

microSD-CONF1 データシート

(for Xilinx ユーザ)

(USDCNF-MD1F-001A)

(USDCNF-MD1H-001A)

(USDCNF-MD1V-001A)

Rev 0.85 2011.09.07

Rev 0.81 2011.08.09

Rev 0.80 2011.07.07

Rev 0.7 2011.06.30

Rev 0.6 2011.06.24

Rev 0.4 2011.06.09

Rev 0.3 2010.09.21

悟空株式会社

改訂履歴

日時	担当者	Rev	改訂内容概要
2010/08/05	大庭	0.1	新規作成 (草案)
2010/08/23	大庭	0.2	
2010/09/21	大庭	0.3	5.接続例修正。 6.タイムチャート追加。 8.制約事項⑤⑥⑦追加。
2011/06/30	大庭	0.7	全面変更。
2011/07/07	大庭	0.80	Lattice 追加。誤記修正。
2011/08/09	大庭	0.81	Pin 機能表 内部 Pullup 記述追加。 Altera 接続例追加。
2011/09/07	大庭	0.85	図番号修正。誤記修正、フォント修正。表記統一。

目次

1. 機能・特徴	4
2. 形状	5
3. 電気特性	6
3.1. 絶対定格	6
3.2. 推奨動作範囲	6
3.3. 消費電流	6
3.4. 電源シーケンス	6
4. Pin 機能表	7
4.1. USDCNF-MD1F-001A ピン機能表	7
4.2. USDCNF-MD1V-001A ピン機能表	8
4.3. USDCNF-MD1H-001A ピン機能表	9
4.4. XAREA 詳細	10
5. 接続例	11
5.1. XILINX	11
5.1.1. USDCNF-MD1F-001A 接続例	11
5.1.2. USDCNF-MD1V-001A 接続例	12
5.1.3. USDCNF-MD1H-001A 接続例	12
6. タイムチャート	13
6.1. 信号出力部ブロック図	13
6.2. タイムチャート	13
7. LED	14
7.1. LED の点灯条件	14
7.2. バージョン情報表示機能	15
8. “CONFIG.TXT”の準備	16
8.1. “CONFIG.TXT”とは	16
8.2. “CONFIG.TXT” サンプル	17
8.3. Commands	18
9. 機能詳細	25
9.1. ワードアライナ機能	25
9.2. bit 連結による Multi FPGA Configuration 機能	26
10. モジュールのソケットからの取り外しについて	27
10.1. モジュール裏面	27
11. 制約事項	29
12. アップデート機能(順次対応予定)	31

1. 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ギュ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 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 microSD カードや、2GByte を超える SDHC には現状対応していません。)
- ・ 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。
- ・ “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~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 をコンフィギュレーションすることが可能です。
- ・ 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(Passive Serial)によるコンフィギュレーション。
 - microSD カードから FPGA への SlaveSelectMAP8(FPP)によるコンフィギュレーション。
 - ・ FPGA への転送クロック CCLK(DCLK)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、及び、Master モード(CCLK が入力)
(Master モードは近日対応予定)
- ・ 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、バイナリデータとして、“.bin”ファイルは元より“.bit”ファイルも指定できます。
- ・ 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。
- ・ PROG B の立ち上がりから INIT B の立ち上がりまでの間隔や、INIT B 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 Daisy-Chain させたシステムにも対応可能です。

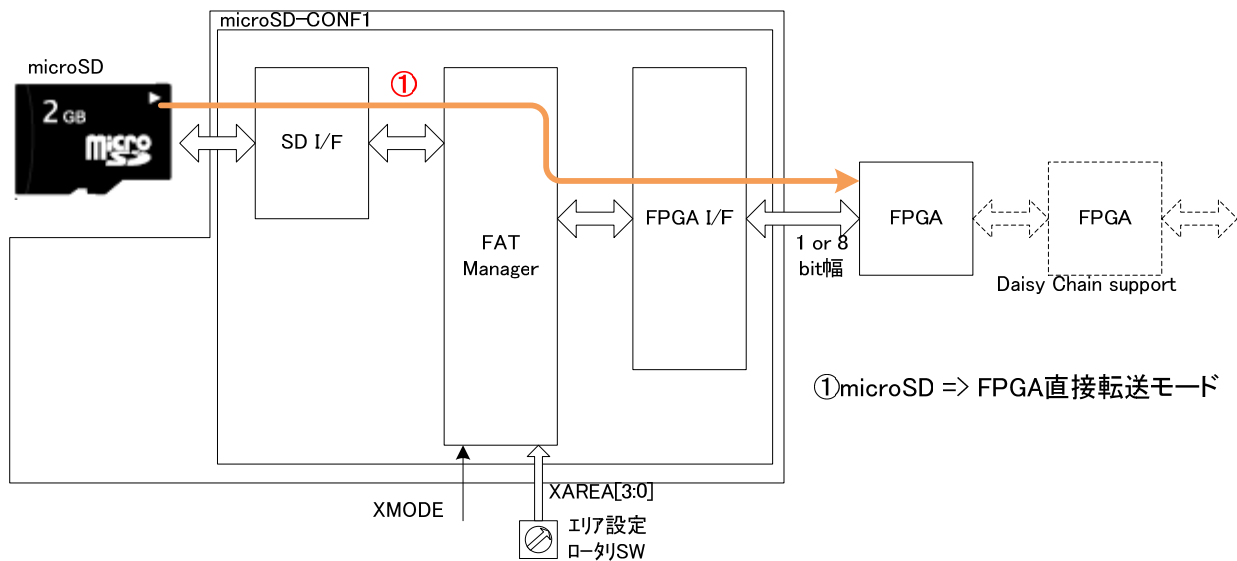


図 1.1 uSDCONF1 ブロックイメージ図

2. 形状

uSD-CONF1 モジュールの形状を掲載します。

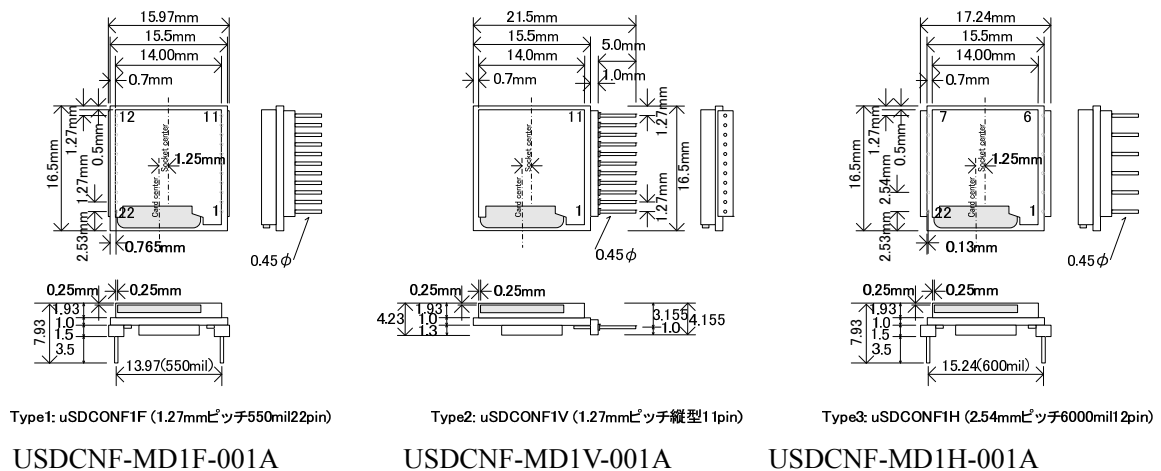


図 2.1 uSDCONF1 形状図

- (1) USDCNF-MD1F-001A の使用 pin は、MAC8 の OQ-11-5-11P。

(<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-10-10-11P(金メッキ品)、MAC8 ME-1-10-11P(スズメッキ品)。

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

Neltron 2200Series

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

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

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-CONF1 モジュールのピン機能表を掲載します。

4.1. USDCNF-MD1F-001A 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	VCCIO	—		入出力信号の VCCIO 電源入力。
2	XAREA0	I		バイナリデータ選択用ロータリーSW(LSB)
3	VCC3V	—		3.3V 電源入力
4	XAREA1	I		バイナリデータ選択用ロータリーSW
5	GND	—		GND
6	XAREA2	I		バイナリデータ選択用ロータリーSW
7	DONE	I	330Ω	FPGA DONE 信号
8	CCLK	O		FPGA CCLK 信号
9	PROGB	O	4.7K	FPGA PROGB 信号
10	XAREA3	I		バイナリデータ選択用ロータリーSW(MSB)
11	INITB	O	4.7K	FPGA INITB 信号
12	XRST	I	4.7K	パワーオンリセット信号
13	D0/SOUT	O		FPGA 8bit Parallel data0 / Serial data
14	D1	O		FPGA 8bit Parallel data1
15	D2	O		FPGA 8bit Parallel data2
16	D3	O		FPGA 8bit Parallel data3
17	D4	O		FPGA 8bit Parallel data4
18	GND	—		GND
19	D5	O		FPGA 8bit Parallel data5
20	D6	O		FPGA 8bit Parallel data6
21	D7	O		FPGA 8bit Parallel data7
22	XMODE	I		モードピン H:Slave Serial / L:Select Map8

表 4.1 USDCNF-MD1F-001A ピン機能表

- 注1) XAREA[3:0]のエリア選択信号、XMODE のモード選択信号には内部 Pullup は実装されていませんので、外部で Pullup して下さい。(例 10KΩ、4.7KΩ)
- 注2) DONE,PROGB,INITB の 3 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がありません。

4.2. USDCNF-MD1V-001A ピン機能表

垂直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	VCCIO	—		入出力信号の VCCIO 電源入力。
2	XAREA0	I		バイナリデータ選択用ロータリーSW(LSB)
3	VCC3V	—		3.3V 電源入力
4	XAREA1	I		バイナリデータ選択用ロータリーSW(MSB)
5	GND	—		GND
6	SOUT	O		FPGA Serial Data
7	DONE	I	330Ω	FPGA DONE 信号
8	CCLK	O		FPGA CCLK 信号
9	PROGB	O	4.7K	FPGA PROGB 信号
10	XRST	I		パワーオンリセット信号
11	INITB	O	4.7K	FPGA INITB 信号

表 4.2 USDCNF-MD1V-001A ピン機能表

- 注1) XAREA[1:0]のエリア選択信号には内部 Pullup は実装されていないので、外部で Pullup して下さい。(例 10KΩ、4.7KΩ)
- 注2) 本モジュールではエリアは 0 から 3 までの選択範囲となります。
- 注3) DONE,PROGB,INITB の 3 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がありません。
- 注4) 本モジュールは受注生産となります。

4.3. USDCNF-MD1H-001A ピン機能表

水平 2.54mm 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	VCCIO	—		入出力信号の VCCIO 電源入力。
2	VCC3V	—		3.3V 電源入力
3	GND	—		GND
4	DONE	I	330Ω	FPGA DONE 信号
5	PROGB	O	4.7K	FPGA PROGB 信号
6	INITB	O	4.7K	FPGA INITB 信号
7	XRST	I	4.7K	パワーオンリセット信号
8	CCLK	O		FPGA CCLK 信号
9	SOUT	O		FPGA Serial Data
10	GND	—		GND
11	XAREA0	I		バイナリデータ選択用ロータリーSW(LSB)
12	XAREA1	I		バイナリデータ選択用ロータリーSW(MSB)

表 4.3 USDCNF-MD1H-001A ピン機能表

- 注1) XAREA[1:0]のエリア選択信号には内部 Pullup は実装されていないので、外部で Pullup して下さい。(例 10KΩ、4.7KΩ)
- 注2) 本モジュールではエリアは 0 から 3 までの選択範囲となります。
- 注3) DONE,PROGB,INITB の 3 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がありません。
- 注4) 本モジュールは受注生産となります。

4.4. 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 進数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 (uSDCONF1 ではバイナリデータ)の対応表です。
- uSDCONF1V や uSDCONF1H は XAREA[1:0]しか入力がなく、モジュール内部では XAREA[3:2]=2'b11; として処理され、エリア 0 からエリア 3 までが有効範囲となります。

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. USDCNF-MD1F-001A 接続例

5.1.1.1. Slave Serial/SelectMap8 モード選択式

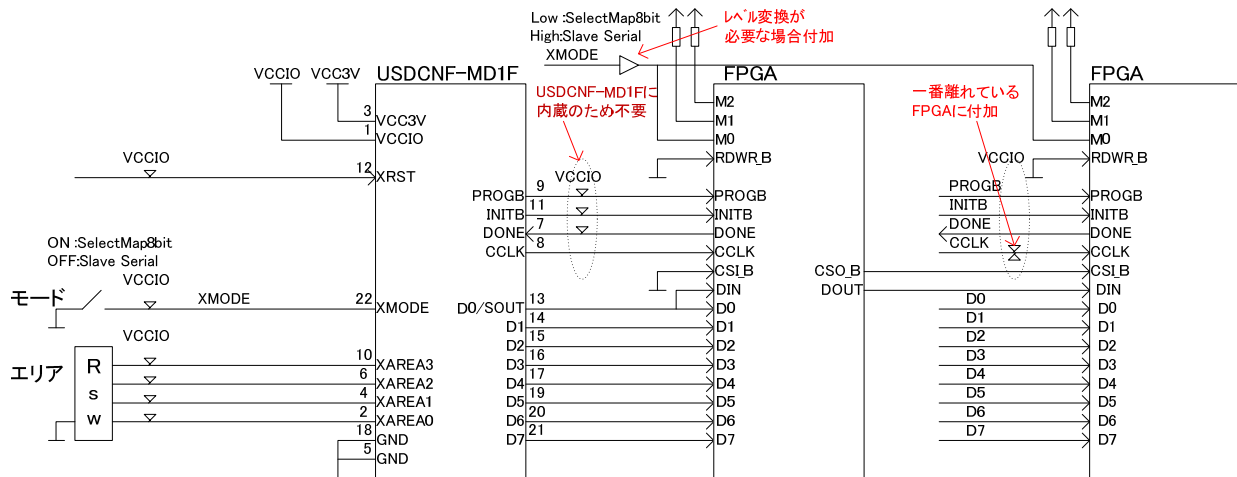


図 5.1.1.1 SlaveSerial/SelectMap8 モード選択式の接続図

5.1.1.2. Slave Serial モード固定

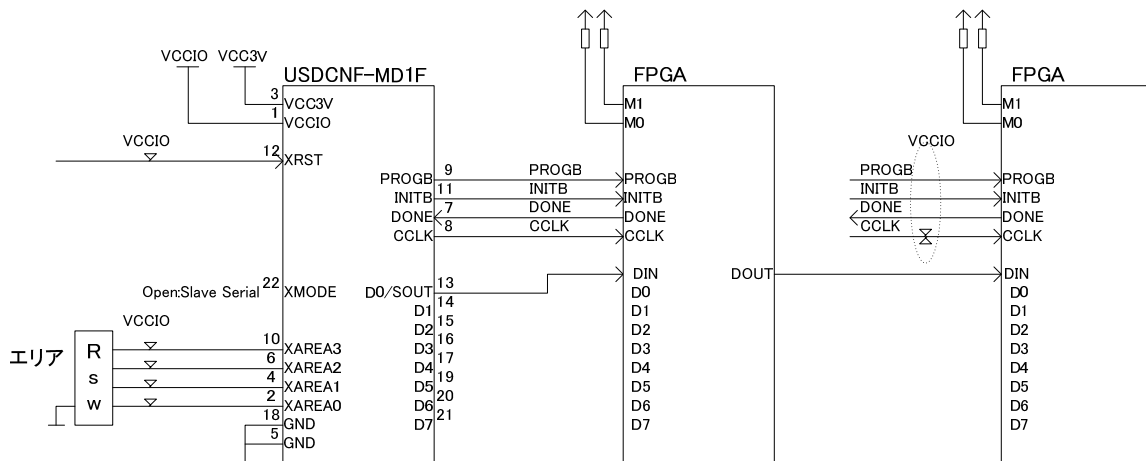


図 5.1.1.2 SlaveSerial モード固定式の接続図

5.1.2. USDCNF-MD1V-001A 接続例

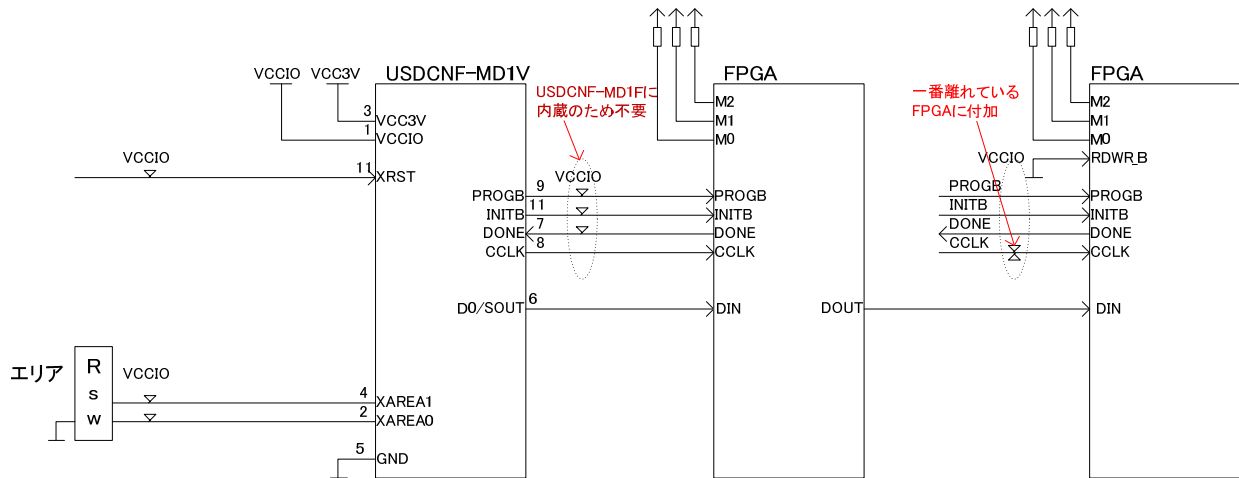


図 5.1.2 USDCNF-MD1V-001A 接続例

5.1.3. USDCNF-MD1H-001A 接続例

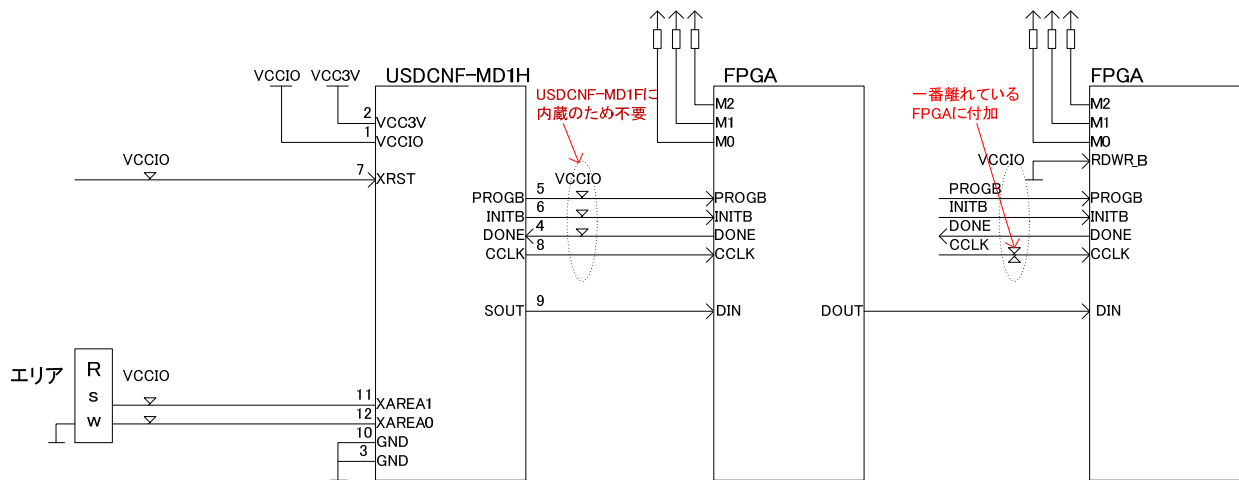
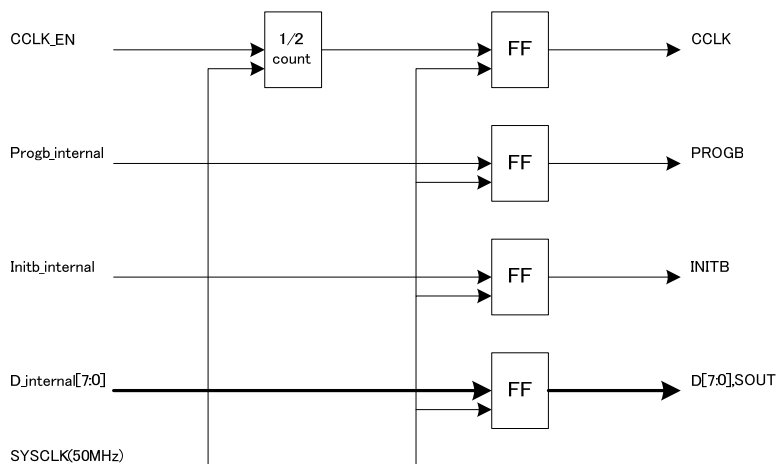


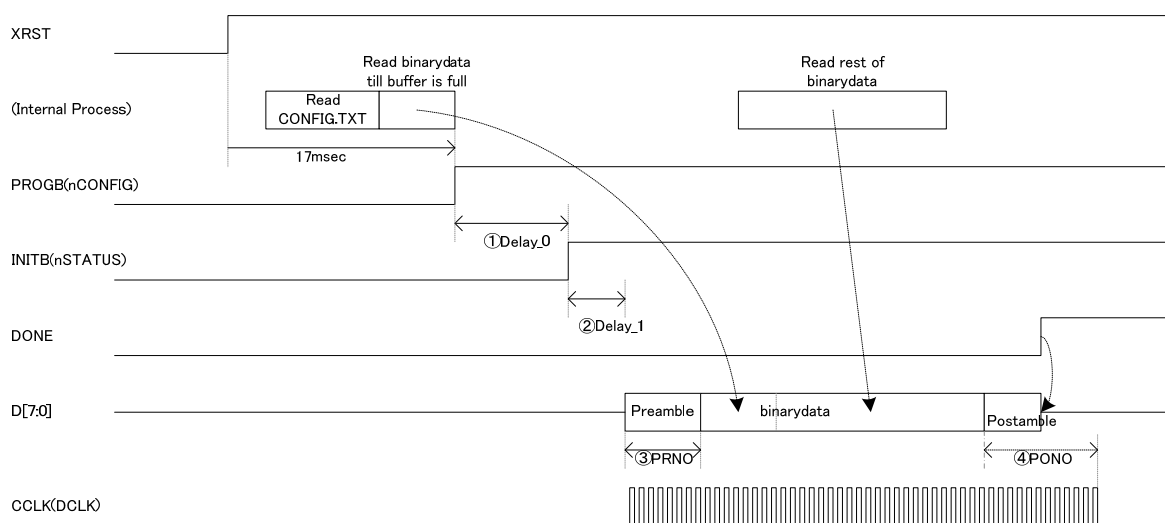
図 5.1.3 USDCNF-MD1H-001A 接続例

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



- ①Delay_0 parameter is set by the command #P : D0 = xxxx_xxxx
- ②Delay_1 parameter is set by the command #P : D1 = xxxx_xxxx
- ③PRNO parameter is set by the command #P : PR = xxxx_xxxx
- ④PONO parameter is set by the command #P : PO = xxxx_xxxx

- Delay_0 = D0 * 20nsec
- Delay_1 = D1 * 20nsec
- Insert all "1" PRNO times by using CCLK(DCLK).
- Insert all "1" PONO times by using CCLK(DCLK).

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。

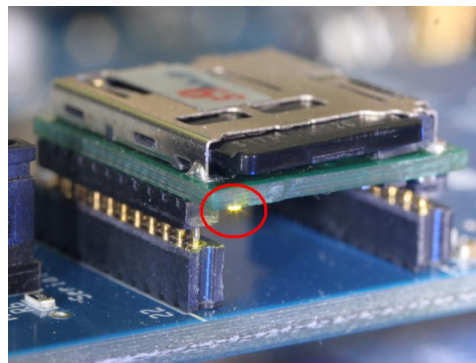


図 7.1 LED の実装位置

7.1. LED の点灯条件

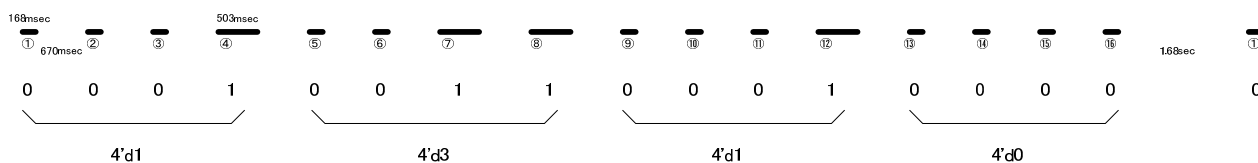
- LED の点灯条件は以下の通りです。

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、若しくは、microSD 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、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。

```
// 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                      //bit file
#3 : TESTLED3.BIT                      //bit file
#4 : TESTLED4.bin
#5 : testled5.bin
...
#F : TESTLEDF.bin
//end
```

図 7.2. “CONFIG.TXT” サンプル

8.3. 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(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。(近日対応)
- ・ 本パラメータで F を選択した場合、FPGA 側からの入力クロック速度は microSD カードからのデータの読み出し速度の範囲内である必要があり、下記の表を目安に設定して下さい。

Master モード幅	推奨 CCLK 上限値	備考
8bit パラレル	20MHz	
4bit パラレル	40MHz	
2bit パラレル	80MHz	
シリアル	100MHz	

- ・ これらの値は、ご利用になる microSD 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、microSD 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、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。
- ・ 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
す。

(i) #P : D0 = xxxx_xxxx

- ・ PROGB がリリースされて INITB 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

- ・ 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。

(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
 - ・ 未定義(Reserved)
- ⑥ #P : C5 = 0/1
 - ・ 未定義(Reserved)
- ⑦ #P : C6 = 0/1
 - ・ 未定義(Reserved)
- 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。(ファイル名の間にスペースや TAB を挿入することはできません。)

(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。(ファイル名の間にスペースや TAB を挿入することはできません。)

8.4. 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)	

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)

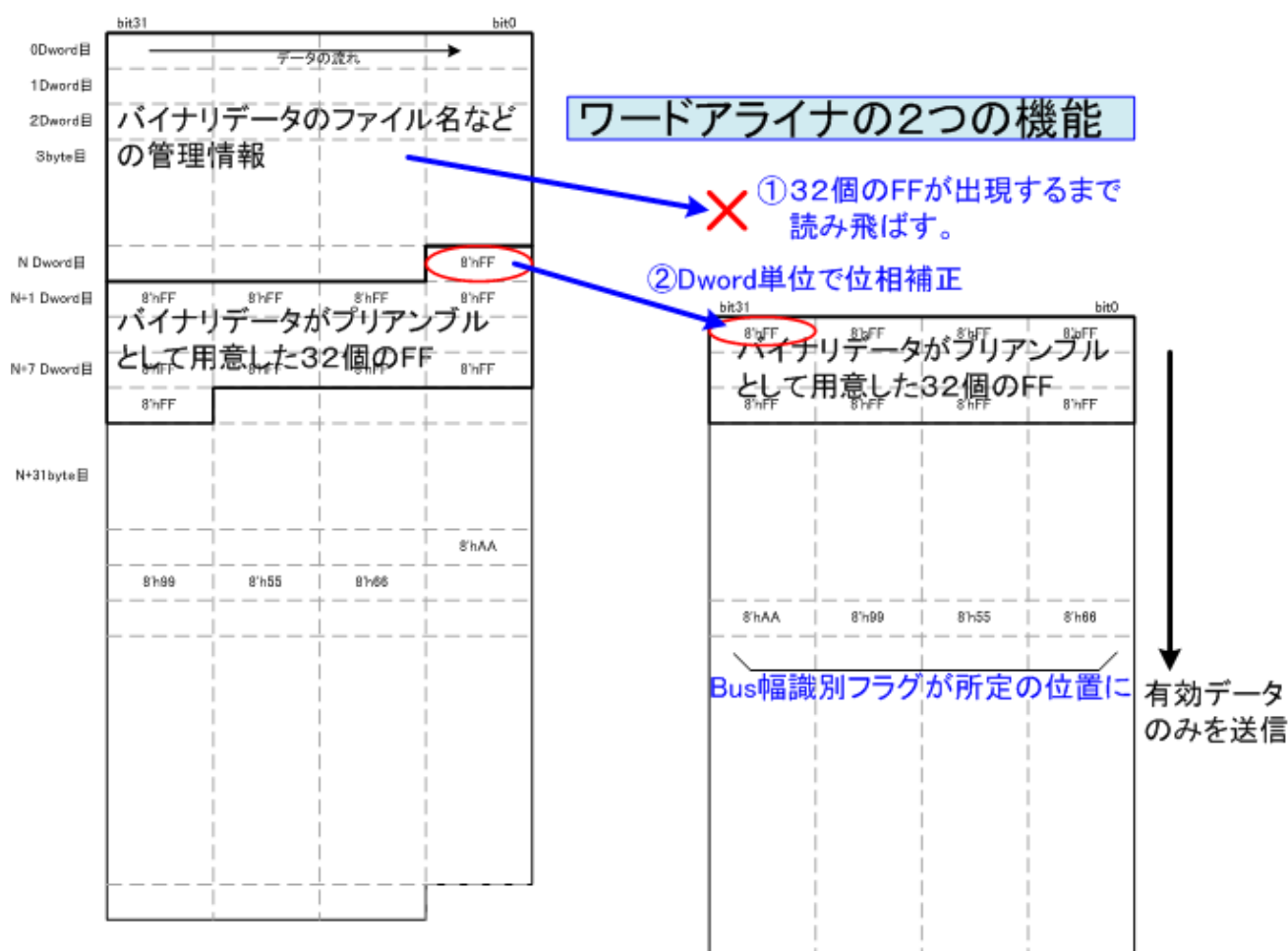


図 9.1. ワードアライナ機能説明図

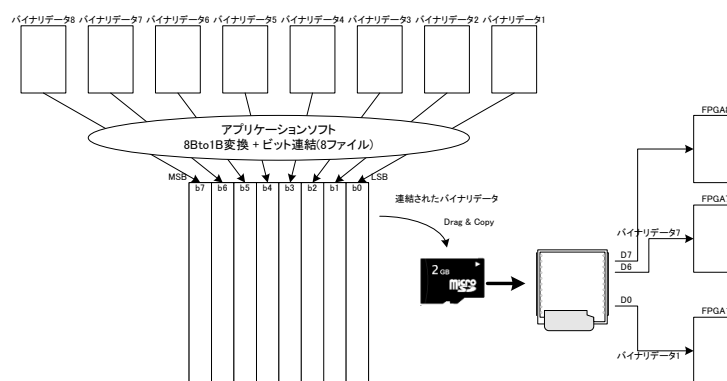
9.2. bit 連結による Multi FPGA Configuration 機能

- 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。
- 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。

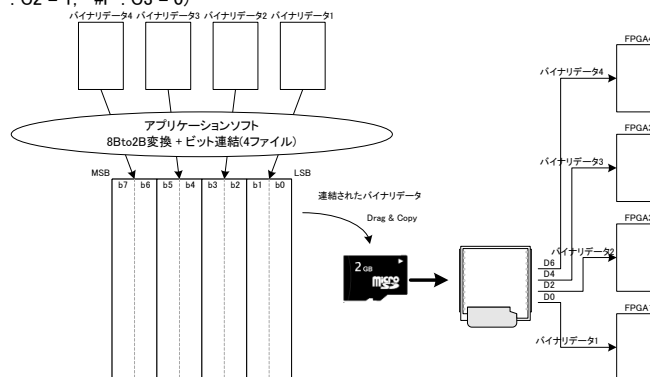
分割個数が 8 個の場合で、FPGA 割り当てに空きがある場合や、それぞれの FPGA のサイズが異なる場合、Daisy-Chain 方式より効率が悪くなる場合があります。

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)

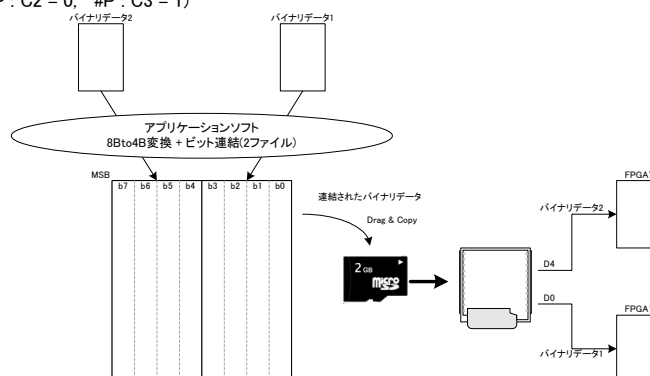


図 9.2. bit 連結による Multi FPGA Configuration の動作原理図

10. モジュールのソケットからの取り外しについて

- ・ uSDCONF1(2)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 10.1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

- ・ “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。

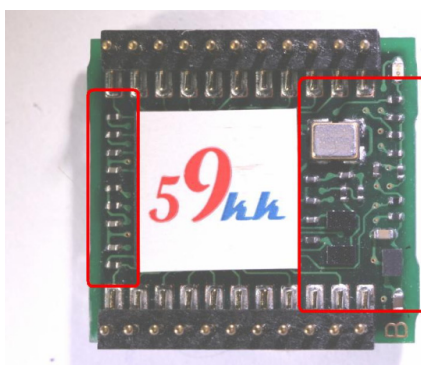


図 10.1. uSDCONF1F Bottom View

10.2. 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 10.1 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。
- ・ ドライバ等で撮り外す場合、図 10.2.1 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 10.1 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 10.2.2 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。

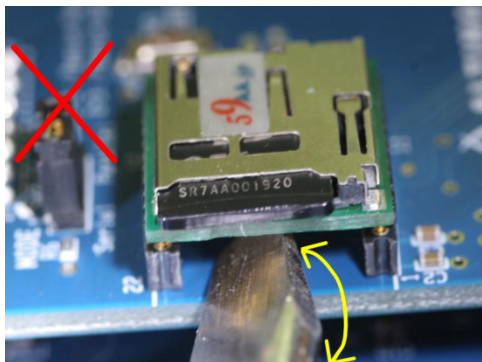


図 10.2.1 取り外しの悪い例

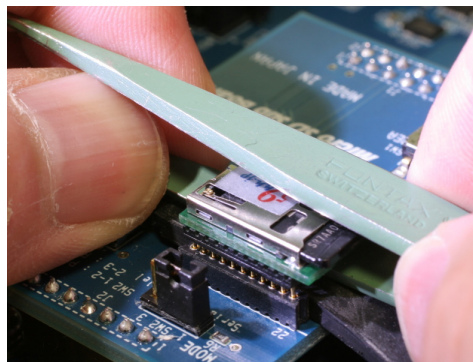


図 10.2.2 ピンセットを使った例

- ・ 図 10.2.3 は D.I.Y で使用する“目地用コーキングヘラ”です。
- ・ 図 10.2.4 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 10.2.3 目地用コーキングヘラ

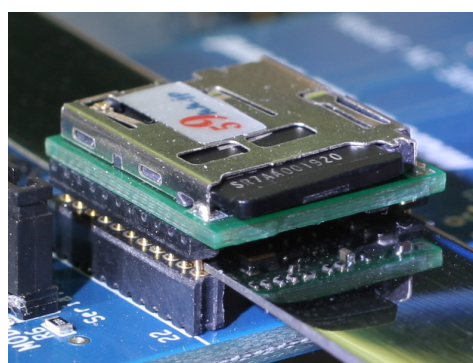


図 10.2.4 ヘラ使用例

11. 制約事項

- ① 使用する microSD 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 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 “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。
- 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。)

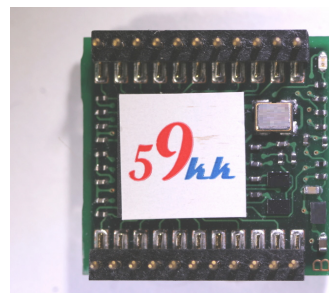
- ⑧ bit 連結による Multi FPGA Configuration 機能はインプリメントされていますが、バイナリデータを連結するためのアプリケーションソフトを準備中です。
- ⑨ CCLK のクロック速度選択機能で、CCLK を受信して動作するマスタモードは定義されていますが、現在インプリメント作業中です。

12. 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。

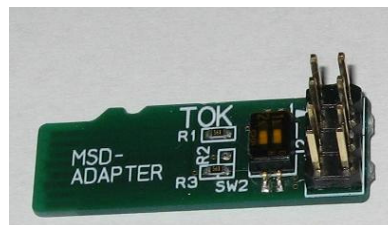
“59kk”のシールの貼られていないモジュールにつきましては、暗号 Key の書き込みが必要です。弊社もしくは担当営業にご相談下さい。

(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 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



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

横浜市港北区新羽町1824

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

電話 : 045-590-6227 Fax : 050-3156-1404

Email : info01@59kk.jp URL : <https://www.59kk.jp>