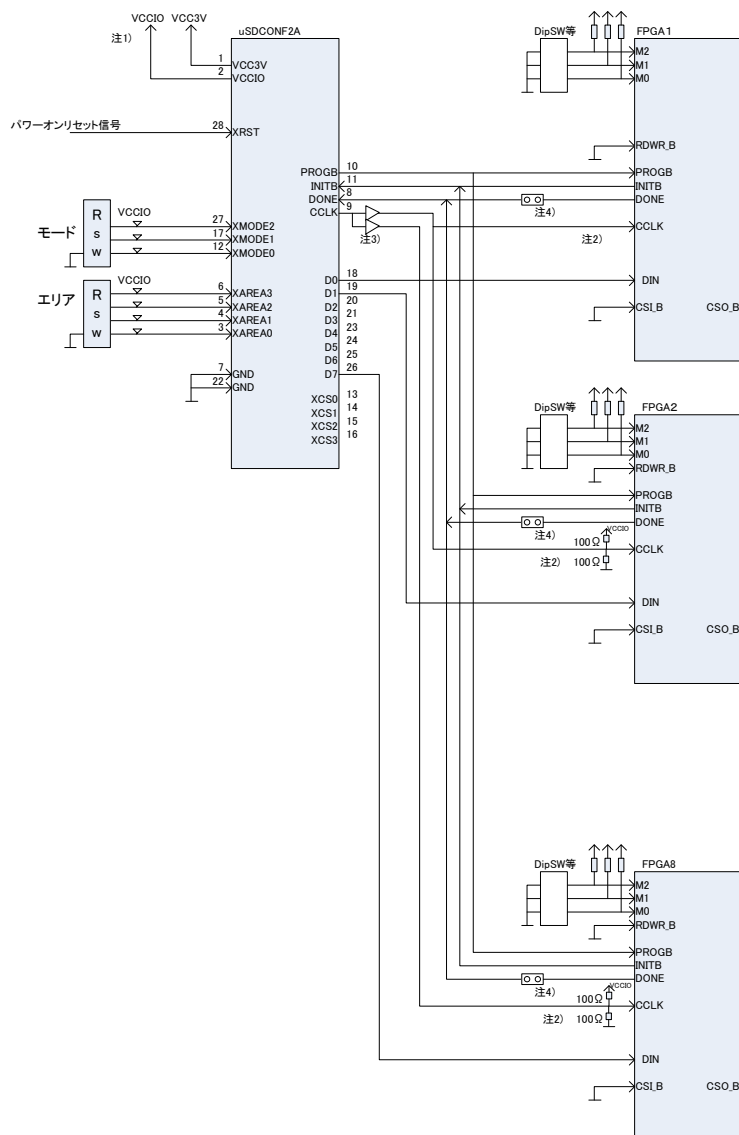


図 5.2.4.1 bit 多重(8 多重)による構成例

- 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 と接続されるときは、最遠端の FPGA の近傍に配置して下さい。FPGA のシリーズによってユーザガイドの終端方法の推奨が異なる場合があります。  
その場合は、ユーザガイドの指定を優先して下さい。
- 注3) CCLK と D[7:0]については、Fanout を 2 に抑えて下さい。
- 注4) デバッグ等で、立ち上げない FPGA が存在する場合、CONF\_DONE 信号を切り離せるようにしておく必要があります。

#### 5.2.4.2. 4 多重の場合

- ・ バイナリデータを、2 ビット単位で 4 列多重した合成バイナリデータを準備することにより、下記の構成を組むことができます。
- ・ config.txt で、#P:C2=1 に設定する必要があります。

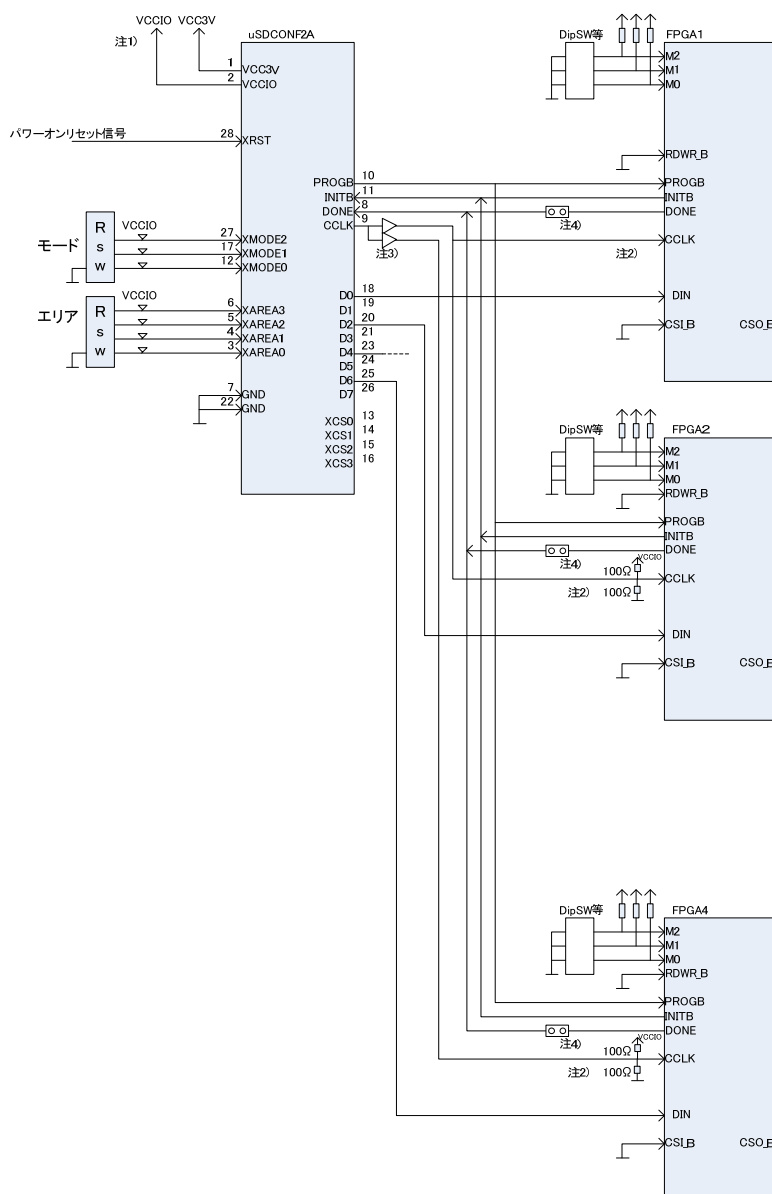


図 5.2.4.2 bit 多重(4多重)による構成例

- 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 と接続されるときは、最遠端の FPGA の近傍に配置して下さい。FPGA のシリーズによってユーザガイドの終端方法の推奨が異なる場合があります。  
その場合は、ユーザガイドの指定を優先して下さい。
- 注3) CCLK と D[7:0] については、Fanout を 2 に抑えて下さい。
- 注4) デバッグ等で、立ち上げない FPGA が存在する場合、CONF\_DONE 信号を切り離せるようにしておく必要があります。

### 5.2.4.3. 4 多重の場合

- ・ バイナリデータを、4 ビット単位で2 列多重した合成バイナリデータを準備することにより、下記の構成を組むことができます。
- ・ config.txt で、#P:C3=1 に設定する必要があります。

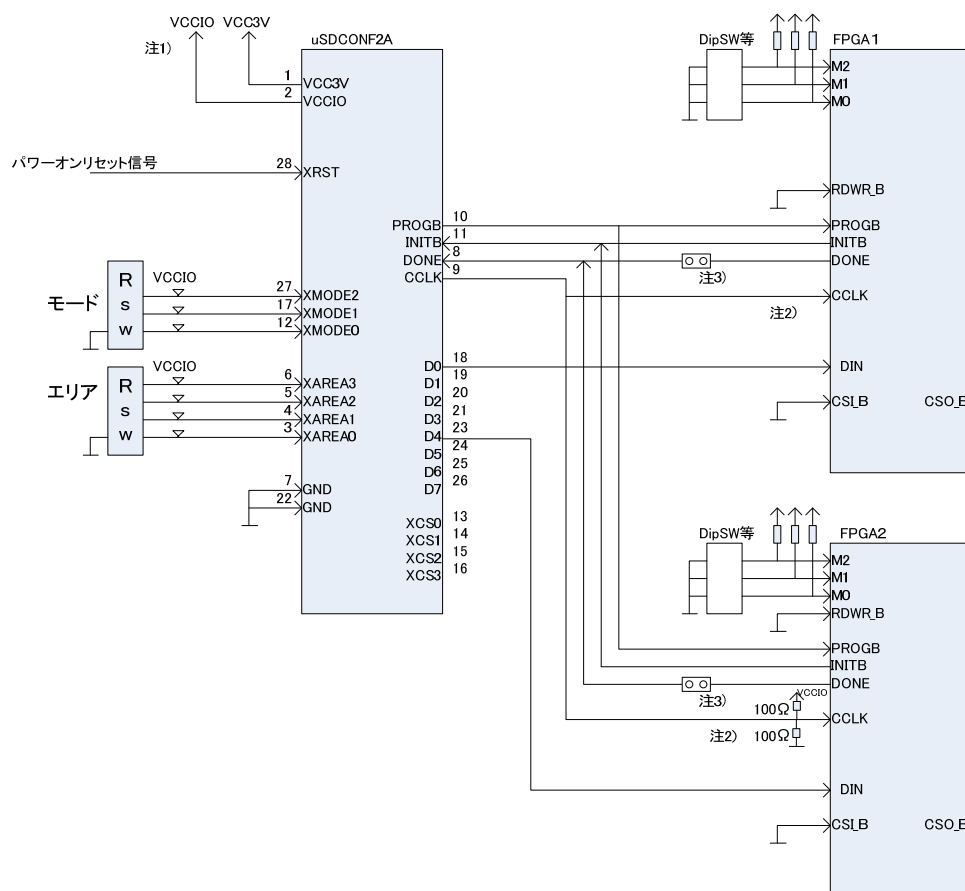


図 5.2.4.3 bit 多重(2多重)による構成例

- 注5) 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6) FPGA の近傍に配置して下さい。また複数の FPGA と接続されるときは、最遠端の FPGA の近傍に配置して下さい。FPGA のシリーズによってユーザガイドの終端方法の推奨が異なる場合があります。  
その場合は、ユーザガイドの指定を優先して下さい。
- 注7) CCLK と D[7:0] については、Fanout を 2 に抑えて下さい。
- 注8) デバッグ等で、立ち上げない FPGA が存在する場合、CONF\_DONE 信号を切り離せるようにしておく必要があります。

## 6. タイムチャート

### 6.1. 信号出力部ブロック図

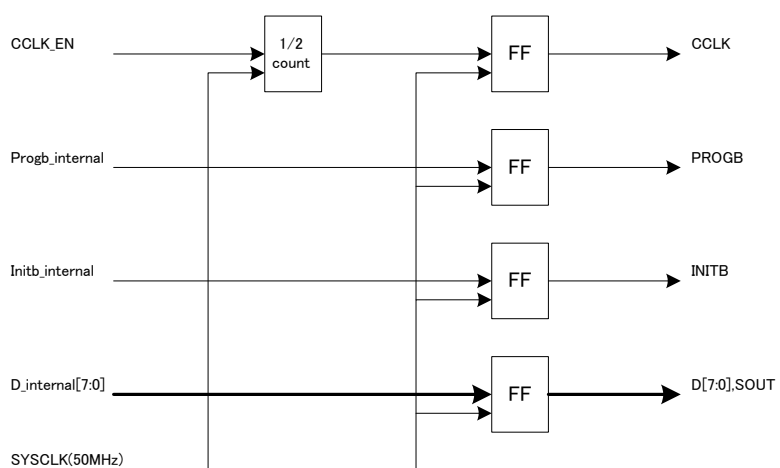


図 6.1. 出力部

## 6.2. タイムチャート

### 6.2.1. SD カードから BinaryData を 1 個読み出し Config する場合

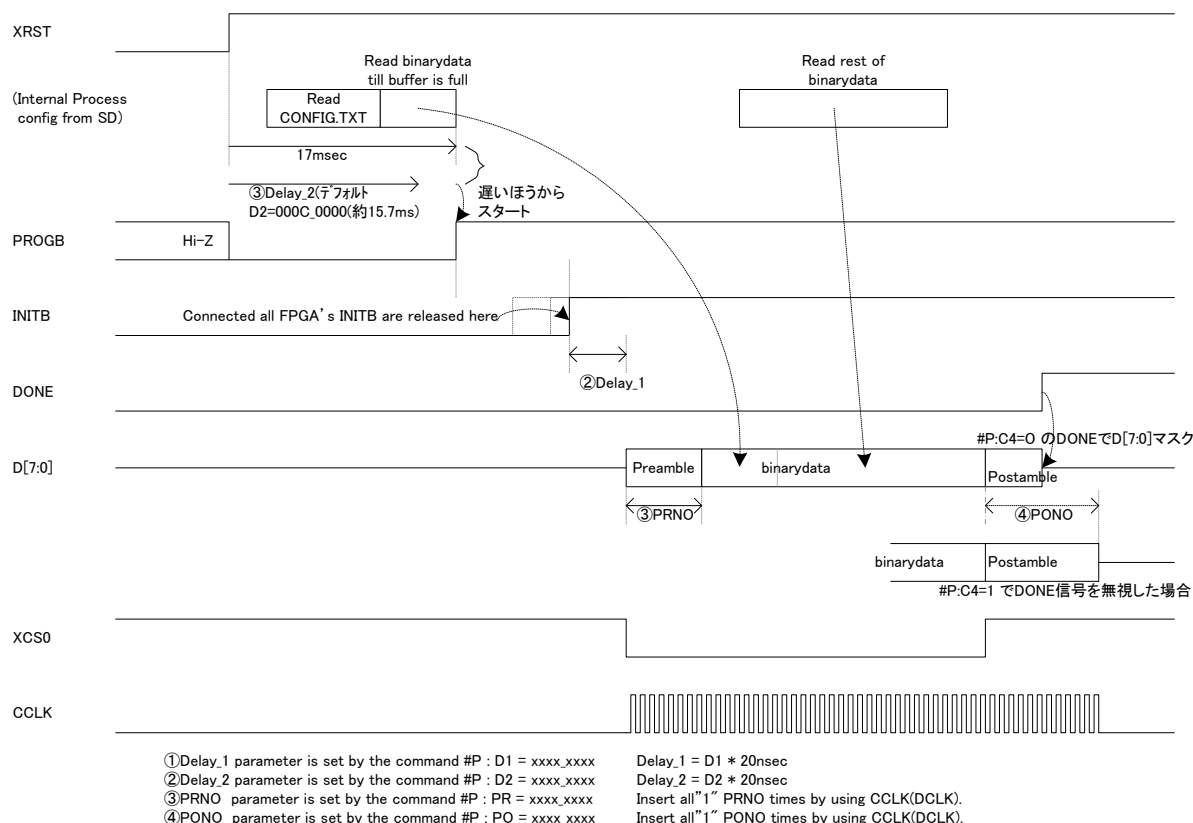


図 6.2.1. SDCard からの読み出しタイミング

- 6.1. 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。
- 6.2. リセット期間中、PROGB は High-Z の状態となります。
- 6.3. リセットが解除されたところから、D2 カウンタがカウントを開始し、config.txt で #P : D2= XXXX\_XXXX で設定したの設定時間を経過するか、SDCard からのデータの読み出し準備が出来た時点の、どちらか遅いほうの事象成立により PROGB が High-Z にリリースされ、外部 Pullup で High 論理が見えます。
- 6.4. INITB は、Ver1.800 より出力から入力に変更となりました。これに伴い、今まで D0 パラメータで INITB のリリースまでの時間設定を行う必要がなくなり、パラメータは未使用となります。  
 接続されている全ての FPGA の準備が整って、全ての FPGA が INITB をリリースしたとき、外部 Pullup により INITB は High 論理と見え、これをトリガに SDCONF は D1 カウンタのカウントを開始します。
- 6.5. D1 カウンタの値が config.txt で設定した値か、デフォルト値に達したとき、XCS0 を Low に落とすと同時に Preamble 信号を #P : PR=XXXX\_XXXX で設定した値だけ CCLK とともに Data バスに ALL"1"信号を出力します。
- 6.6. バイナリデータが送出し終わると、XCS0 を High に戻し、#P:PR=XXXX\_XXXX で設定した値分、CCLK をを出力するとともに、Data バスに ALL"1"を出力します。
- 6.7. もし、#P:C4=0(デフォルト)設定の場合、DONE 信号が High になったことを検出して、Data バスを High-Z に開放して端子を FPGA に空け渡します。ただし、CCLK については、PO で設定されたクロック数、最後まで出力します。#P:C4=1 に設定された場合、DONE 信号によらず、PO の設定値分、Data バスに ALL"1"を出し続けます。

## 6.2.2. NandFlash から BinaryData を 1 個読み出し Config する場合

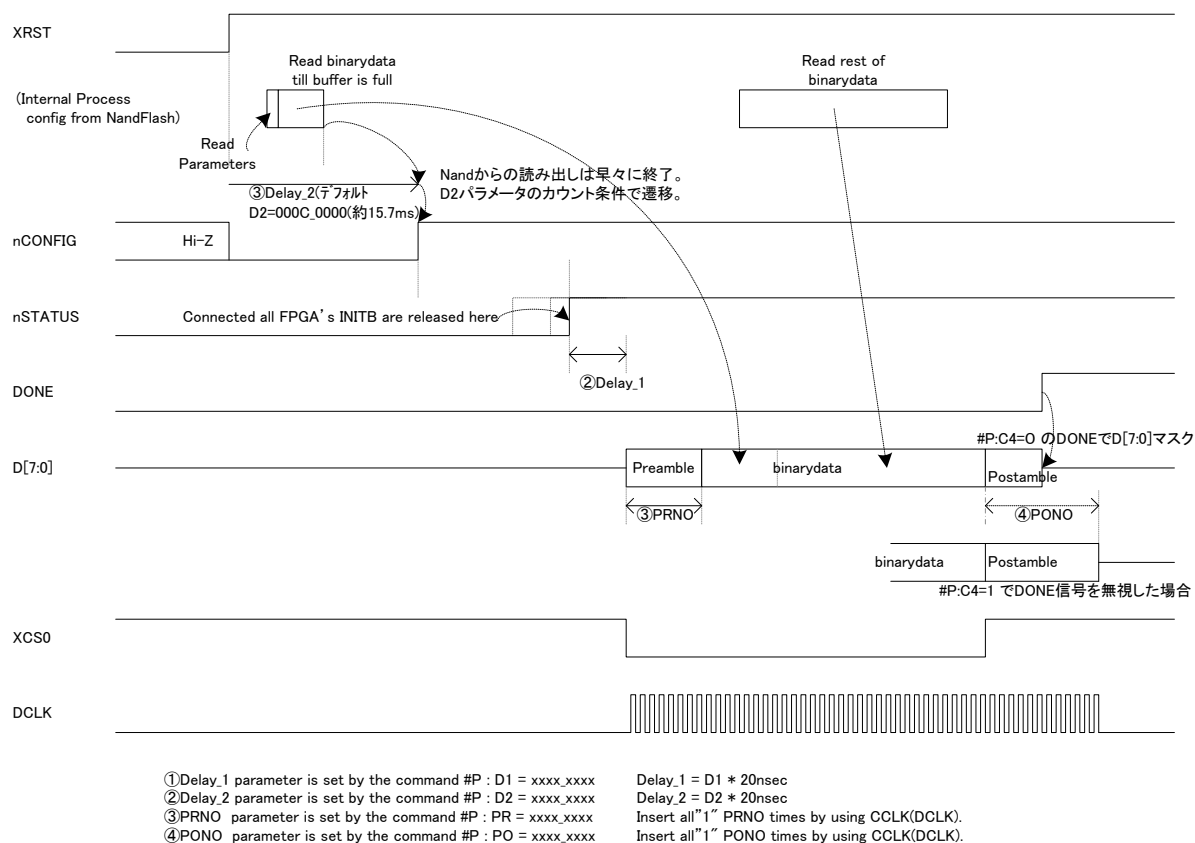


図 6.2.2. NandFlash からの読み出しタイミング

- 6.8. 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。
- 6.9. リセット期間中、PROGB は High-Z の状態となります。
- 6.10. リセットが解除されたところから、D2 カウンタがカウントを開始し、config.txt で #P : D2= XXXX\_XXXX で設定したの設定時間を経過するか、NandFlash からのデータの読み出し準備が出来た時点の、どちらか遅いほうの事象成立により PROGB が High-Z にリリースされ、外部 Pullup で High 論理が見えます。Nand のデータ準備は高速に行われるため、通常 D2 の値が採用されます。
- 6.11. INITB は、Ver1.800 より出力から入力に変更となりました。これに伴い、今まで D0 パラメータで INITB のリリースまでの時間設定を行う必要がなくなり、パラメータは未使用となります。  
 接続されている全ての FPGA の準備が整って、全ての FPGA が INITB をリリースしたとき、外部 PullUp により INITB は High 論理と見え、これをトリガに SDCONF は D1 カウンタのカウントを開始します。
- 6.12. D1 カウンタの値が config.txt で設定した値か、デフォルト値に達したとき、XCS0 を Low に落とすと同時に Preamble 信号を #P : PR=XXXX\_XXXX で設定した値だけ CCLK とともに Data バスに ALL"1"信号を出力します。
- 6.13. バイナリデータが送出し終わると、XCS0 を High に戻し、#P:PR=XXXX\_XXXX で設定した値分、CCLK をを出力するとともに、Data バスに ALL"1"を出力します。
- 6.14. もし、#P:C4=0(デフォルト)設定の場合、DONE 信号が High になったことを検出して、Data バスを High-Z に開放して端子を FPGA に空け渡します。ただし、CCLK については、PO で設定されたクロック数、最後まで出力します。#P:C4=1 に設定された場合、DONE 信号によらず、PO の設定値分、Data バスに ALL"1"を出し続けます。

## 6.2.3. SD カードから BinaryData を 3 個読み出し転送の場合

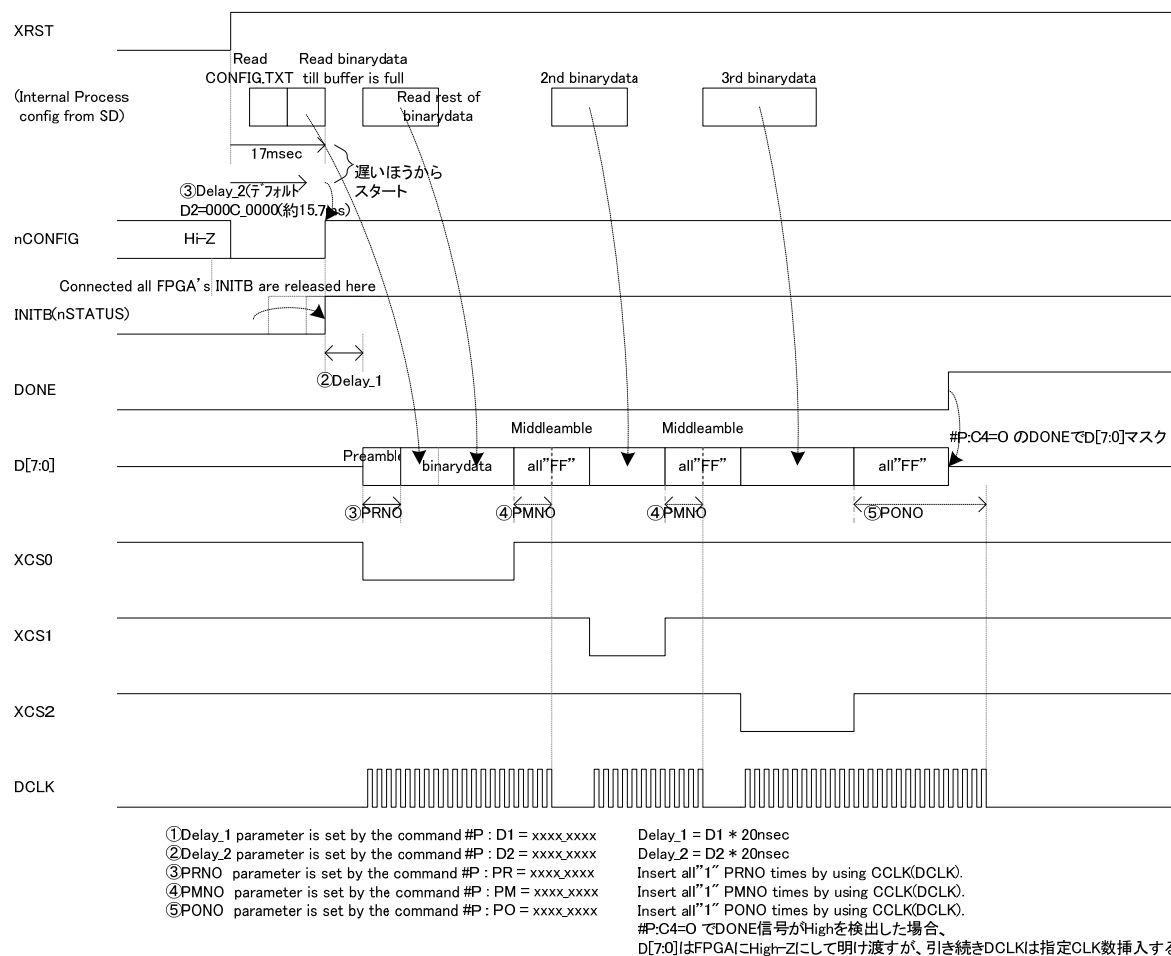


図 6.2.3. SDCard からの読み出しタイミング(3 連続)

- 6.15. 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。
- 6.16. リセット期間中、PROGB は High-Z の状態となります。
- 6.17. リセットが解除されたところから、D2 カウンタがカウントを開始し、config.txt で #P : D2= XXXX\_XXXX で設定したの設定時間を経過するか、NandFlash からのデータの読み出し準備が出来た時点の、どちらか遅いほうの事象成立により PROGB が High-Z にリリースされ、外部 Pullup で High 論理が見えます。Nand のデータ準備は高速に行われるため、通常 D2 の値が採用されます。
- 6.18. INITB は、Ver1.800 より出力から入力に変更となりました。これに伴い、今まで D0 パラメータで INITB のリリースまでの時間設定を行う必要がなくなり、パラメータは未使用となります。  
 接続されている全ての FPGA の準備が整って、全ての FPGA が INITB をリリースしたとき、外部 Pullup により INITB は High 論理と見え、これをトリガに SDCONF は D1 カウンタのカウントを開始します。
- 6.19. D1 カウンタの値が config.txt で設定した値か、デフォルト値に達したとき、XCS0 を Low に落とすと同時に Preamble 信号を #P : PR=XXXX\_XXXX で設定した値だけ CCLK とともに Data バスに ALL "1" 信号を出力します。
- 6.20. 1 個目のバイナリデータが送出し終わると、XCS0 を High に戻し、#P:PM=XXXX\_XXXX で設定した値分、CCLK をを出力するとともに、Data バスに ALL "1" を出力します。  
 PM で設定した値分 CCLK を出し終わると、次のデータの送信準備ができるまで、CCLK は停止した状態で、Data バスに ALL "1" を出し続けます。



- 6.21. PM のカウントが終了して、データの準備ができると、次の XCS1 を Low に落とすと同時にバイナリデータを送出し始めます。  
このとき、PM で指定した CCLK の挿入が、Preamble と Postamble の代わりとなります。
- 6.22. 最後のバイナリデータを送信し終わると、PO で指定したカウンタ分 CCLK と Data バスに ALL"1"を出力します。
- 6.23. もし、#P:C4=0(デフォルト)設定の場合、DONE 信号が High になったことを検出して、Data バスを High-Z に開放して端子を FPGA に空け渡します。ただし、CCLK については、PO で設定されたクロック数、最後まで出力します。#P:C4=1 に設定された場合、DONE 信号によらず、PO の設定値分、Data バスに ALL"1"を出し続けます。

## 7. LED

### 7.1. LED の実装位置

- ・ uSDCONF2A 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。



図 7.1 LED の実装位置

### 7.2. LED の点灯条件

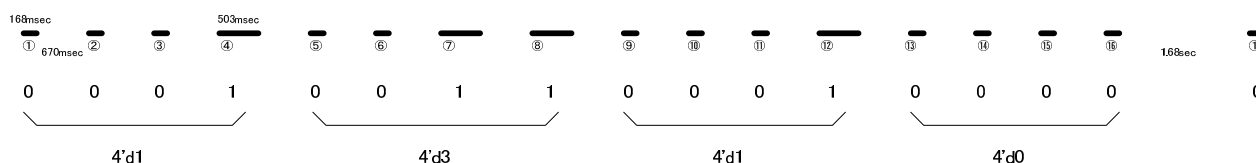
- ・ LED の点灯条件は以下の通りです。

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

### 7.3. 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、太線と太線の間の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ド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

## 8. “CONFIG.TXT”の準備

### 8.1. “CONFIG.TXT”とは

- SD カードから FPGA へ直接コンフィグレーションを行う場合、若しくは、SD カードから NandFlash へ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。

## 8.2. “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。

```
#M:X                //Maker Code Xilinx
//following are Xilinx's default setting.
//If you would like to change the parameter,please remove "/" and change the value.
//#s:0              //FPGACLK Speed 0:50M(def) 1:25MHz 2:12.5M 3:8.333M 4:6.25M 5:5M
6:3.125M
//                  //              7:1.56M 8:0.78M 9:0.39M

//Swap parameters
//#p:ss=1           //Bit swap (LSB first)
//#p:sb=0           //Byte swap(not used)
//#p:sw=0           //Wrod swap
//Delay setting
//#p:pr=0000_0100   //Preamble insert No. 256clk
//#p:po=0010_0000   //Postamble insert No. 1048576clk
//#p:d0=0008_0000   //PROGB => INITB 10msec(removed from V1.800)
//#p:d1=0000_0010   //INITB => CCLK 320nsec
//#p:d2=000C_0000   //XRST => PROGB 15.7msec
//#p:c0=1           //Word aligner on
//#p:c1=0           //nCONFIG,nSTAUS Pump_ON off
//#p:c2=0           //Multi FPGA config P2bit to Serial x 4 off
//#p:c3=0           //Multi FPGA config P4bit to Serial x 2 off
//#p:c9=0           //LED Ver disp mode off
//#r:0              //Retry No.(0-F default:0)

//Binary file area(up to 8 + 3)
#0 : file0.bin //bit or bin
#1 : file1.bin
#2 : file2.bit
.....
#E : fileE.bit
#F : fileF.bit
//#F:fileF.bin + file0.bin + file1.bin + file2.bin // + 連結が可能です。
//end
```

図 7.2. “CONFIG.TXT” サンプル

## 8.3. 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 出力開始までのディレイ間隔
	D2=	xxxx_xxxx	XRST~PROGB までのディレイ間隔
	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	FPP x4 モード ON(Xilinx では使用しません。)常時"0"
	C6=	0/1	XCS copy merge detect on
	C7=	0/1	外部回路 74138 の追加による XCS[3:0]>XCS[7:0]拡張モード ON
	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

## 8.4. 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(CCLK)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	

## (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 をコンフィグ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。
- ・ XILINX の bin 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 に開放されます。(DONE 信号を無視するためには、#p:C4=1 を設定して下さい。)

(i) #P : PM = xxxx\_xxxx

- ・ +連結で、複数のバイナリデータを FPGA に送信する場合に、バイナリデータを送信し終わって、次のバイナリデータを送信する際に、Data Bus を All“1”にした状態で、PM で指定した回数 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。
- ・ 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。
- ・ Data バスはミドルアンブル送信後も、次のバイナリデータ送信開始まで ALL“1”を出力し続けます。

(j)#P : DR = xxxx\_xxxx => #P:D2=xxxx\_xxxx に変更。

(j) #P : D0 = xxxx\_xxxx

- ・ ROM バージョン V1.800 より nSTATUS 信号が出力から入力に変更となりました。
- これに伴い、nSTATUS の出力の時間タイミングを規定する D0 パラメータは削除されました。
- ・ ~~nCONFIG がリリースされて nSTATUS 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。~~

(k) #P : D1 = xxxx\_xxxx

- ・ INITB 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。

(l) #P : D2 = xxxx\_xxxx

- ・ XRST が High になって PROGB を High にするまでの間隔を指定します。
- ・ 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。
- ・ SD カードからのコンフィグレーションの場合、SD カード読み出し準備時間の 17msec より短い時間を設定した場合、機能しません。
- ・ Nand からのコンフィグレーションの場合、SD カード読み出し準備時間による待ちが存在しないため、本パラメータが有効になります。
- ・ デフォルトは 000D\_0000 が設定され、約 15.7msec の遅延が挿入され、SD カード読み出し準備時間による遅延による動作と、ほぼ等価となります。

(m) #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になります。
- ・ “.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になります。
- ・ 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。  

$$D[1:0] \Rightarrow D[0]$$

$$D[3:2] \Rightarrow D[2]$$

$$D[5:4] \Rightarrow D[4]$$

$$D[7:6] \Rightarrow D[6]$$
- ・ 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

- 本パラメータに“1”を設定すると、Done 信号を無視します。
- DONE 信号を uSDCONF2A に接続しない回路構成系において、DONE 信号端子をオープンのままにしておくと、内部 PullUp によって常に High と見えるため、コンフィグレーションが完了したと判断し、Data バスを High-Z にして、ユーザーに開放します。
- 本設定は、DONE 信号を無視し、FPGA にデータを出し続けるためのパラメータです。

## ⑥ #P : C5 = 0/1

- ALTERA 専用のパラメータで、Xilinx では使いません。常に“0”に設定して下さい。

## 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と、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

- ・ "1"にセットすると、XCS[3:0]の4本の制御信号により、外付け回路を制御し、8本の XCSext[7:0]信号を生成し、最大8個の FPGA に対してマルチにコンフィギュレーションを行うことが可能となります。
- ・ デフォルト(0)

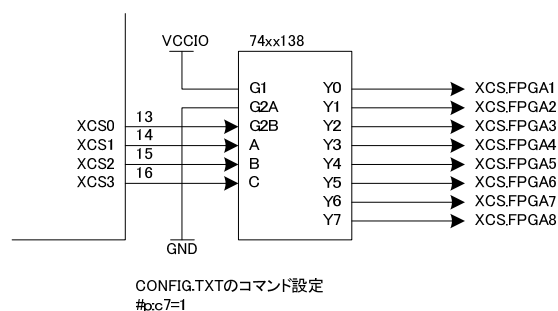


図 8.4 #P : C7=1 のときの外部回路

## 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更新されます。
- ・ リビジョンは、簡易な変更があった場合に更新されます。
- ・ 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。

## (m) #R : 0~F

- ・ コンフィグ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)
- ・ ポストアンプル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グ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グレーションをやり直します。
- ・ INITB によるコンフィグレーションの失敗の監視は行っていません。

## (n) 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。(ファイル名の間にスペースや 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 コンフィグ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。

## (o)#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グ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。(ファイル名の間にスペースや 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 コンフィグ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。

## 8.5. D1,D2,PO の設定について

- D2 パラメータは、XILINX のパラメータの tPOR(PowerOnReset) - tPL(ProgramLatency)に該当し5～45msec で、デフォルトは 15.7msec が設定されています。  
 uSDCONF1A では SDCard から config.txt とバイナリファイルの先頭を読み出し終わるまでの間、約 17msec 間 必要とし、NandFlash からの読み出し時も、ほぼこれに合わせるため、D2=000C\_0000(=15.7msec)をデフォルトとしています。(uSDCONF1A には NandFlash は搭載されていませんが、Nand 搭載版と共通の論理を使用しているため、共通のデフォルト値が採用されています。)
- D1 パラメータは、XILINX のパラメータの tICCK(CCLK outputDelay)に類似します。(MasterMode の Output Delay として定義されていますが、入力側の定義は見当たらないため、この値を利用しています。)
 

デフォルトとして、D1=0000\_0010 を採用しており、320nsec の遅延が挿入されます。
- PO パラメータは、XILINXパラメータの USRCLK 数とみなしてそれより大きい数値を設定して下さい。  
 デフォルトは PO=0010\_0000 と 1048576CLK 挿入され、十分大きな値が設定されています。
- 表 8.1 は、代表的な FPGA のシリーズのこれらのパラメータを列記した表です。
- FPGA のシリーズにより、これらの値は異なり、共通に使える値はありません。このため、これらのパラメータを理解し、条件に合致した値を選択する必要があります。
- 一覧にない FPGA をコンフィギュレーションする場合は、これらの値を調査し、最適な値を設定して下さい。

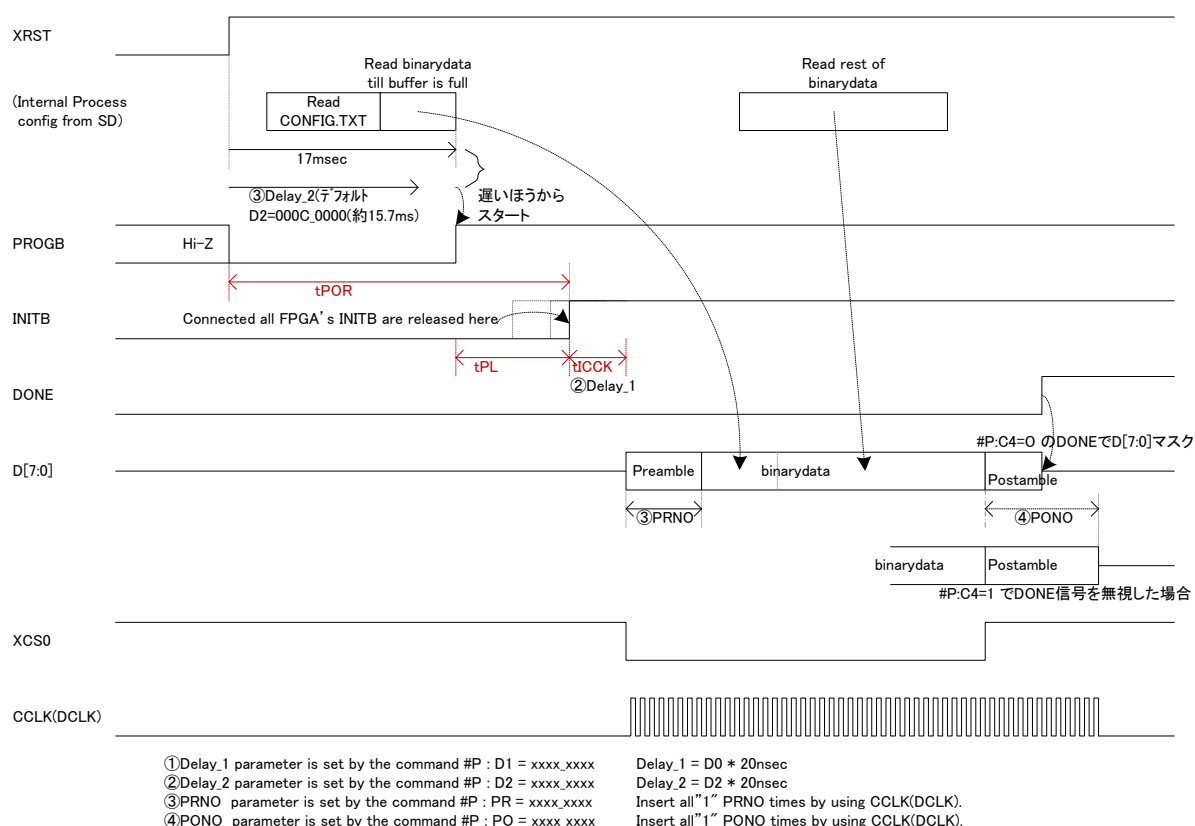


図 8.3 パラメータ位置関係

## 8.6. 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 = 1 (Swap 無し) #P : SW = 0 (Swap 無し) #S : 0 (50MHz Passive モード) #P : PR = 0000_0100 (プリアンブルなし) #P : PO = 0001_0000 (ポストアンブル 65536CLK 挿入) #P : PM = 0000_1000 (ミドルアンブル 4096CLK 挿入) <del>#P : D0 = 0000_1000 (nCONFIG ~ nSTATUS : 82usec)</del> #P : D1 = 0000_0010 (nSTATUS ~ DCLK : 320nsec) #P : D2 = 000C_0000 (XRST~nCONFIG : 15.7msec) #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)	

## (2) メーカー指定をした場合のデフォルト値(XILINX)

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 = 0010_0000 (ポストアンブル 1048576CLK 挿入) <del>#P : D0 = 0000_1000 (PROGB ~ INITB : 82usec)</del> #P : D1 = 0000_0010 (INITB ~ CCLK : 320nsec) #P : D2 = 000C_0000 (XRST~PROGB : 15.7msec) #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)	



## 9. 機能詳細

### 9.1. 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)

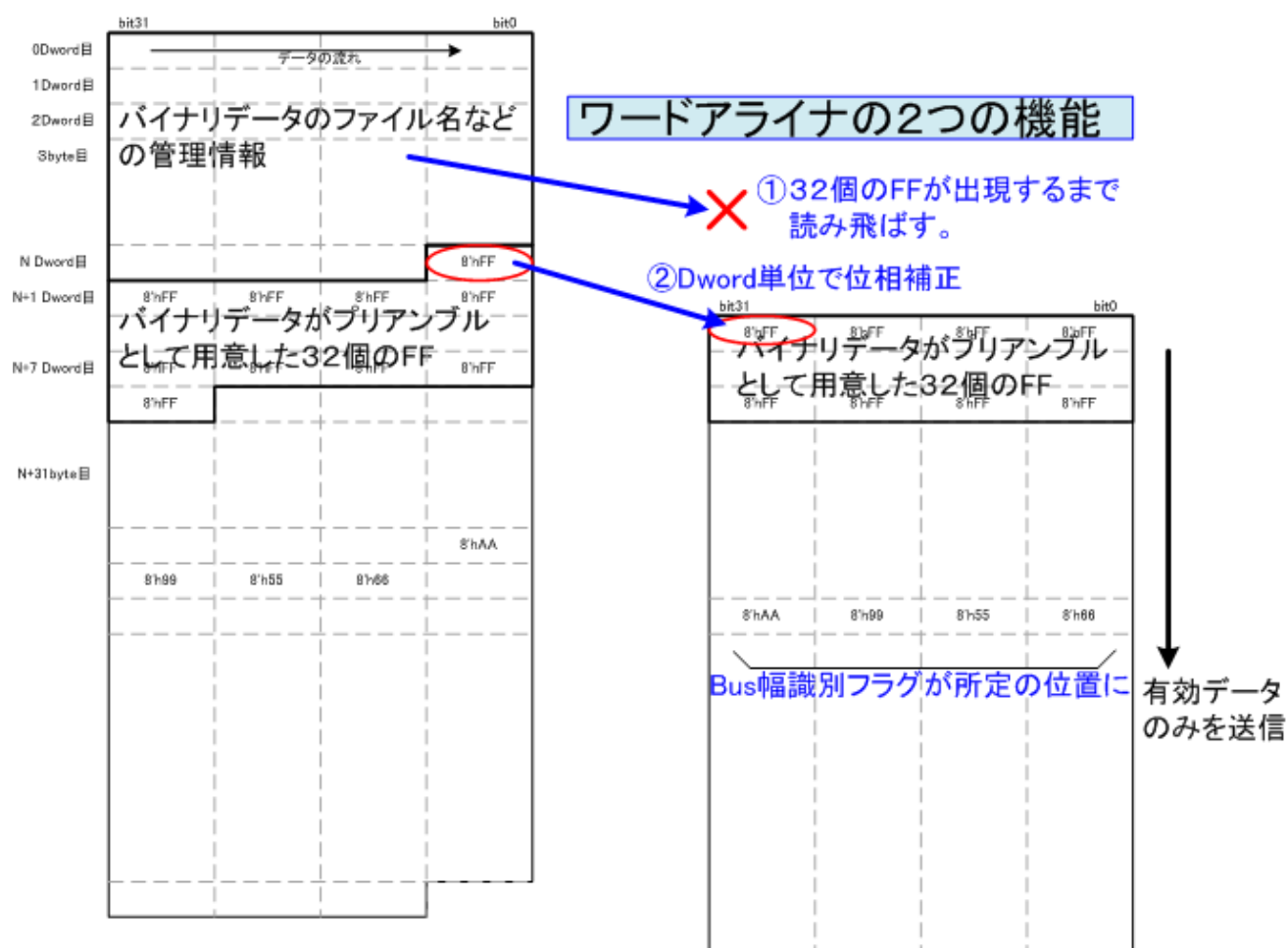


図 9.1. ワードアライナ機能説明図

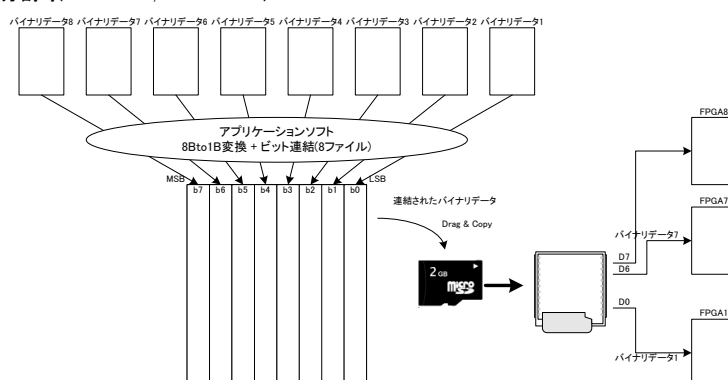
## 9.2. bit 連結による Multi FPGA Configuration 機能

- ・ 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 をコンフィグ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 をコンフィグ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。

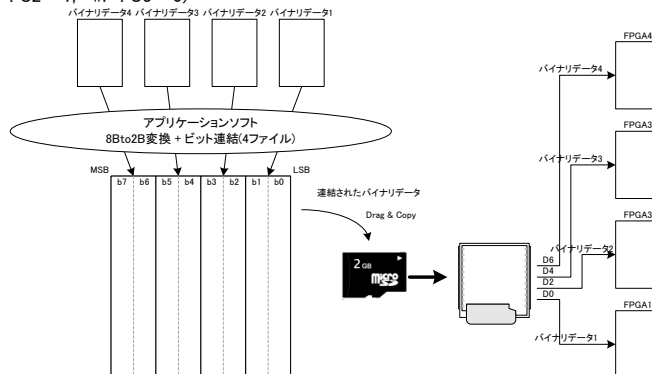
分割個数が 8 個の場合で、FPGA 割り当てに空きがある場合や、それぞれの FPGA のサイズが異なる場合、Daisy-Chain 方式より効率が悪くなる場合があります。

### 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)

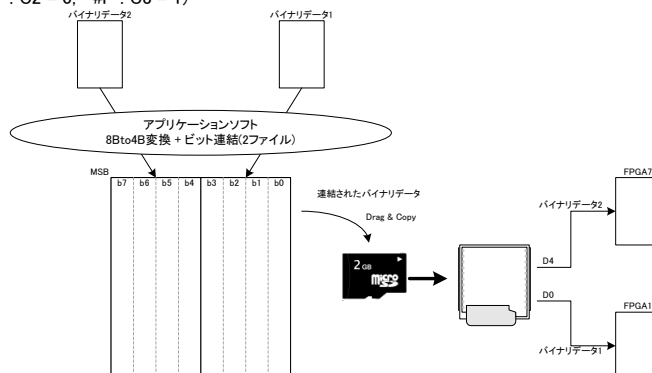


図 9.2. bit 連結による Multi FPGA Configuration の動作原理図

### 9.3. “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。

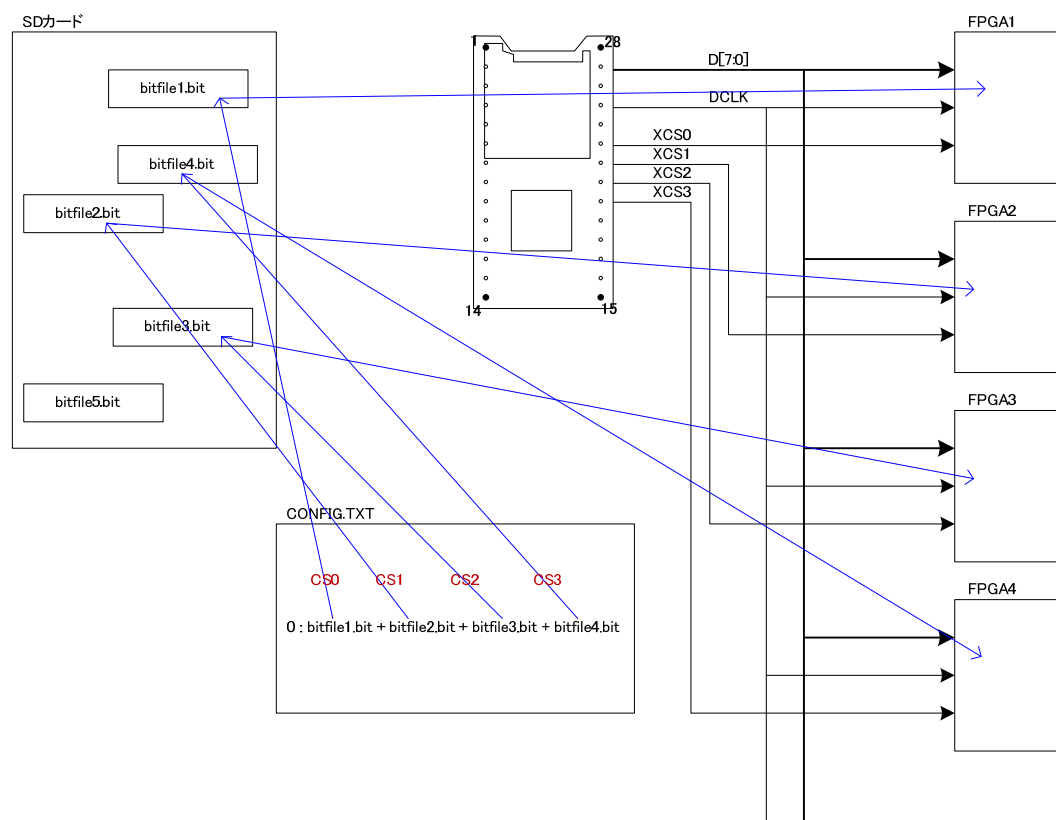


図 9.3. “CONFIG.TXT”による Multi FPGA Configuration の動作原理図

## 9.4. SDCard の活線挿抜について

### 9.4.1. 構成図

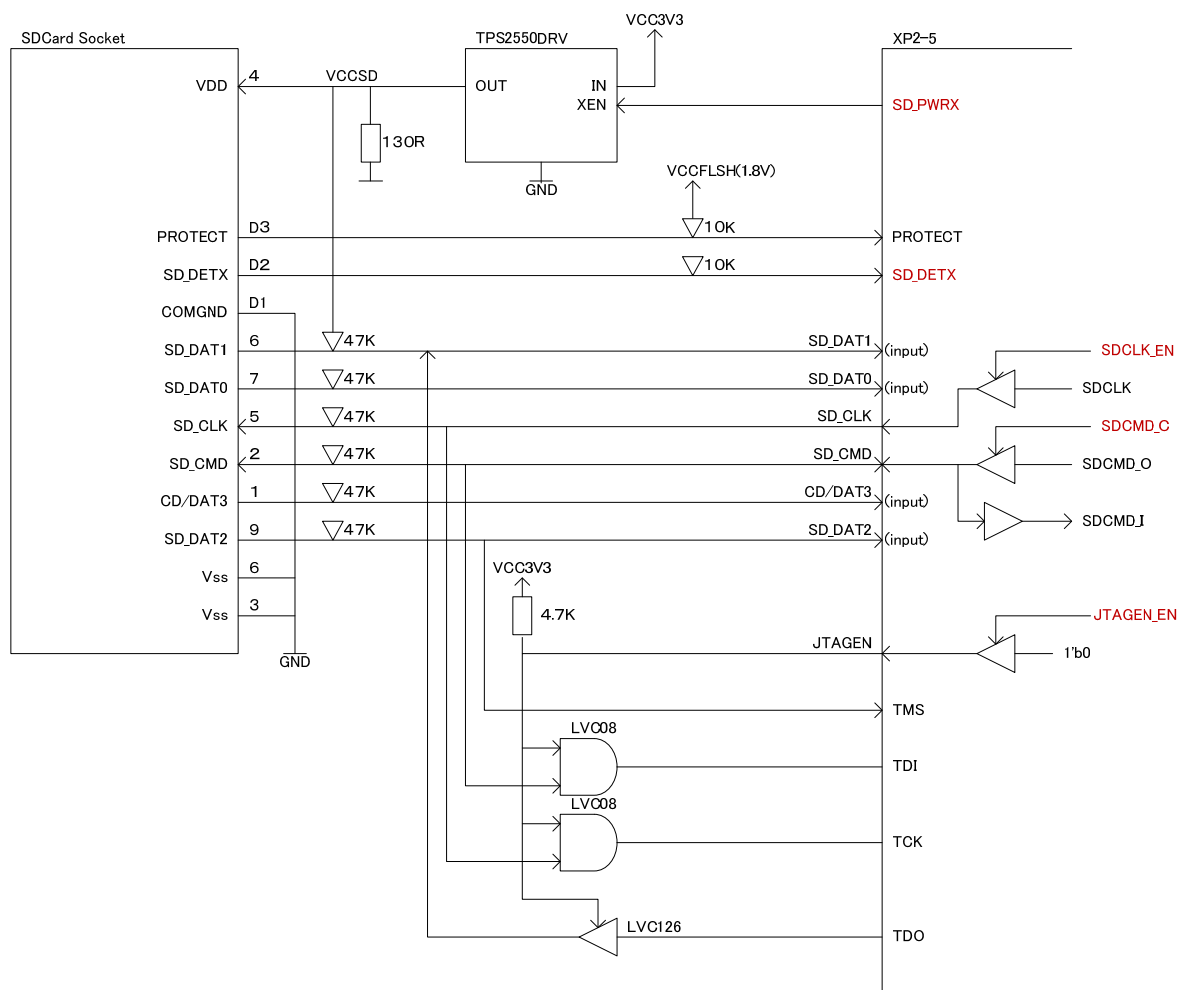


図 9.5.1 SDCard インタフェース部構成図

### 9.4.2. SDCard の活線挿抜対策

#### ① SD カード電源の制御

- ・ XRST 信号、SD\_DET<sub>X</sub> 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<sub>X</sub> 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<sub>X</sub> == 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<sub>X</sub>==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<sub>X</sub> がアサート時、Hi-Z 出力となります。
- ・ 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 出力となります。

#### 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チェックします。  
 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、TDL,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。

## 10. 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。

### 10.1. 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。

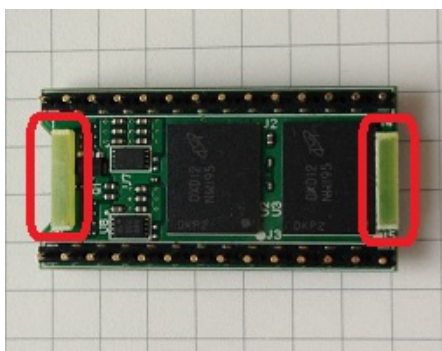


図 10.1. uSDCONF2A Bottom View

## 10.2. モジュールの取り外し方

- Sunhayato の IC EXTRACTOR GX-7 のヘッド部分を有償でご用意しています。
- 治具のアームのフック金具をモジュールとソケットの間に差し込みます。
- 両側のアームの中心が、モジュールの中央ラインにくるように、また、フックがしっかりとモジュール下部に入り込むように調整します。
- 写真矢印の方向に、片方の手でアーム部分を挟みこみ、フックがずれることなくモジュールを引っ掛けるよう、しっかりと押さえ込みます。
- この状態で、もう片方の手で、ピストルの引き金を引くようにレバーを持ち上げて下さい。

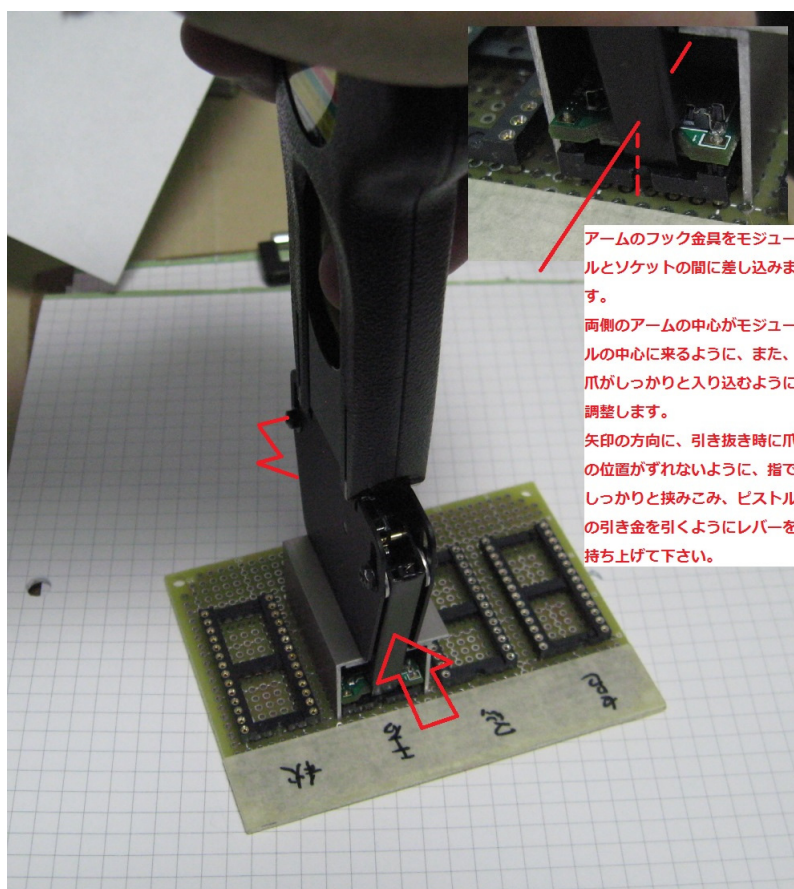


図 10.2. 取り外し用金具



## 11. 添付品

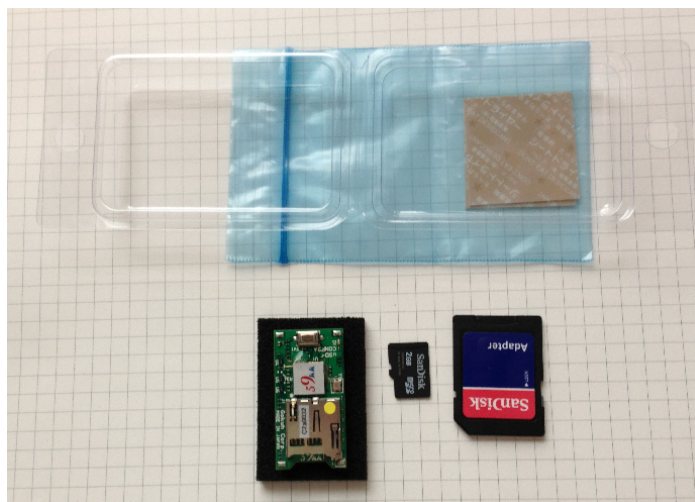


図 11.1



図 11.2

- 図 11.1 は製品として出荷される構成要素です。
  - uSDCONF2A 本体
  - 2 Gbyte microSDCard ( メーカーはそのときの入手状況によって異なる場合があります。 )
  - SDCard 中の readme ディレクトリ内に、下記のデータが書き込まれています。
    - (ア) Datasheet
    - (イ) CONFIG.TXT サンプル(ALTERA、Xilinx 用)
  - SDCard アダプタ
  - 静電マット
  - 乾燥剤＋静電袋＋ブリスターケース
- 図 11.2 はケースに梱包された状態です。
  - ケースのサイズは、突端部分を含めて 50mm × 85mm × 20mm です。
- 数量が多く、即実装される場合には、梱包しないで出荷も可能です。



## 12. VCCIO 電圧とシールの色

- ・ uSDCONF2A の VCCIO 電圧を識別できるよう、microSD ソケット表面に丸いシールが張られています。(表 13.1 参照)
- ・ 3.3V と 2.5V のモジュール内部の設定が共通のため、3.3V/2.5V を青色シール 1 枚で代用する場合があります。

VCCIO	インターフェース	シール
3.3V	LVC MOS3.3V	青色
2.5V	LVC MOS2.5V	緑色
1.8V	LVC MOS1.8V	黄色

表 12.1

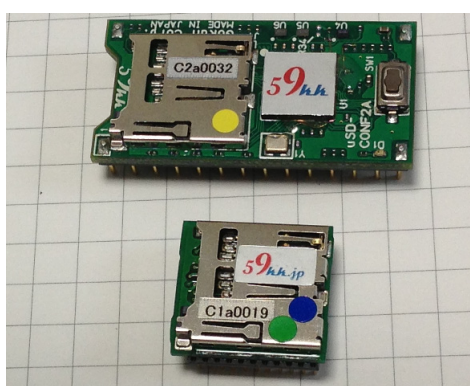


図 12.1 シール添付例

## 13. 発注型格

(2) 発注時、下記の型格でご注文下さい。

(3) 数量が多い場合などで、即実装される場合は、静電マット平面梱包が便利です。

VCCIO 電圧	梱包形態	発注型格
3.3V/2.5V	ピリスターケース単体梱包	uSDCONF2A8-001-33V
	静電マット平面梱包	uSDCONF2A8-001-33VN
1.8V	ピリスターケース単体梱包	uSDCONF2A8-001-18V
	静電マット平面梱包	uSDCONF2A8-001-18VN

表 13.1

## 14. 制約事項

### ① SD カードは、添付の2G バイト品をご使用下さい。

市販の2G バイト以下の FAT16 でフォーマットされた SD カードもご利用いただけますが、初期のころの SD カードインタフェース速度が極端に遅いものや、最近のものでも、極まれに相性が合わない場合があります。この場合、SD カードからのコンフィギュレーション時、転写時に LED が点灯したままの状態になります。

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 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。

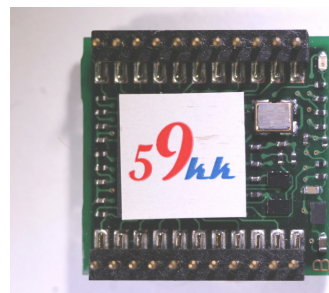
### ⑤ CCLK のクロック速度選択機能で、CCLK を受信して動作するマスタモードは定義されていますが、現在インプリメントしていません。

## 15. 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。

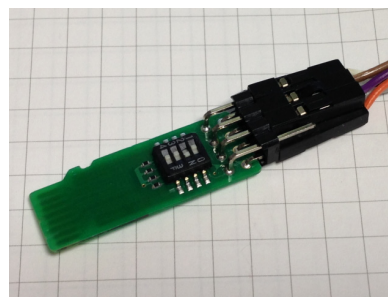
(JTag ケーブルは、悟空株式会社にて販売する予定です。)



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
- ✧ コンフィグ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 からのコンフィグレーション
  - 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>