

microSD-CONF2 データシート
(USDCNF-MD2-002A)

Rev 0.5 2011.03.07

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	
2010/09/01	大庭	0.3	
2010/09/21	大庭	0.4	6.タイムチャート追加。 8.制約事項⑤⑥⑦追加。
2011/03/07	大庭	0.5	2.形状 寸法図更新、写真更新。

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. 電源シーケンス	6
4. Pin 機能表	7
4.1. uSD-CONF2 ピン機能表	7
4.2. XAREA 詳細	8
4.3. XMODE 詳細	8
4.3.1. microSDCard=>FPGA コンフィグレーションモード [*] (Slave Serial)	8
4.3.2. microSDCard=>FPGA コンフィグレーションモード [*] (SelectMap 8bit)	9
4.3.3. NandFlash=>FPGA コンフィグレーションモード [*] (Slave Serial)	9
4.3.4. NandFlash=>FPGA コンフィグレーションモード [*] (SelectMap 8bit)	9
4.3.5. microSDCard=>NandFlash 転写モード [*]	9
4.3.6. NandFlash エリアイレースモード [*]	9
4.3.7. NandFlash フォーマットモード [*]	9
5. 接続例	11
6. タイムチャート	12
6.1. 信号出力部ブロック図	12
6.2. タイムチャート	12
6.3. 全体的シーケンス	12
7. CONFIG.TXT の準備	13
7.1. CONFIG.TXT とは	13
7.2. CONFIG.TXT サンプル	13
7.3. Commands	14
8. 制約事項	15

1. 機能・特徴

- 本モジュールは、microSDCard と、NandFlash を用いたコンフィグレーションが可能です。
- MODE ピンに所定のレベルを与えてやることにより、下記のオペレーションが選択できます。
 - microSDCard から FPGA への直接コンフィグレーション。
 - microSDCard から NandFlash へのコンフィグレーションデータの転写。
 - NandFlash から FPGA へのコンフィグレーション。
 - NandFlash のデータのイレース。
 - NandFlash のフォーマット。
- microSDCard には、最大 2GB までの Card を使用でき、容量の許す限りビットデータを格納することができます。
- microSDCard 上に格納されたビットデータは、同じく microSDCard 上に格納される“CONFIG.TXT”の名称のテキストファイル上で、ビットデータ名と AREA[3:0]ピンによってカバーできる 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]ピンの設定により、最大 16 個のビットファイルを瞬時に選択し、コンフィグレーションすることが可能です。
- NandFlash は、16 個のエリアに分割され、AREA[3:0]ピンがカバーできる 0~F の 16 個の 16 進数と関連付けられ、AREA[3:0]ピンの設定により、瞬時に NandFlash 上のエリアを選択することができ、NandFlash から FPGA へのコンフィグレーション、microSDCard から NandFlash へのビットデータ転写、指定のエリアのビットデータのイレースを行うことが可能です。
- NandFlash へのデータの書き込みも、microSDCard を介して転写することができるため、JTag ケーブルを接続して書き込むような煩わしさはありません。
- 16 個のエリアは、開発段階の複数のバージョンのビットデータを保管しておくことが可能で、AREA[3:0]ピンに接続されたロータリーSW を操作することで、ビットデータを書き換えることなく、バージョン毎の差分を瞬時に比較することができ、デバッグの効率を大幅に高めることが可能です。
- 複数の FPGA をデジチェーンさせたシステムにも使用可能です。
- モジュールの論理部は、Lattice 社の XP2-5 FPGA を使用しているため、機能のカスタマイズにも対応が可能です。(別途費用がかかります。)

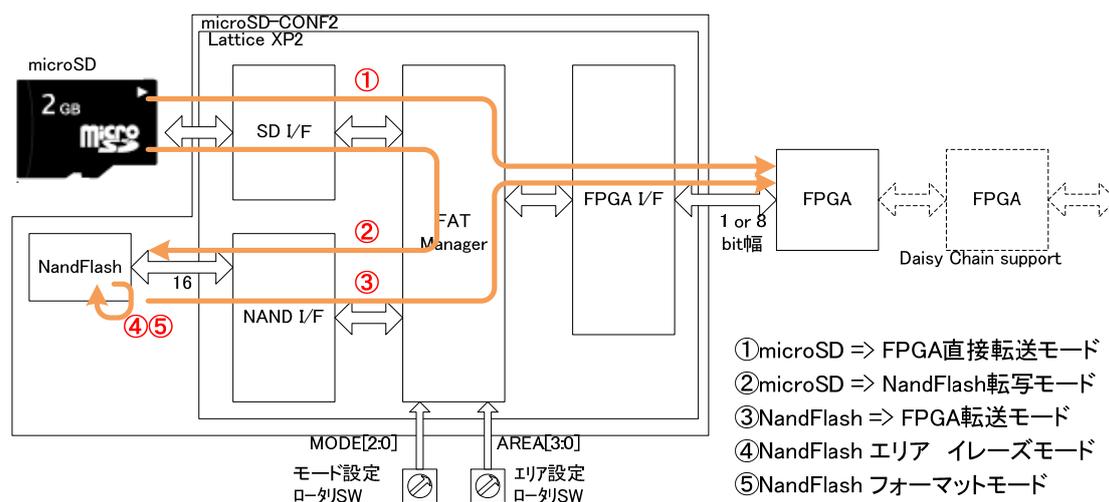
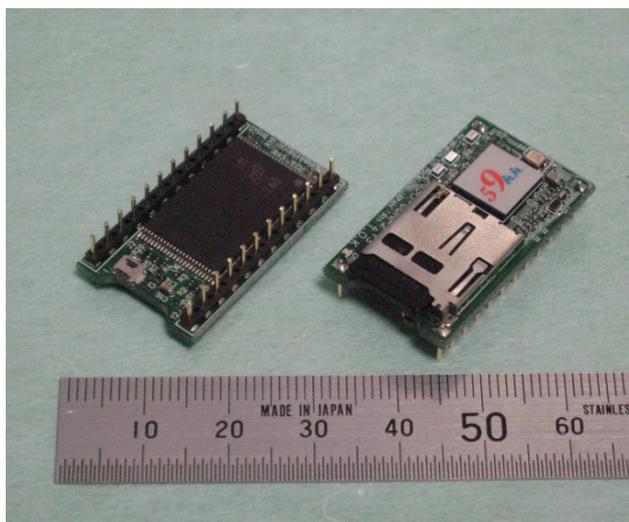
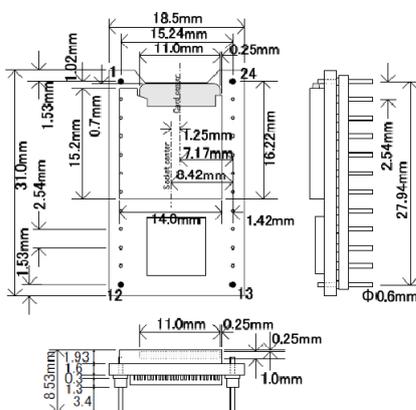


図 1-1.uSD-CONF2 モジュールの機能イメージ

2. 形状

uSD-CONF2 モジュールの形状と写真を掲載します。



使用 pin は、MAC8 の HQS-2-5-12P。(<http://www.mac8sdk.co.jp/mac8/pdf/HQS.pdf>)

丸型 0.6mm ϕ 。推奨 Pad 径は 0.8mm ϕ 。

ソケット例

(1) AR 24 HZL/7-TT (ASSMANN WSW) (RS 品番 674-2491)

(<http://docs-asia.origin.electrocomponents.com/webdocs/0da4/0900766b80da42ae.pdf>)

(2) IC26-2406-GG4 (山一電機)

(<http://www.yamaichi.co.jp/products/picsocket/ic26dip/pdf/ic26dip.pdf>)

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. 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。

4. Pin 機能表

uSD-CONF2 モジュールのピン機能表を掲載します。

4.1. uSD-CONF2 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	無し	4.7kΩPullup	Bitdata 選択用ロータリ SW 入力(LSB)
4	XAREA1	I	無し	4.7kΩPullup	Bitdata 選択用ロータリ SW 入力
5	XAREA2	I	無し	4.7kΩPullup	Bitdata 選択用ロータリ SW 入力
6	XAREA3	I	無し	4.7kΩPullup	Bitdata 選択用ロータリ SW 入力(MSB)
7	GND	—			GND
8	DONE	I	330ΩPullup	不要	FPGA DONE 信号
9	CCLK	O	無し	100ΩPullup+ 100ΩPulldn	FPGA CCLK 信号
10	PROGB	O	4.7kΩPullup	不要	FPGA PROGB 信号
11	INITB	O	4.7kΩPullup	不要	FPGA INITB 信号
12	XMODE0	I	4.7kΩPullup	4.7kΩPullup	動作モード選択信号 0
13	XMODE1	I	4.7kΩPullup	4.7kΩPullup	動作モード選択信号 1
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	GND	—			GND
19	D4	O			FPGA 8bit Parallel data4
20	D5	O			FPGA 8bit Parallel data5
21	D6	O			FPGA 8bit Parallel data6
22	D7	O			FPGA 8bit Parallel data7
23	XMODE2	I		4.7kΩPullup	動作モード選択信号 2
24	XRST	I	4.7kΩPullup	不要	パワーオンリセット信号

Pullup は VCCIO で、PullDown は GND と行なって下さい。

4.2. XAREA 詳細

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

4.3. XMODE 詳細

XMODE			詳細
2	1	0	
H	H	H	microSDCard=>FPGA コンフィグレーションモード [*] (Slave Serial)
H	H	L	microSDCard=>FPGA コンフィグレーションモード [*] (Select Map 8bit)
H	L	H	NandFlash=>FPGA コンフィグレーションモード [*] (Slave Serial)
H	L	L	NandFlash=>FPGA コンフィグレーションモード [*] (Select Map 8bit)
L	H	H	microSDCard=>NandFlash 転写モード [*]
L	H	L	NandFlash エリアイレースモード
L	L	H	NandFlash フォーマットモード [*]
L	L	L	--

4.3.1. microSDCard=>FPGA コンフィグレーションモード^{*}(Slave Serial)

XMODE[2:0]=(H,H,H)の状態ではパワーオンリセットが解除されると立ち上がるモード。

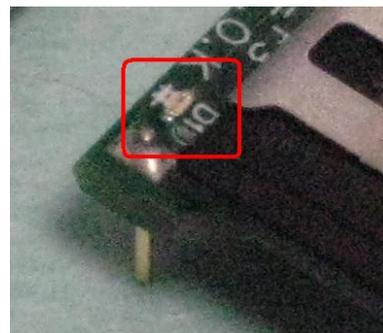
XAREA[3:0]で指定されたエリア番号と関連付けられたビットファイルを microSDCard より読み出し、FPGA をコンフィグレーションするモード。

Slave Serial モードで 14 番ピンの D0(SOUT)によりシリアル転送される。

- 4.3.2. microSDCard=>FPGA コンフィグレーションモード(SelectMap 8bit)
XMODE[2:0]=(H,H,L)の状態ではパワーオンリセットが解除されると立ち上がるモード。
XAREA[3:0]で指定されたエリア番号と関連付けられたビットファイルを microSDCard より読み出し、FPGA をコンフィグレーションするモード。
Select Map の 8bit モードで転送される。
- 4.3.3. NandFlash=>FPGA コンフィグレーションモード(Slave Serial)
XMODE[2:0]=(H,L,H)の状態ではパワーオンリセットが解除されると立ち上がるモード。
XAREA[3:0]で指定されたエリア番号の NandFlash エリアに格納されたビットファイルをり読み出し、FPGA をコンフィグレーションするモード。
Slave Serial モードで 14 番ピンの D0(SOUT)によりシリアル転送される。
- 4.3.4. NandFlash=>FPGA コンフィグレーションモード(SelectMap 8bit)
XMODE[2:0]=(H,L,L)の状態ではパワーオンリセットが解除されると立ち上がるモード。
XAREA[3:0]で指定されたエリア番号の NandFlash エリアに格納されたビットファイルをり読み出し、FPGA をコンフィグレーションするモード。
Select Map の 8bit モードで転送される。
- 4.3.5. microSDCard=>NandFlash 転写モード
XMODE[2:0]=(L,H,H)の状態ではパワーオンリセットが解除されると立ち上がるモード。
XAREA[3:0]で指定されたエリア番号と関連付けられたビットファイルを microSDCard より読み出し、同じエリア番号の NandFlash エリアに転写を行うモード。
転写前に、同エリアに対してイレース処理が行われる。
- 4.3.6. NandFlash エリアイレースモード
XMODE[2:0]=(L,L,H)の状態ではパワーオンリセットが解除されると立ち上がるモード。
XAREA[3:0]で指定されたエリア番号の NandFlash エリアを初期化するモード。
に格納されたビットファイルをり読み出し、FPGA をコンフィグレーションするモード。
- 4.3.7. NandFlash フォーマットモード
XMODE[2:0]=(L,L,H)の状態ではパワーオンリセットが解除されると立ち上がるモード。
本モジュールでは、現段階において、簡易的な FAT 管理を行っています。
2秒以上のパワーオンリセット状態が続いたあとにパワーオンリセットが解除されるとフォーマットモードが動作を始めます。
出荷時に Bad Block を回避しながら NandFlash の全エリアに対して連続した一つの FAT チェインを構築します。
FAT チェインが構築されたあと、1つのエリアに割り当てられたサイズ毎に、先頭から16個のエリアに分割し、各エリアの先頭の FAT アドレスによる管理テーブルを作成します。
初回のフォーマットは NandFlash 出荷時の Bad Block のみを走査するため、1秒未満で完了します。
2回目以降のフォーマットにおいては、5555h,AAAAh のパターンを write/readして block の正当性の試験を行いますので、約5分要します。
Bad Block と判断された Block は FAT チェインから切り離されます。

5. LED

uSD-CONF2 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。

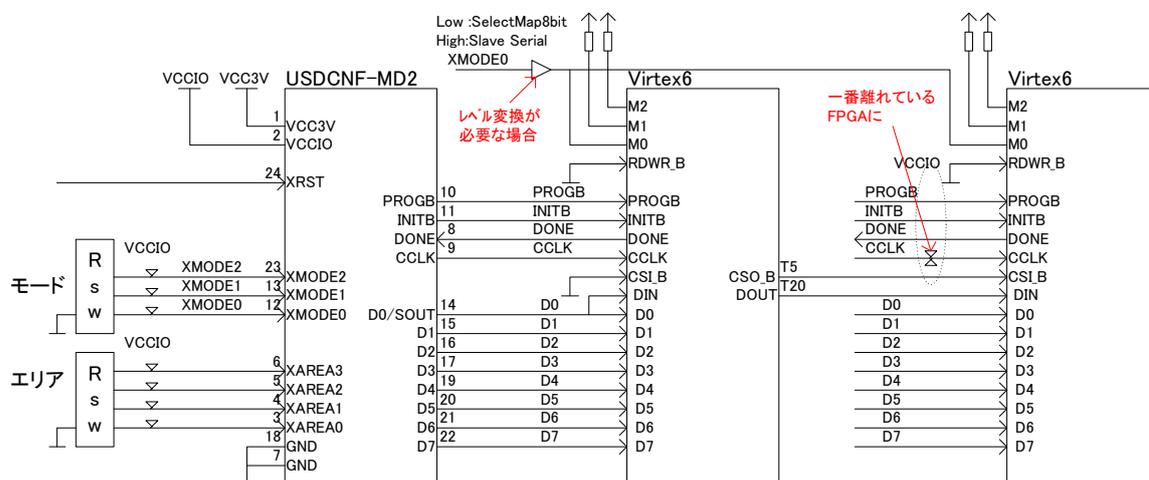


LED の点灯条件は以下の通りです。

LED 状態	条件
点灯	SD=>FPGA Slave Serial モードで SD よりデータを転送している間。
	SD=>FPGA Slave SelectMap モードで SD よりデータを転送している間。
	Nand=>FPGA Slave Serial モードで Nand よりデータを転送している間。
	Nand=>FPGA Slave SelectMap モードで Nand よりデータを転送している間。
	SD=>Nand 転送モードで、SD よりデータを転送している間。
	Erase モードで Nand の所定のエリアを消去している間。
点滅 167msec 点灯+ 167msec 消灯の 繰り返し。	Format モードで Nand を Format している間。
	SD=>FPGA Slave Serial/SelectMap、SD=>Nand 転送モードで、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、rename した場合など。)
その他	SD=>FPGA Slave Serial/SelectMap、SD=>Nand 転送モードで、config.txt によって指定された bit ファイルがルートディレクトリ上で認識できなかった場合。 原因: bit ファイルが存在しない。 スペルが間違っている。 bit 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 8 文字以下に rename した場合など。)
	※SD が挿入されていない場合は消灯のままです。

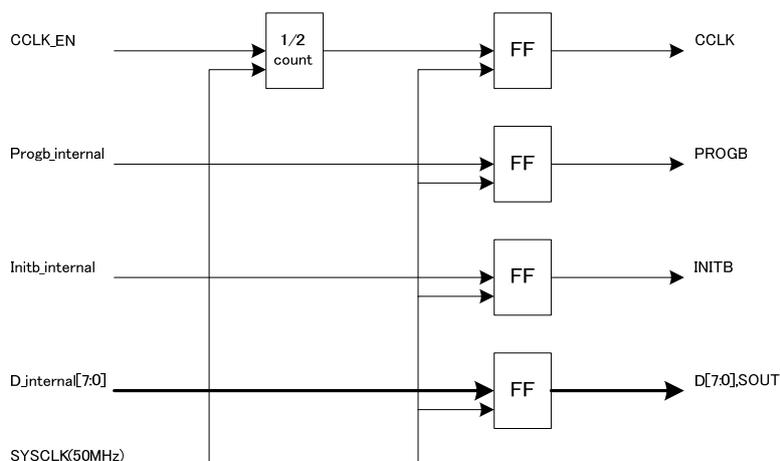
6. 接続例

Slave SelectMap8/Slave Serial選択式

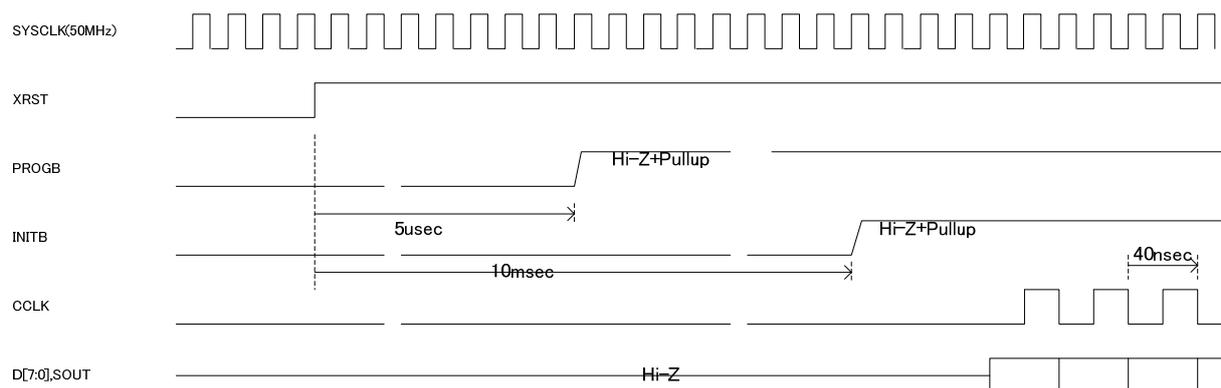


12. タイムチャート

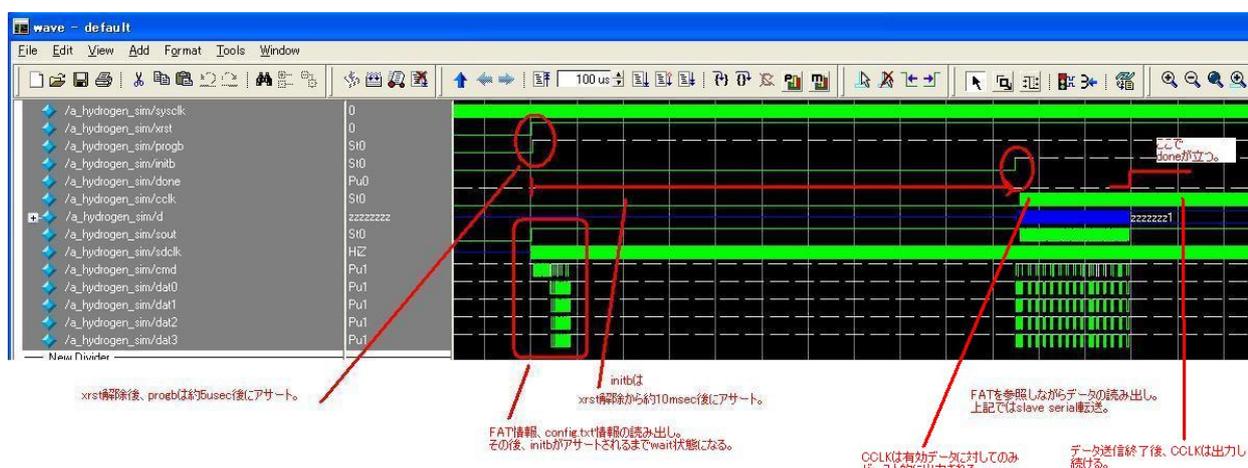
7.1. 信号出力部ブロック図



7.2. タイムチャート



7.3. 全体的シーケンス



8. CONFIG.TXT の準備

8.1. CONFIG.TXT とは

- ・ microSDCard 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、microSDCard から NandFlash ヘデータを転写する場合に参照されるテキストファイルです。
- ・ ファイル名は 半角アスキーの“CONFIG.TXT”(小文字、大文字小文字混在も可)に固定化されており、microSDCard 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 ファイルには、下記のコマンドが存在します。

8.2. CONFIG.TXT サンプル

microSDCard 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          <= ‘/’でコメントアウト。左記は verilog 風に 2 つ並べてい  
ます  
  
//#W : S                            <= コメントアウトを外すと slave serial mode となります  
  
//TEST_LED.BIN                      <= コメントアウトを外すと TEST_LED.BIN が SW4 に関  
係なく  
                                     選択されます  
  
#0 : TESTLED0.BIN  
#1 : TESTLED1.BIN  
#2 : TESTLED2.BIN  
#3 : TESTLED3.BIT                   <= BIT ファイルも選択可能です  
#4 : TESTLED4.BIN  
#5 : TESTLED5.BIT  
#6 : TESTLED6.BIT                   <= ファイル名は半角英数字 8 文字以下として下さい
```

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。

(b) #W : S or B or W or D (本モジュールでは使われていません。)

- コンフィグレーションのバス幅を指定します。
- サフィックスは下記の通りです。

D(d)	: 32bit width (default)
W(w)	: 16bit width
B(b)	: 8bit width
S(s)	: 1bit width
- 各々のキャラクタの間や前後に、スペースや TAB を含むことができます。
- 本コマンドは、ビットファイルを指定するコマンド列の前に配置する必要があります。
- 記述がない場合は D(32bit width)を選択します。

(c) 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 はビットファイルとして扱います。
- ビットファイルは、”.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。
- ビットファイル名は 8 文字以下である必要があります。
- ビットファイル名の前後にスペースや TAB を含むことができます。(ファイル名の中にスペースや TAB を挿入することはできません。)

(d)#0 ~ #F : bitfile name

- ビットファイル名を 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、関連付けされたビットファイル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。
- ビットファイルは、”.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。
- ビットファイル名は 8 文字以下である必要があります。
- ビットファイル名の前後にスペースや TAB を含むことができます。(ファイル名の中にスペースや TAB を挿入することはできません。)

9. 制約事項

- ① 使用する microSDCard 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 microSDCard 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。
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 microSDCard 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”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。
- ④ NandFlash メモリ使用中、Bad ブロックが発生する可能性があります。Bad ブロック置換処理は行いません。
- 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 Configuration 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 VerUP 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 bitcopy.bat をコピーする。

Bitcopy.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。

F: <=ドライブ名
Chdir F:\work\bitfiles <=chdir のあとに Explorer のアドレス部分をコピーします。

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。)

Bitcopy.bat をダブルクリックする。

- 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。

15. アップデート(近日常対応予定)

“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。

- 制約事項の対応
 - ◇ config.txt の 512 バイト制約 など
- 機能改版(有償)
 - ◇ FAT32 対応
 - ◇ SDHC 対応
 - ◇ config 終了後の SD へのインタフェース機能
 - ◇ など
- 同一モジュールによる別機能の実装(有償)
 - ◇ SPI(Master/Slave)終端機能
 - ◇ JTag 終端機能
 - JTag からのコンフィグ
 - ROM データの書き換え

