

uSDCONF1A03 データシート  
(for XILINX ユーザ)  
(uSDCONF1A0303-5/8-xxV)

Rev1.00 2014.07.22(ROMVer.1.805)

悟空株式会社

注)データシートとして Xilinx 用 ALTERA 用がありますが、モジュール自体は共通で差異はありません。  
本データシートでは、Xilinx 用 FPGA に特化して、その使い方を説明しています。

## 改訂履歴

日時	担当者	Rev	改訂内容概要
2014.10.14	大庭	1.0	1.2. uSDCONF1F,1A01,02 からの修正点追記 図 2.1 uSDCONF1A03 形状図 差し替え 添付ソケット記述追加。 3.3. 消費電流 ※記述追加。 4.1. uSDCONF1A03 ピン機能表 注3)追加。 12.添付品 ④ 2200SA-12G-34 追記。 13. ゲートサイズ、VCCIO 電圧とシールの色 全面変更。 14. 発注型格 ゲートサイズ追加。

## 目次

1. 機能・特徴	5
1.1. uSDCONF1A03 の機能・特徴	5
1.2. uSDCONF1F、1A01,1A02 からの修正点	6
2. 形状	7
3. 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
3.5. リセットに関して	8
4. Pin 機能表	9
4.1. uSDCONF1A03 ピン機能表	9
4.2. XAREA 詳細	10
5. 接続例	11
5.1. uSDCONF1A03 接続例	11
5.1.1. SelectMap8/Slave Serial モード選択式	11
5.1.2. Slave Serial モード固定	13
5.2. uSDCONF1A03 Multi FPGA 接続例	16
6. タイムチャート	19
6.1. 信号出力部ブロック図	19
6.2. タイムチャート	20
7. LED	23
7.1. LED の点灯条件	23
7.2. バージョン情報表示機能	24
8. “CONFIG.TXT”の準備	25
8.1. “CONFIG.TXT”とは	25
8.2. “CONFIG.TXT” サンプル	26
8.3. Commands 速見表	27
8.4. Commands	28
8.5. D1,D2,PO の設定について	36
9. 機能詳細	39
9.1. ワードアライナ機能	39
9.2. bit 連結による Multi FPGA Configuration 機能	40
9.3. “CONFIG.TXT”による Multi FPGA Configuration 機能	41
9.4. 外付け回路による最大 7 個の Multi FPGA Configuration 機能	41

9.5.	SDCard の活線挿抜について.....	42
9.5.1.	構成図.....	42
9.5.2.	SDCard の活線挿抜対策.....	43
10.	uSDCONF1A03 と uSDCONF1F の共存について.....	44
10.1.	uSDCONF1F 用の基板に uSDCONF1A03 を実装する場合.....	44
10.1.1.	uSDCONF1F 用基板に uSDCONF1A03 を新規にアセンブリする場合.....	44
10.1.2.	uSDCONF1F 用ソケットが既に実装されている場合.....	45
11.	モジュールのソケットからの取り外しについて.....	46
11.1.	モジュール裏面.....	46
12.	添付品.....	48
13.	VCCIO 電圧とシールの色.....	49
14.	発注型格.....	49
15.	制約事項.....	50
16.	アップデート機能(順次対応予定).....	51

## 1. 機能・特徴

### 1.1. uSDCONF1A03 の機能・特徴

- ・ 本モジュールは microSD カードを媒体として、Windows システムによって Card に記憶されたバイナリデータを FPGA に転送してコンフィグレーションを行う機能を供します。
- ・ microSD カードには、最大 2GB までの FAT16<sup>注1)</sup> でフォーマットされた Card を使用でき、容量の許す限り<sup>注2)</sup> バイナリデータを格納することができます。
- ・ 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、コンフィグレーションに関する信号間の遅延量など、各種パラメータを指定することができ、専用のソフトを使うことなく、使い慣れたテキストエディタで編集するだけです。
- ・ “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。
- ・ 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 から 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 をコンフィグレーションすることが可能です。(例えば #3 : data3.bin で SW 設定3と data3.bin を関連付けします。)
- ・ 本機能は、例えばデバッグの場面で開発段階の複数のバージョンのバイナリデータの動作比較や、デモの場面で複数のアプリケーションを瞬時に切り替えてデモンストレーションする場面などで威力を発揮します。
- ・ microSD カードとのインタフェースは 50MHz の HighSpeed モードで転送を行います。
- ・ FPGA への転送バス幅を 1bit と 8bit から選択することが可能です。(MODE ピン参照)
  - microSD カードから FPGA への Slave Serial によるコンフィグレーション。
  - microSD カードから FPGA への SelectMap8 によるコンフィグレーション。
    - ・ 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
- ・ 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。サフィックスが“.bit”のファイルには先頭にファイル名を含む管理情報が含まれており、ダブルワードアクセス時の位相を不確定にしてしまいますが、この管理情報を取り除くことができ、ダブルワード単位での位相を揃えることが可能になります。
- ・ 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。
- ・ リセット解除から PROGB の立ち上がりまでの間隔(D2 パラメータ)や、PROGB 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。

注 1)FAT32 でフォーマットされた microSD カードや、2GByte を超える SDHC には現状対応していません。

注 2)FAT16 のルートディレクトリへの書き込みの上限個数は 512 個までです。

## 1.2. uSDCONF1F、1A01,1A02 からの修正点

- ① uSDCONF1A01,02 から、microSD カードの電源制御が追加されました。これにより、パワーオンリセット時に必ず電源が初期化されるため、リセット信号のチャタリング等で microSD カードが INACT 状態に陥ることを回避できるようになりました。uSDCONF1F では INACT 状態に陥ったときは、電源を一旦 OFF するか、microSD カードを挿抜する必要がありましたが、uSDCONF1A03 ではリセット信号だけで INACT 状態から復帰できるようになりました。
- ② uSDCONF1F の前面方向の空きスペースを利用して、ピンの Dimension を変更することなく、新たに 2Pin 追加し、XCS0,XCS1 に割り当てました。(将来的に搭載する機能によっては別の信号に割り当てることもあります。)

これにより、SDCONF ソリューションのひとつのアドバンテージ機能である + 連結をご利用いただけます。

外付けの追加回路なしに、2つまでの FPGA に xCS 信号により選択コンフィグレーションできます。

また、外付けに回路を追加することにより、最大7個までの FPGA に対して xCS 信号により選択コンフィグレーションすることが可能です。

- ③ uSDCONF1A03 より、ソケットを JAE の ST11S008V4H に変更になりました。これにより、microSDCard の誤挿入による爪の破損障害が改善され、脱落防止機構により、衝撃や重力による microSDCard の脱落の心配が無くなりました。
- ④ 搭載の制御用 FPGA の論理サイズが uSDCONF1F,1A01,02 より 1.6 倍大きいモジュールを選択できるようになりました (uSDCONF1A03-8-xxV)。これにより、今後のより高度な機能追加に、モジュールの買い替えなしに、論理データの変更だけでアップグレードが可能です。

(uSDCONF1A03-5-xxV に比べ、uSDCONF1A03-8-xxV は 1.6 倍の論理サイズとなります。)

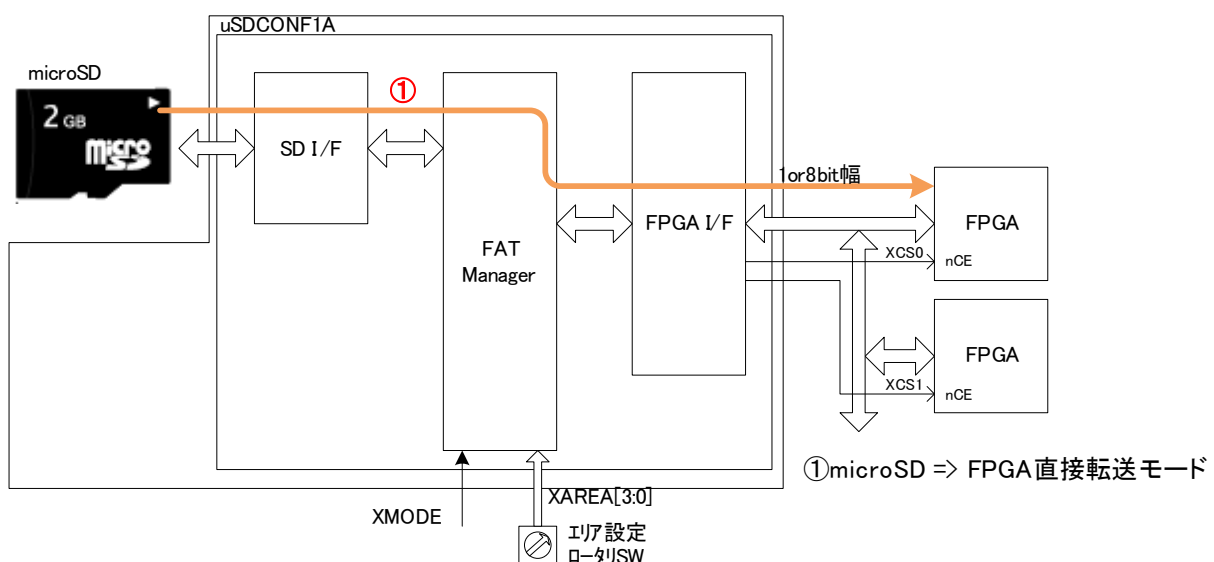


図 1.1 uSDCONF1A03 ブロックイメージ図

## 2. 形状

uSDCONF1A03 モジュールの形状を掲載します。

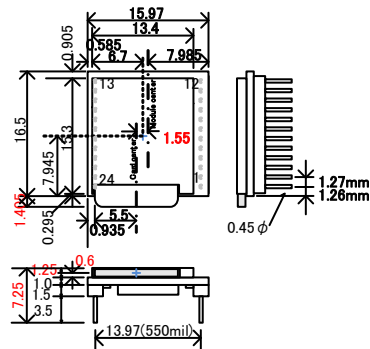


図 2.1 uSDCONF1A03 形状図

- (1) uSDCONF1A03 の使用 pin は、MAC8 の OQ-11-5-12P 。

( <http://www.mac8sdk.co.jp/mac8/pdf/OQ.pdf> )

ソケット実装の場合の推奨ソケットは、

MAC8 ME-10-10-12P(金メッキ品)、MAC8 ME-1-10-12P(スズメッキ品)。

( <http://www.mac8sdk.co.jp/mac8/pdf/MEMF.pdf> )

Neltron 2200Series

( <http://akizukidenshi.com/download/2200-pinsocket.pdf> )

※製品には、MAC8 ME-10-10-12P か ME-1-10-12P か 2200SA-12G-34 のいずれかが  
2 個添付されます。

プリント板取付穴径は、直接実装、ソケット実装とも 0.6~0.7mm φ。

### 3. 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 Tristage 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)		90mA	
	VCCIO(3.3V 時)		10mA	
microSD カード	VDD(3.3V)		100mA	規格上の Max 値
合計			200mA	

表 3.3 消費電流表

※消費電流は ROM のバージョンによって変化する場合があります。

#### 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

uSDCONF1A03 モジュールのピン機能表を掲載します。

### 4.1. uSDCONF1A03 ピン機能表

水平 1.27mm ハーフピッチ実装タイプのピン機能表を示します。

Pin No.	Signal name	Dir	内部 Pullup	Descriptions
1	XCS0	O		FPGA0 制御用チップセレクト信号
2	VCCIO	—		入出力信号の VCCIO 電源入力。
3	XAREA0	I	4.7K	バイナリデータ選択用ロータリーSW(LSB)
4	VCC3V	—		3.3V 電源入力
5	XAREA1	I	4.7K	バイナリデータ選択用ロータリーSW
6	GND	—		GND
7	XAREA2	I	4.7K	バイナリデータ選択用ロータリーSW
8	DONE	I	330Ω	FPGA DONE 信号
9	CCLK	O		FPGA CCLK 信号
10	PROGB	O	4.7K	FPGA PROGB 信号
11	XAREA3	I	4.7K	バイナリデータ選択用ロータリーSW(MSB)
12	INITB	O	4.7K	FPGA INITB 信号
13	XRST	I	4.7K	パワーオンリセット信号
14	D0/SOUT	O		FPGA 8bit Parallel data0 / Serial data
15	D1	O		FPGA 8bit Parallel data1
16	D2	O		FPGA 8bit Parallel data2
17	D3	O		FPGA 8bit Parallel data3
18	D4	O		FPGA 8bit Parallel data4
19	GND	—		GND
20	D5	O		FPGA 8bit Parallel data5
21	D6	O		FPGA 8bit Parallel data6
22	D7	O		FPGA 8bit Parallel data7
23	XMODE	I	4.7K	モードピン H:Slave Serial / L:Select Map8
24	XCS1	O		FPGA1 制御用チップセレクト信号

表 4.1 uSDCONF1A03 ピン機能表

注1) XAREA[3:0],XMODE, DONE,PROGB,INITB の信号には内部に Pullup 抵抗が実装されており、外部で実装する必要はありません。

注2) XCS 制御する場合は、外部に終端抵抗が必要です。

注3) uSDCONF1F からの設計移行時、XCS[0:1]の追加と、XAREA[3:0],XMODE に 4.7KΩPullup が付加されている点にご注意下さい。

#### 4.2. XAREA 詳細

- ・ “CONFIG.TXT”ファイル上で microSD 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 4.2 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 4.2 XAREA[3:0]とエリア番号表

## 5. 接続例

### 5.1. uSDCONF1A03 接続例

#### 5.1.1. SelectMap8/Slave Serial モード選択式

##### (1) FPGA と 1 対 1 接続の場合

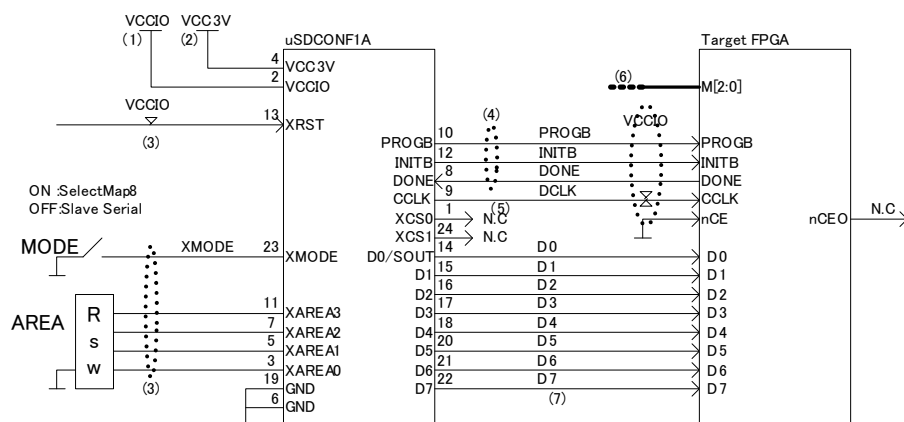


図 5.1.1.1 uSDCONF1A03 SelectMap8 / Slave Serial 選択式 1:1 接続例

- 注1) VCCIO は 3.3V~2.5V、1.8V に対応しています。  
VCCIO に 1.8V を使用する場合は、Module のデータを 1.8V 用に書き変える必要があります。
- 注2) 3.3V ± 5% (0.165V) が推奨入力電圧となります。
- 注3) XRST, XMODE, XAREA[3:0] ピンには、内部 Pullup 抵抗 (4.7kΩ) が実装されており、外付けで終端抵抗を実装する必要はありません。
- 注4) PROGB, INITB, DONE 信号には、Module 内部に Pullup 抵抗が実装されており、外部に実装する必要はありません。  
ただし、uSDCONF1A03 をソケットから取り外して運用することがある場合は、外付けに終端抵抗を取り付けて下さい。
- 注5) CCLK は最短でレイアウトするようにして下さい。また、GND でシールドして下さい。  
FPGA の近傍に、Pullup50Ω、Pulldown50Ω のテブナン終端をして下さい。
- 注6) XMODE ピンの論理と適合した M[2:0] 信号の設定値を選択して下さい。

## (2) XCS 選択接続の場合

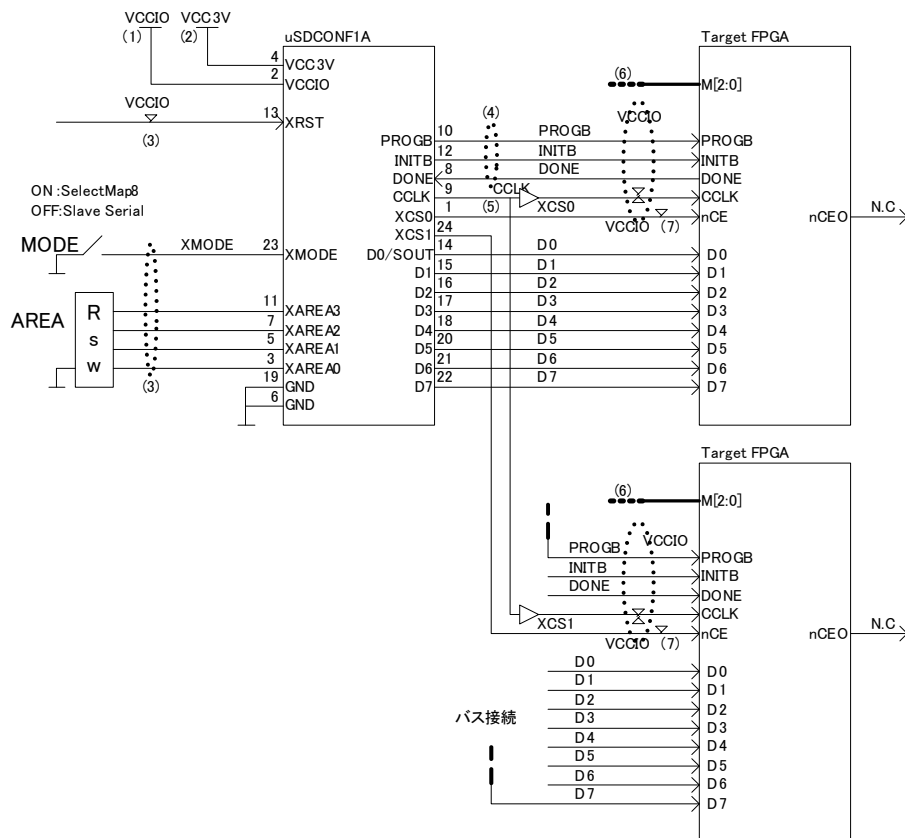


図 5.1.1.2 uSDCONF1A03 SelectMap8 / Slave Serial 選択式 1:2 接続例

- 注1) VCCIO は 3.3V~2.5V、1.8V に対応しています。  
VCCIO に 1.8V を使用する場合は、Module のデータを 1.8V 用に書き変える必要があります。
- 注2) 3.3V ± 5% (0.165V) が推奨入力電圧となります。
- 注3) Xrst, XMODE, XAREA[3:0] ピンには、内部 Pullup 抵抗 (4.7K Ω) が実装されており、外付けで終端抵抗を実装する必要はありません。
- 注4) PROG, INITB, DONE 信号には、Module 内部に Pullup 抵抗が実装されており、外部に実装する必要はありません。  
ただし、uSDCONF1A03 をソケットから取り外して運用することがある場合は、外付けに終端抵抗を取り付けて下さい。
- 注5) CCLK は最短でレイアウトするようにして下さい。また、GND でシールドして下さい。  
FPGA の近傍に、Pullup 50 Ω、Pulldown 50 Ω のテブナン終端をして下さい。
- 注6) XMODE ピンの論理と適合した M[2:0] 信号の設定値を選択して下さい。
- 注7) XCS0, XCS1 を + 連結で制御する若番の FPGA から順番に接続します。  
それぞれの制御線には、FPGA の近傍に VCCIO で PullUp 抵抗 (4.7K Ω) を接続して下さい。

## 5.1.2. Slave Serial モード固定

## (1) FPGA と 1 対 1 接続の場合

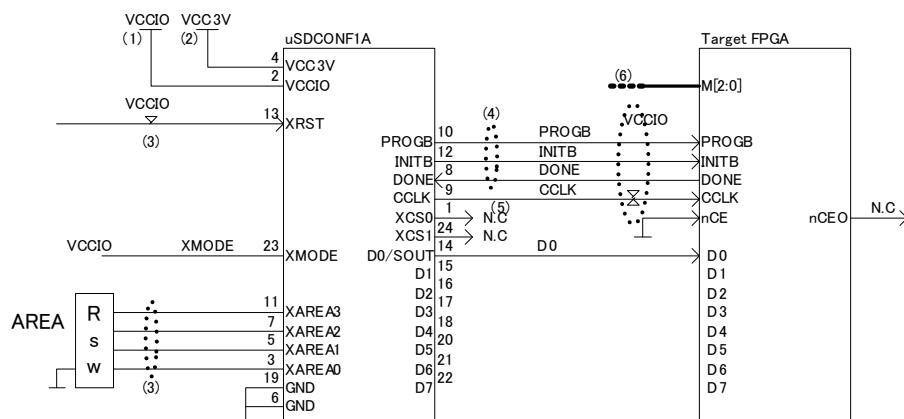


図 5.1.2.1 uSDCONF1A03 Slave Serial 固定 1:1 接続例

- 注1) VCCIO は 3.3V~2.5V、1.8V に対応しています。  
VCCIO に 1.8V を使用する場合は、Module のデータを 1.8V 用に書き変える必要があります。
- 注2) 3.3V ± 5% (0.165V) が推奨入力電圧となります。
- 注3) XRST, XMODE, XAREA[3:0] ピンには、内部 Pullup 抵抗 (4.7K Ω) が実装されており、外付けで終端抵抗を実装する必要はありません。
- 注4) PROGB, INITB, DONE 信号には、Module 内部に Pullup 抵抗が実装されており、外部に実装する必要はありません。  
ただし、uSDCONF1A03 をソケットから取り外して運用することがある場合は、外付けに終端抵抗を取り付けて下さい。
- 注5) CCLK は最短でレイアウトするようにして下さい。また、GND でシールドして下さい。  
FPGA の近傍に、Pullup50 Ω、Pulldown50 Ω のテブナン終端をして下さい。
- 注6) XMODE ピンの論理と適合した M[2:0] 信号の設定値を選択して下さい。

## (2) XCS 選択接続の場合

Xilinx の SlaveSerial モードでは nCE が無効なため、図 5.1.2.2a の構成では最初のバイナリデータが2つの FPGA に書き込まれてしまいます。

Xilinx の SlaveSerial モードで複数の FPGA をコンフィグする場合は、DIN,DOUT による伝播を行うか、5.2 uSDCONF1A03 Multi FPGA 接続例を参考にして下さい。

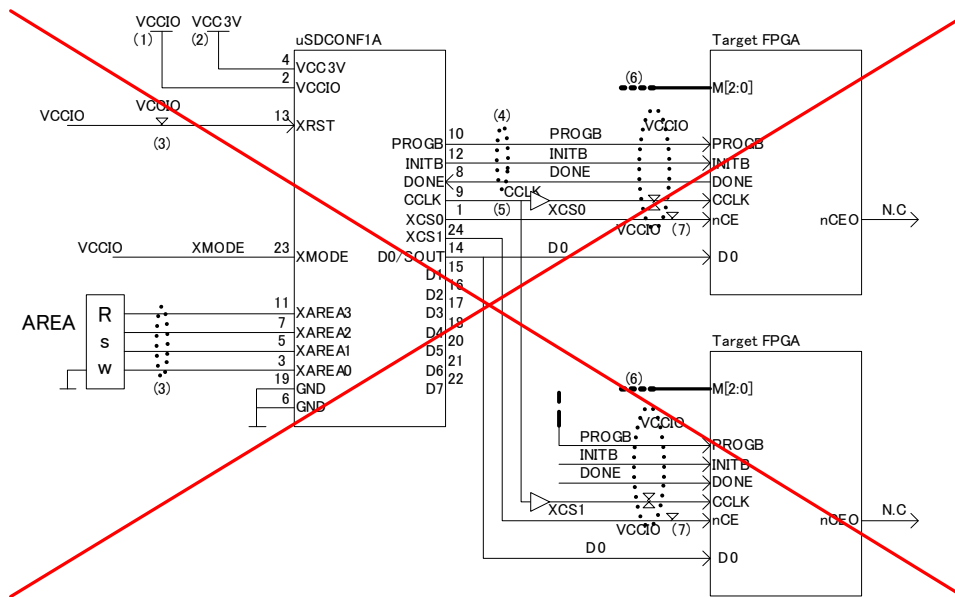


図 5.1.2.2a uSDCONF1A03 Slave Serial 固定 1:2 接続例 (動作しません。)

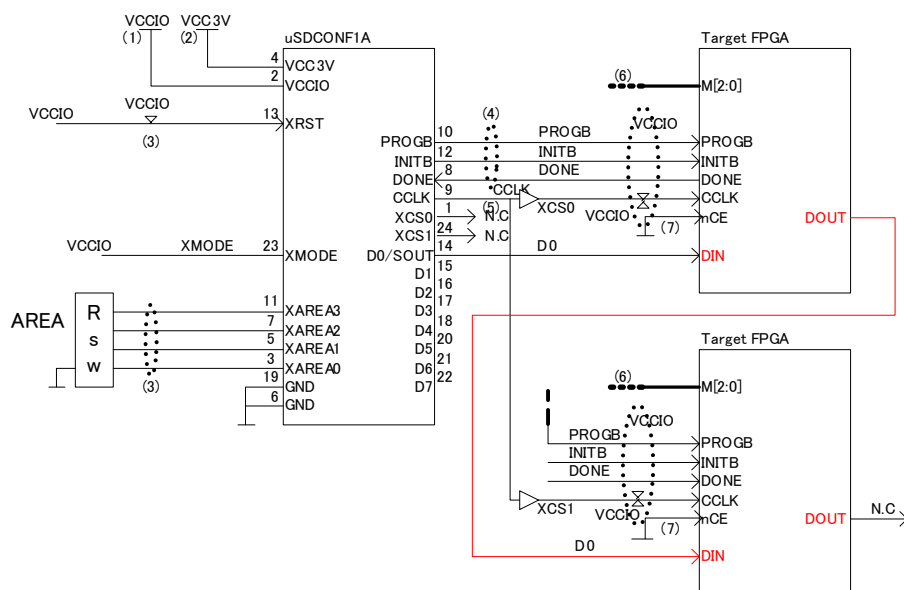


図 5.1.2.2b uSDCONF1A03 Slave Serial 固定 1:2 接続例

- 注1) VCCIO は 3.3V~2.5V、1.8V に対応しています。  
VCCIO に 1.8V を使用する場合は、Module のデータを 1.8V 用に書き変える必要があります。
- 注2) 3.3V  $\pm$  5% (0.165V) が推奨入力電圧となります。
- 注3) XRST, XMODE, XAREA[3:0]ピンには、内部 Pullup 抵抗(4.7K  $\Omega$ )が実装されており、外付けで終端抵抗を実装する必要はありません。
- 注4) PROGB, INITB, DONE 信号には、Module 内部に Pullup 抵抗が実装されており、外部に実装する必要はありません。  
ただし、uSDCONF1A03 をソケットから取り外して運用することがある場合は、外付けに終端抵抗を取り付けて下さい。
- 注5) CCLK は最短でレイアウトするようにして下さい。また、GND でシールドして下さい。  
FPGA の近傍に、Pullup50  $\Omega$ 、Pulldown50  $\Omega$  のテブナン終端をして下さい。
- 注6) M[2:0]信号は Slave Serial に設定して下さい。
- 注7) Xilinx の SlaveSerialモードでは nCE が機能しないため、XCS0, XCS1 による制御は行えません。DIN, DOUT によるシリアルデータ線のチェーンの構成をとって下さい。

## 5.2. uSDCONF1A03 Multi FPGA 接続例

## (1) 8 分割時

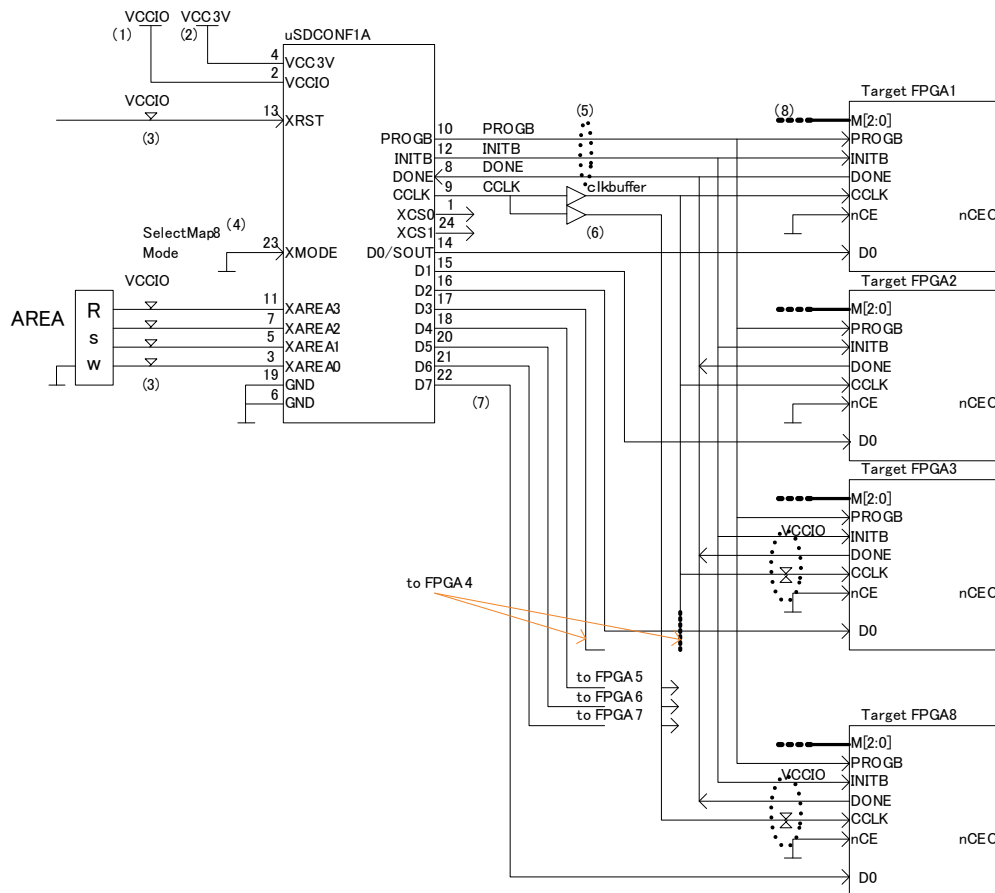


図 5.4.1 Multi FPGA x8 接続例

- 注1) VCCIO は 3.3V~2.5V、1.8V に対応しています。  
VCCIO に 1.8V を使用する場合は、Module のデータを 1.8V 用に書き変える必要があります。
- 注2) 3.3V ± 5%(0.165V)が推奨入力電圧となります。
- 注3) Xrst,XMODE,XAREA[3:0]ピンには、内部 Pullup 抵抗(4.7K Ω)が実装されており、外付けで終端抵抗を実装する必要はありません。
- 注4) 本モードでは XMODE ピンは SelectMap8 に設定して下さい。
- 注5) PROG,INIT,DONE 信号には、Module 内部に Pullup 抵抗が実装されており、外部に実装する必要はありません。
- 注6) CCLK は最短でレイアウトするようにして下さい。また、GND でシールドして下さい。  
FPGA4 個に対して 1 個の CLK バッファを挿入して下さい。  
CLK バッファ毎に一番離れた FPGA の近傍に、Pullup50 Ω、Pulldown50 Ωのテブナン終端をして下さい。
- 注7) FPGA の接続個数が 8 個に満たない場合、分割損が生じます。
- 注8) M[2:0]の設定は Slave Serial モードに設定して下さい。



## (2) 4 分割時

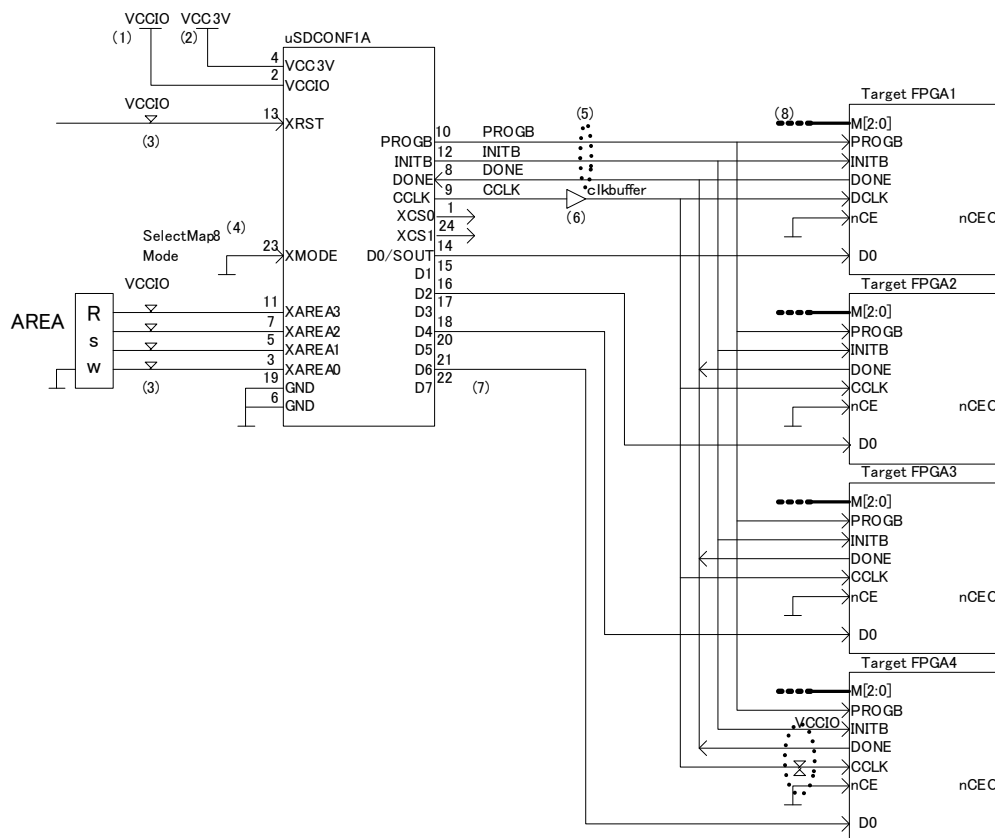


図 5.4.2 Multi FPGA x4 接続例

- 注1) VCCIO は 3.3V~2.5V、1.8V に対応しています。  
VCCIO に 1.8V を使用する場合は、Module のデータを 1.8V 用に書き変える必要があります。
- 注2) 3.3V ± 5%(0.165V)が推奨入力電圧となります。
- 注3) Xrst,XMODE,XAREA[3:0]ピンには、内部 Pullup 抵抗(4.7K Ω)が実装されており、外付けで終端抵抗を実装する必要はありません。
- 注4) 本モードでは XMODE ピンは SlaveSelectMap8 に設定して下さい。
- 注5) PROG B,INIT B,DONE 信号には、Module 内部に Pullup 抵抗が実装されており、外部に実装する必要はありません。  
DCLK は最短でレイアウトするようにして下さい。また、GND でシールドして下さい。  
FPGA4 個に対して 1 個の CLK バッファを挿入して下さい。  
CLK バッファ毎に一番離れた FPGA の近傍に、Pullup50 Ω、Pulldown50 Ωのテブナン終端をして下さい。
- 注6) M[2:0]の設定は、SlaveSerial モードに設定して下さい。
- 注7) microSD カードの実効読出速度が 200Mbps に対して、50Mbps×2=100Mbps で、PS モードを 1 ~ 4 個において、書き込み速度を犠牲にすることなく効率的にコンフィグレーションすることが可能です。

## (3) 2 分割時

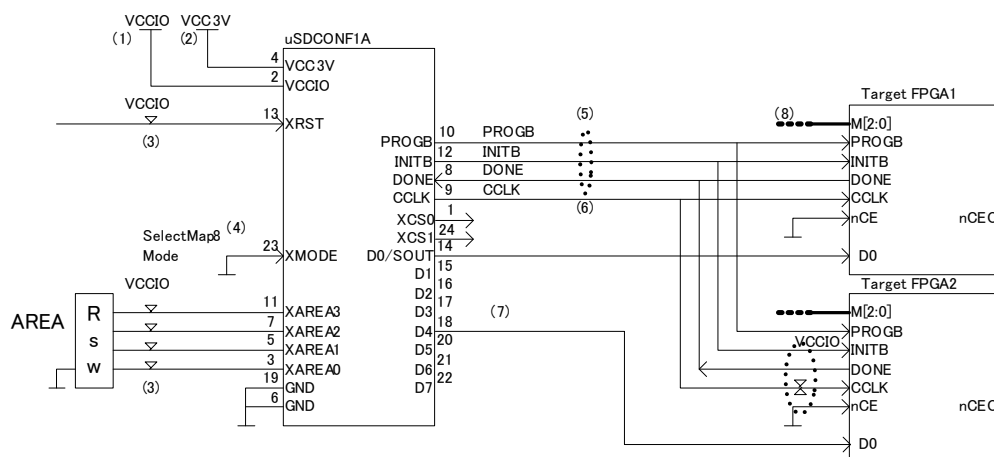
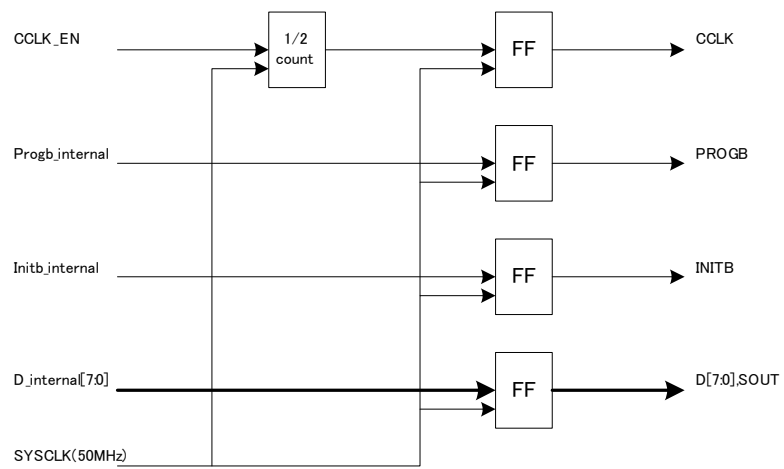


図 5.4.3 Multi FPGA x2 接続例

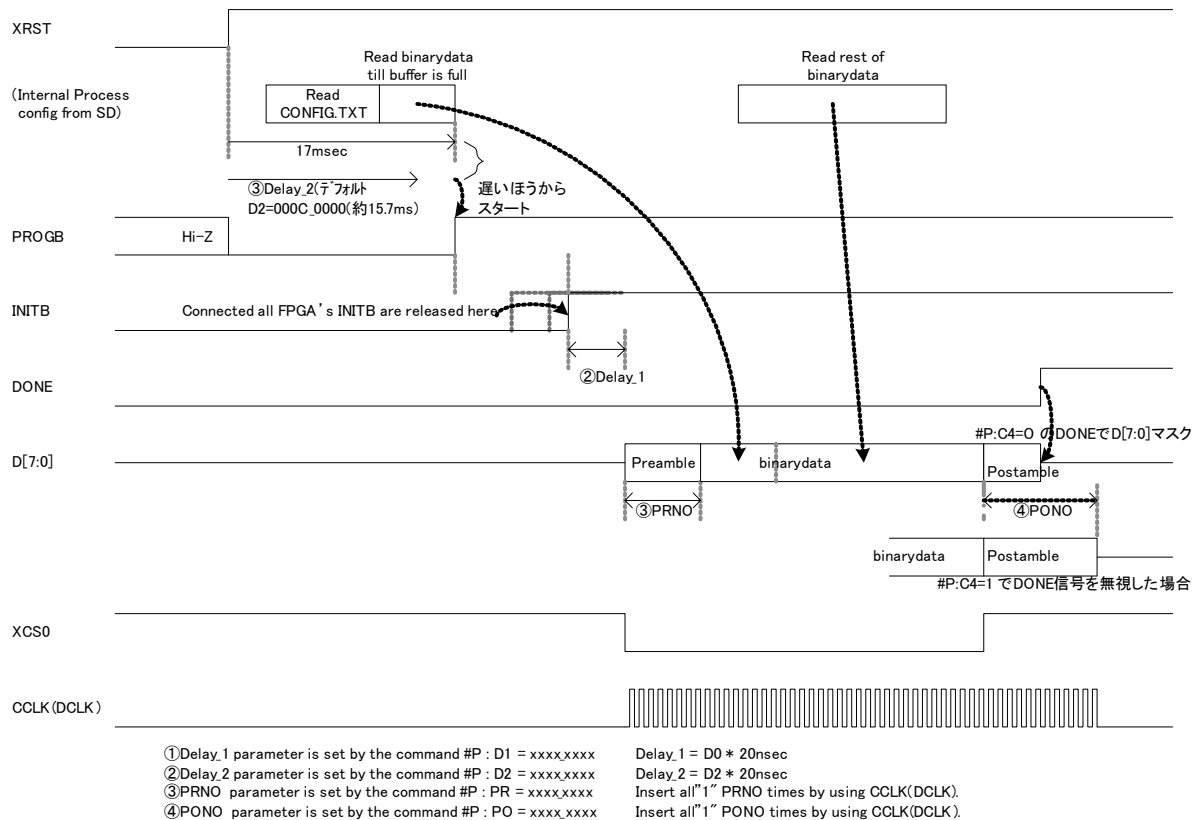
- 注1) VCCIO は 3.3V~2.5V、1.8V に対応しています。  
VCCIO に 1.8V を使用する場合は、Module のデータを 1.8V 用に書き変える必要があります。
- 注2) 3.3V ± 5%(0.165V)が推奨入力電圧となります。
- 注3) Xrst,XMODE,XAREA[3:0]ピンには、内部 Pullup 抵抗(4.7KΩ)が実装されており、外付けで終端抵抗を実装する必要はありません。
- 注4) 本モードでは XMODE ピンは SelectMap8 に設定して下さい。
- 注5) PROG B,INIT B,DONE 信号には、Module 内部に Pullup 抵抗が実装されており、外部に実装する必要はありません。  
CCLK は最短でレイアウトするようにして下さい。また、GND でシールドして下さい。  
FPGA4 個に対して 1 個の CLK バッファを挿入して下さい。  
CLK バッファ毎に一番離れた FPGA の近傍に、Pullup50Ω、Pulldown50Ωのテブナン終端をして下さい。
- 注6) M[2:0]の設定は、Slave Serial モードに設定して下さい。
- 注7) microSD カードの実効読出速度が 200Mbps に対して、50Mbps×2=100Mbps で、PS モードを 1~4 個において、書き込み速度を犠牲にすることなく効率的にコンフィグレーションすることが可能です。

## 6. タイムチャート

### 6.1. 信号出力部ブロック図

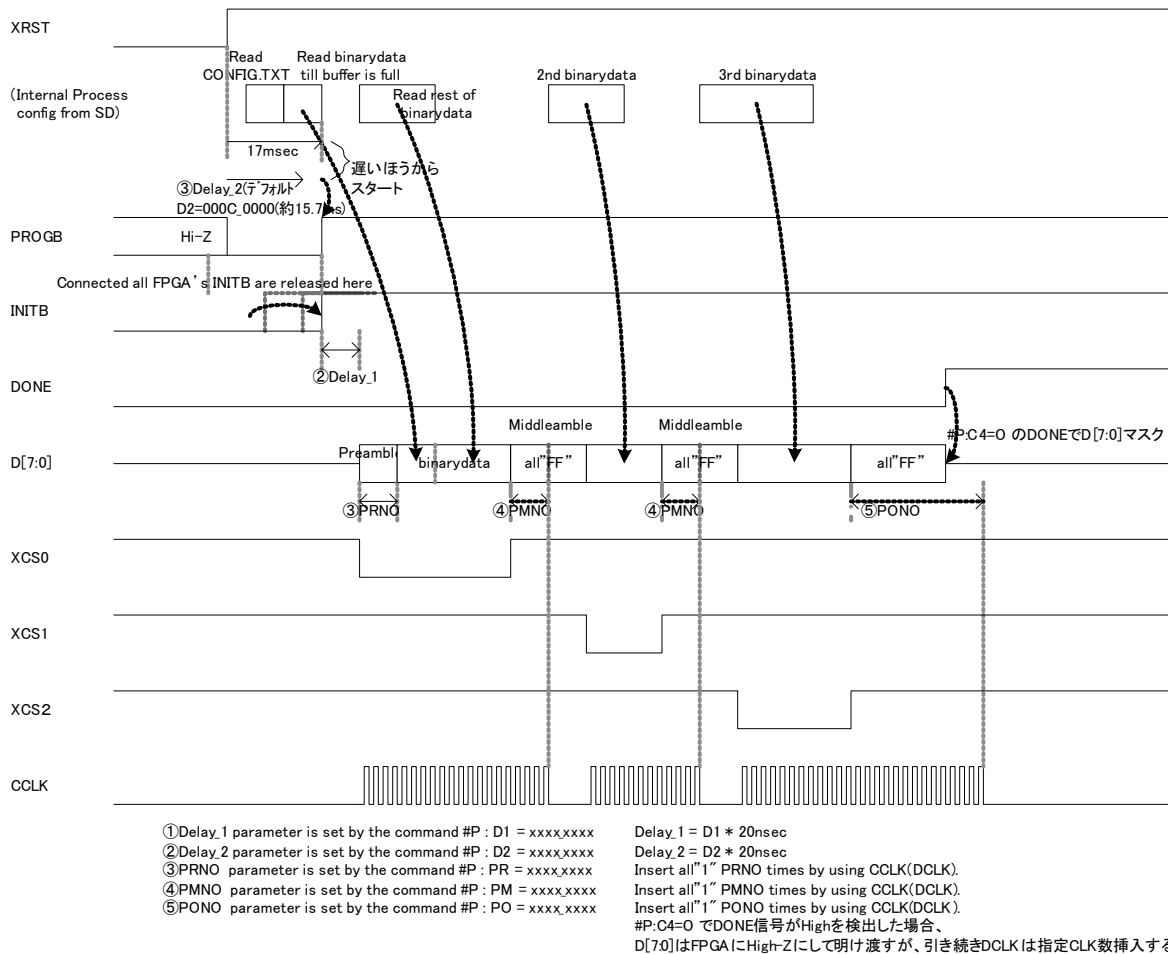


## 6.2. タイムチャート



- XRST 信号は 10msec 以上アサートを推奨しています。また、チャタリングのない信号を入力して下さい。
  - リセット期間中、PROGB は High-Z の状態となります。
  - 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。
  - INITB は、Ver1.800 より出力から入力方向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。(指定しても読み飛ばされます。)
- 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。
- 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。
  - 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 Postamble として ALL"1"を出力します。
  - 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。

## (2) SD カードから BinaryData を3個読み出し転送の場合



- XRST 信号は 10msec 以上アサートを推奨しています。また、チャタリングのない信号を入力して下さい。
  - リセット期間中、PROGB は High-Z の状態となります。
  - 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。
  - 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。(指定しても読み飛ばされます。)
- 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。
- 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。
  - 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、DCLK をを出力するとともに、Data バスに ALL"1"を出力します。
- PM パラメータは、バイナリデータと次のバイナリデータとの間(Middle)に ALL"1"を PM で指定した個数 CCLK を挿入します。(Middle amble)

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。

- 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。

- 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。
- 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 に受け渡します。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

- ・ uSD-CONF1 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。

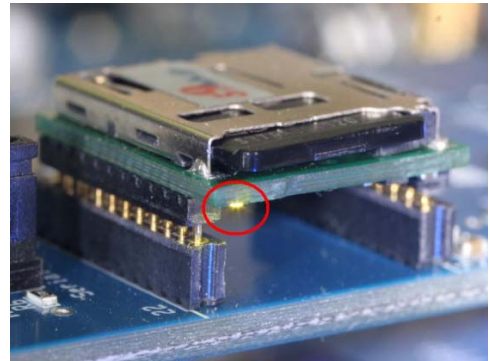


図 7.1 LED の実装位置

### 7.1. LED の点灯条件

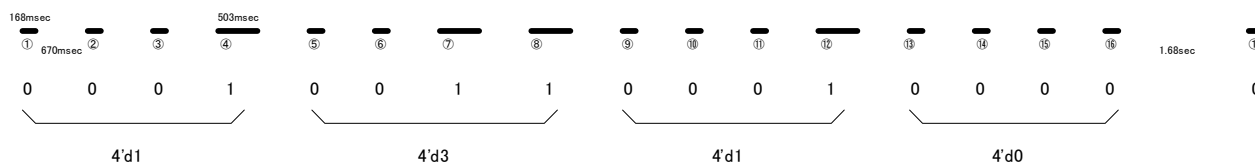
- ・ LED の点灯条件は以下の通りです。

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 7.1 LED 点灯条件

## 7.2. 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 7.2 LED 点滅例



## 8. “CONFIG.TXT”の準備

### 8.1. “CONFIG.TXT”とは

- microSD 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、microSD 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

- microSD 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:		0~9	DCLK 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	nCONFIG~nSTATUS までのデレイ間隔(削除)
	D1=	xxxx_xxxx	nSTATUS~Data 出力開始までのデレイ間隔
	D2=	xxxx_xxxx	XRST~nCONFIG までのデレイ間隔
	C0=	0/1	ワードアライナ機能
	C1=	0/1	nCONFIG,nSTATUS 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

### (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 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、microSD 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。
- 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。
- ・ 指定されても読み飛ばされ、エラーとはなりません。
- ・ ~~PROGB がリリースされて INITB がリリースされるまでのデ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。
- ・ デフォルトは 000C\_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] => 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

- 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 uSDCONF1A03 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 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 固定

- ・ ”1”にセットすると、XCS0,XCS1 の 2 本の制御信号により、外付け回路を制御し、7本の XCS[6:0]信号を生成し、最大7個の FPGA に対してマルチにコンフィグレーションを行うことが可能となります。

## 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。

## (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。)

## (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。
- ・ 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外付け無しで2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
- ・ 外付け回路を追加することにより、最大 7 個までの FPGA のコンフィグレーションを行うことが可能です。
- ・ “+”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。

uSDCONF1A03 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。(uSDCONF1A03 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。

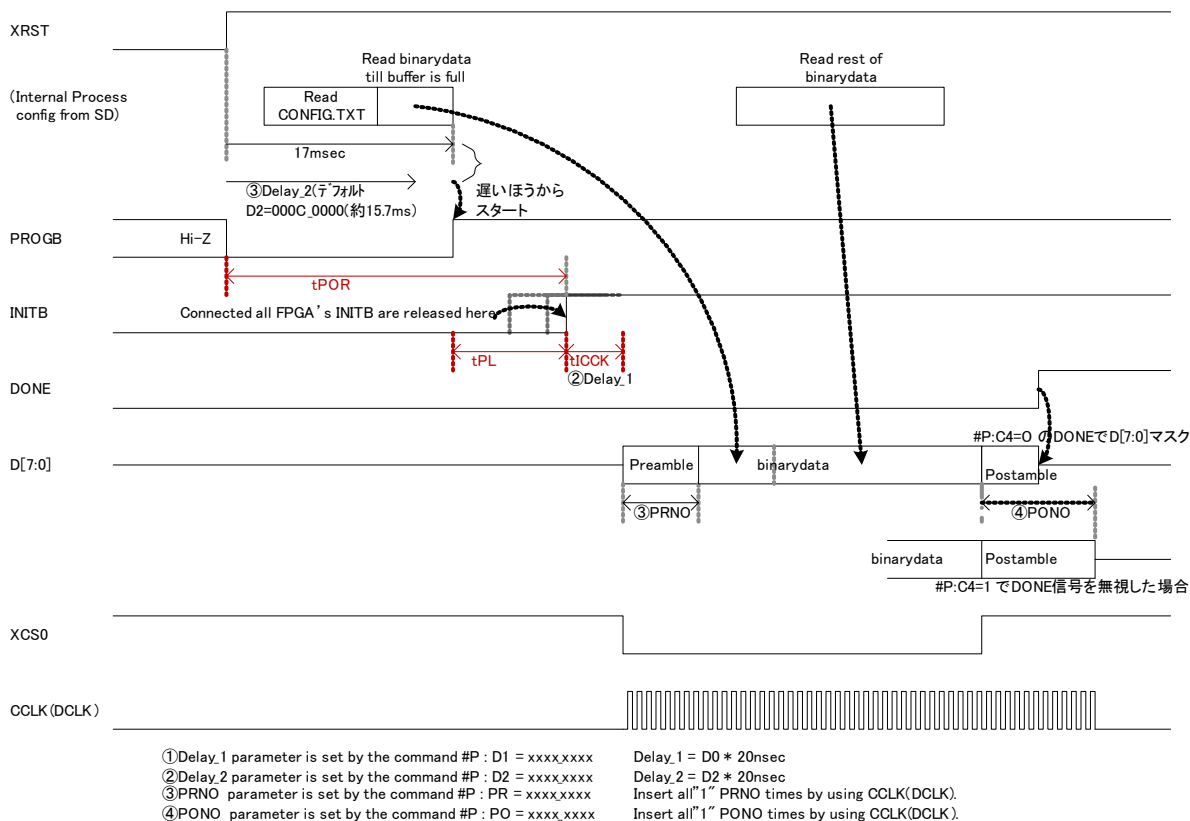


図 8.3 パラメータ位置関係

FPGA Series	tPOR	tPL	tICCK	CLKUSR	D1	D2	PO
Virtex-5	10 $\leq$ , $\leq$ 50ms	$\leq$ 3ms	(400ns $\leq$ )		0000_0010 =320nS(def)	000C_0000 =15.7mS(def)	0010_0000 (def)
Virtex-6	15 $\leq$ , $\leq$ 55ms	$\leq$ 5ms	(400ns $\leq$ )		0000_0010 =320nS(def)	000C_0000 =15.7mS(def)	0010_0000 (def)
Virtex-7	10 $\leq$ , $\leq$ 50ms	$\leq$ 5ms	(150ns $\leq$ )		0000_0010 =320nS(def)	000C_0000 =15.7mS(def)	0010_0000 (def)
Kintex-7	10 $\leq$ , $\leq$ 50ms	$\leq$ 5ms	(150ns $\leq$ )		0000_0010 =320nS(def)	000C_0000 =15.7mS(def)	0010_0000 (def)
Artix-7	10 $\leq$ , $\leq$ 50ms	$\leq$ 5ms	(150ns $\leq$ )		0000_0010 =320nS(def)	000C_0000 =15.7mS(def)	0010_0000 (def)
Spartan-6	5 $\leq$ , $\leq$ 40ms	$\leq$ 4ms	(-- )		0000_0010 =320nS(def)	000C_0000 =15.7mS(def)	0010_0000 (def)

表 8.5 D1,D2,P0 設定例

## 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(XILINX)

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

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)

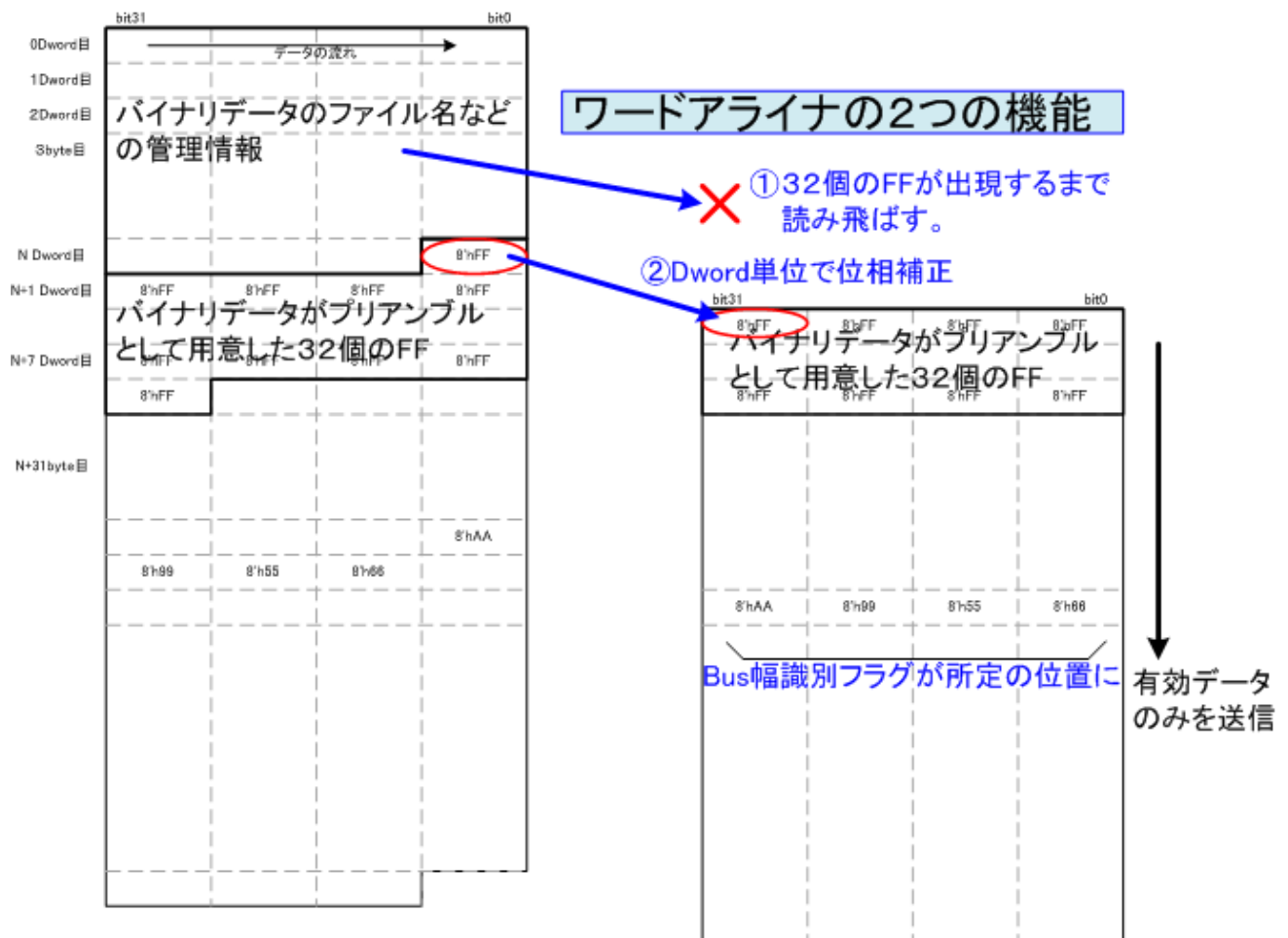


図 9.1. ワードアライナ機能説明図

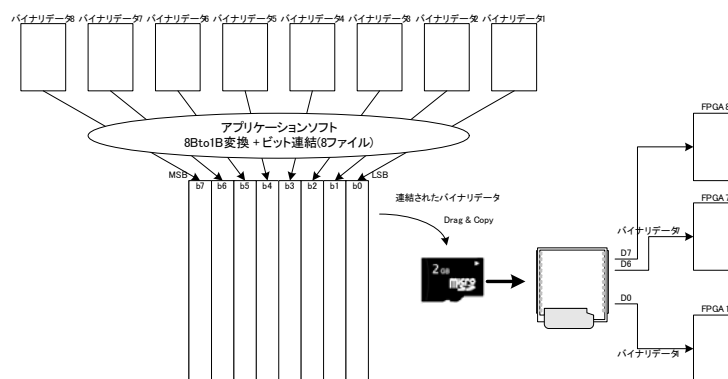
## 9.2. bit 連結による Multi FPGA Configuration 機能

- 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。
- microSD 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。

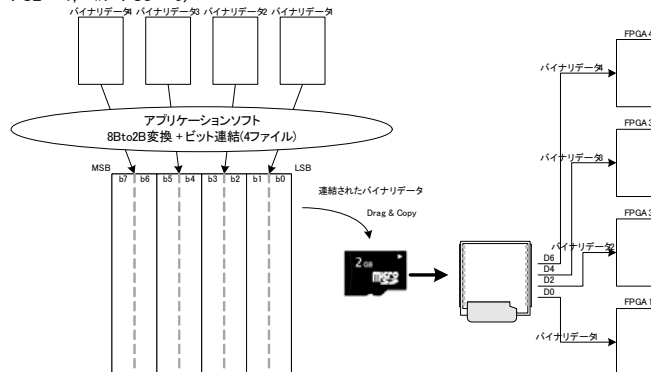
分割個数が 8 個の場合で、FPGA 割り当てに空きがある場合や、それぞれの FPGA のサイズが異なる場合、Daisy-Chain 方式より効率が悪くなる場合があります。

### 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)

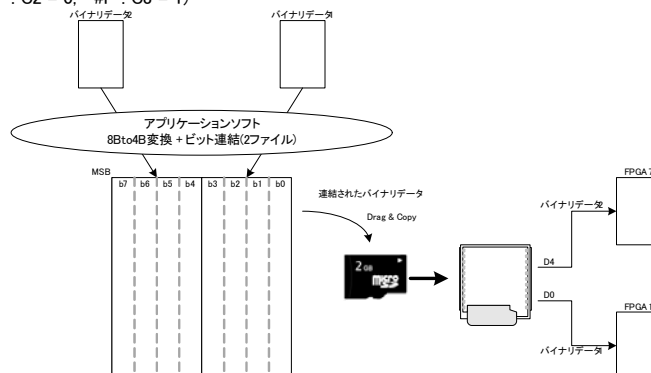


図 9.2. bit 連結による Multi FPGA Configuration の動作原理図



### 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。

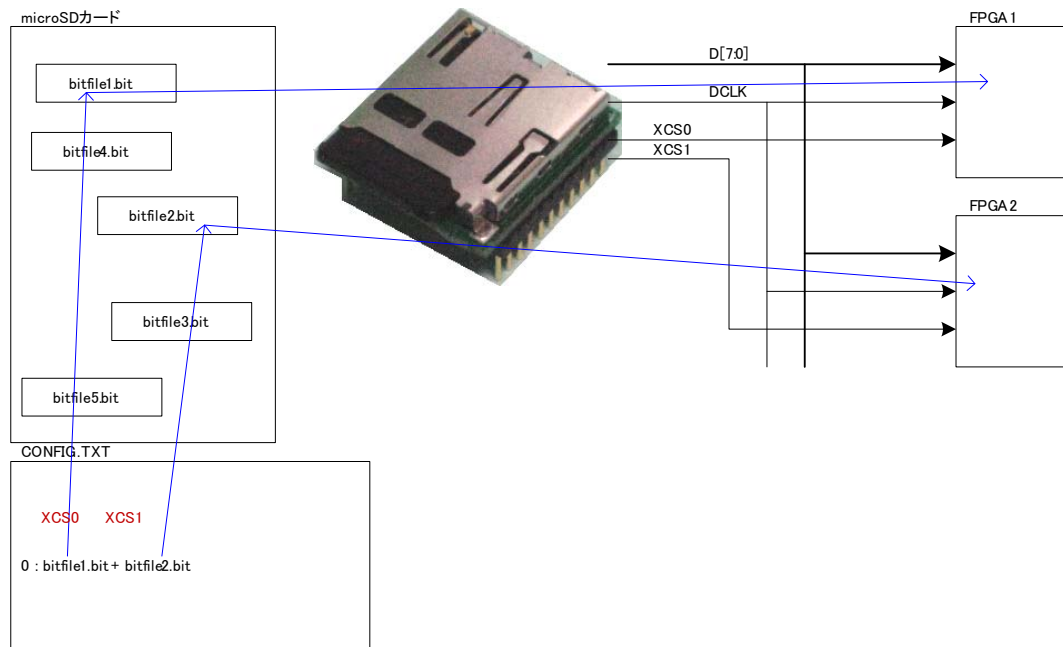


図 9.3 “CONFIG.TXT”による Multi FPGA Configuration の動作原理図

### 9.4. 外付け回路による最大 7 個の Multi FPGA Configuration 機能

- 下記外部回路を実装し、かつ、“CONFIG.TXT”ファイル上で#p:C7=1 を設定すると、外部で生成した XCS[6:0]信号により、最大7個の FPGA をマルチにコンフィグレーションすることが可能になります。

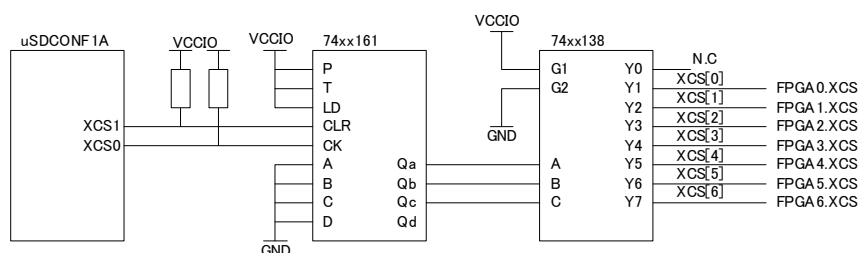


図 9.4

## 9.5. SDCard の活線挿抜について

### 9.5.1. 構成図

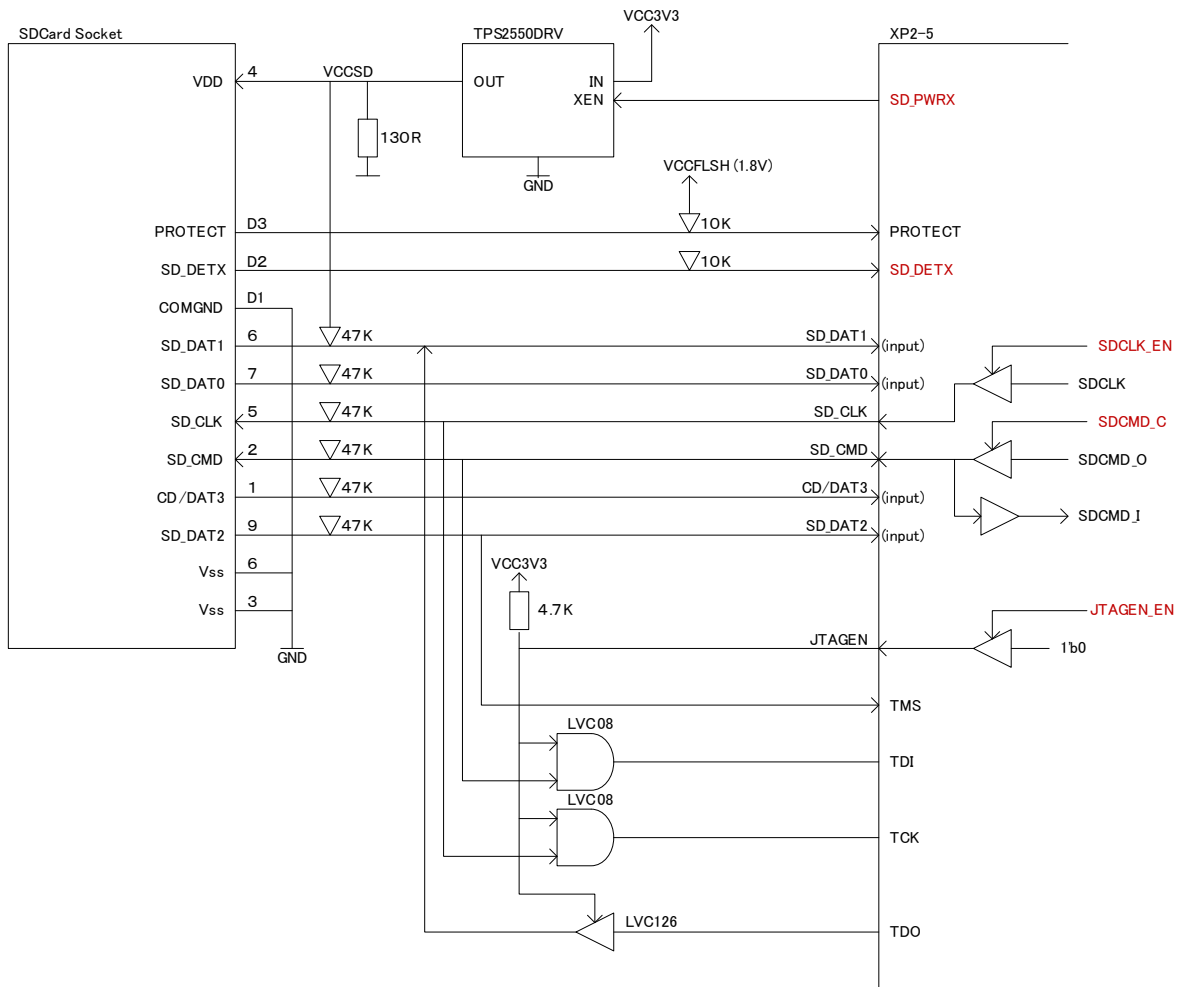


図 9.5.1 SDCard インタフェース部構成図

## 9.5.2. 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 4.7KΩ 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。

## 10. uSDCONF1A03 と uSDCONF1F の共存について

### 10.1. uSDCONF1F 用の基板に uSDCONF1A03 を実装する場合

#### 10.1.1. uSDCONF1F 用基板に uSDCONF1A03 を新規にアセンブリする場合



図 10.1

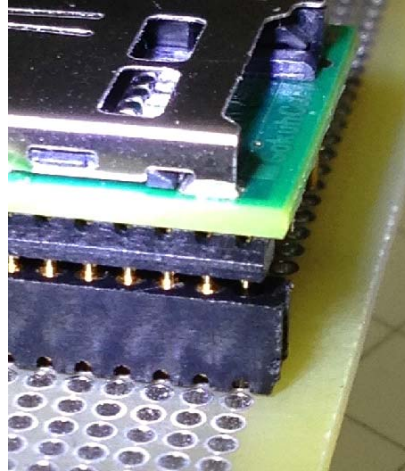


図 10.2

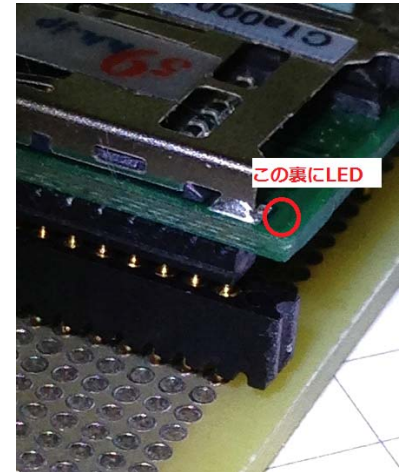


図 10.3

- ・ 図 10.1 は、添付されるの MAC8 の ME-10-10-12P ソケットから 1pin を取り外した写真です。取り外しは、ラジオペンチの先で pin 先を挿入方向に沿って押すと簡単に外せます。
- ・ 取り外した ME-10-10-12P ソケットを、取り外した側を前面にして実装します。(図 10-2)
- ・ ソケットに uSDCONF1A03 を実装すると、前面側の手前の pin(図 10.2 の写真では PIN24)が、図 10.1 でピンを抜いたモールドの穴部分に隠れ、静電気や外圧から守られます。
- ・ ソケット前面端は、ME-10-10-11P ソケットを実装した場合より、前面に約 1.27mm ほどモールド部分が飛び出します。
- ・ 図 10.3 は、uSDCONF1F を実装した写真です。uSDCONF1A03 の 24pin が存在する箇所に LED が存在しますが、コネクタ間との距離も十分保たれています。
- ・ ME-10-10-12P ソケットは、uSDCONF1A03 前面先端内に位置していますが、ME-10-10-11P ソケット実装時の前面先端から 1.27mm 以内に部品が配置されている場合は、モールド部分に部品が干渉して本方式を適用できないかもしれません。その場合は、モールド部分の干渉部分をカットするか、uSDCONF1A03 の 1pin、24pin をカットする必要があります。
- ・ ご購入検討時にご相談下さい。

## 10.1.2. uSDCONF1F 用ソケットが既実装されている場合



図 10.4

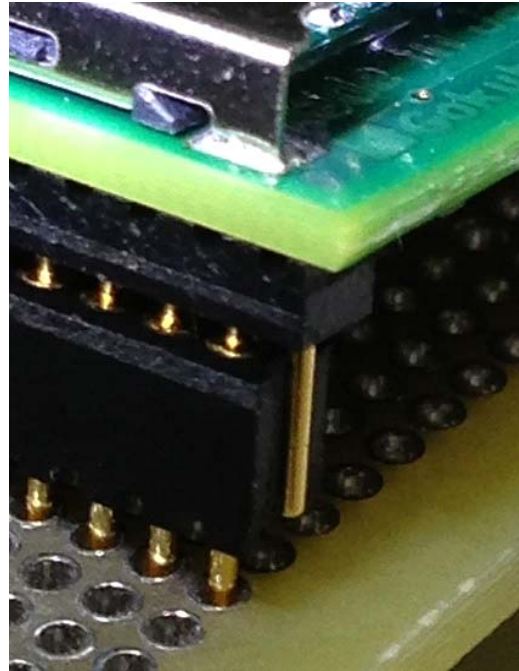


図 10.5

- ・ 図 10.4 のように、ソケットの両端は、下側のようにピンを取り外してカットされて曲面を持った側と、上側のように、ベースとなるソケットの終端部分で曲面のない先端をもつタイプがあります。
- ・ 図 10.4 の下側の場合、図 10.5 のようにモールドの内側に湾曲した部分にピンが潜り込みますが、図 10.4 の上側の先端が前面にきている場合、ピンが外側に押し出されて曲がってしまいます。
- ・ この場合、下記より方法を選択して下さい。
  - ソケットを取り外し、12pin 用に付け直す。
  - モールド部分の接触部分をルーター等で削り、接触しないようにする。
  - uSDCONF1A03 の前面外側のピンをカットする。
- ・ 上記処理は、悟空株式会社で行うことが可能です。(有償となります。)
- ・ また、カットしたピンの付け直しも行います。(有償となります。)

## 11. モジュールのソケットからの取り外しについて

- ・ uSDCONF1A03 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 11.1.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。

### 11.1. モジュール裏面

- ・ “59kk”のシールの下に制御用のチップが実装されています。  
その両側の赤枠で囲った部分には、チップ抵抗やチップコンデンサ、ロジックチップが密集して実装されています。



図 11.1. uSDCONF1A03 Bottom View

### 11.2. モジュールの取り外し方

- ・ ドライバ等でモジュールをソケットから抜き取る場合、図 11.1 の部品配置を常にイメージして、赤枠部分に力がかからないよう十分に注意して下さい。
- ・ “59kk”のシールの貼られた制御チップの高さは 1.3mm あり、他の部品より背高です。  
ドライバ等で引き抜かれる場合は、こちらのチップに垂直に持ち上げる方向に力がかかるように、ゆっくりと、そして四隅が均等に持ち上がるように少しずつ操作を繰り返して下さい。
- ・ 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。
- ・ ドライバ等で取り外す場合、図 11.2 のようにモジュール下への差し込みが十分でない場合、図 11.1 の赤枠の部分に作用点が働いてしまうとチップ部品を破損してしまいます。  
また、下の基板に部品が配置されている場合、下の基板の部品を破損してしまう恐れがあります。
- ・ 図 11.3 のように、ピンセットをモジュールの下を通し、両側から少しずつ持ち上げるとスムーズに取り外せます。

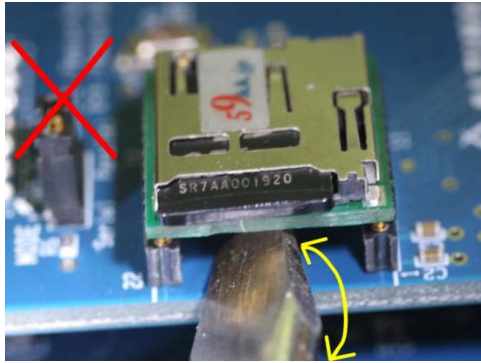


図 11.2 取り外しの悪い例

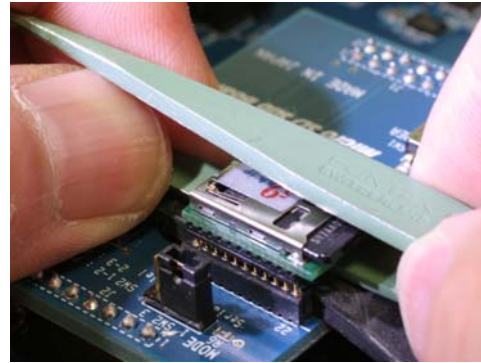


図 11.3 ピンセットを使った例

- ・ 図 11.4 は D.I.Y で使用する“目地用コーキングヘラ”です。
- ・ 図 11.5 はヘラをモジュール下に滑り込ませた写真です。  
ソケットの台座部分に滑り込んでいるのが判ります。  
この状態で両側から両手で少しずつ持ち上げると部品に接触することなく取り外しが可能です。
- ・ こちらのヘラは、Amazon やコーナン PRO などでお買い求めいただけます。  
下記品名で検索をかけると Amazon やコーナン PRO が上位にヒットします。  
尚、モジュールご注文の際に弊社で調達することも可能です。ご入り用の際は弊社までお問い合わせください。

品名：目地用コーキングヘラ曲り  
 発売元：株式会社ハンディ・クラウン  
 商品コード: 309018 0000  
 注文型番：4905533-152782



図 11.4 目地用コーキングヘラ



図 11.5 ヘラ使用例

## 12. 添付品



図 12.1



図 12.2

- 図 12.1 は製品として出荷される構成要素です。
  - uSDCONF1A03 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
  - MAC-8 ME-10-10-12P 若しくは ME-1-10-12P 若しくは 2200SA-12G-34 ソケット 2個
  - 静電マット
  - 乾燥剤+静電袋+プリスターケース
- 図 12.2 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。



### 13. VCCIO 電圧とシールの色

- ・ uSDCONF1A03 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)
- ・ ゲートサイズの識別を行う必要があるため、uSDCONF1A03 より、表 13.1 のように割り振りを変更しました。

ゲートサイズ	VCCIO	インターフェース	シール
5	2.5V ~ 3.3V	LVC MOS2.5V ~ 3.3V	青色
5	1.8V	LVC MOS1.8V	黄色
8	2.5V ~ 3.3V	LVC MOS2.5V ~ 3.3V	白色
8	1.8V	LVC MOS1.8V	黄色

表 13.1

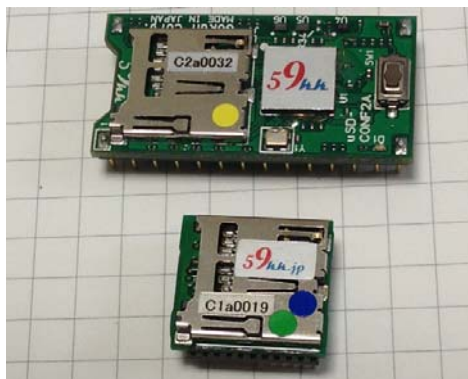


図 13.1 シール添付例

### 14. 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。

ゲートサイズ	VCCIO 電圧	梱包形態	発注型格
5	2.5V ~ 3.3V	プリスターケース単体梱包	uSDCONF1A03-5-33V
		静電マット平面梱包	uSDCONF1A03-5-33VN
	1.8V	プリスターケース単体梱包	uSDCONF1A03-5-18V
		静電マット平面梱包	uSDCONF1A03-5-18VN
8	2.5V ~ 3.3V	プリスターケース単体梱包	uSDCONF1A03-8-33V
		静電マット平面梱包	uSDCONF1A03-8-33VN
	1.8V	プリスターケース単体梱包	uSDCONF1A03-8-18V
		静電マット平面梱包	uSDCONF1A03-8-18VN

表 14.1 発注型格表

## 15. 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 microSD 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 microSD 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 microSD 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 microSD 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。)

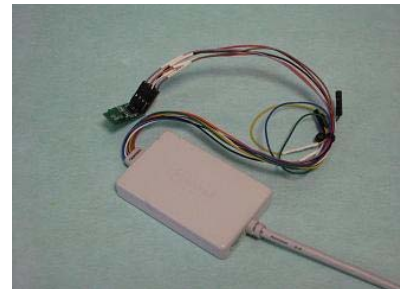
- ④ bit 連結による Multi FPGA Configuration 機能はインプリメントされていますが、バイナリデータを連結するためのアプリケーションソフトを準備中です。
- ⑤ CCLK のクロック速度選択機能で、CCLK を受信して動作するマスタモードは定義されていますが、現在インプリメント作業中です。

## 16. 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。

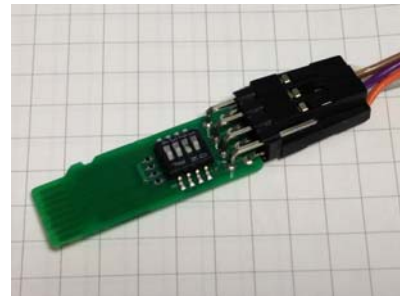
“59kk”のシールの貼られていないモジュールにつきましては、暗号 Key の書き込みが必要です。弊社もしくは担当営業にご相談下さい。

(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
  - ◇ コンフィグレーション終了後の microSD 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



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)

〒 222-0033

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>