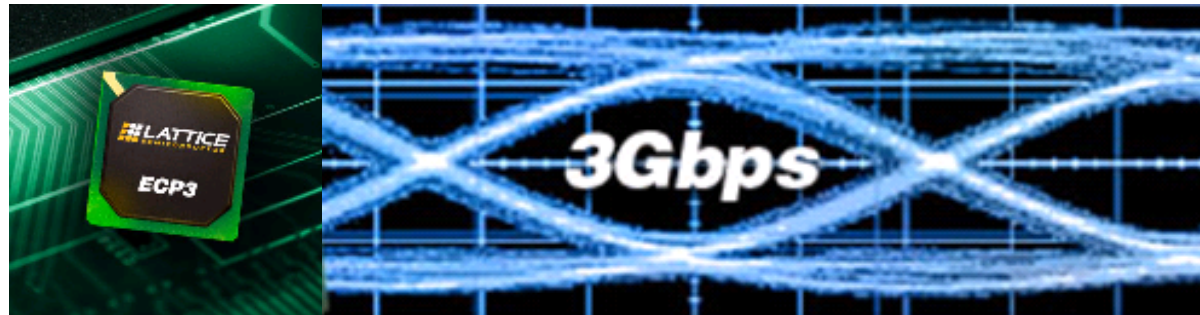


ECP3 ユーザーズガイド



TecStar
Company

- ・ 本資料は、Lattice社 ECP3の基板設計時の注意事項、使用時の注意事項をまとめたものです。実際の動作等詳細、最終の確認は、別途データシートを参照頂けるようお願い申し上げます。
- ・ Lattice社データシートと本資料との間に差異があった場合には、Lattice社データシートを正としてお取り扱い下さい。

ECP3保有ピンの機能説明・ピン処理について

1-1 電源ピン	P5
1-2 コンフィグ関連ピン	P10
1-3 クロック専用ピン、PLL専用ピン	P11
1-4 その他のピン	P14
1-5 Serdes信号ピン	P15

ECP3のSISIOについて

2-1 sysI/O Bankの構成について	P17
2-2 ECP3 ECLK Clocking構造について	P18
2-3 ECP3 Mixed Voltage Input left/right	P19
2-4 ECP3 Mixed Voltage Input top	P20
2-5 バンク毎のStandardI/Oサポート	P21
2-6 VCCIO,VREF,VTT電源の扱いについて	P22
2-7 On Chip Termination,Equalizerについて	P23
2-8 SysI/O をLVDSで使用する場合	P24
2-9 疑似LVDSの外部抵抗処理	P26
2-10 LVDS33Dを使用した各種差動IO Standardを実現する方法	P27

ECP3のIOレジスタについて

3-1 DDR Genericモード及びDDR memoryモードについて	P31
3-2 ECP3 DQS Buffer Blockについて(DDR Genericモード時)	P32
3-3 ECP3 DQS Buffer Blockについて(x1DDR Genericモード時)	P33
3-4 ECP3 DQS Buffer Blockについて(x2DDR Genericモード時)	P34
3-5 入力サイドのDDR Genericインプリメント例(x1 DDR)	P35
3-6 入力サイドのDDR Genericインプリメント例(x2 DDR)	P36
3-7 ECP3 Output Register Block (x1 DDR)	P37
3-8 ECP3 Output Register Block(x2 DDR)	P38
3-9 出力サイドのDDR Genericインプリメント例 (x1 DDR)	P39
3-10 出力サイドのDDR Genericインプリメント例(x2 DDR)	P40
3-11 各バンク毎のIDDRX,IDDRX2,ODDRX,ODDRX2の制約条件	P41

ECP3での7:1LVDSデザインの実装方法

4-1 Lattice 7:1LVDSリファレンスデザインについて	P43
4-2 デザインに関する注意点その1	P44
4-3 デザインに関する注意点その2	P46
4-4 デザインの配置上の制約について	P48
4-5 ピン配置の制約について	P49

ECP3のコンフィグレーションについて

5-1 コンフィグレーションの電源、バンク	P51
5-2 コンフィグレーションモードについて	P52
5-3 コンフィグレーションモード選択	P53
5-4 SPIモード詳細(1Flash+1FPGAの場合)	P55
5-5 SPIモード詳細(1Flash+2FPGAの場合)	P56
5-6 SPIモード詳細(CPUからSPI Flashにアクセス)	P57
5-7 SPIモード詳細	P58
5-8 MCPMモード詳細	P59
5-9 Slave SPIモード詳細①	P60
5-10 Slave SPIモード詳細②	P61
5-11 Slave SPIモード詳細③	P62
5-12 Slaveパラレルモード詳細①	P63
5-13 Slave パラレルモード詳細(XO ,XP使用)	P64
5-14 JTAGモード詳細	P65

ECP3のコンフィグレーション詳細説明

6-1 ECP3纏め encryption and compress	P68
6-2 ECP3纏め encryption Key - Data relationship	P69

ECP2/Mのコンフィグレーションフロー

7-1 コンフィグレーションフロー(全体)	P71
7-2 Initializeシーケンス(電源ON時)	P72
7-3 Initializeシーケンス(Programnピン)	P73
7-4 コンフィグレーションシーケンス	P74
7-5 Wake Up(コンフィグレーション完了後のデバイスの起動)	P75
7-6 WakeUpシーケンス	P76
7-7 WakeUpタイムチャート	P77

ECP3のSPI-Flash,Serial-ROMについて

8-1 コンフィグレーションビットサイズ一覧	P79
8-2 SPI Serial Flash Vendor List	P80

ispVMsystemを使用したECP3への書き込みについて

9-1 ispVM Systemでのコンフィグデータの書き込み	P82
9-2 Bitstreamの圧縮について	P86
9-3 Bitstreamのマージについて	P87
9-4 デュアルブートについて	P88

ECP3保有ピンの機能説明・ピン処理について

■ 電源関係ピン概要

ピン名	電圧	備考
Vcc	1.2V	1.2V電源専用ピン。FPGAコア用電源。
VCCA	1.2V	SERDES用Tx,Rx,PLL,リファレンスクロック用の電源。この電源はノイズの少ないクリーンな独立した電源であることを推奨します。
Vccpll	3.3V	PLL用の電源。使用しない場合でも3.3Vを印加してください。この電源はノイズの少ないクリーンな独立した電源であることを推奨します。
Vccaux	3.3V	3.3V電源。参照電圧を用いる差動入力バッファを動作させるための補助電源です。コンフィグレーション用の電圧としても使われます。
Vccio(0-7)	1.2V/2.5V/3.3V	各バンク毎に用意されています。バンク毎にそこで使用するインターフェースによって、印加する電源を決定します。使用しないBankについても必ず1.2V,2.5V,3.3Vを印加してください。Vccioが3.3Vの場合、Vccauxと同じ電源を使用することを推奨します。 (Vccio8は可能な限りVccauxと繋げて使用するようになさい)
Vccj	1.2V/2.5V/3.3V	VCCJはLVCMOS JTAGピンの特性を決定します。つまり、JTAGピンの電源となります(VCCIOからは独立しています)3.3Vを繋ぐ場合はVccauxと繋ぐことを推奨します。
VCCIB	1.2V/1.5V	CML入力終端用電源。
VCCOB	1.2V/1.5V	CML出力終端用電源。
VREF1_x、VREF2_x	-	リファレンス電圧が必要なインターフェース(HSTL等)を使用したときに入力します。各バンクで2種類のVREF(VREF1/VREF2)を設定できます。 ※ただし、DDRメモリ/Fを使用する場合はVREF1を使用してください
VTT	-	オンチップ終端(SSTL15,1.25GLVDS,DDR3)用電源。 使用しない場合はフローティングにしてください。

■ 電源系統と電圧値 (データシートDS1021, p.3-1, Recommended Operating Conditions参照)

電 源	推奨電圧 [V]	記 述
V_{CC}	1.2V $\pm$ 5%	FPGAコア用電源
V_{CCAUX}	3.3V $\pm$ 5%	補助電源 (各種基準電源)、SERDES終端抵抗など
$V_{CCPLL} [L,R]$	3.3V $\pm$ 5%	FPGAファブリック内蔵PLL用電源
V_{CCIO}	1.2/1.5/1.8/2.5/3.3V $\pm$ 5%	各I/Oバンク毎の電源。バンクは0~3, 6~8。
V_{CCJ}	1.2/1.5/1.8/2.5/3.3V $\pm$ 5%	JTAG I/F (TAPコントローラ) 用電源
SERDES部		
V_{CCA}	1.2V $\pm$ 5%	PLLと基準クロック用バッファ、トランシーバの電源
V_{CCIB}	1.2V $\pm$ 5%、1.5V $\pm$ 5%	入力バッファ用電源。未使用チャンネルはオープン
V_{CCOB}	1.2V $\pm$ 5%、1.5V $\pm$ 5%	出力バッファ用電源。未使用チャンネルはオープン

- 注: ① V_{CCIO} または V_{CCJ} が1.2Vの場合は V_{CC} と同じ電源に、或いは3.3Vの場合は V_{CCAUX} と同じ電源にそれぞれ接続すること
- ② V_{CCPLL} はリニアレギュレータを用いてリップルを適切に除去することを強く推奨。PLL未使用時も電源を供給する必要があるが、この場合は V_{CC} と同じ電源から供給も可。
- ③ SERDES用の V_{CCA} は供給元にリニアレギュレータを用いてかつLCフィルタを用いるなど、リップルを適切に除去することを強く推奨。
 V_{CCIB} と V_{CCOB} にはチャンネル毎にLCフィルタを用いる事を推奨。
 未使用チャンネルの V_{CCIB} と V_{CCOB} 、およびHDINP/N, HDOUTP/N端子はオープンとする。
 V_{CCIB} や V_{CCOB} に1.5Vで使用し、かつ電源を供給する場合は常に V_{CCA} にも1.2Vを供給すること。
- ④ I/Oバンク8は主にコンフィグレーション関連ピンのバンク。

■電源のランプレート(傾き)、立ち上がり時間

電源投入後に着目している供給電源が有効レベルに至るまでの傾き、または時間に対する項目です。LatticeECP3では以下を守ることを推奨します。なお、 V_{CC} と V_{CCAUX} は**単調増加**でなければなりません(必須。単調増加については5.3.1を参照)。

■電源ランプレート・立ち上がり時間要件

電 源	条 件	最少有効電圧に達するまでの電源要件
V_{CCAUX}	0 ~ 3.3V、ほぼ線形に増加する場合	立ち上がり時間100usec ~ 100msec
	0 ~ 3.3V、非線形に増加する場合	立ち上がりレートが最大30mV/usec
V_{CC}	0 ~ 1.2V	立ち上がり時間100usec ~ 100msec
V_{CCA} V_{CCIO8}	0 ~ 規定値	立ち上がり時間100usec ~ 100msec

注:この表の値はキャラクタライズの最悪値をもとに決定しているが保証値ではない

■電源オン・オフ(パワーサイクリング)時などの残留電圧

デバイスが動作中に V_{CC} や V_{CCAUX} 電源が瞬断などで一時的に低下した場合、或いは意図的にオフとオンの操作/制御(パワーサイクリング)をする場合の要件です。

電源が復帰した後の動作(再コンフィグレーション)を確実にするため、本表の残留電圧値以下にする事を推奨します。

パワーサイクリング時の推奨最低電圧値

電 源	条 件	最低残留電圧値 [V]
V_{CC} 、 V_{CCAUX}	表 2-2の ランプ要件下	1usec以上の時間にわたり V_{CC} が0.5V以下、または V_{CCAUX} が2.0V以下となること

注:この表の値はキャラクタライズの最悪値をもとに決定しているが保証値ではない

■電源シーケンス

これは複数システムを必要とする供給電源が有効レベルに達する(パワーアップ)順序、もしくは電源オフ(パワーダウン)時の順序についての要件です。

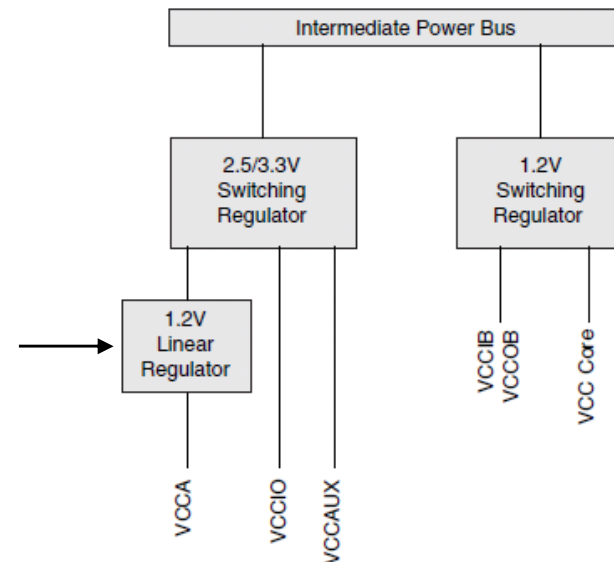
LatticeECP3では特に守らなければならないシーケンス要件はありませんが、デバイスとしての突入電流を最小にするという意味で、以下を推奨します。

$V_{CCAUX} \rightarrow V_{CCIO} \rightarrow V_{CC} \rightarrow V_{CCA}$

【重要】SERDES用電源ピンの取り扱いに関する注意事項

- ・VCCAはSERDESを使用する場合は必ず、推奨電圧レンジを遵守してはなりません。SERDESを未使用の場合でもVCCAは1.2Vを供給しておく必要があります。
- ・VCCIB VCCOBはSERDESを使用しない場合はフローティングで構いません。未使用チャネルはHizであり、10K Ω の内部抵抗で差動ペアのピン間が接続されています。
- ・1.5V VCCIB VCCOBを使用する際は、必ず1.2V VCCAが立ち上がった状態でなくてはなりません。1.5V、1.2Vは同時に立ち上げることを推奨しています。ランプレートは現状未確定。
- ・VCCAはクリーン且つ独立した電源でなくてはなりません。

VCCAはリニアレギュレータで
リップルを除去。



ピン名	I/Otype(コンフィグ時)	PinType	コンフィグモード(*1)	備考
CFG[2:0]	入力	専用	全て	内部Pullup有効
PROGRAMN	入力	専用	全て	内部Pullup有効
INITN	双方向	専用	全て	内部Pullup設定は、Option。
DONE	双方向	専用	全て	内部Pullup設定は、Option。
CCLK	入力	専用	SLAVE	CFG0 = 0の時有効。
MCLK	双方向	I/O or 専用	MASTER	CFG0 = 1の時有効。
D[0]/SPIFASTN	双方向	I/O or 専用	SPI/SPI _m /MPCM/SPCM	—(*2)
D1	双方向	I/O or 専用	MPCM/SPCM	—(*2)
D2	双方向	I/O or 専用	MPCM/SPCM	—(*2)
D3/SI	双方向	I/O or 専用	MPCM/SSPI/SPCM	—(*2)
D4/SO	双方向	I/O or 専用	MPCM/SSPI/SPCM	—(*2)
D5	双方向	I/O or 専用	MPCM/SPCM	—(*2)
D6/SPID1	双方向	I/O or 専用	SPI/MPCM/SSPI/SPCM	—(*2)
D7/SPID0	双方向	I/O or 専用	SPI/MPCM/SSPI/SPCM	—(*2)
CSN/SN/OEN	双方向	I/O or 専用	MPCM/SSPI/SPCM	内部Pullup有効
CS1N/HOLDN/RDY	双方向	I/O or 専用	MPCM/SSPI/SPCM	内部Pullup有効
WRITEN	入力(LowActive)	I/O or 専用	MPCM/SPCM	—
BUSY/SISPI	双方向	I/O or 専用	SPI/SPI _m /MPCM/SSPI/SPCM	内部Pullup有効
DI/CSSPI0N/CEN	双方向	I/O or 専用	SPI/SPI _m /SSPI/SCM	内部Pullup有効
DOUT/CSON/CSSPI1N	双方向	I/O or 専用	SPI/SPI _m /MPCM/SSPI/SCM/SPCM	内部Pullup有効
XRES	—	専用	—	10KΩ精度±1%の抵抗でPull-Down処理が必要

(*1) コンフィグモード＝
 SPI : SPIマスター
 SPI_m : SPIマスターマルチ
 MPCM : パラレルrEEPROM
 SSPI : スレーブSPI
 SCM : スレーブシリアル
 SPCM : スレーブパラレル

(*2) D[7:0]はPowerUP時、
 コンフィグレーション時には
 ブルアップされていません。
 PowerUp中、コンフィ
 グレーション中の状態を確
 定させる為に、外部ブルア
 ップ、もしくはブルダウン抵
 抗を設置してください。

ピン名	説明
[LOC][num]_GPLL[T, C]_IN_A	GPLLへのクロック入力ピン。PLL未使用時は通常Inputとして使用可能。
[LOC][num]_GPLL[T, C]_FB_A	GPLLへのフィードバック入力ピン。PLL未使用時は通常Inputとして使用可能
[LOC][num]_GDLL[T, C]_IN_A	DLLへのクロック入力ピン。DLL未使用時は 通常Input として使用可能
[LOC][num]_GDLL[T, C]_FB_A	DLLへのフィードバック入力ピン。DLL未使用時は 通常Input として使用可能
PCLK[T,C]_[n:0]_[3:0]	Globalクロック入力ピン。クロックピンとして使用しない場合、通常I/Oとして使用可能

※上記のピンは未使用時はOPENで構いません

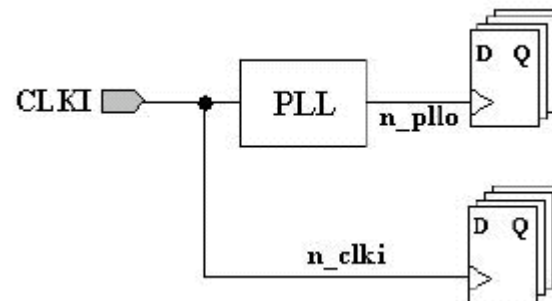
[LOC]	・・PLLのロケーション(ULM、LLM、URM、LRM)
[T/C]	・・T(True)、C(Complement) 差動のP/N
[Edge]	・・L(Left)、B(Bottom)、R(Right)、T(Top)
[Row/Column]	・・行または列の番号
[A/B]	・・I/Oセル(PIC)のうちどちらのI/O(PIO)を使っているか

■クロックピンの取り扱いに関する注意事項

- ・ クロックピン、PLLピンについてシングルエンド入力で使用する場合にはT(True),C(Complement)のT側を使用するようにして下さい。(この場合、C側はUser I/Oとして使用できます。クロックピンとしては使用できません)
- ・ クロック信号は必ずクロックピンに入力するようにしてください。I/Oから入力も可能ですが、内部専用クロックラインに乗せるまでの遅延が生じます。(クロックピンから内部専用クロックラインまでのパスは最小遅延になります)
- ・ 基板設計時、PLLを使用するかどうか判断に迷った場合には、PLL専用ピン、クロックピン両方にクロック信号を供給しておくことを推奨します。(使わない方のピンはパターンがつながっていても特に問題はありません)

■PLL専用ピンの取り扱いに関する注意事項

- ・ クロックピン、PLLピンについてシングルエンド入力で使用する場合にはT(True),C(Complement)のT側を使用するようにして下さい。(この場合、C側はUserI/Oとして使用できます。)
- ・ 下記のようなデザインの場合、n_clkiを内部専用クロック配線に乗せることができなくなります。ツール上ではn_clkiを内部専用クロック配線に乗せてもエラーにはなりません、n_clkiがPLLの入力に入る形に展開されてしまいます。つまり、一度中央の内部専用クロック配線を経由してからPLLにn_clkiが入る形になるので、その分遅延が生じてしまいます。もし、Tco改善のため、このようなデザインを考えた場合には上記の様にPLL入力までの遅延がかさむ為、逆にTcoが遅くなってしまうことが考えられます。このような場合には、CLKIをPLL専用ピンとクロック専用ピンに入力してあげることで回避することができます。



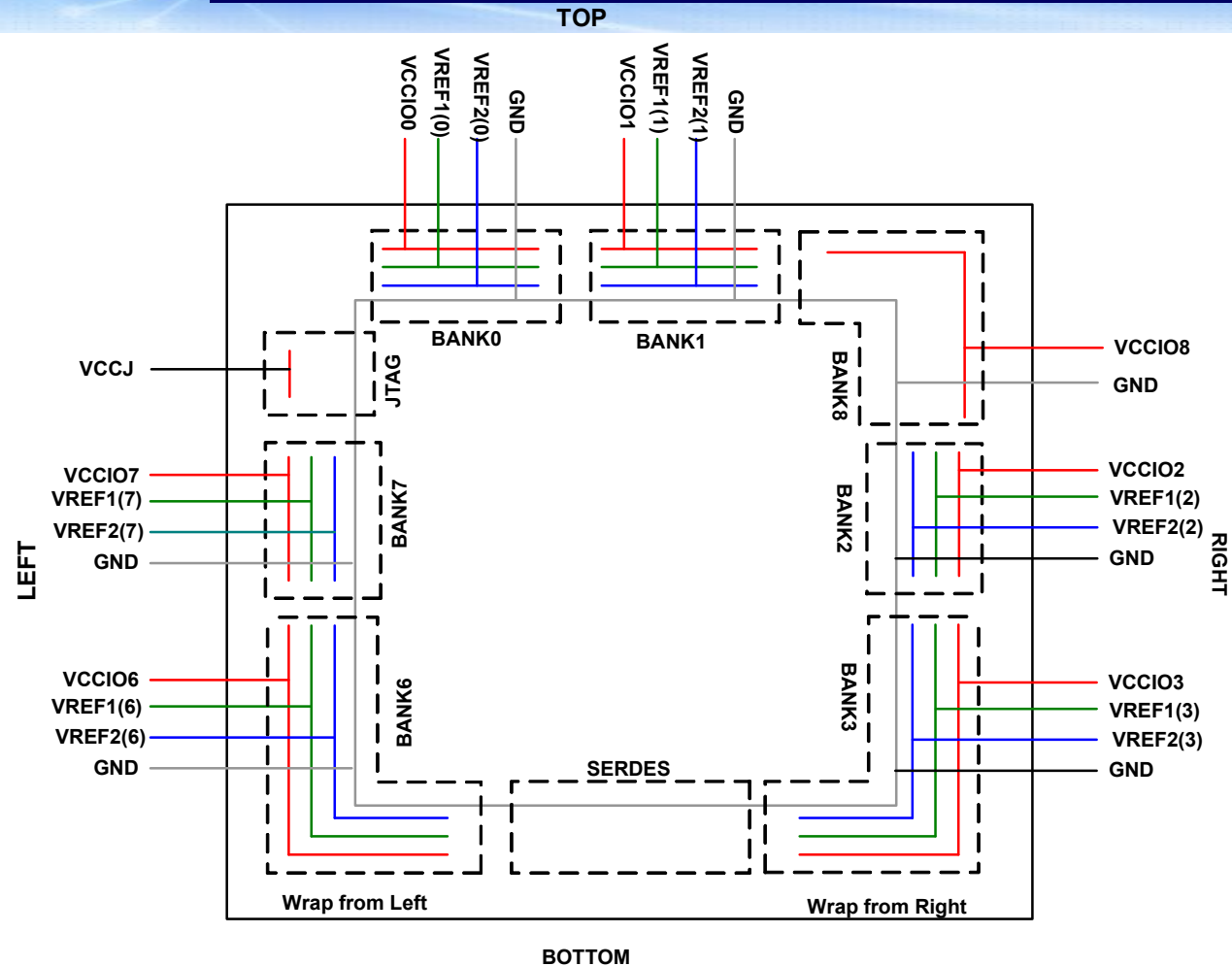
ピン名	説明
GSRN	Global RESETピン。(通常I/Oと兼用。)全ての通常I/OはGlobal RESETとして使用可能。特にこのピンからリセットを入れなければいけないわけではない。未使用時はオープンでOK。
RESERVED	リザーブピン(ECP3-150のみ)。このピンは必ずオープンにする。

- ・ GSRNは基本的には論理合成時に自動で割り当てられます。但し、自動で割り当てられる条件があり、Distributed-Ramを含む全てのFFの初期化信号であることが条件になります。もし、うまくGSRNに割り当てられない場合には、ソース中にGSRNマクロブロックを明示する必要があります。
- ・ GSRNに割り当てられたかどうかの確認はPlace & Route ReportのDevice utilization summaryの項目中をご覧ください。割り当てられた信号がRSTという信号だった場合以下のように記載されます。
 - Signal RST is selected as Global Set/Reset

ピン名	説明
PCS[index]_HDINN _m	入力ピン。ネガティブチャネル。
PCS[index]_HDOUTN _m	出力ピン。ネガティブチャネル。
PCS[index]_REFCLKN	リファレンスクロックピン。ネガティブ。
PCS[index]_HDINP _m	入力ピン。ポジティブチャネル。
PCS[index]_HDOUTP _m	出力ピン。ポジティブチャネル。
PCS[index]_REFCLKP	リファレンスクロックピン。ポジティブ。

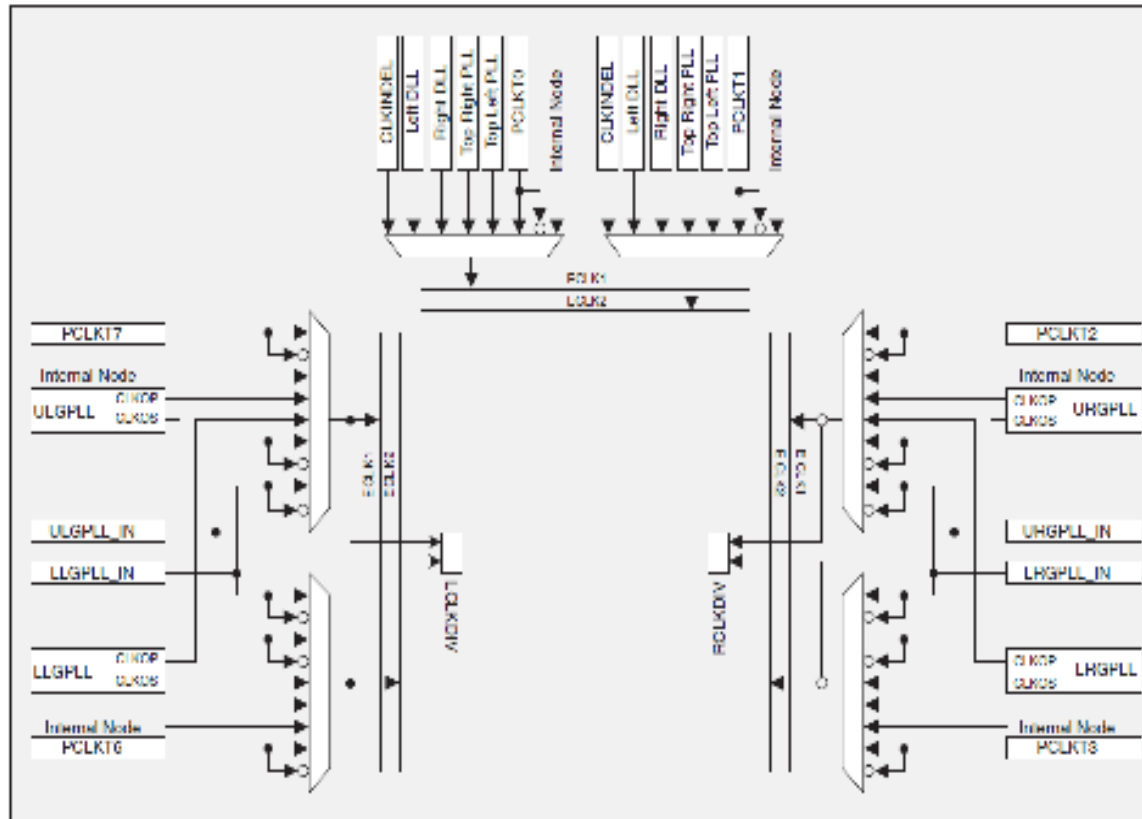
ECP3のSYSIOについて

ECP3のデバイスにおいてIO及びIO周辺機能をSYSIOと呼んでいます。このSYSIOを使用する際に必要となる情報及び注意点について記載します



- ・7つのSYSIO BANKでデバイスを構成しています
- ・それぞれのBankはVCCIO,VREF1,2を保有しています
- ・Bank 8 はsysCONFIG専用BANK。Bank8についてはVrefはBank3と共有しています
- ・SERDESはBOTTOMに配置しております (ECP2Mの場合TOP,BOTTOM)
- ・DDRメモリを使う場合、SSTL用リファレンス電圧は必ずVREF1を使用してください

◆ECP3デバイスはSYSIO周辺にエッジクロックを保有しています。このエッジクロックはIOと最短パスで繋がっており、高速信号をラッチしやすいアーキテクチャを構成しています。



■2つのECLKがTOP,Left/Rightサイドに用意されています。

■Left/Rightサイドの入力できるソースリスト

- ・ 専用クロックピン
- ・ ULGPLL出力,LLGPLL出力*注1
- ・ ULGPLL,LLGPLL入力ピン
- ・ DLLからの出力ノード
- ・ 内部ノード

■TOPサイドの入力できるソースリスト

- ・ 専用クロックピン
- ・ Top Left/Top Right PLL出力
- ・ Left/Right DLL出力
- ・ 内部ノード
- ・ CLKINDEL (DLL)

* 注1) GPLL出力のうちELCK1はCLKOPのみ、ECLK2はCLKOSのみ。UL,LLはPLLの位置を示す

◆Left/RightのBankについてSYSIOに繋げることができる信号の電圧レベルとVCCIOの関係について説明します

Left/Right Banks -

- ・ PCIクランプダイオードは常にON
- ・ LVCMOS33入力時にはVCCIO=3.3Vでなければならない
(VCCIO=3.3V時にはLVCMOS25は入力可能)
- ・ LVCMOS12はVCCIOの電圧に関係なく入力可能
- ・ Hotcoket未対応
- ・ イコライザーオプション、ターミネーションオプションが使用可能
(詳しくは別ページ参照)

V _{CCIO}	Input sysIO Standards					Output sysIO Standards				
	1.2V	1.5V	1.8V	2.5V	3.3V	1.2V	1.5V	1.8V	2.5V	3.3V
1.2V	Yes					Yes				
1.5V	Yes	Yes					Yes			
1.8V	Yes		Yes					Yes		
2.5V	Yes			Yes					Yes	
3.3V	Yes			Yes	Yes					Yes

◆TOPのBankについてSYSIOに繋げることができる信号の電圧レベルとVCCIOの関係について説明します

Top Banks-

- ・ PCIクランプはON,OFF選択可能(ONの場合は前頁を参照)
- ・ VCCIOの電圧に関わらずLVCMOS12,25,33は入力可能
- ・ Hotsocketing対応

Mixed Voltage on Top Banks when PCI Clamp is OFF

V _{CCIO}	Input sysIO Standards					Output sysIO Standards				
	1.2V	1.5V	1.8V	2.5V	3.3V	1.2V	1.5V	1.8V	2.5V	3.3V
1.2V	Yes			Yes	Yes	Yes				
1.5V	Yes	Yes		Yes	Yes		Yes			
1.8V	Yes		Yes	Yes	Yes			Yes		
2.5V	Yes			Yes	Yes				Yes	
3.3V	Yes			Yes	Yes					Yes

Description	Top Side Banks 0-1	Right Side Banks 2-3	Bottom Side Banks 4-5	Left Side Banks 6-7
Types of I/O Buffers	Single-ended	Single-ended and Differential	Single-ended	Single-ended and Differential
Single-Ended Standards Outputs	LVTTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12 SSTL15 SSTL18 Class I, II SSTL25 Class I, II SSTL33 Class I, II HSTL15 Class I HSTL18_I, II	LVTTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12 SSTL15 SSTL18 Class I, II SSTL25 Class I, II SSTL33 Class I, II HSTL15 Class I HSTL18 Class I, II	LVTTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12 SSTL15 SSTL18 Class I, II SSTL2 Class I, II SSTL3 Class I, II HSTL15 Class I HSTL18 Class I, II	LVTTTL LVCMOS33 LVCMOS25 LVCMOS18 LVCMOS15 LVCMOS12 SSTL15 SSTL18 Class I, II SSTL2 Class I, II SSTL3 Class I, II HSTL15 Class I HSTL18 Class I, II
Differential Standards Outputs	LVCMOS33D SSTL15D SSTL18D Class I, II SSTL25D Class I, II SSTL33D Class I, II HSTL15D Class I HSTL18D Class I, II LVDS ² RSOS ² Mini-LVDS ² PPLVDS ² (point-to-point) LVDS25E ¹ LVPECL ¹ BLVDS ¹ RSDSE ¹	LVCMOS33D SSTL15D SSTL18D Class I, II SSTL25D Class I, II SSTL33D Class I, II HSTL15D Class I HSTL18D Class I, II LVDS ² RSOS ² Mini-LVDS ² PPLVDS ² (point-to-point) LVDS25E ¹ LVPECL ¹ BLVDS ¹ RSDSE ¹	LVCMOS33D SSTL15D SSTL18D Class I, II SSTL25D Class I, II SSTL33D Class I, II HSTL15D Class I HSTL18D Class I, II LVDS ² RSOS ² Mini-LVDS ² PPLVDS ² (point-to-point) LVDS25E ¹ LVPECL ¹ BLVDS ¹ RSDSE ¹	LVCMOS33D SSTL15D SSTL18D Class I, II SSTL25D Class I, II SSTL33D_I, II HSTL15D Class I HSTL18D Class I, II LVDS ² RSOS ² Mini-LVDS ² PPLVDS ² (point-to-point) LVDS25E ¹ LVPECL ¹ BLVDS ¹ RSDSE ¹
Inputs	All Single-ended and Differential 1RLVDS (Transition Reduced LVDS)	All Single-ended, Differential	All Single-ended, Differential	All Single-ended, Differential
Clock Inputs	All Single-ended, Differential	All Single-ended, Differential	All Single-ended, Differential	All Single-ended, Differential
Hot Socketing	Yes	Only on Bank 8	Yes	No
Equalization on Inputs	No	Yes	No	Yes
ISI Correction	For DDR3 memory	For DDR3 memory	For DDR3 memory	For DDR3 memory
On Chip Termination	No	On-Chip Parallel Termination On-Chip Differential Termination	No	On-Chip Parallel Termination On-Chip Differential Termination
PCI Support	PCI33 with or without clamp	PCI33 with clamp	PCI33 with clamp	PCI33 with clamp

1. These differential standards are implemented by using a complementary LVCMOS driver with the external resistor pack.

2. Available on 50% of the I/Os in the bank.

・差動ペア出力はLeft/Rightサイドのみ
 ーLeft/Rightサイド以外で差動出力を使う場合にはLVCMOS+外付け抵抗でエミュレートさせることも可能です。

・内部Equalizer、Onchip TerminationはLeft/Rightサイドのみ保有

・HotsocketはTOP,Bottomのみ

◆ECP3デバイスのSYSIOはStandardIOに応じてVCCIO,VREFを考慮する必要があります

Input Standard	V _{REF} (Nominal)	V _{CCIO} (Nominal)
Single Ended Interfaces		
LVTTTL	—	—
LVCMO533	—	—
LVCMO525	—	—
LVCMO518	—	—
LVCMO515	—	—
LVCMO512	—	—
PCI33	—	—
HSTL18 Class I, II	0.9	—
HSTL15 Class I	0.75	—
SSTL33 Class I, II	1.5	—
SSTL25 Class I, II	1.25	—
SSTL18 Class I, II	0.9	—
SSTL15	0.75	—
Differential Interfaces		
Differential SSTL33 Class I, II	1.5	—
Differential SSTL18 Class I, II	0.9	—
Differential SSTL25 Class I, II	1.25	—
Differential HSTL15 Class I	0.75	—
Differential HSTL18 Class I, II	0.9	—
Differential SSTL15	0.75	—
LVDS	—	—
Transition Reduced LVDS	—	—
LVPECL	—	—
Bus LVDS	—	—
MLVDS	—	—

* SSTL用VREF電源はVREF1を使用すること

Output Standard	Drive	V _{CCIO} (Nominal)
Single Ended Interfaces		
LVTTTL	4mA, 5mA, 12mA, 18mA, 20mA	3.3
LVCMO533	4mA, 5mA, 12mA, 18mA, 20mA	3.3
LVCMO525	4mA, 5mA, 12mA, 18mA, 20mA	2.5
LVCMO518	4mA, 5mA, 12mA, 18mA, 20mA ¹	1.8
LVCMO515	4mA, 5mA, 12mA, 18mA ¹ , 20mA ²	1.5
LVCMO512	2mA, 4mA ² , 6mA, 8mA ² , 12mA ² , 18mA ² , 20mA ²	1.2
PCI33	N/A	3.3
HSTL18 Class I	5mA, 12mA	1.8
HSTL18 Class II	N/A	1.8
HSTL15 Class I	4mA, 5mA	1.5
SSTL33 Class I, II	N/A	3.3
SSTL25 Class I	8mA, 12mA	2.5
SSTL25 Class II	16mA, 20mA	2.5
SSTL18 Class I	N/A	1.8
SSTL18 Class II	8mA, 12mA	1.8
SSTL15	10mA	1.5
Differential Interfaces		
Differential HSTL18 Class I	5mA, 12mA	1.8
Differential HSTL18 Class II	N/A	1.8
Differential HSTL15 Class I	4mA, 5mA	1.5
Differential SSTL33 Class I, II	N/A	3.3
Differential SSTL25 Class I	8mA, 12mA	2.5
Differential SSTL25 Class II	16mA, 20mA	2.5
Differential SSTL18 Class I	N/A	1.8
Differential SSTL18 Class II	8mA, 12mA	1.8
Differential SSTL15	10mA	1.5
LVDS	N/A	2.5
Point to Point LVDS (PPLVDS)	N/A	2.5
RSDS	N/A	2.5
Mini-LVDS ¹	N/A	2.5
MLVDS ²	N/A	2.5
BLVDS ²	N/A	2.5
LVPECL ²	N/A	3.3

1. Multiple Drive supported using DriDrive and MultiDrive.

2. Emulated with LVCMO5 drivers and external resistors.

3. This drive strength is only available when the output is configured as open-drain.

VTT電源

VTT電源は内部Terminationのリファレンス電源になります。(次項参照)使用するIOスタンダードに応じてVTT電源を接続して下さい。SSTLのTerminationとして使う場合、Termination用レギュレータを接続することをお奨めします

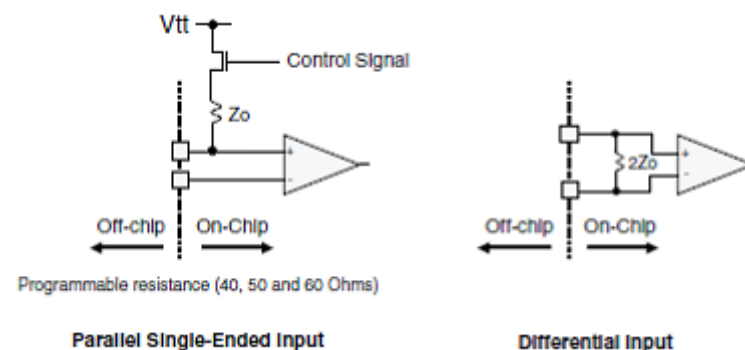
◆SYSIOのLeft/RightBANKにはターミネーション、イコライザーが入っております。使い方について記載します

On chip termination

- Differential Inputの場合、LVDS用に終端抵抗が内蔵されております
- SingleEnd Inputの場合、SSTLxx用に終端抵抗が内蔵されております
- それぞれの抵抗において以下の通り設定可能です
- IO毎の設定ではなくBANK毎の設定となります

Attribute	Values	Default
TERMINATEVTT	OFF, 40, 50, 60	OFF

Attribute	Values	Default
DIFFRESISTOR	OFF, 80, 100, 120	OFF



Equalizer

- SingleEnd入力の場合、或いはTrueのDifferential入力の場合、Equalizerを使用することが可能です。
- 設定可能範囲は以下の通りです

Attribute	Values	Default
EQ_CAL	0, 1, 2, 3, 4	0

- これらの各種設定はDesignPlannerのSpreadSheetViewから設定を行います

◆ECP3デバイスでのLVDS入力に関する使い方、注意点を記載します

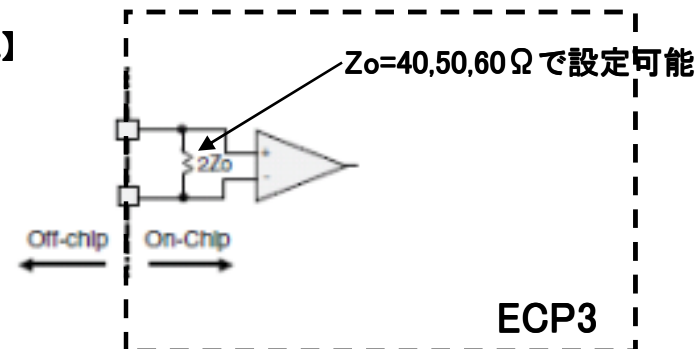
- 入力の場合には全てのバンクで入力可能。
- 使用するIOは必ず差動ペアにして下さい。差動ペアとは下記のピンリストのPL3A,PL3Bになります。T側がTrue,C側がComplementaryになります。
- DQSピンについてはLVDS入力としては使用しないでください(DDRのDQS専用回路の為。Input Registerがありません)
- 但し、Left/RightサイドはOnChipTerminationを保有しているので外付け抵抗なしで入力可能
- 差動LVDS入力として扱った場合、内部Pull-upが有効となりません。その為、転送用ケーブルが外れるようなアプリケーションの場合には、Floating防止の為、外部で処理が必要になります。処理については弊社FAEまでお問い合わせ下さい。

	Pin/Ball Function	Bank	Dual Funct	Differential
1	TMS	40		
2	TDO	40		
3	VCCJ	40		
4	TCK	40		
5	TDI	40		
6	PL3A	7	T	
7	PL5A	7	T(LVDS)*	
8	PL3B	7	C	
9	VCCIO7	7		
10	PL5B	7	C(LVDS)*	
11	PL6A	7	T	
12	GNDIO7	7		
13	PL8A	7	T(LVDS)*	
14	PL6B	7	C	
15	PL8B	7	C(LVDS)*	
16	PL9A	7	T	
17	PL11A	7	T(LVDS)*	
18	PL9B	7	C	

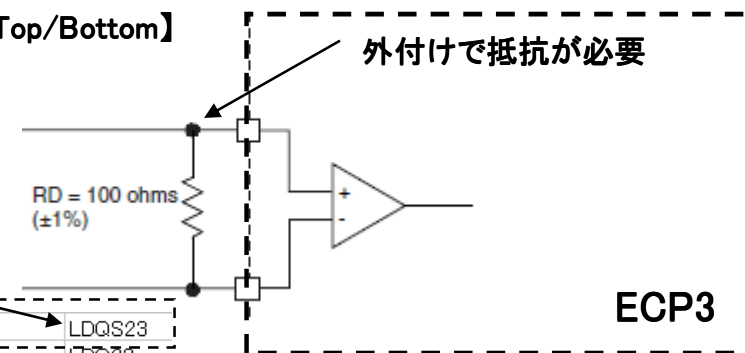
DQS専用ピンはLVDSとして使えないので注意

37	PL23A	7	T	J8	LDQS23	H3	LDQS23
38	PL21B	7	C(LVDS)*	H8	LDQ23	R4	LDQ23
39	GNDIO7	7					

【Left/Right】



【Top/Bottom】



- ・ ECP3デバイスでのLVDS出力を行う際の使い方、注意点を記載します

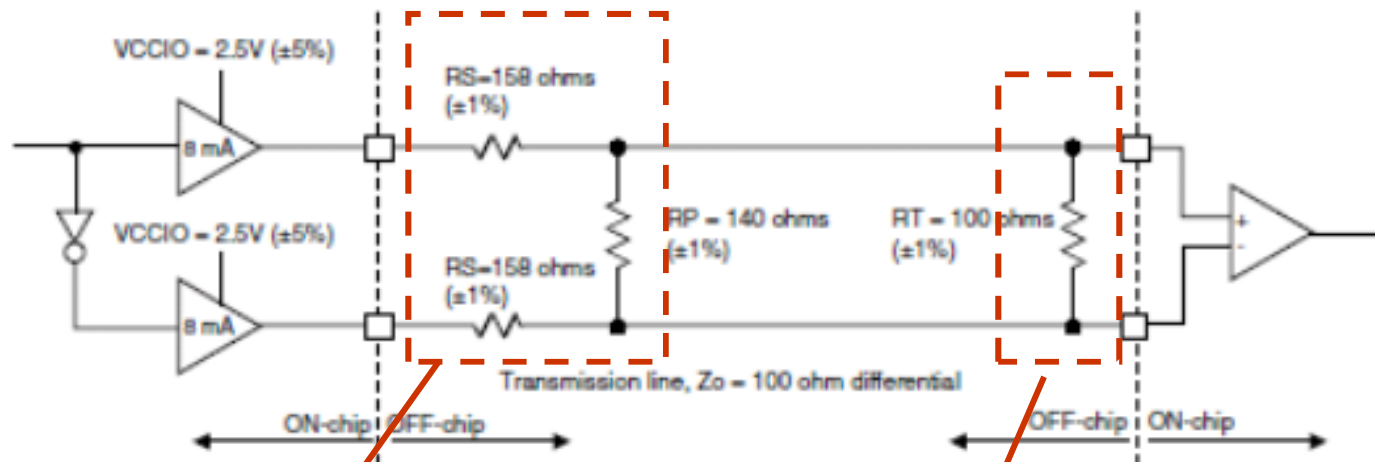
真LVDSと擬似LVDSの二つで構成可能です

■真LVDS

- LEFT,Rightバンクで対応しております。ピンリストで確認した場合Differencialの項目に * 印がついてる個所が真のLVDS対応I/Oです。
- 使用するI/Oは、差動入力と同じく差動ペアにしてください。
- DQSピンはLVDS出力としては使用できません
- 真のLVDSで使用する場合、VCCIOには2.5Vを印加する必要があります。必然的に、LVDSを使用するこのバンクは2.5Vインターフェイスでしか使用できませんのでご注意ください。
- 開発ツール上では、DesignPlannerを使用してI/O Typeを“LVDS25”に設定してください。
- 外付け抵抗は必要ありません。

■擬似LVDS(EmulateLVDS)

- Top,Bottom,Left,Right全てのバンクで擬似LVDS出力として使用できます。
- 使用するI/Oは、差動入力と同じく差動ペアにしてください。(真のLVDSと違い * 印がついていなくとも問題ございません)
- DQSピンはLVDS出力としては使用できません
- 擬似LVDSで使用する場合、VCCIOには2.5Vを印加する必要があります。必然的に、LVDSを使用するこのバンクは2.5Vインターフェイスでしか使用できませんのでご注意ください。
- 開発ツール上では、DesignPlannerを使用してI/O Typeを“LVDS25E”に設定してください。ドライブ電流は8mAにしてください。
- 外付け抵抗が必要になります。詳細は次ページをご覧ください。

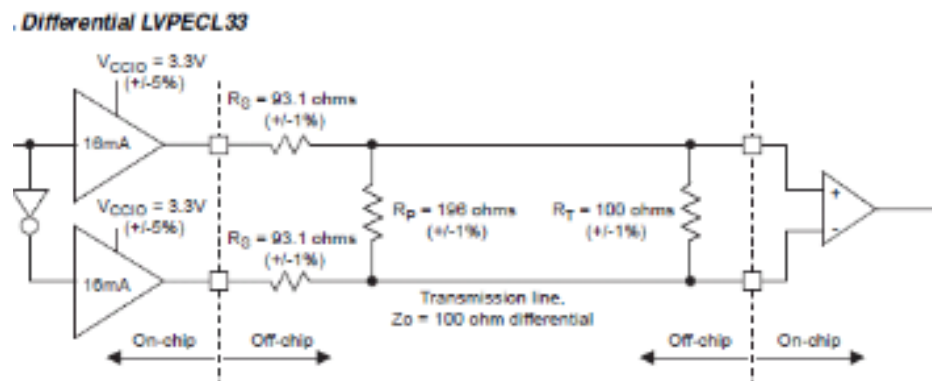


この外付け抵抗は擬似LVDSのときのみ必要となります。真のLVDSの時には必要ありません。

この終端抵抗は真のLVDSでも、擬似LVDSでも必要になります。

◆LVCMOS33DはコンプリメンタリLVCMOSバッファでの出力なので、外部抵抗を組み合わせることで各種差動スタンダードを実現することが可能です

◆以下の例ではLVPECLをLVCMOS33Dを使用してエミュレートする方法について記載します



上図はデータシートに記載されている外付け抵抗値の推奨値になりますが、これ以外の抵抗値でも実現することができます。その際は抵抗 R_s 及び R_d の値を次のように設定します。

① R_s と $R_d/2$ の並列合成抵抗値を約 50Ω にします。

⇒ インピーダンスマッチングのため、出力抵抗を線路特性インピーダンスの 50Ω に近づけます。

②出力部の外付け抵抗と終端抵抗を考慮した上で出力電圧(V_{oh}, V_{ol})が出力オフセット電圧を基準としてスレッシュホールドを満し、かつ出力電圧範囲に収まるようにします。出力オフセット電圧はドライバがLVCMOS33バッファであるため、 $3.3/2 = 1.65(V)$ になります。スレッシュホールド、出力電圧範囲はLVPECL規格に沿います。
($V_{th} = \pm 200 \sim 300mV$, $V_{od} = 600 \sim 1000 mV$)

データシートの回路図ではどのようなになるかを以下に示します。

条件①に関して

$R_s = 100\Omega$, $R_d = 187\Omega$ のとき、

$$\frac{1}{(1/100) + \{1/(187/2)\}} = 48.3(\Omega)$$

条件②に関して、右図のような出力部の等価回路を考え、 V_{oh} , V_{ol} を求めます。

$R_s = 100\Omega$, $R_d = 187\Omega$ のとき、

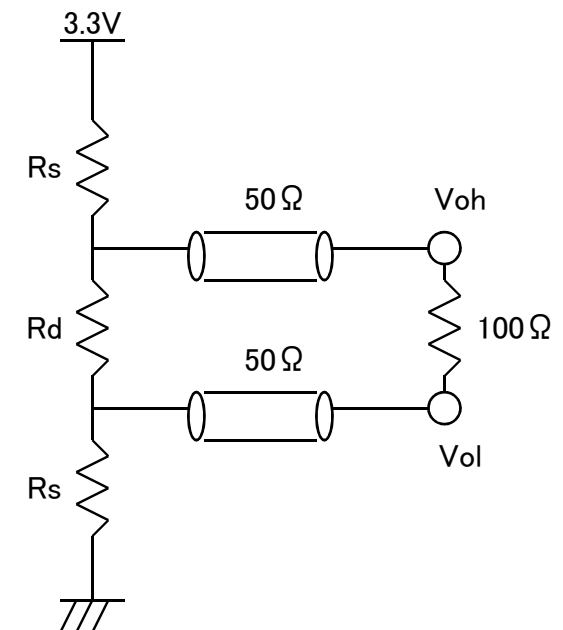
$$R_d \text{と終端抵抗}(100\Omega) \text{の並列合成抵抗} = \frac{1}{(1/187) + (1/100)} = 65.156(\Omega)$$

分圧式より、

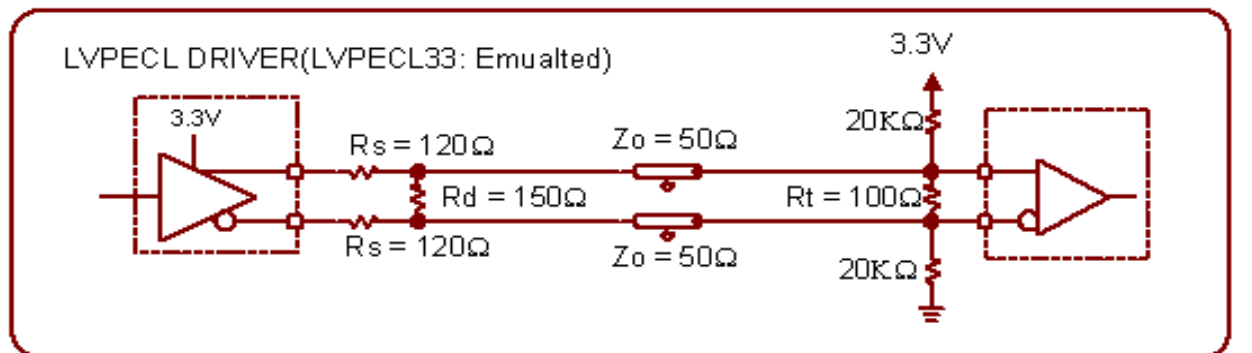
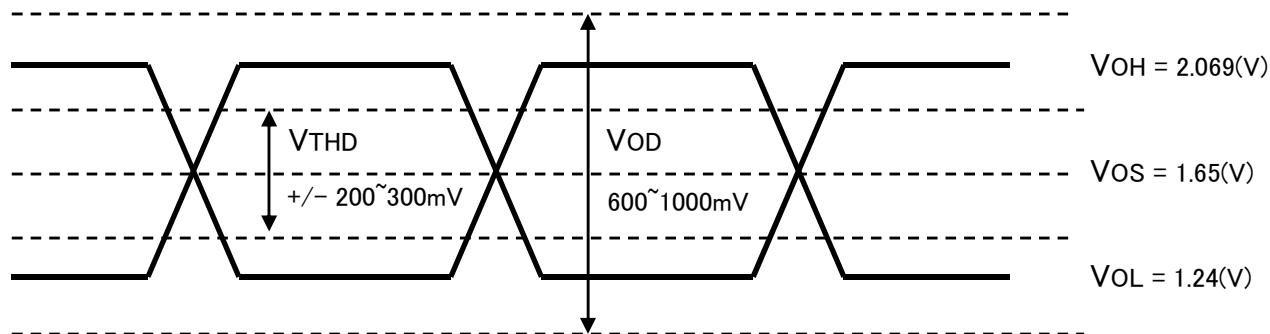
$$V_{oh} = 3.3 \times \frac{100 + 65.156}{100 + 65.156 + 100} = 2.069(V)$$

$$V_{ol} = 3.3 \times \frac{100}{100 + 65.156 + 100} = 1.24(V)$$

※線路特性インピーダンスは10MHz以上の高周波域では $Z_0 = \sqrt{L/C}$ となり、安定したDCレベルを考えると無視できるので、上式に含めません。



条件②で考えた式より求めた出力電圧値とLVPECLの各種規定電圧を合わせて図示すると下図のようになり、LVPECLの規定内に収まっていることがわかります。



ECP3のIOレジスタについて

ECP3のデバイスにおいてSYSIOには特徴のあるIOレジスタ機能が付いております。
このIOレジスタを使用して高速信号を安定してラッチすることが可能です。IOレジスタの使い方、注意点を説明します

3-1 DDR Genericモード及びDDR memoryモードについて

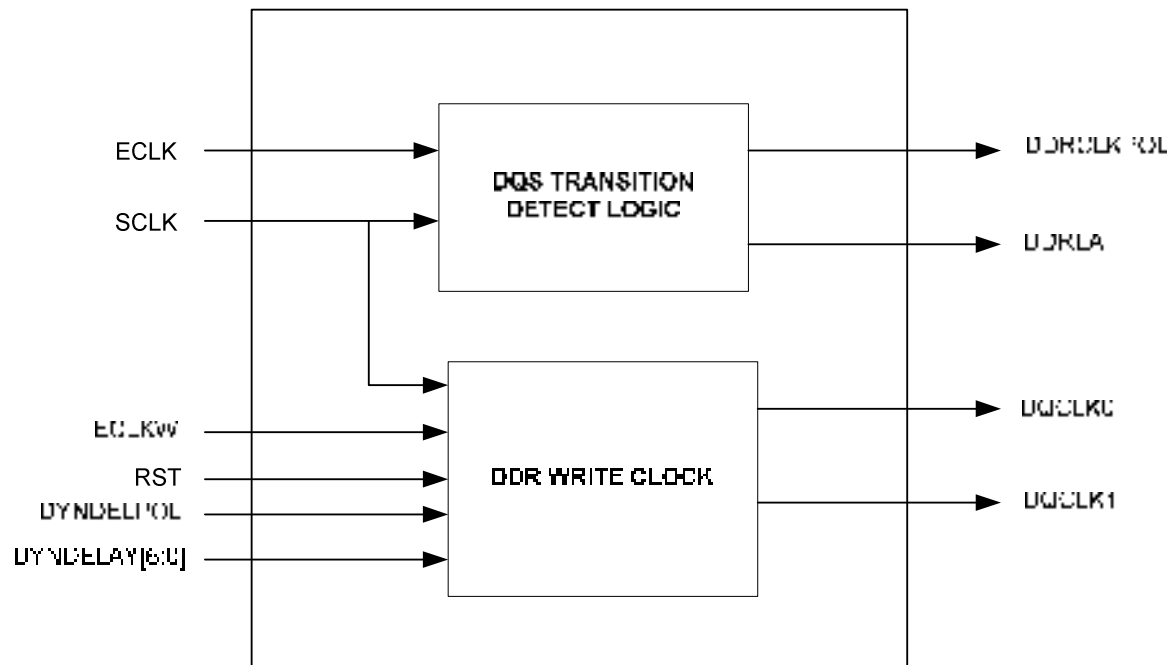
Type	Primitive	Usage
Data Input	IDDRXD	Generic DDRX1 DDR1/DDR2 Memory
	IDDRX2D	Data Input Generic DDRX2 DDR3 Memory
Data Output	ODDRXD	Generic DDRX1 DDR1/DDR2 Memory
	ODDRX2D	Generic DDRX2 DDR3 Memory
Data Tristate	ODDRTDQA	Generic DDRX2 DDR3 Memory
	OFD1S3AX	Generic DDRX1 DDR1/DDR2 Memory
DQS Output	ODDRXDQSA	DDR1/DDR2 Memory
	ODDRX2DQSA	DDR3 Memory
DQS Tristate	ODDRTDQSA	DDR1/DDR2 Memory DDR3 Memory
DQSBUF Logic	DQSBUFD	DDR3 Memory
	DQSBUFF	DDR1/DDR2 Memory
	DQSBUFE	Generic DDRX2
	DQSBUFG	Generic DDRX1
DQSDLL	DQSDLL	DDR1/DDR2/DDR3 Memory
Input Delay	DELAYB	Delay block for Generic DDRX1/X2
ECLK Stop	ECLKSYNCA	ECLK Synchronization for DDR3 Memory

DDR Genericモードで使用する
各種プリミティブ

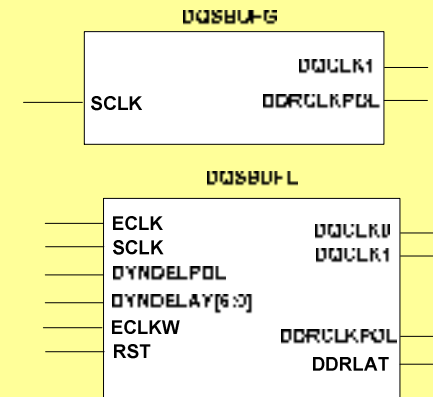
SYSIOについているIOレジスタはDDR Genericモード、DDR memoryモードで動作させることが可能です。

- ・DDR Generic モード・・・4:1シリパラパラシリ回路、2:1シリパラパラシリ回路として動作
- ・DDR memoryモード・・・DDR1,2,3用の専用IOレジスタとして動作

次ページよりDDR Genericモードで使用する際の上記プリミティブの情報及び使い方、注意点について記載します



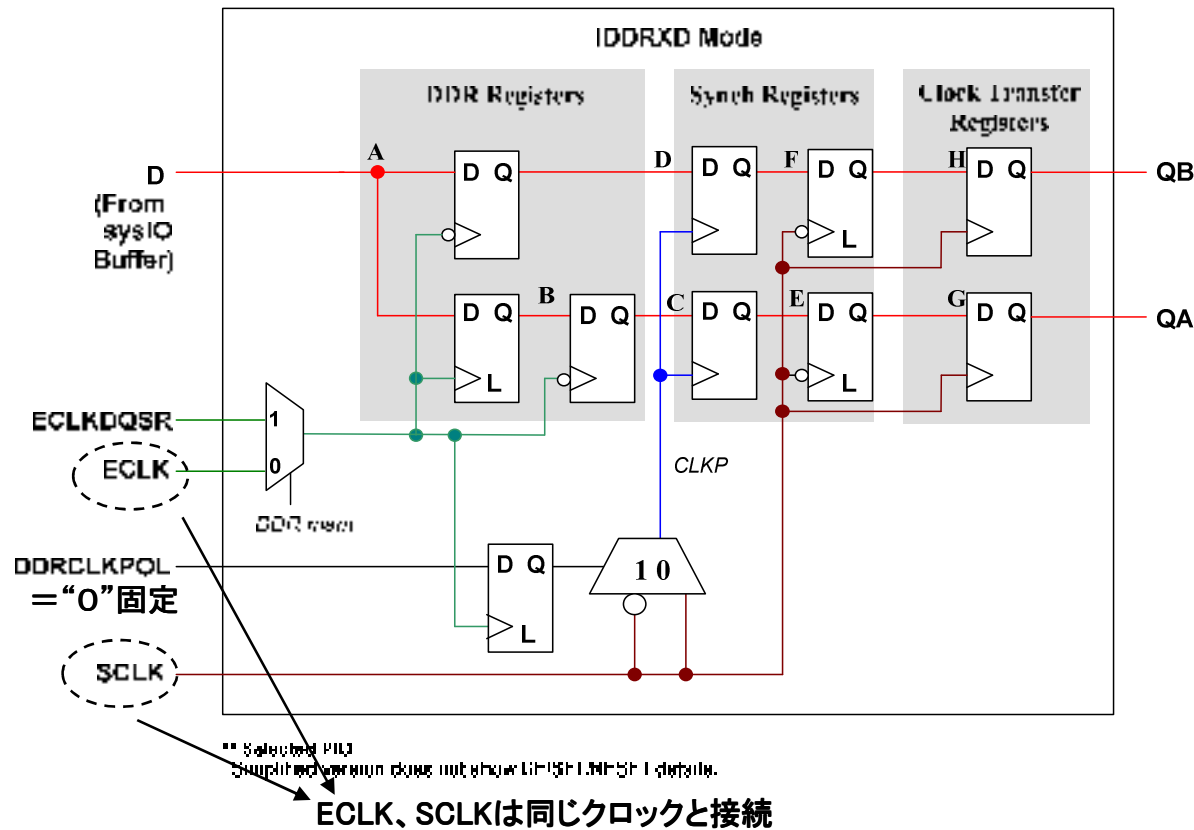
【デザインで使用するプリミティブ】



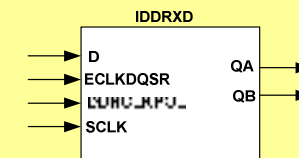
■x1 Generic DDR modeの場合
DQSBUFFGが使われます
(2:1GearBox)

■x2 Generic DDR modeの場合DQSBUFFLが使われます
(4:1GearBox)

- ・ ECP3デバイスではDDRGenericを使用する場合にはDQSバッファが必ず使用されます(ECP2/Mとの変更点)
- ・ ECLK,SCLK
 - ECLKにはエッジクロックソースが繋がります。SCLKにはFPGAロジックで使われているクロックが繋がります
- ・ DDRCLKPOL,DDRLAT
 - DDRGenericモードでは使用されず“0”固定になります(DDRメモリモードで使用)
- ・ DQCLK0,DQCLK1
 - ODDR2Dの4:1LogicでのMuxのコントロール信号として使われます (ODDR2Dは4:1のパラシリ回路ブロック)
 - ODDR2DではDQCLK1のみ使われます(ODDR2Dは2:1のパラシリ回路ブロック)
- ・ ECLKW,DYNDELPOL,DYNDELAY
 - メモリモードでのみ使用されます
- ・ TOPサイドについてはIDDR2D,IDDR2DでのみDQSBUFFが使われます(IDDR2D,2Dは1:2,1:4のシリパラ回路ブロック)
(ODDR2D,ODDR2DではDQSBUFFが使われません)



【デザインで使用するプリミティブ】

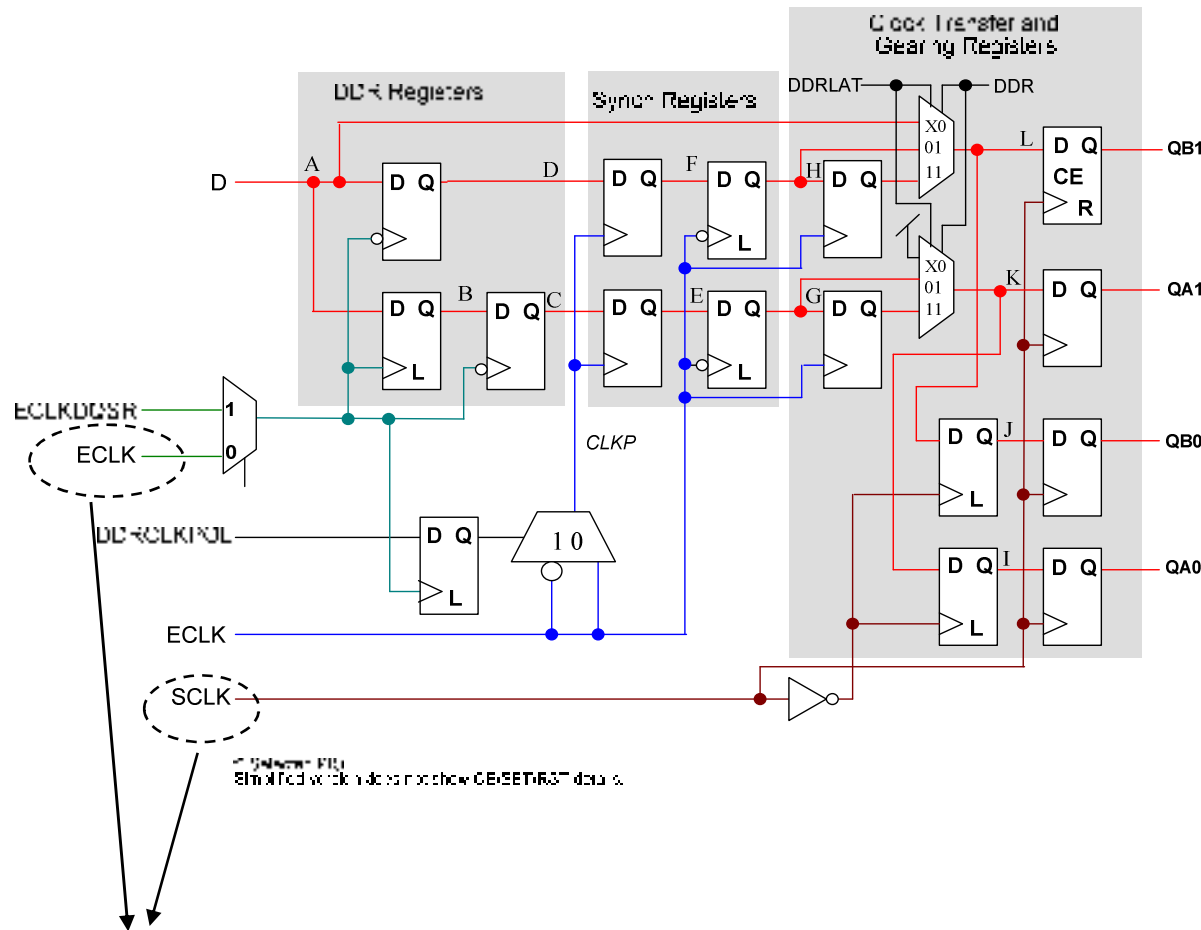


■X1 DDR Genericモードの場合

- IDDRXD が使用されます
- ECLKDQSR と SCLK は同じクロックで接続されます
- IDDRXDにおいては、ECLKDQSRはECLKルートを
使います(ECP2ではSCLKを使用していた)

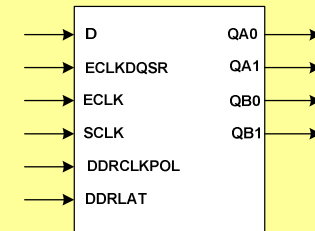
- ・ DDR Genericモードにおいて、ECLKDQSR (ECLK tree) とSCLK は同じクロックソースで接続します
- ・ ECLKDQSRはECLKが繋がります
- ・ DDRCLKPOLはDDR Genericモードにおいては“0”固定となります(メモリモードでのみ使用されます)

* DDR Memoryモードの時のみ、ECLKDQSRはDQSBUFのECLKDQSR出力と接続される



【デザインで使用するプリミティブ】

IDDRX2D

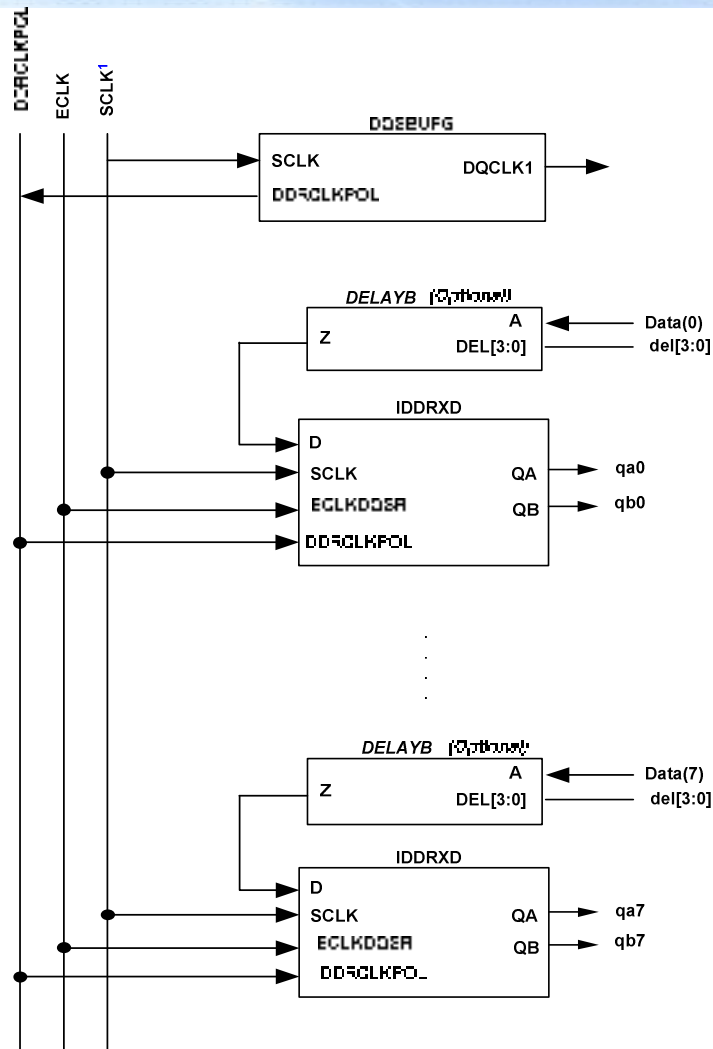


■ X2 DDR Genericモードの場合

- ・ IDDRX2Dが使用されます
- ・ Generic DDRモードにおいては、ECLKQSRとECLKは同じクロックで接続します

同じECLKで接続

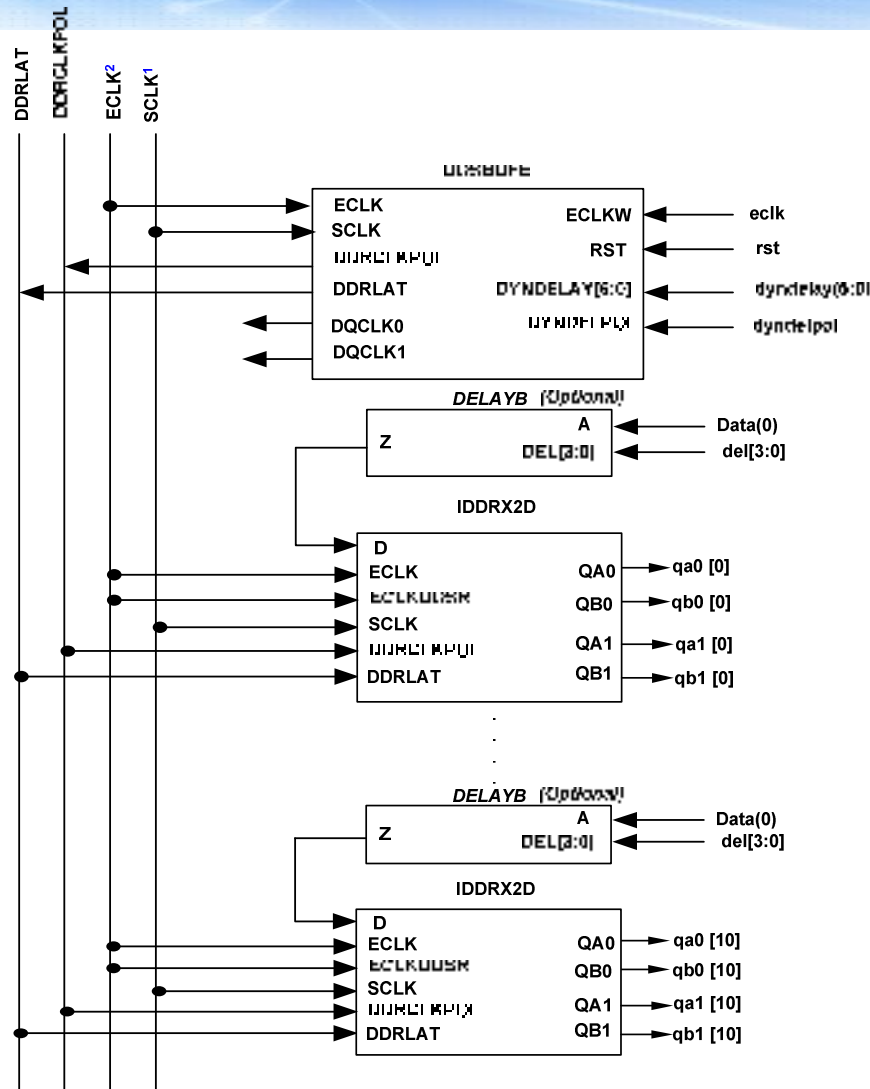
- ・ DDR Genericモードにおいて、the 1st and 2nd stage registersは同じクロックで接続します
- ・ DDR Genericモードにおいては、DDRCLKPOLと DDRLAT は“0”固定で使います



1 SCLK should be connected to the same clock source as ECLK(SCLK=ECLK)

■ 特定の1DQSグループ内にインプリメントした例

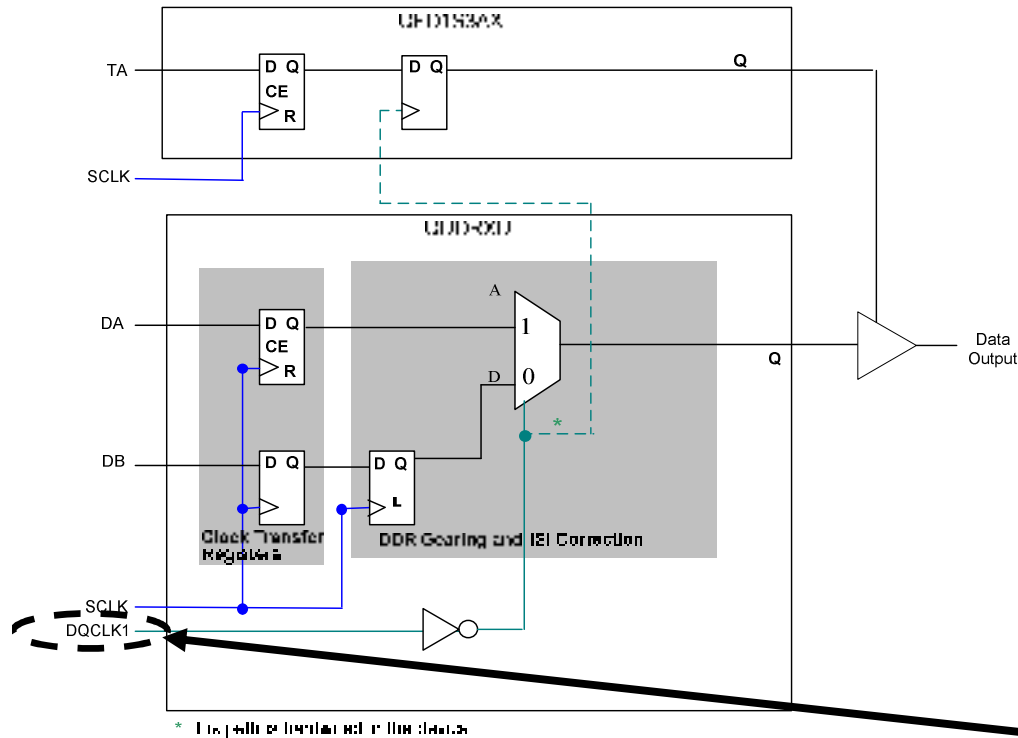
- 10ビット以上のLVDSグループを配置する場合、複数のDQSグループを使用してインプリメントされる。このとき、クロック信号はすべて同じものが使用されます
- DELAYB block はデータラインの遅延調整回路です。IPEXPRESSでx1 DDR Genericモードブロックを生成した際、自動的に追加されます。
- DDRCLKPOL入力はDQSBUFGのDDRCLKPOL出力と接続します。但し、DDR GenericモードではDDRCLKPOLは“0”固定となっています
- SCLK とECLK は同じクロックで接続されます



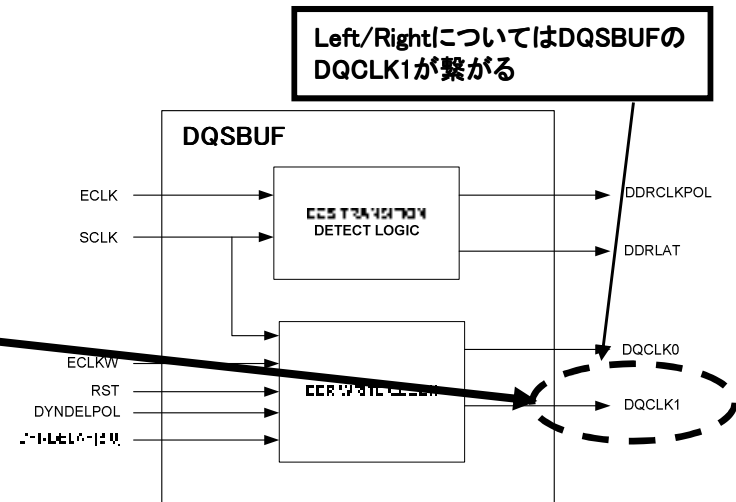
1. SCLK should be generated from ECLK or RST (SCLK = % + 10 %)
2. ECLKDQSR and ECLK inputs to the IDDRX2D should be connected to the same clock

■特定の1DQSグループ内にインプリメントした例

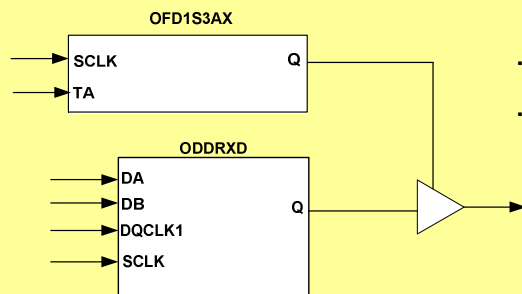
- 10ビット以上のLVDSグループを配置する場合、複数のDQSグループを使用してインプリメントされます。このとき、ECLKについては全て共通のECLKが各DQSBUFEに繋がります
- ECLK、ECLKDQSRはエッジクロックルーティングを使用します
- DELAYB block はデータラインの遅延調整回路です。IPEXRESSでGenericDDRブロックを生成した際に自動的に追加されます
- DDRCLKPOL入力はDQSBUFGのDDRCLKPOL出力と接続します。但し、ここではDDRCLKPOLは“0”固定となっています
- SCLKはECLKと同一ソースのクロック信号から生成してたものを使用して下さい(例えば、1つのPLLを使用してCLKOS,CLKOKをECLK,SCLKとして使ったり、CLKDIVを使用する等)



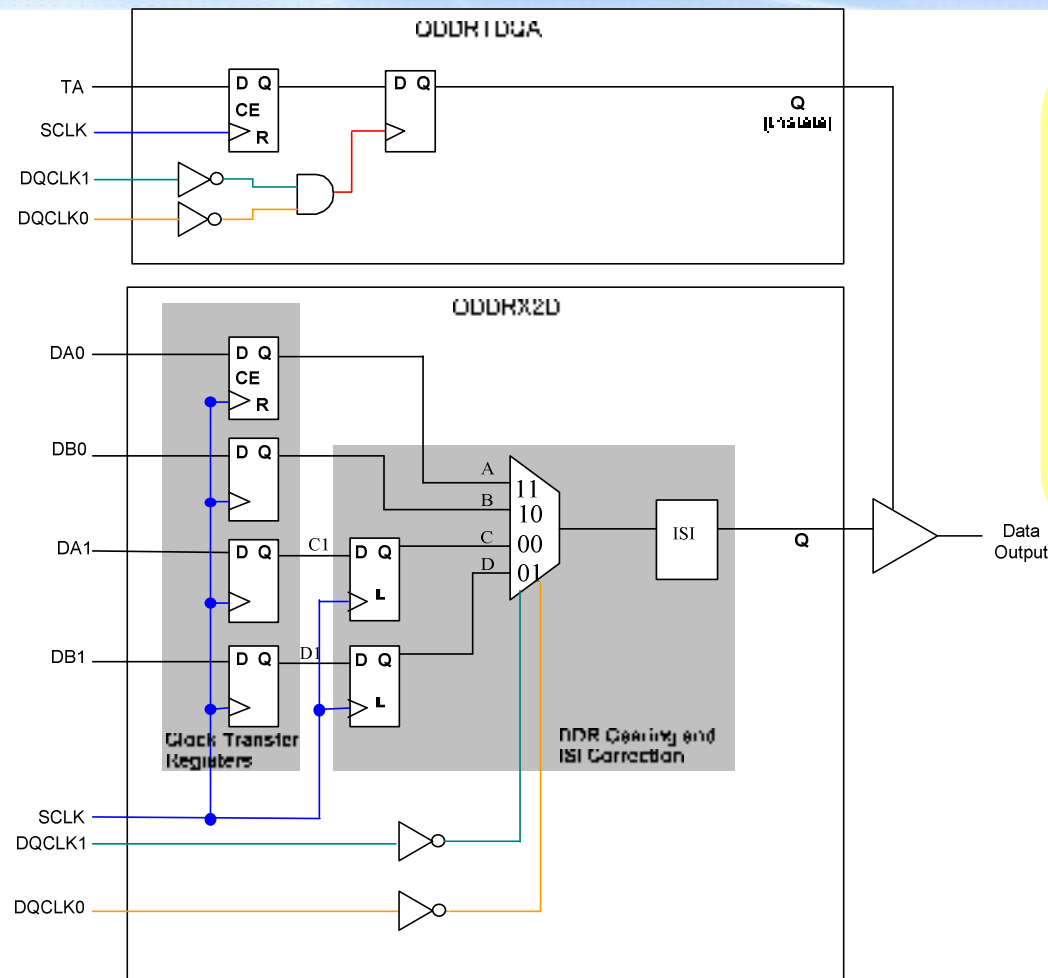
- ・ DQCLK1はDQSBUFから生成された信号が繋がります。出力信号のMuxを行います。
- ・ TOPサイドについてDQBUFが実装されていない為、DQCLK1についてはSCLKがそのまま繋がります。



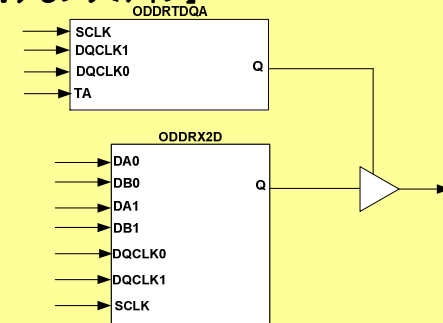
【デザインで使用するプリミティブ】



- ・ DQCLK0、DQCLK1はDQSBUFの出力に接続します
- ・ SI Calibration機能はISL_CALアトリビュートで設定を行います

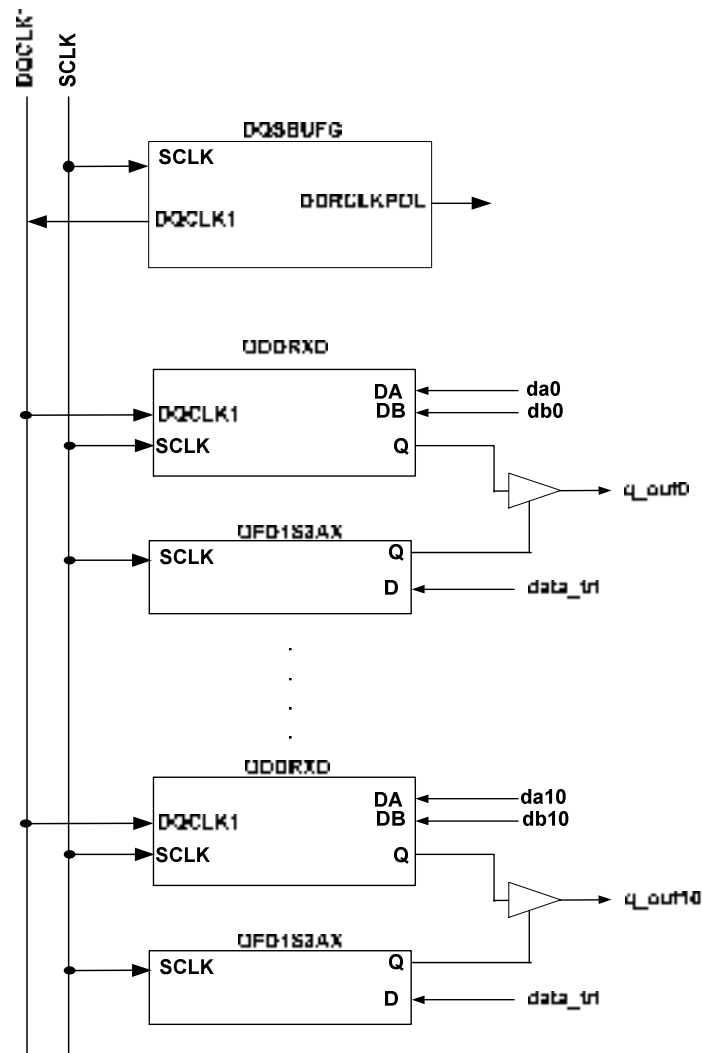


【デザインで使用するプリミティブ】



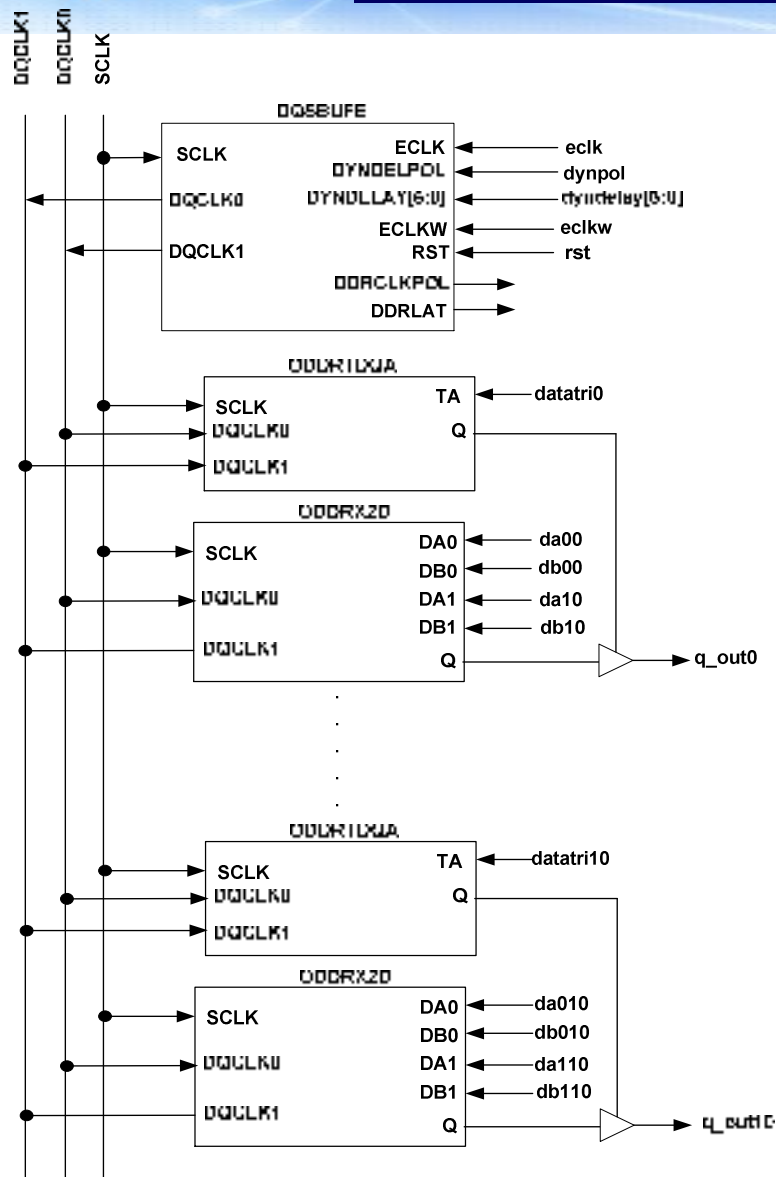
- DQCLK0、DQCLK1はDQSBUFの出力に接続します
- SI Calibration機能はISI_CALアトリビュートで設定を行います

- ・ DQCLK0、そしてDQCLK1はDQSBUFから生成され、出力の4:1Muxのセレクタを制御します
- ・ ISI Correction機能はODDR2Dに実装されており使用可能です。Left/Rightサイドのみ実装されています



■特定の1DQSグループ内にインプリメントした例

- ・ 10ビット以上のLVDSグループを配置する場合、複数のDQSグループを使用してインプリメントされる。このとき、クロック信号はすべて同じものが使用されます
- ・ DQCLK1はDQSBUF0が生成したDQCLK1と接続されます
- ・ TOPサイドについてはDQSBUF0が実装されておりません。よってDQCLK1はSCLK信号が接続されます



■特定の1DQSグループ内にインプリメントした例

- 10ビット以上のLVDSグループを配置する場合、複数のDQSグループを使用してインプリメントされる。このとき、クロック信号はすべて同じものが使用されます
- DQSBUFGから生成されたDQCLK1、DQCLK0はODDRX2DのDQCLK0,1と接続します
- TOPサイドについてDQSBUFGが実装されておりません。よってDQCLK0,1が生成できない為ODDRX2Dはサポートされません

Left/Right sidesで使用する際の注意点 -

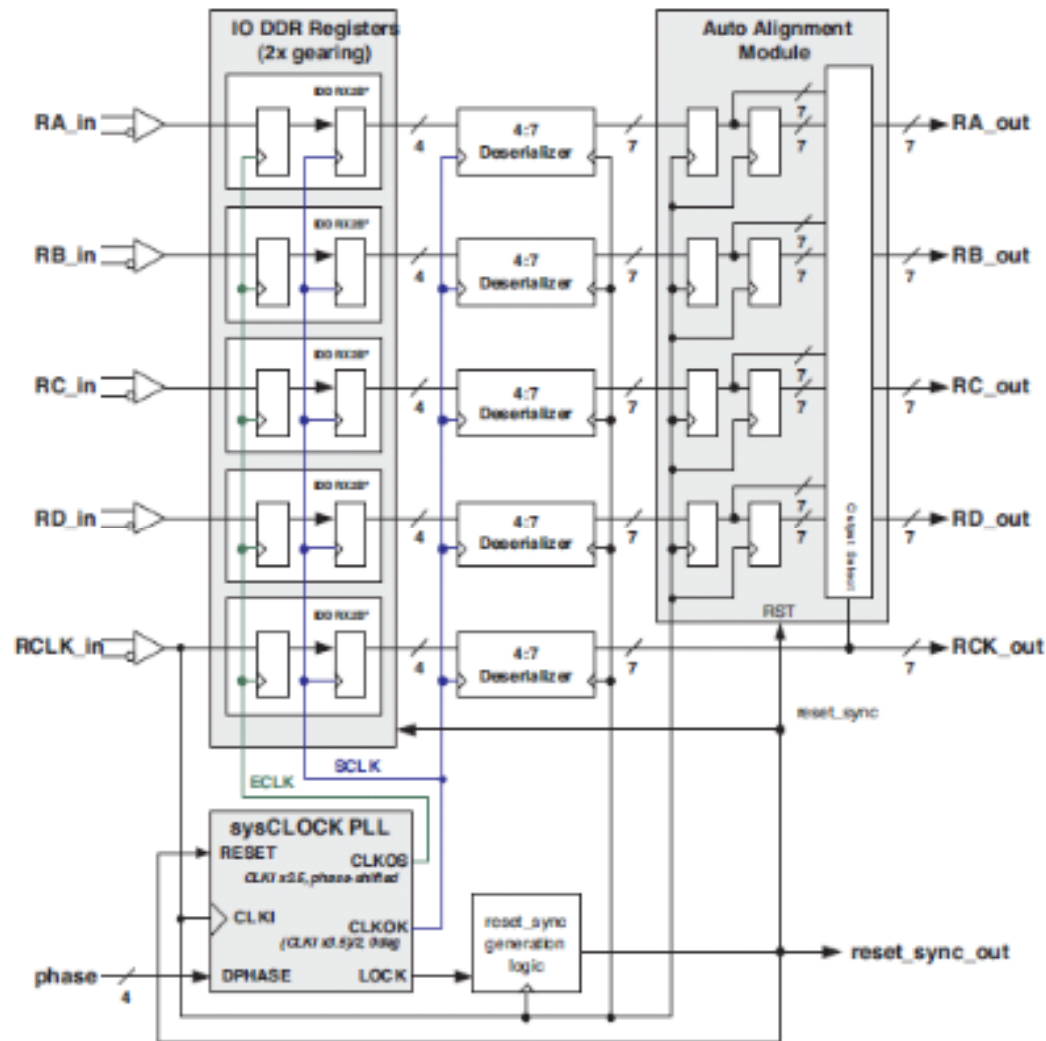
- ・ IDDRX, IDDRX2, ODDRX and ODDRX2全て使用可能です
- ・ 上記モードにおいてDQSBUFは必ず使用する必要があります。DQSBUFはDQSグループ毎に1つ用意する必要があります。
- ・ IDDRX, IDDRX2を使用する際、エッジクロックはECLK2が使われます (ECLK1は使えません) によって各辺には同一クロックソースのIDDRX, IDDRX2しか繋げることができません
- ・ DQSピンにはIDDRX, IDDRX2, ODDRX and ODDRX2が用意されていません。Inputレジスタは存在しますので、単純なInputRegisterとしては使用可能です

Top Sideで使用する際の注意点 -

- ・ IDDRX, IDDRX2, ODDRXがサポートされています。ODDRX2はDQCLK0,1ラインが存在しない為サポートされていません
- ・ IDDRX, IDDRX2を使用する際にはDQSBUFを用意する必要があります。ODDRXを使用する場合にはDQSBUFは使用する必要はありません。この時、DQCLK0はSCLKを繋げてください
- ・ IDDRX, IDDRX2を使用する際、エッジクロックはECLK2が使われます (ECLK1は使えません) によって片辺には同一クロックソースのIDDRX, IDDRX2しか繋げることができません
- ・ DQSピンにはIDDRX, IDDRX2, ODDRX and ODDRX2が用意されていません。Inputレジスタは存在しますので、単純なInputRegisterとしては使用可能です
- ・ TOPサイドではTrueLVDSを使用できません。EmulateLVDSでの対応になりますのでご注意ください (P13参照)

ECP3での7:1LVDSデザインの実装方法

前項で説明したDDR Genericモードを使用して7:1LVDSデザインを実装する際の実装方法、注意点について記載します。このデザインを使用してチャネルリンク、カメラリンクをエミュレートすることが可能です

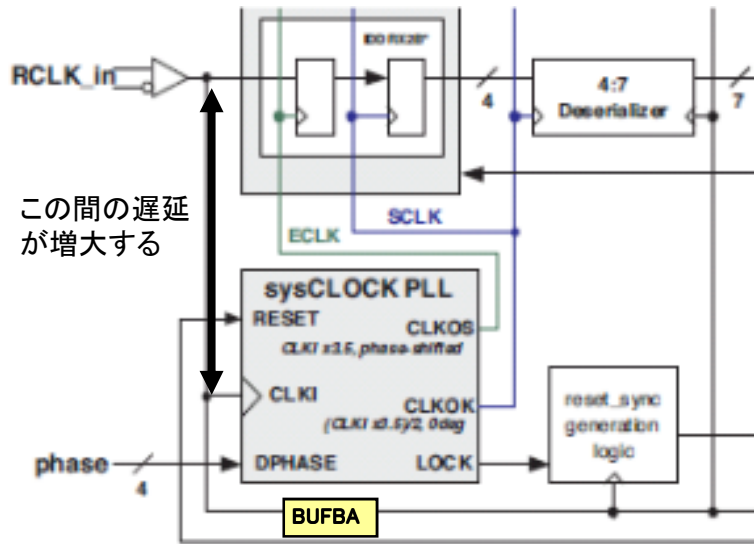


Lattice社ではチャネルリンク、カメラリンク機能をエミュレートするためのリファレンスデザインを用意しています。このデザインでは先に説明したIDDRX2, ODDRX2を使用したデザインとなっています。

この7:1LVDSデザインをインプリメントする際の注意点について記載します

7:1LVDSリファレンスデザインブロック図

・ RCLK_in信号の扱い方



本デザインでは、RCLK_in信号の扱い方に注意が必要です。以下に記載のポイントに関して処理を行わない場合、RCLK_in信号とPLL間の配線パスが内部一般配線を経由してしまう為遅延が増大してしまいます

(ポイント1)IDDRX2Dモジュールの修正

IDDRX2DはIPEXPRESSで生成します。この生成されたソース中では、RCLK_in信号及びその他の信号(RA_in, RB_in, RC_in, RD_in)が一度入力Buffer(IB)でバッファリングされています。このIBを通さないような編集を行ってください。(例)IBをコメントアウト、アサイン文で繋げます

```
// IB buf_Data0_in_inst (.I(Data[0]), .O(buf_Data0));
assign buf_Data0 = Data[0];
```

(ポイント2)BUFBAの追加

デザインのTOP等において上記BUFBAモジュールを追加し、RCLK_in信号をつなげます。このBuffer出力の信号をRCLK_inの代わりに使用します。(例)RCLK_in_bufをRCLK_inの代わりに使用します。この時、冗長回路として論理圧縮されないようにAttributeを付けてください

■記述例

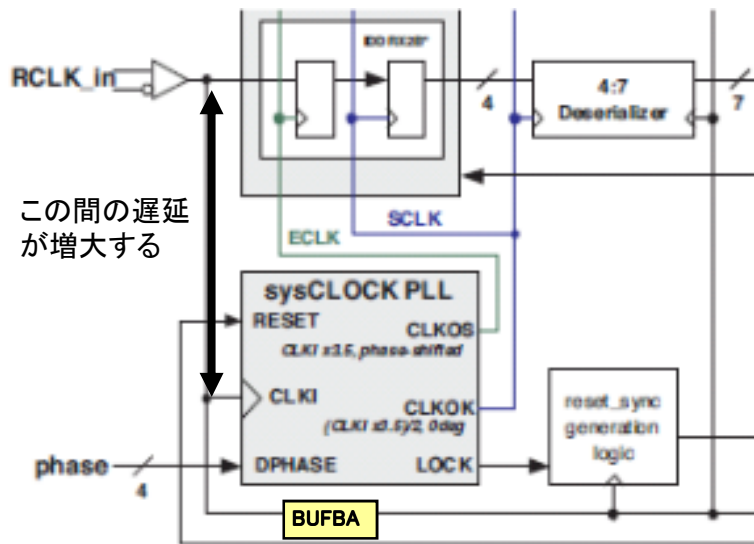
```
wire RCLK_in_buf /* synthesis syn_keep=1 nomerge="on" */;
```

```
BUFBA rclk_buf(.Z(RCLK_in_buf), .A(RCLK_in))/* synthesis syn_noprune=1 */;
```

■追加するBUFBAモジュール

```
module BUFBA (Z, A);
    output Z ;
    input A ;
```

・ RCLK_in信号の扱い方



(ポイント3)RCLK_in_buf信号への制約

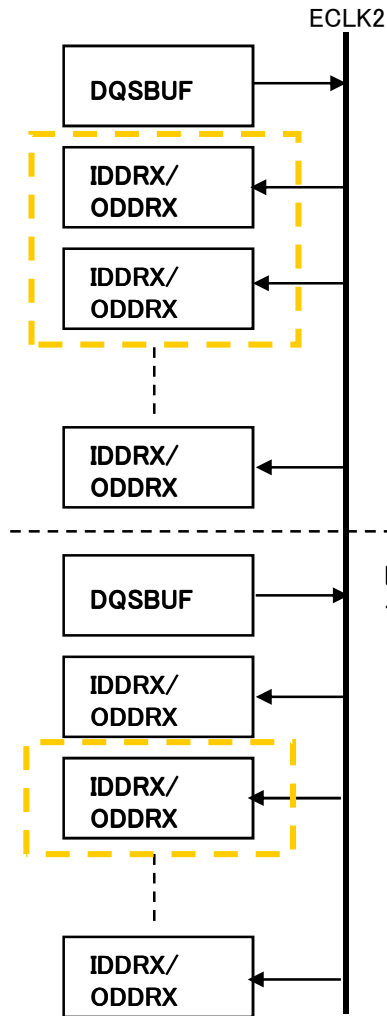
最後にRCLK_in_bufを必ずPrimaryクロックライン、或いはSecondaryクロックラインに乗せるようにしてください。クロックラインへの制約についてはDesignPlannerにて行います。

・ PLLの扱い方

(ポイント1)PLLフィードバック信号への制約

PLLのフィードバック信号はInternalにすることをお奨めします

・ DQSBUFの扱い方



ECP3デバイスでは前述の通り、1:4(2).4(2):1専用回路+DQSBUFで構成されています。又、IOにはDQSグループが存在し、そのグループ内において、1つのDQSBUFを共通で使用します。(下記の例参照)

注意すべきは、例えば7:1LVDSデザインをインプリメントする際に、左記のDQSグループ1、DQSグループ2の両グループのIOを使用した場合です。IPEXPRESSで生成したGenericDDRマクロ(IDDRX,ODDRX)内にはDQSBUFのインスタンスが一つしか存在しない為、上記の場合にはDQSBUFが一つしかない為エラーが表示されます。

よってDQSグループを跨いで7:1LVDSデザインを使用する際には、DQSBUFを手動で追加する必要があります。

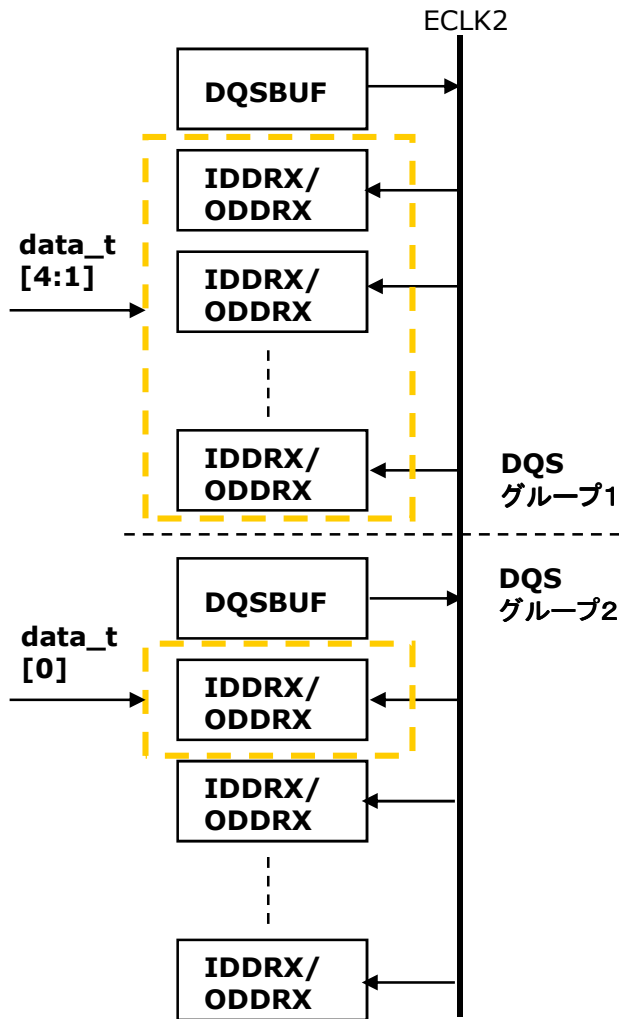
次ページに具体的な例を記載します。

DQS14グループ

22	PL14A	7	F5	LDQS14
23	PL12B	7	C2	LDQ14
24	GNDIO7	7		
25	PL14B	7	F4	LDQ14
26	PL15A	7	D2	LDQ14
27	PL17A	7	E1	LDQ14
28	PL15B	7	D1	LDQ14
29	GNDIO7	7		
30	PL17B	7	F1	LDQ14
31	PL18A	7	G1	LDQ23
32	PL20A	7	J7	LDQ23
33	PL18B	7	H1	LDQ23
34	PL20B	7	J6	LDQ23
35	PL21A	7	H2	LDQ23
36	VCCIO7	7		
37	PL23A	7	J3	LDQS23
38	PL21B	7	H3	LDQ23

DQS23グループ

• DQSBUFの扱い方



```

DQSBUF Inst2_DQSBUF (.ECLK(eclk), .SCLK(sclk_t), .ECLKW(eclk), .RST(rst),
.DYNDELPOL(t_dyndelpol), .DYNDelay6(t_dyndelay6), .DYNDelay5(t_dyndelay5),
.DYNDelay4(t_dyndelay4), .DYNDelay3(t_dyndelay3), .DYNDelay2(t_dyndelay2),
.DYNDelay1(t_dyndelay1), .DYNDelay0(t_dyndelay0), .DQCLK0(dqclk0),
.DQCLK1(dqclk1), .DDRCLKPOL(ddrclkpol), .DDRlat(ddrlat));

IDDRX2D Inst1_IDDRX2D4 (.D(data_t4), .SCLK(sclk_t), .ECLK(eclk), .ECLKDQSR(eclk),
.DDRlat(1'b0), .DDRCLKPOL(1'b0), .QA0(qa04), .QB0(qb04),
.QA1(qa14), .QB1(qb14));

IDDRX2D Inst1_IDDRX2D3 (.D(data_t3), .SCLK(sclk_t), .ECLK(eclk), .ECLKDQSR(eclk),
.DDRlat(1'b0), .DDRCLKPOL(1'b0), .QA0(qa03), .QB0(qb03),
.QA1(qa13), .QB1(qb13));

IDDRX2D Inst1_IDDRX2D2 (.D(data_t2), .SCLK(sclk_t), .ECLK(eclk), .ECLKDQSR(eclk),
.DDRlat(1'b0), .DDRCLKPOL(1'b0), .QA0(qa02), .QB0(qb02),
.QA1(qa12), .QB1(qb12));

IDDRX2D Inst1_IDDRX2D1 (.D(data_t1), .SCLK(sclk_t), .ECLK(eclk), .ECLKDQSR(eclk),
.DDRlat(1'b0), .DDRCLKPOL(1'b0), .QA0(qa01), .QB0(qb01),
.QA1(qa11), .QB1(qb11));

IDDRX2D Inst1_IDDRX2D0 (.D(data_t0), .SCLK(sclk_t), .ECLK(eclk), .ECLKDQSR(eclk),
.DDRlat(1'b0), .DDRCLKPOL(1'b0), .QA0(qa00), .QB0(qb00),
.QA1(qa10), .QB1(qb10));
    
```

修正前

左の図でIOのアサインメントを見ると
DQSグループ1
DQSBUF+4IO
DQSグループ2
DQSBUF+1IO
となっている。

現在のGenericDDRのソースを
見ると
DQSBUF+5IOとなっており配置
配線できずエラーが出る。。

```

DQSBUF Inst3_DQSBUF (.ECLK(eclk), .SCLK(sclk_t), .ECLKW(eclk), .RST(rst),
.DYNDELPOL(t_dyndelpol), .DYNDelay6(t_dyndelay6), .DYNDelay5(t_dyndelay5),
.DYNDelay4(t_dyndelay4), .DYNDelay3(t_dyndelay3), .DYNDelay2(t_dyndelay2),
.DYNDelay1(t_dyndelay1), .DYNDelay0(t_dyndelay0), .DQCLK0(dqclk0),
.DQCLK1(dqclk1), .DDRCLKPOL(ddrclkpol), .DDRlat(ddrlat));

DQSBUF Inst2_DQSBUF (.ECLK(eclk), .SCLK(sclk_t), .ECLKW(eclk), .RST(rst),
.DYNDELPOL(t_dyndelpol), .DYNDelay6(t_dyndelay6), .DYNDelay5(t_dyndelay5),
.DYNDelay4(t_dyndelay4), .DYNDelay3(t_dyndelay3), .DYNDelay2(t_dyndelay2),
.DYNDelay1(t_dyndelay1), .DYNDelay0(t_dyndelay0), .DQCLK0(dqclk0_1),
.DQCLK1(dqclk1_1), .DDRCLKPOL(ddrclkpol_1), .DDRlat(ddrlat_1));

IDDRX2D Inst1_IDDRX2D4 (.D(data_t4), .SCLK(sclk_t), .ECLK(eclk), .ECLKDQSR(eclk),
.DDRlat(ddrlat), .DDRCLKPOL(ddrclkpol), .QA0(qa04), .QB0(qb04),
.QA1(qa14), .QB1(qb14));

IDDRX2D Inst1_IDDRX2D3 (.D(data_t3), .SCLK(sclk_t), .ECLK(eclk), .ECLKDQSR(eclk),
.DDRlat(ddrlat), .DDRCLKPOL(ddrclkpol), .QA0(qa03), .QB0(qb03),
.QA1(qa13), .QB1(qb13));

IDDRX2D Inst1_IDDRX2D2 (.D(data_t2), .SCLK(sclk_t), .ECLK(eclk), .ECLKDQSR(eclk),
.DDRlat(ddrlat), .DDRCLKPOL(ddrclkpol), .QA0(qa02), .QB0(qb02),
.QA1(qa12), .QB1(qb12));

IDDRX2D Inst1_IDDRX2D1 (.D(data_t1), .SCLK(sclk_t), .ECLK(eclk), .ECLKDQSR(eclk),
.DDRlat(ddrlat), .DDRCLKPOL(ddrclkpol), .QA0(qa01), .QB0(qb01),
.QA1(qa11), .QB1(qb11));

IDDRX2D Inst1_IDDRX2D0 (.D(data_t0), .SCLK(sclk_t), .ECLK(eclk), .ECLKDQSR(eclk),
.DDRlat(ddrlat_1), .DDRCLKPOL(ddrclkpol_1), .QA0(qa00), .QB0(qb00),
.QA1(qa10), .QB1(qb10));
    
```

修正後

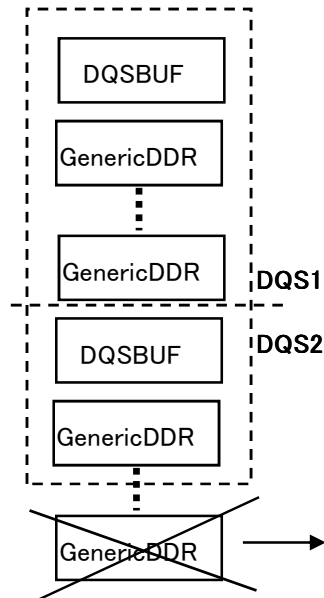
手動でDQSBUFを追加する。
この例ではDQSBUFIInst3が
DQSグループ1のDQSBUF、
DQSBUFIInst2がDQSグループ
2のDQSBUFとなる。

DQSグループ2用のDQSBUFを
作成したので、後はdata_t[0]が
繋がるIDDRXとグループ2用の
DQSBUFのインスタンスを繋げる

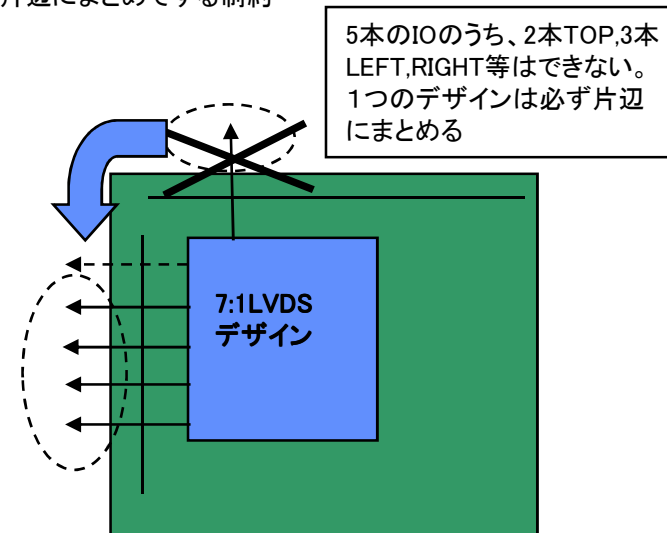
- ・ 受信側の7:1LVDSデザインに関しては必ずエッジクロックラインのECLK2を使用します。各辺にはECLK2はひとつずつしか存在しない為受信の7:1LVDSデザインは各辺毎に1つしか配置できません
- ・ 送信側の7:1LVDSデザインについてはECLK1,ECLK2のどちらでも使用可能なため各辺毎に2つ配置可能です(但し、内部のPLLの数、内部クロック配線リソースに注意が必要です)
- ・ 送信7:1LVDSデザインに関して、True LVDS、内部終端抵抗は左右BANKIにしか配置されていません。TOPBANKはEmulateLVDS、外部終端となりますのでご注意ください。
- ・ 又TOPBANKではODDRX2D(4:1パラシリ)がサポートされていません。高速で送信7:1LVDSデザインを使用する際には左右BANKを使用するようにして下さい。
- ・ あるDQSグループでDDRメモリ回路を実装している場合、そのブロックにはGenericDDRマクロを配置できません(DQSBUFの使い方が違う為)

- 7:1LVDSデザイン中のソースシンクロナスクロック(RCLK_IN)については必ずPLL専用ピンにアサインしてください。データと同じ辺に存在するPLL専用ピンを使用して下さい。
- DQSピンについてはGenericDDR(IDDRX,ODDRX)機能がありません。ピンを使用しないで下さい
- またがったDQSグループのIOを使用している場合で、残ったIOはGenericDDRとしては使用できないのでご注意ください(DQSBUFを使ってしまうので、残ったGenericDDR回路にクロックを供給できないため。下記例を①参照)
- TrueLVDS出力ピンは、特定ピンでしかサポートされておりませんのでご注意ください
- 送信、受信どちらにおいても1つの7:1LVDS回路を配置する際には左右、上どちらか1つの辺にまとめて配置するようにピン配置をしてください。

①またがったDQSグループを使った場合の制約



②ピン配置は片辺にまとめる制約



ECP3のコンフィグレーションについて

■コンフィグレーション用のバンク

- ・Bank8はコンフィグレーション専用バンクとなります。
 - ECP2は Bank8は ConfigとI/O兼用。

■コンフィグレーション開始電源

- ・電源は 3つの条件が全部揃えたら、コンフィグレーションを開始します。
 - $V_{cc} > 0.8V$
 - $V_{cc} > 2.7V$
 - $V_{ccio8} > 0.8V$

ECP2/Mデバイスがサポートしているコンフィグレーションモードは以下の通りです

SPIモード (SPI Master Single)

SPI I/Fを保有しているFlashメモリと直接接続し、コンフィグレーションを行います。

SPImモード (SPIM)

SPIフラッシュメモリの記憶領域を2つに分け、デュアルブート機能を用いたコンフィグレーションを行います。

デュアルブートに関する詳細はデータシート及びテクニカルノート(TN1108)を参照してください。

MPCMモード (MPCM)

8ビットワイドパラレルコンフィグレーション

SSPIモード (SSPI)

スレーブSPIコンフィグレーション

スレーブシリアルモード (SCM)

FPGAがスレーブ(クロックCCLKから入力)となり、外部シリアルROM、或いはCPUからのデータを読み込みコンフィグレーションを行うモードです。

スレーブパラレルモード (SPCM)

FPGAがスレーブ(クロックCCLKから入力)となり、外部シリアルROM、或いはCPUからのデータを読み込みコンフィグレーションを行うモードです。

ispJTAGモード (JTAG)

ispJTAGポートを介してコンフィグレーションを行うモードです。

■コンフィグレーション選択ピン

CFG[2:0](後述)を以下のように設定することで、各種コンフィグレーションモードを決定します。

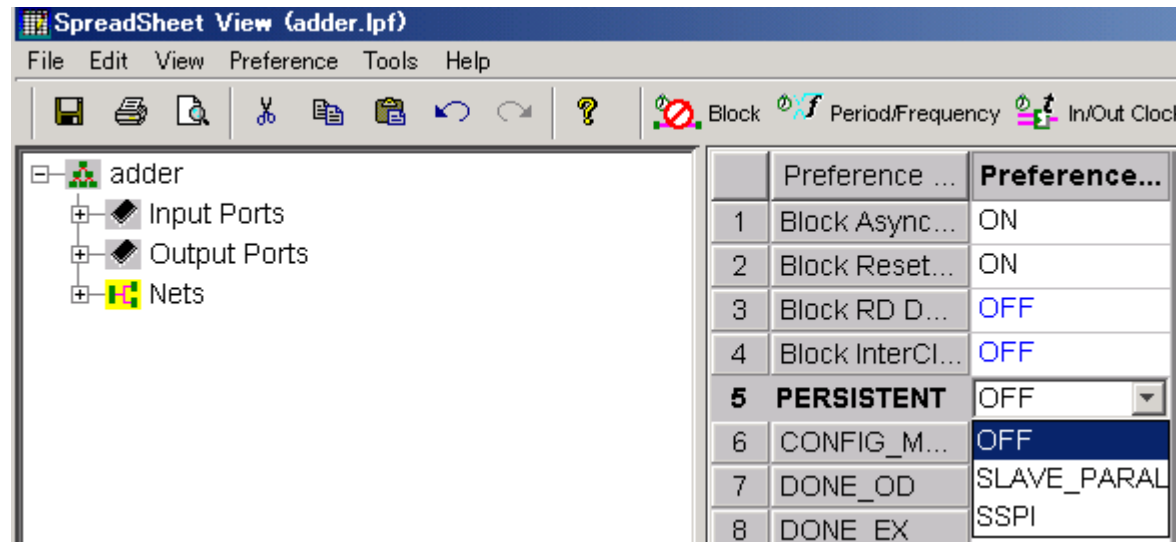
モード		Bus size (bit)	MaxFreq (MHz)	CFG[2]	CFG[1]	CFG[0]	Clock Pin	備考
SPI Master	single	1	33	0	0	0	MCLK	
	Multiple	1	33	0	1	0	MCLK	
MPCM		8	33	0	1	1	MCLK	
Slave SPI		1	33	1	0	0	CCLK	
Slave Serial		1	33	1	0	1	CCLK	
Slave Parallel		8	33	1	1	1	CCLK	
JTAG		1	25	x	x	x	TCK	CFG[2:0]の設定によらず、常に有効

- 通常動作中にコンフィグレーションSRAMにアクセスするにはPersistent設定が必要になります。(次頁参照)
- SPI Masterモード、MPCMモードの最低動作周波数は2.5MHzです。

■ Persistent Optionについて

PERSISTENTは、通常動作中にコンフィグレーションSRAMにアクセスする為のDual Perposeピンを使用できるようにするアトリビュートです。

PERSISTENTアトリビュートはispLEVERのDesignPlannerで設定します。

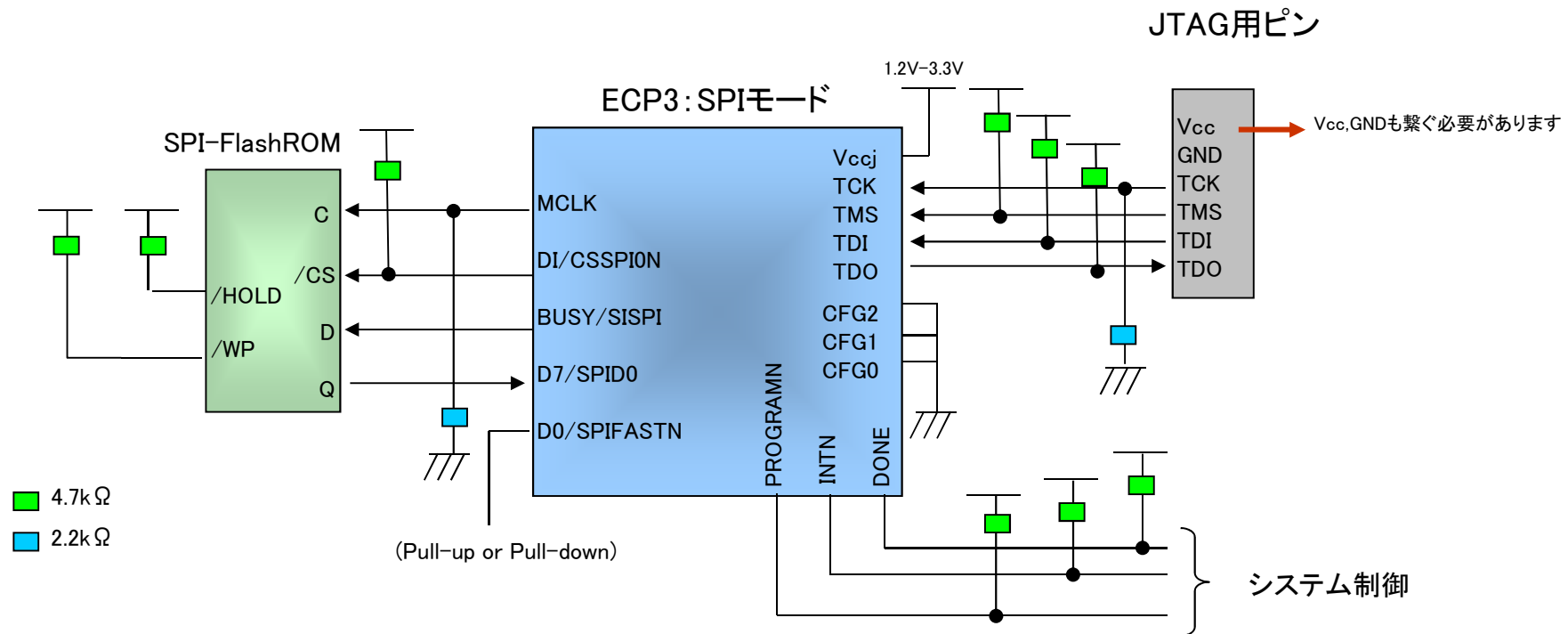


■ Persistent設定で有効になるピン

Persistent Setting	Pins
OFF	<None>
SLAVE_PARALLEL	D [0:7], CSN, CS1N, WRITEN, BUSY, CSON, DI ¹ , MCLK ¹
SSPI	SI, SO, SN, HOLDN

1. Not used for Slave Parallel mode, but enabled through the PERSISTENT setting (no longer user I/O).

- SLAVE_PARALLEL : Slave Parallel経由でアクセスする場合の設定
- SSPI : SlaveSPI経由でアクセスする場合の設定



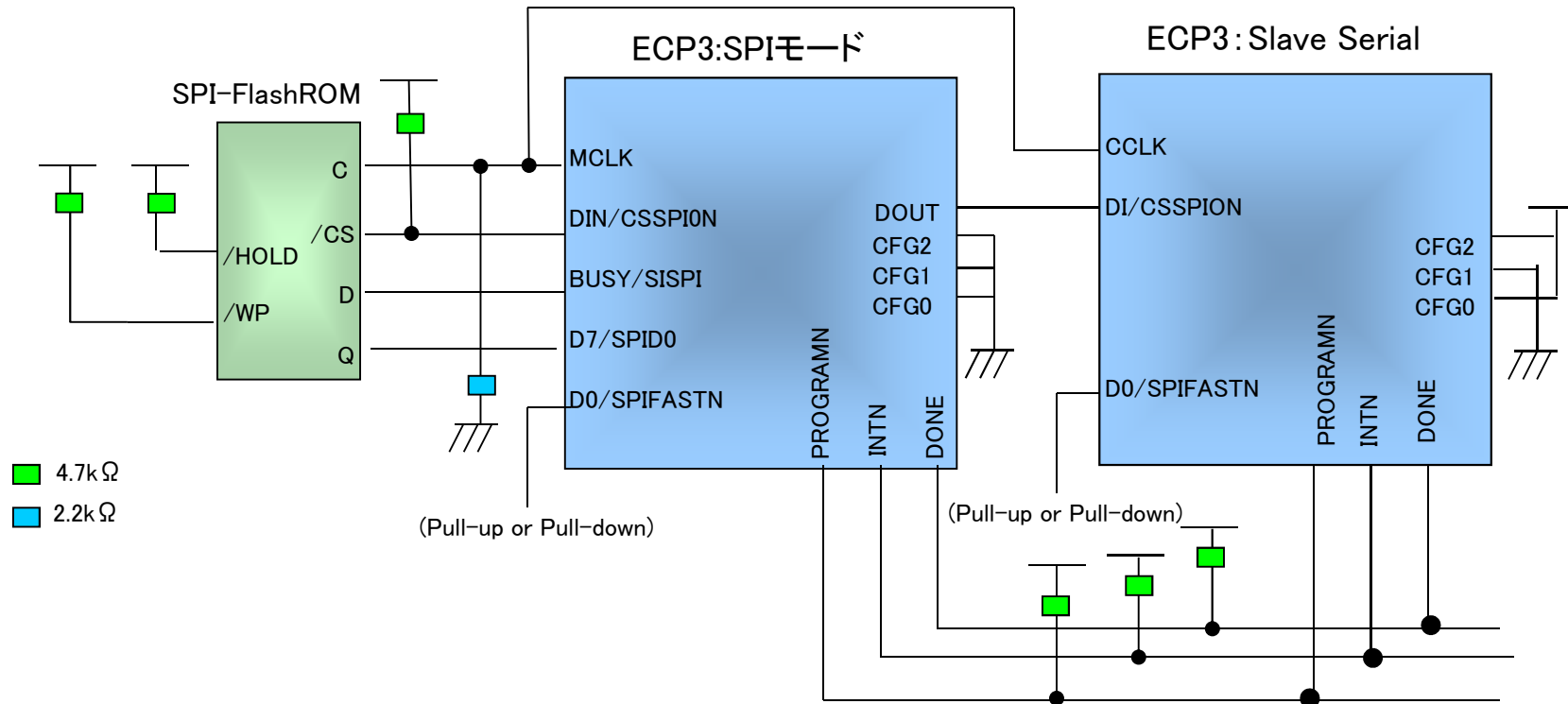
・DOUT信号はコンフィグレーション中出力ピンとなります。もし、このピンをUser I/Oとして使用する場合には出力ピンとして使用することを推奨します。

・D[0]/SPIFASTはお使いになるSPI-FLASHが高速読み出しをサポートしているかどうかを御確認頂き、Pull-down, Pull-up処理を行なってください。

PullUp = 通常SPI Read PullDown = Fast SPI Read

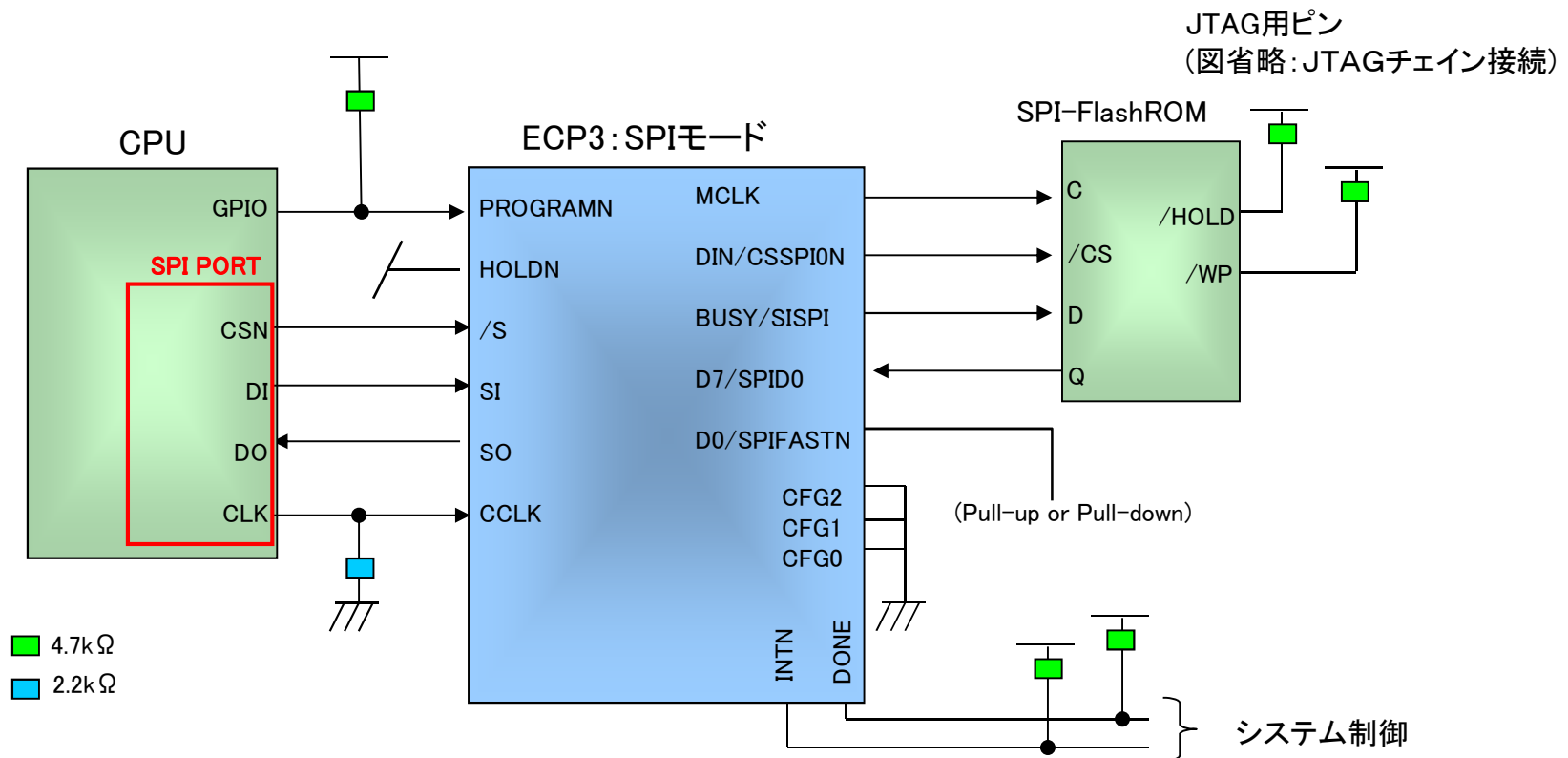
・PROGRAMNはコンフィグ失敗時や、コンフィグレーションシーケンスから抜け出せないようなイレギュラーな現象が起こった場合にトグルすることでFPGAを初期化し、再コンフィグレーション行なうことができます。CPUに接続して管理、制御できるようにしておくことを推奨します。DONE, INITに関してはコンフィグレーションの状態を確認することができます。LEDを接続しておくことでデバック時にとても有効です。

JTAG用ピン
(図省略: JTAGチェーン接続)

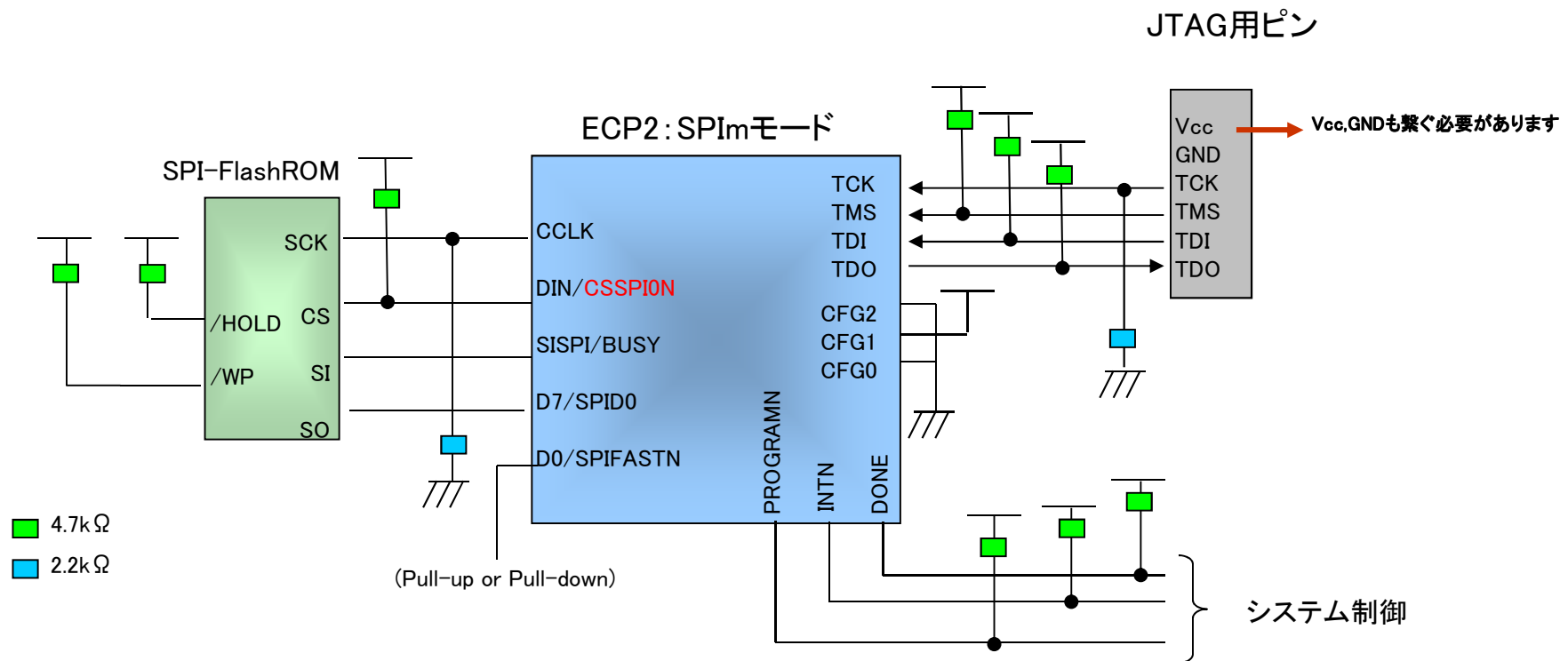


・FPGAを二つ以上繋ぐ場合、CFGピンを 1 番目のデバイスはSPIモードに、2 番目からはスレーブシリアルモードに設定する必要があります。

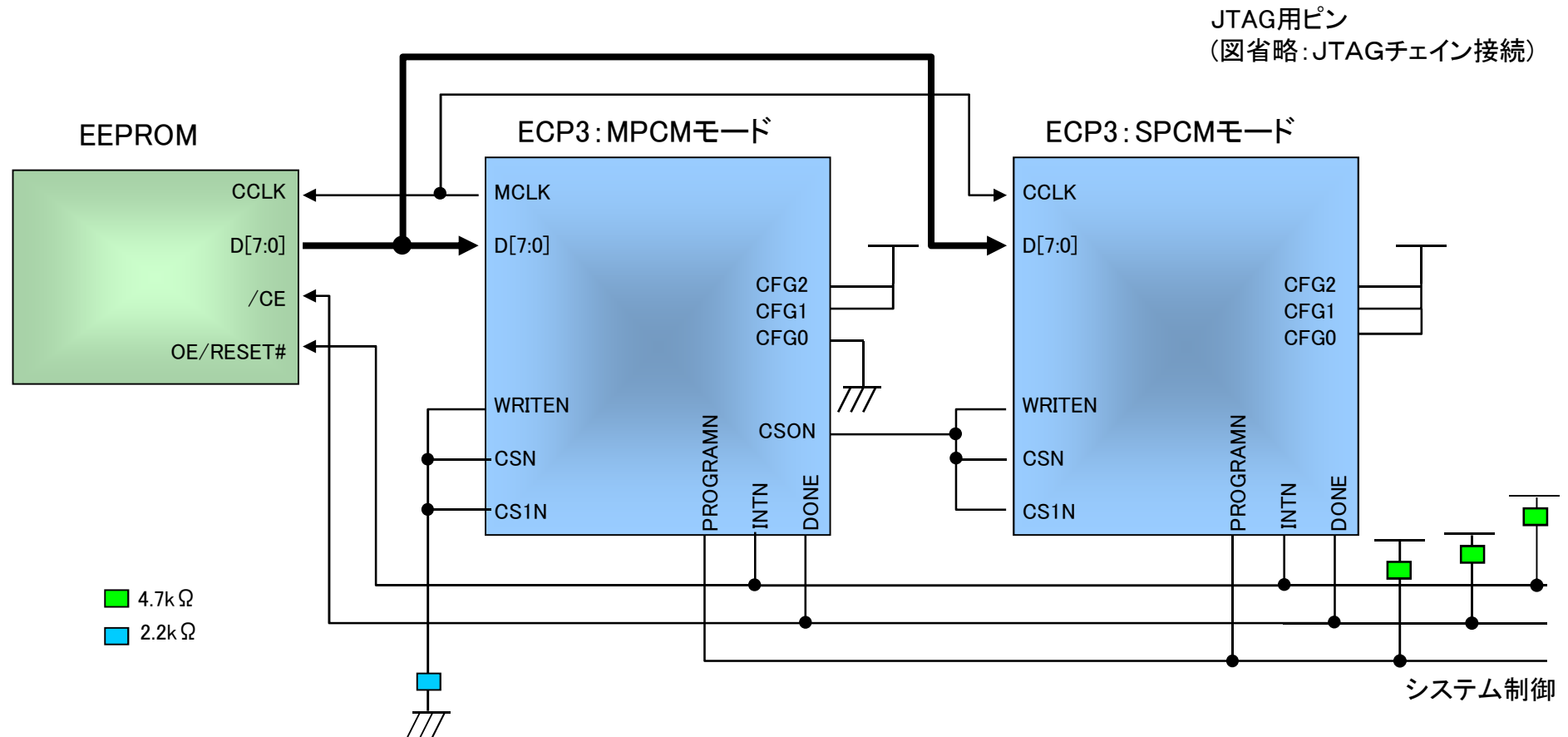
・1 番目 FPGA は、DesignPlanner!においてDONE_EXの扱いをONにしておく必要があります。それに伴って、Wakeupシーケンスを4に変更してください。



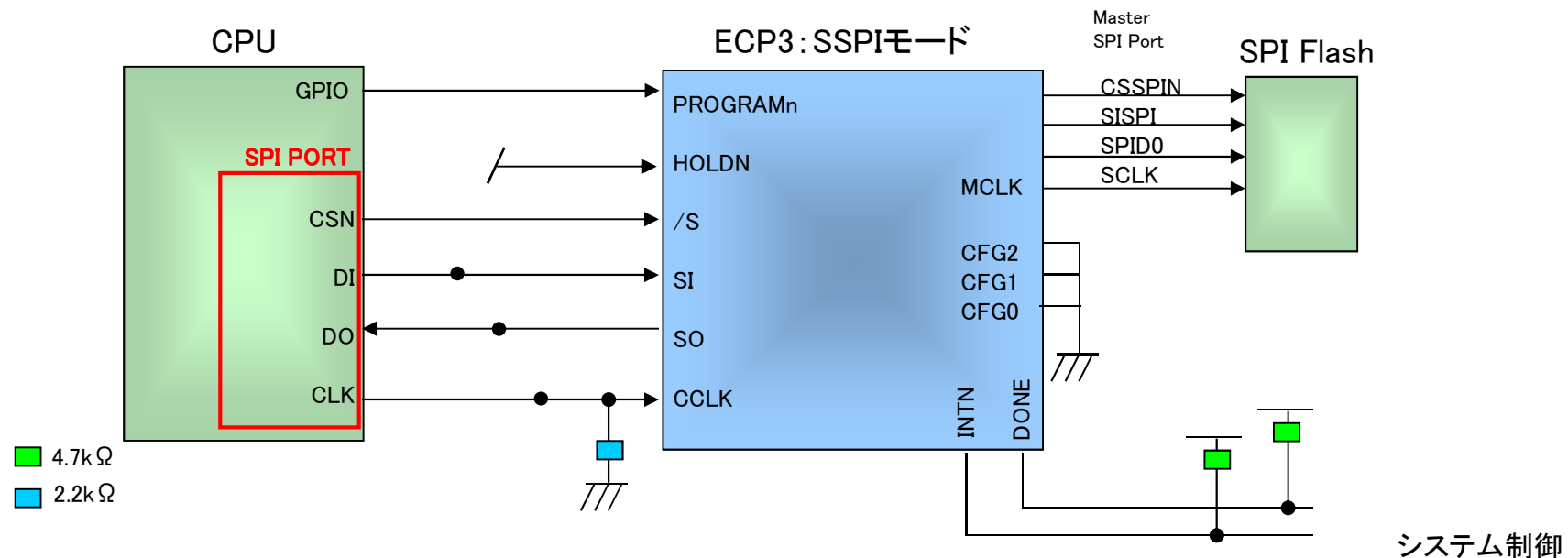
- CPUのSPI PortからFPGAを介してSPI Flashメモリにコンフィグデータを書き込む場合の例です。
CPUから書き込み後、PROGRAMNをトグルすることで再コンフィグ可能です。
- D[0]/SPIFASTはお使いになるSPI-FLASHが高速読み出しをサポートしているかどうかを御確認頂き、Pull-down, Pull-up処理を行なってください。
PullUp = 通常SPI Read PullDown = Fast SPI Read
- PROGRAMNはコンフィグ失敗時や、コンフィグレーションシーケンスから抜け出せないようなイレギュラーな現象が起こった場合にトグルすることでFPGAを初期化し、再コンフィグレーション行なうことができます。CPUに接続して管理、制御できるようにしておくことを推奨します。DONE, INITに関してはコンフィグレーションの状態を確認することができます。LEDを接続しておくことでデバック時にとても有効です。



- ・DOUT信号はコンフィグレーション中出力ピンとなります。もし、このピンをUser I/Oとして使用する場合には出力ピンとして使用することを推奨します。
- ・D[0]/SPIFASTはお使いになるSPI-FLASHが高速読み出しをサポートしているかどうかを御確認頂き、Pull-down, Pull-up処理を行なってください。
- ・PROGRAMNはコンフィグ失敗時や、コンフィグレーションシーケンスから抜け出せないようなイレギュラーな現象が起こった場合にトグルすることでFPGAを初期化し、再コンフィグレーション行なうことができます。CPUに接続して管理、制御できるようにしておくことを推奨します。DONE, INITに関してはコンフィグレーションの状態を確認することができます。LEDを接続しておくことでデバック時にとても有効です。
- ・DualBootイメージの作成方法、概念に関してはデータシート及びテクニカルノート(TN1169)をご参照ください。



- ・外部ロジックが必要無いEEPROMコンフィグレーションです。
- ・Chainする際はFlow-Through制御が可能です。後段のデバイスはスレーブパラレルモードに設定します。



SPIモード

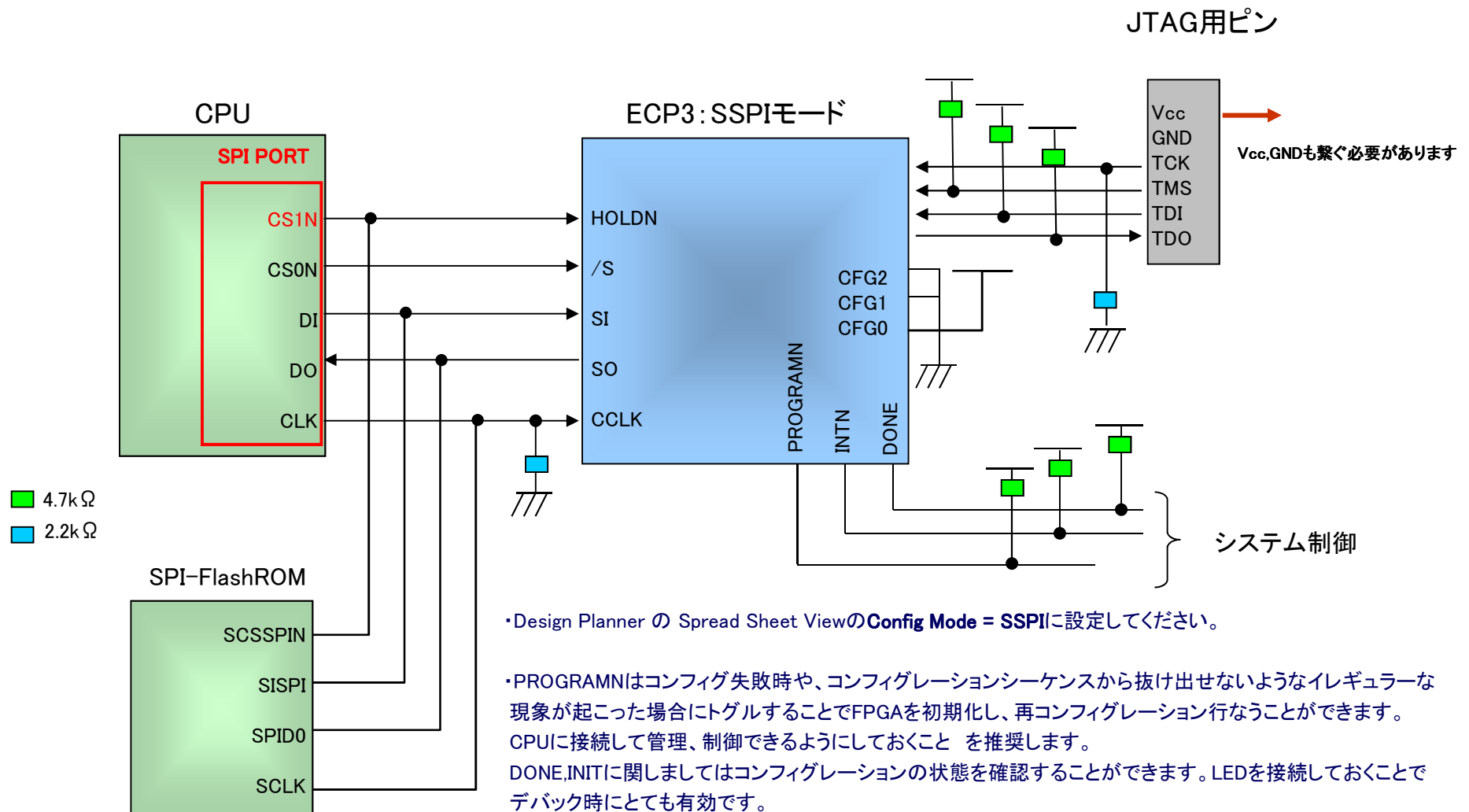
- CPUから見て、FPGAは S-SPI方式
- SPI-ROMから見て、FPGAはMaster-SPI方式。

CLK

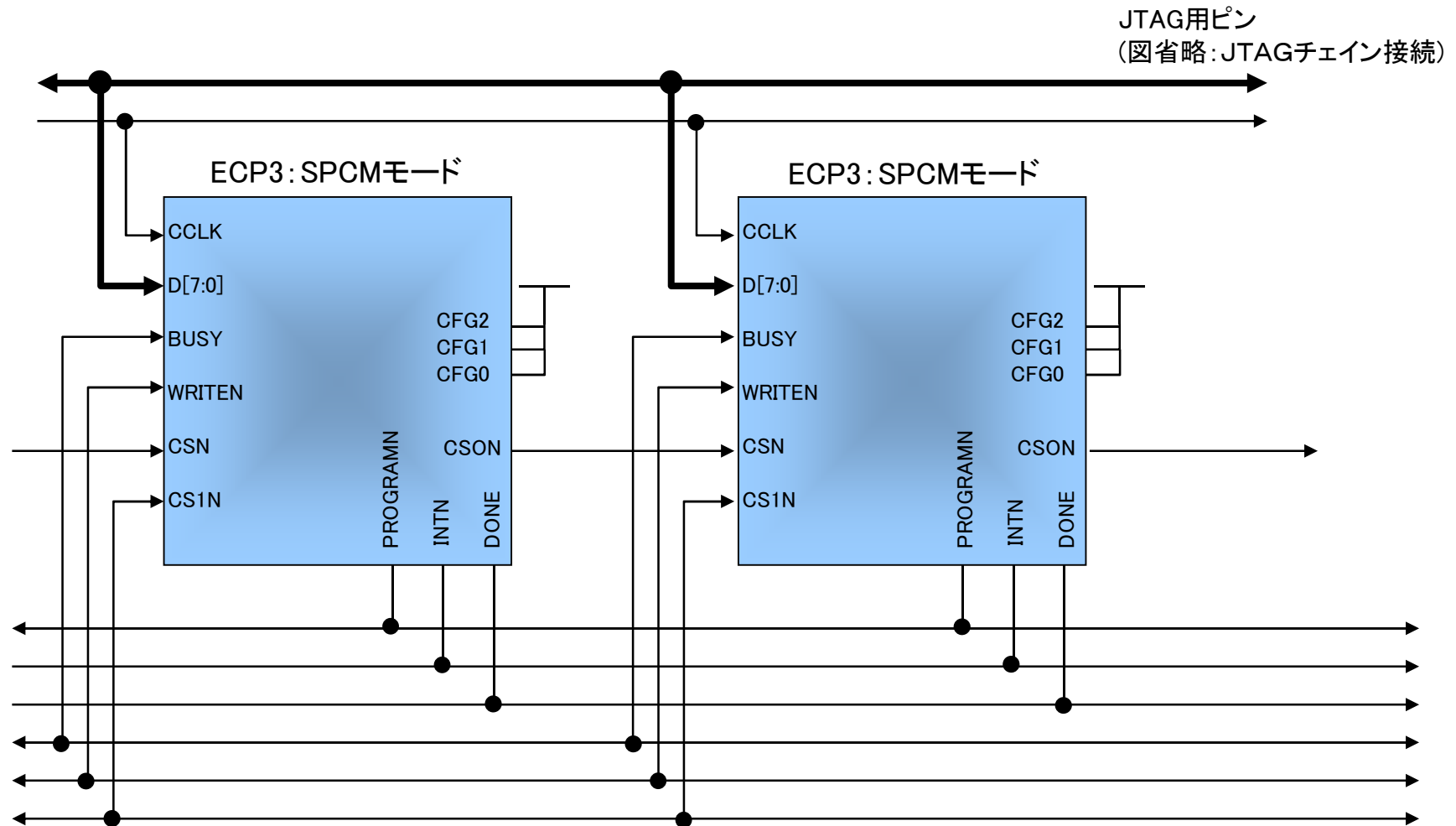
- FPGAの“CCLK”は入力ピン、S-SPI方式用
- FPGAの“SCLK”は出力ピン、M-SPI方式用。

SPI-ROMのアクセス

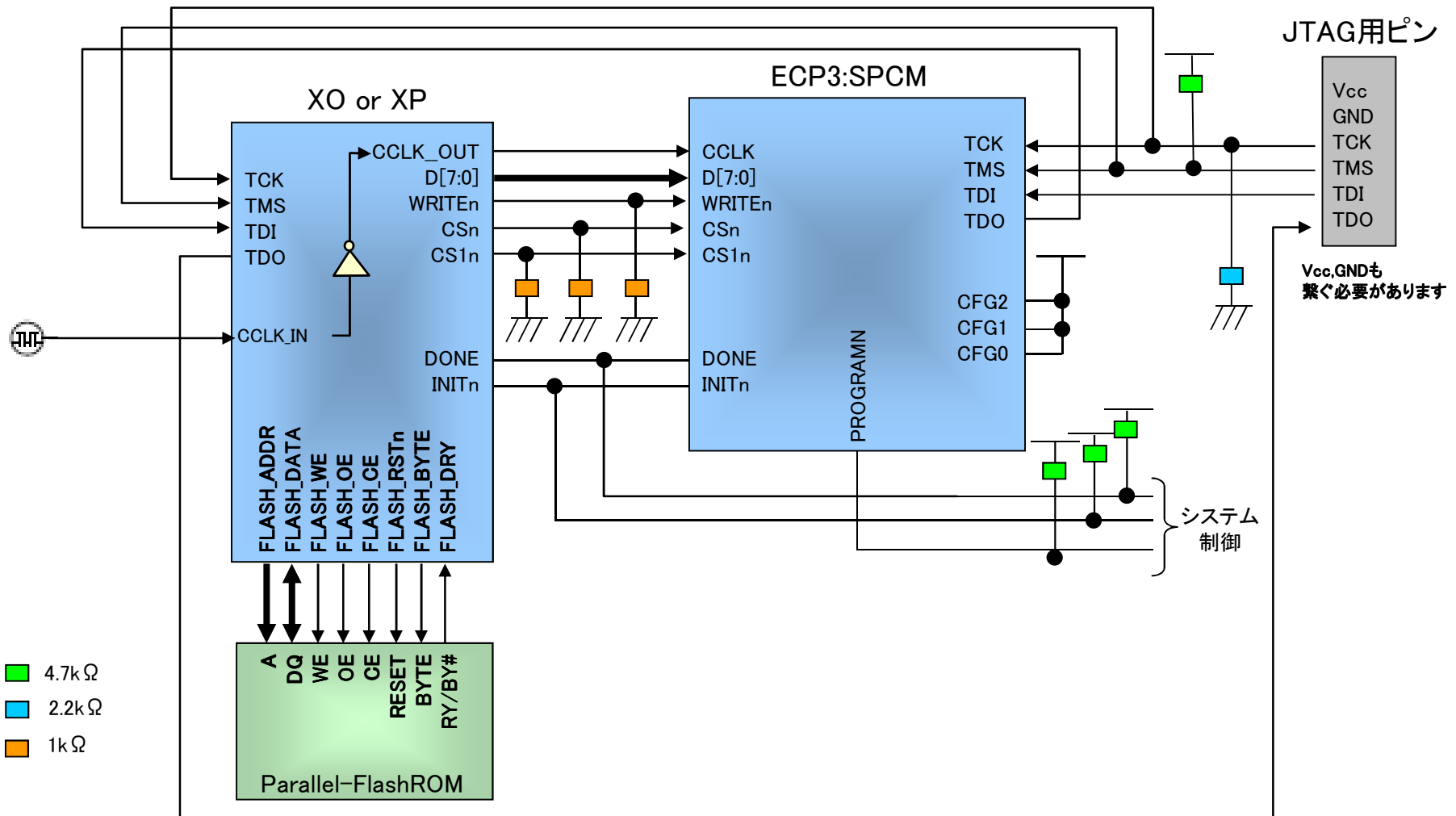
- SPI-ROMへのライト (FPGAデータの更新)は
CPUのソフト(ユーザー作成)でFPGAのS-SPIポートへ制御し、
FPGAを通して、FPGAのM-SPIポートからSPI-ROMへ書き込みする。
- SPI-ROMからのリード (FPGAのConfig)は
Power-On時、FPGAはM-SPIとして、SPI-ROMからデータを読み出し、自己Configする。



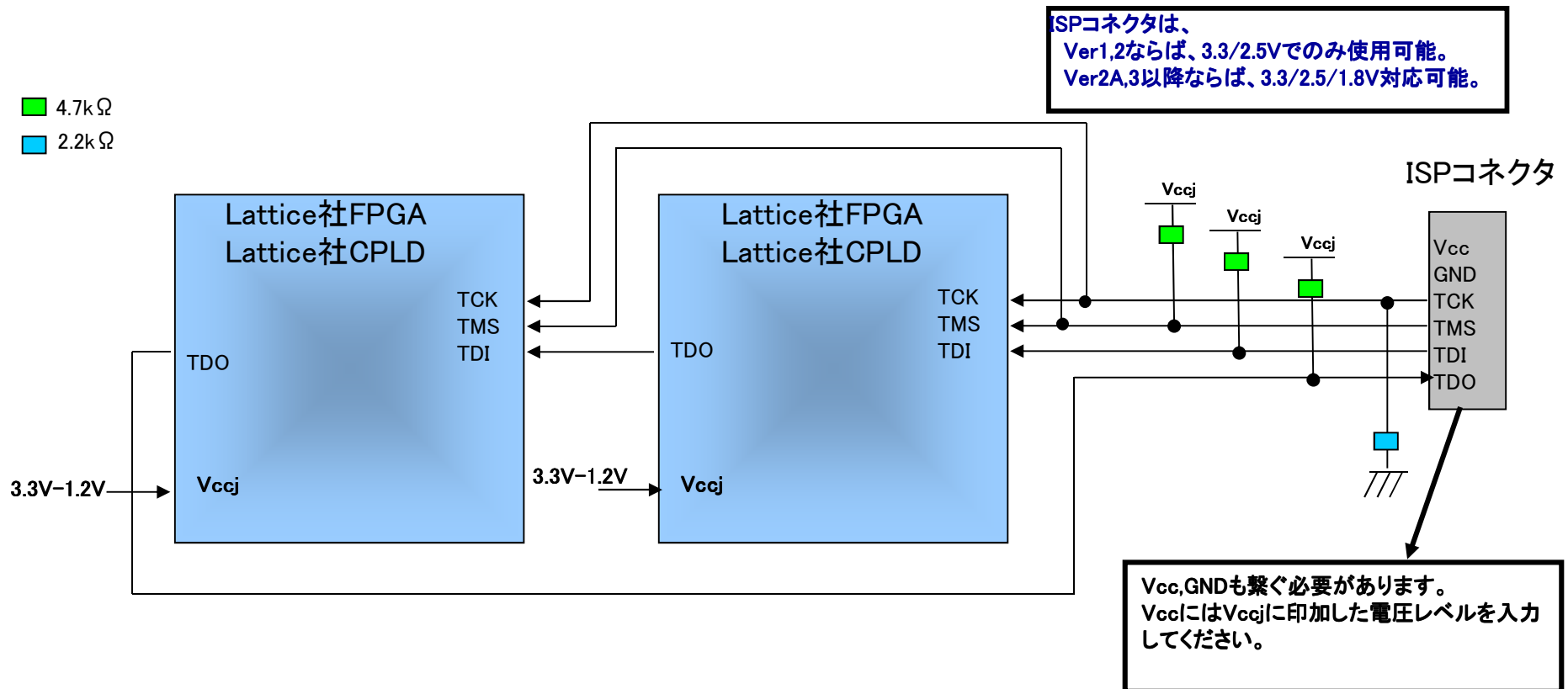
- ・ CS0N、CS1Nとも使用。
- ・ SPIモード：
 - － CPUから見て、FPGAは S-SPI方式
 - － CPUから見て、SPI-ROMは S-SPI方式。
- ・ CLK
 - － FPGAの“CCLK”は入力ピン、S-SPI方式用
 - － SPI-ROMの“SCLK”は入力ピン、S-SPI方式用。
 - － CPUの“CLK”は共通の出力ピンで、
 - ≫ “CS1N”=Low時、SPI-ROMはActive; FPGAはNon-Active
 - ≫ “CS1N”=High且つ“CS0N”=Low時、SPI-ROMはNon-Active; FPGAはActive
- ・ SPI-ROMのアクセス
 - － SPI-ROMへのライト（FPGAデータの更新）は
“CS1N”=Low にし、CPUのソフト（ユーザー作成）で、
CPUが直接FPGAデータをSPI-ROMの指定エリアへ書き込み（連続）
 - － SPI-ROMからのリード（FPGAのConfig）は
 - ≫ A)まず、“CS1N”=Low にし、CPUのソフト（ユーザー作成）で、
CPUがSPI-ROMの指定エリアからFPGAのデータを呼び出し（断続）、
CPUのRAMエリアに保存、
 - ≫ B)つぎ、“CS1N”=High且つ“CS0N”=Low、CPUのソフト（ユーザー作成）で
CPUがRAMのデータをFPGAへ転送する。
 - ≫ C)その繰り返しで、FPGAのデータを転送終了までやる。



- ・ Busyを使う場合は、EncryptionとCompressはサポートできません。



・ Encryption機能はサポートできません。



・Vccjは、JTAGチェーンで統一してください。

・もし、チェーン上に3.3Vデバイスと1.2Vデバイスが混在している場合などは、3.3VデバイスのTDOと1.2VデバイスのTDIのI/F取るためにVccjを3.3Vにする必要があります。

・その他、詳細につきましては、弊社作成のマニュアルJTAG基板設計時資料をご覧ください。

ECP3のコンフィギュレーションの詳細説明

mode	encyption	compress	enc & com
SPI	○	○	?
SPI _m	○	○	?
MPCM	×	○	×
SSPI	○	○	×
SCM	○	○	?
SPCM	○ (1)	○ (1)	○ (1)
1149.1 (2)	○	○	×
1532	×	×	×

Note(1) : BUSY信号を使用しない場合、上記3ケースいずれもサポート可能
ただし、XO、XP経由でのコンフィグレーションにはサポート無し。

Note(2) : Fast Programというコマンドを指します。

コンフィグモード

SPI : SPIマスター

SPI_m : SPIマスターマルチ

MPCM : パラレルrEEPRM

SSPI : スレーブSPI

SCM : スレーブシリアル

SPCM : スレーブパラレル

1149.1 : JTAG

1532 : JTAG

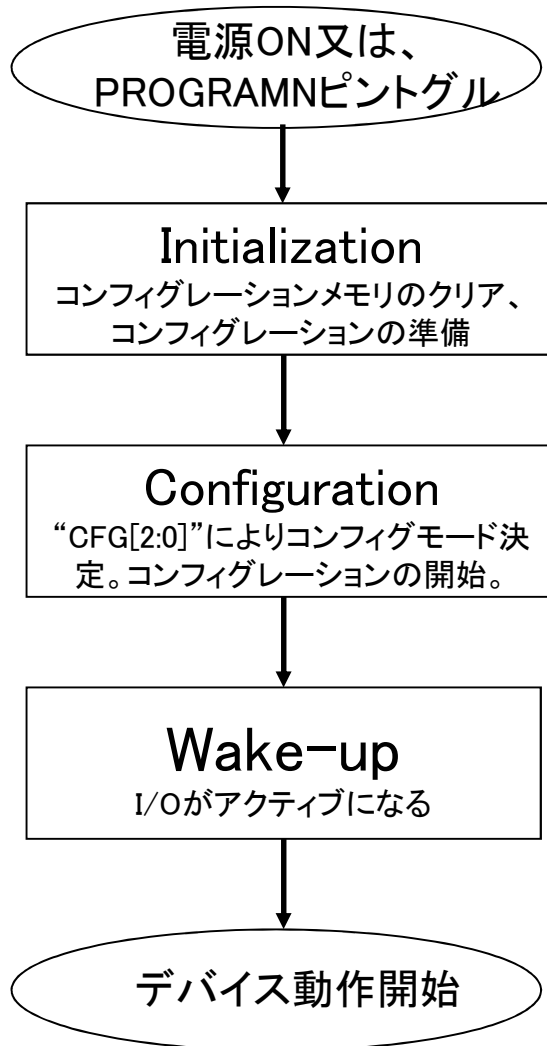
Configuration可否		FPGAデータ	
		Encryption	Not Encryption
FPGA Key	書き込み無し	○	×
	書き込み有り	○	○

○の意味について。

- 暗号化のKeyを書き込まれた ECP3 デバイスは
EncなしのFPGA—DATAを書き込まれた場合、Configできます。
 》 仕組み: Bitstreamデータに、最初はPreambleになります。
 No-EncはBDB3(hex)、EncはBFB3(hex)、
 デバイスのConfigエンジンはこれを識別し、
 Configできるような仕組みになっています。

mode	clock	write to FPGA	read from FPGA
SPI	MCLK	○	×
SPI _m	MCLK	○	×
MPCM	MCLK	○	×
SSPI	CCLK	○	○
SCM	CCLK	○	×
SPCM	CCLK	○	○
1149.1	TCK	○	×
1532	TCK	○	○

ECP3のコンフィグレーションフロー



Initialization

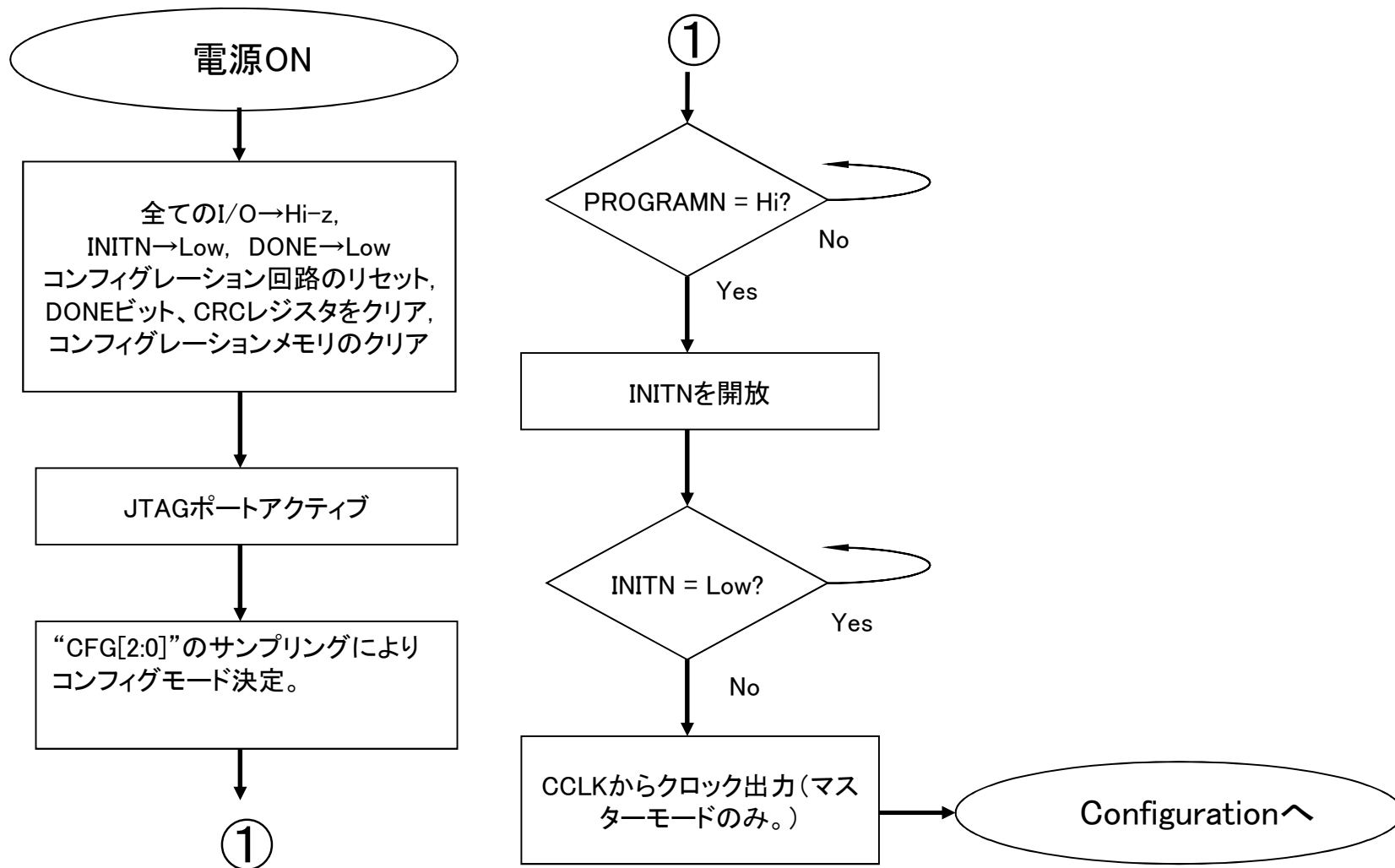
電源投入又はPROGRAMNピンをLowにトグルすることで、FPGA内部コンフィグレーション・メモリ領域がクリアされます。コンフィグレーション・メモリ領域とDoneビットをクリアし終わると、INITNピンはEC/ECPデバイスによって開放されます。外部からINITNピンをLowにドライブすることで、コンフィグレーション・プロセスを遅らせることができます。

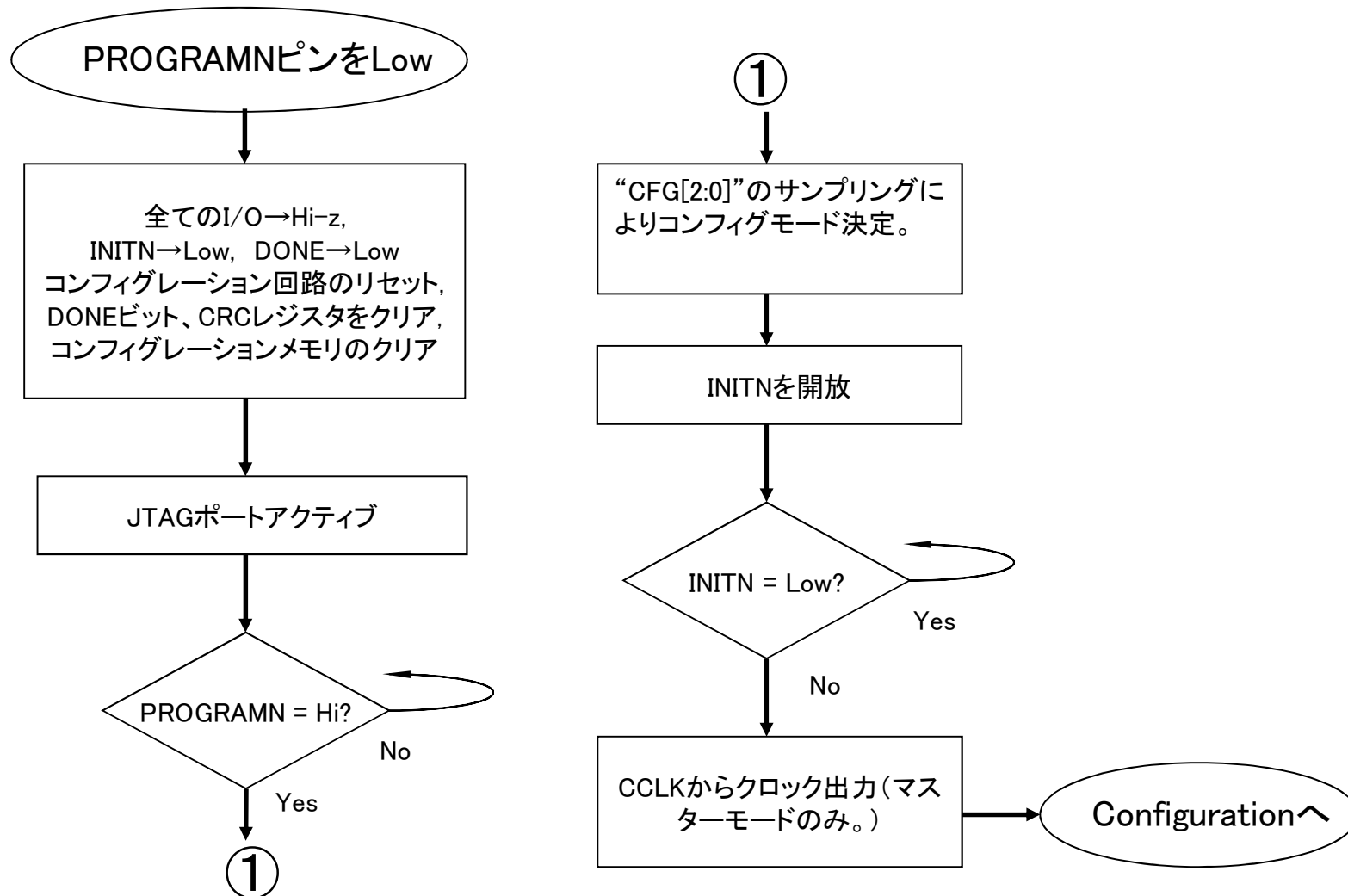
Configuration

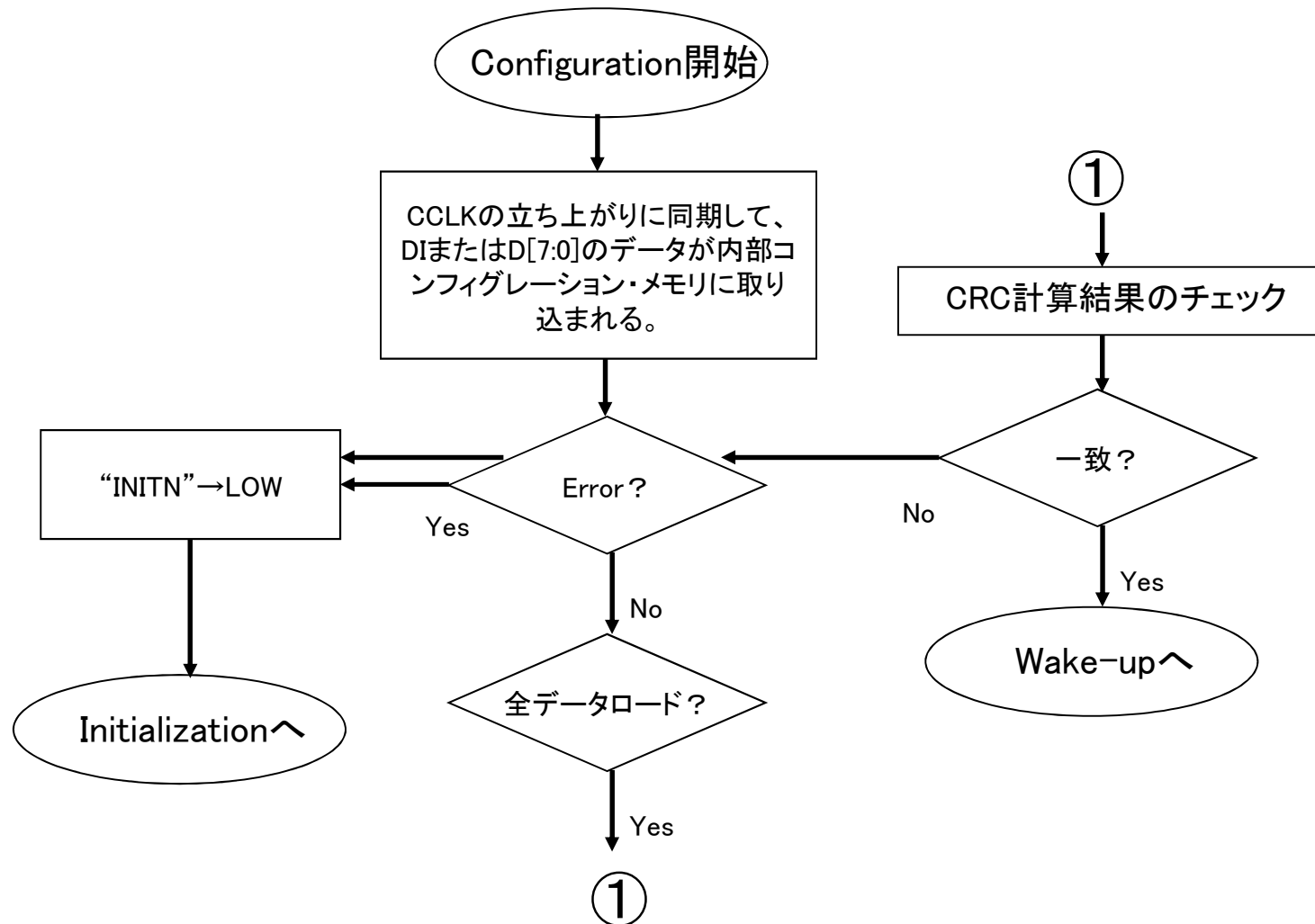
選択されたコンフィグレーション・モードにより、DIかD[0:7]からビット・ストリームをロードします。何らかのエラーがあるとINITNピンはLowになり、エラー発生を知らせます。

Wake-up

ウェイクアップ・シーケンスは、コンフィグレーションの完了後にデバイスを機能モードに投入します。適切なウェイクアップ・シーケンスを選ぶことは、コンテンションを防ぐために重要です。







●ウェイク・アップは25通りのシーケンスから1つを選択

- ビット・ストリーム生成時のWAKE_UPプリファレンスにて指定
- 制御される信号は、
 - ・GOE (グローバル・アウトプット・イネーブル)
 - ・GSR (グローバル・セット・リセット)
 - ・GWD (グローバル・ライト・ディセーブル)
 - ・DONE (DONEピン)

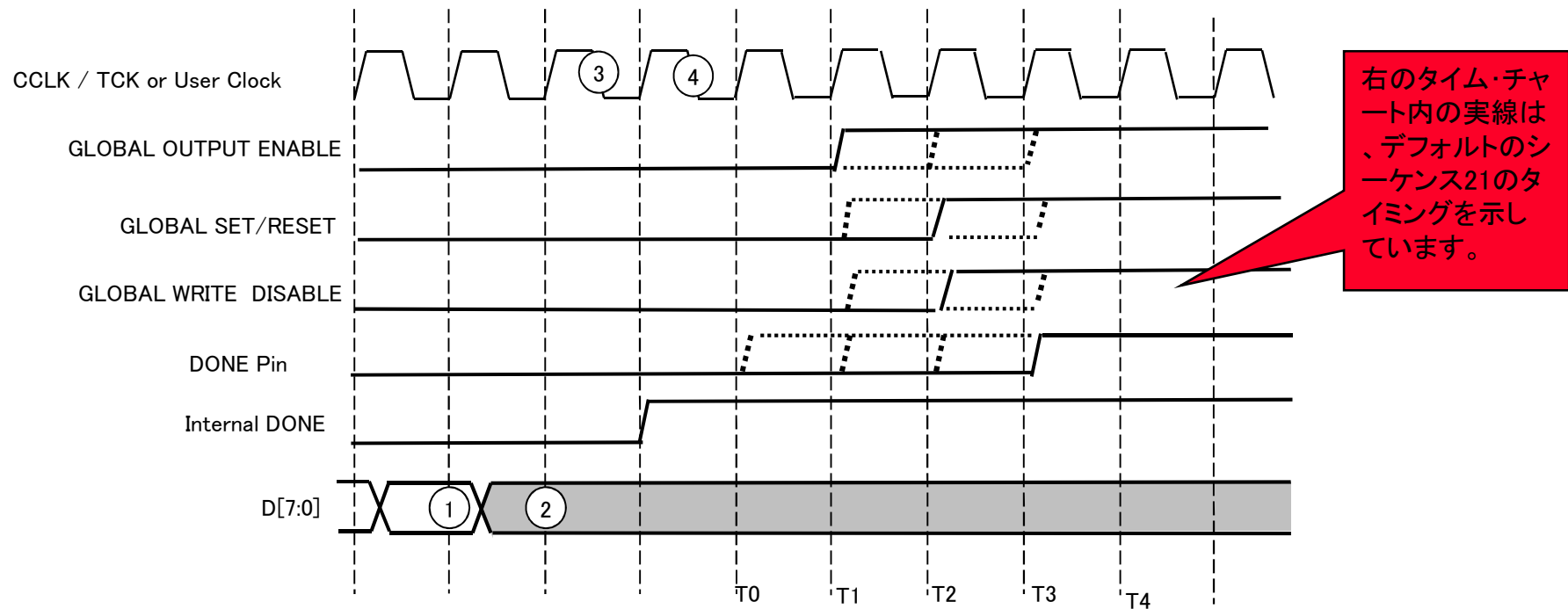
●ウェイク・アップのためのクロックを選択

- ビット・ストリーム生成時のWAKEUP_CLKプリファレンスにて指定
- 指定できるクロックは、
 - ・CCLK (デフォルト)
 - ・ユーザクロック (任意のクロック入力ピン、もしくは任意のネットで指定可能)

●ウェイク・アップ・シーケンス開始をPLLロック後に設定可能

- WAKE_ON_PLLプリファレンスを用いることで、ウェイク・アップ・シーケンス開始をPLLのLock信号が出力するまで待機させることが可能

Sequence	Phase T0	Phase T1	Phase T2	Phase T3
Default		GOE	GSR, GWDIS	DONE
1	DONE	GOE, GWDIS, GSR		
2	DONE		GOE, GWDIS, GSR	
3	DONE			GOE, GWDIS, GSR
4	DONE	GOE	GWDIS, GSR	
5	DONE	GOE		GWDIS, GSR
6	DONE	GOE	GWDIS	GSR
7	DONE	GOE	GSR	GWDIS
8		DONE	GOE, GWDIS, GSR	
9		DONE		GOE, GWDIS, GSR
10		DONE	GWDIS, GSR	GOE
11		DONE	GOE	GWDIS, GSR
12			DONE	GOE, GWDIS, GSR
13		GOE, GWDIS, GSR	DONE	
14		GOE	DONE	GWDIS, GSR
15		GOE, GWDIS	DONE	GSR
16		GWDIS	DONE	GOE, GSR
17		GWDIS, GSR	DONE	GOE
18		GOE, GSR	DONE	GWDIS
19			GOE, GWDIS, GSR	DONE
20		GOE, GWDIS, GSR		DONE
21 (Default)		GOE	GWDIS, GSR	DONE
22		GOE, GWDIS	GSR	DONE
23		GWDIS	GOE, GSR	DONE
24		GWDIS, GSR	GOE	DONE
25		GOE, GSR	GWDIS	DONE



1. PROGRAM_DONEコマンド・フレームの最終バイト
2. PROGRAM_DONEコマンドの実行最終ステート。もしPROGRAM_DONE命令にCRCが付加されていた場合、CRC比較はこの時点で完了される。
3. 内部DONEビット(*)が開放されるには、複数のクロック・サイクルが必要となる場合がある。
4. PLLがロックするまで、複数のクロック・サイクルが必要。もしPLLのロック信号に依存しない(Wake_on_lockを未設定)場合は、1クロック・サイクルのみ。

*: 内部DONEビットはデバイス内部の制御用で、DONEピンと区別されています。またそれらの信号遷移タイミングも異なります。

ECP3のSPI-Flash, Serial-ROMについて

Maximum Configuration Bits - SPI Flash Mode Bitstream File¹

Device	Uncompressed Bitstream Size	SPI Flash	Dual Boot SPI Flash	Units
ECP3-17	4.5	8	16	Mb
ECP3-35	8.2	16	32	Mb
ECP3-70	22.5	32	64	Mb
ECP3-95	22.5	32	64	Mb
ECP3-150	35.7	64	128	Mb

1. Specified sizes include maximum embedded memory block preload.

・“Generate Bitstream Data”の”PropertiesにあるCompress bit stream”をTrueとすることで、生成されるファイル(BITファイル)が圧縮モードで生成されます。圧縮あり時の圧縮率は、設計により変わります。一般的に小規模の設計では、圧縮率が高くなり、大規模で複雑な設計では、圧縮率が低くなります。

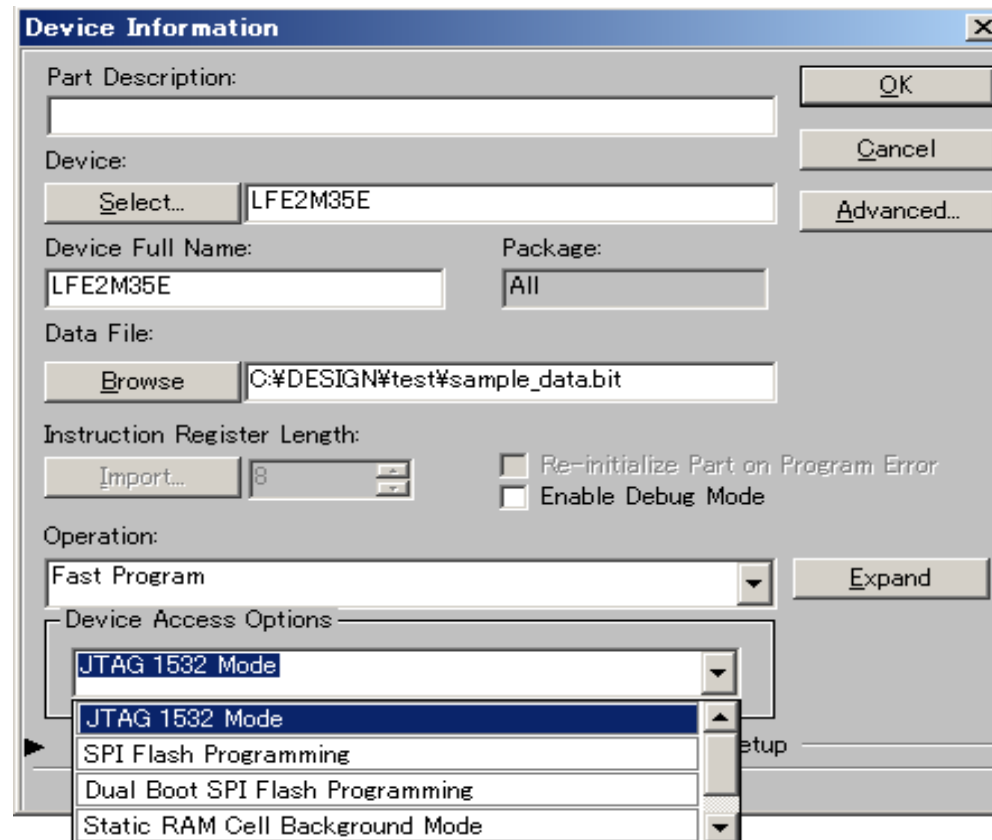
・上記に記載されていないデバイスに関しましては、弊社営業或いは技術までお問い合わせください。

Vendor	Part Number
ST Microelectronics	M25Pxx
Winbond	W25Pxx
Spansion	S25FLxx
NexFlash	NX25Pxx
Atmel	AT25Fxx
Silicon Storage Technology	SST25VFxx SST25LFxx

- ・ 上記はLattice社ECP3とダイレクトに繋げることでできるSPI-Flashのベンダーリストになります。
- ・ 上記Flashに書き込む為にはispVMsystemを使用して書込みを行ないます。SPI-Flashの選定の際にはこちらのソフトが対応しているかどうかにも注意をしてください。(可能であれば、SPI-Flashを選定の際に弊社までご一報をお願いします)
- ・ SPI-FLASHは3.3V品をお使いください。Vccio8には3.3Vを供給してください。

ispVMsystemを使用したECP3への書込みについて

- ・ SPIフラッシュ及びデバイスにコンフィグレーションデータを書き込むにはispVM Systemを使用します。
この項ではSPIフラッシュ及びデバイスへのアクセス方法について説明します。



- ・ Device InformationのウィンドウでDevice Access Optionsの欄でアクセス方法を選択します。

- ・ **JTAG 1532 Mode**
 - デバイスのJTAGポートを経由してデバイスのSRAM領域にアクセスします。アクセス中はデバイスはプログラムモード(全I/Oトライステート)になります。
- ・ **Static RAM Cell Background Mode**
 - デバイスのJTAGポートを経由してバックグラウンドでデバイスからリード、イレース、ベリファイを行います。デバイスの動作は止まりません。
- ・ **SPI Flash Programming**
 - デバイスのJTAGポートを経由してSPIフラッシュにアクセスします。下図のようなフラッシュROMの設定画面が現れますので、お使いになるフラッシュROMを選択後、コンフィグレーションデータを入力し、Load From Fileをクリックしてデータサイズを表示します。その後オペレーションを選択してOKをクリックします。

SPI Serial Flash Device

Flash Device: Select... SST25VF080B Vendor: SSTI

Device Description: SST25VF080B-50-4C-S2F Package: 8-lead SOIC Device Size: 8 MBits

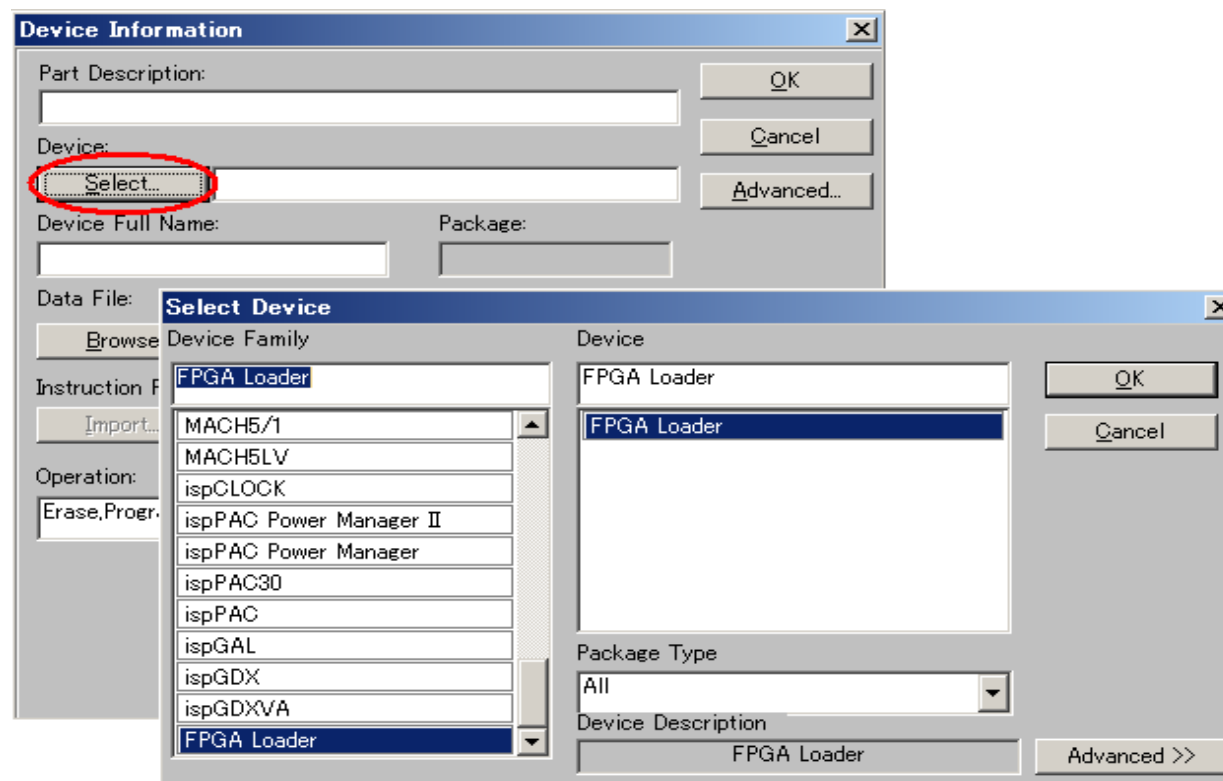
Starting Address: 0x000000 Hex Ending Address: 0x0F0000 Hex Data File Size: 51820 Bytes

Data File: Browse D:\DESIGN\TEST\SAMPLE.BIT Load From File

Operation: SPI Flash Erase,Program,Verify

☒ Erase Part on Programming Error

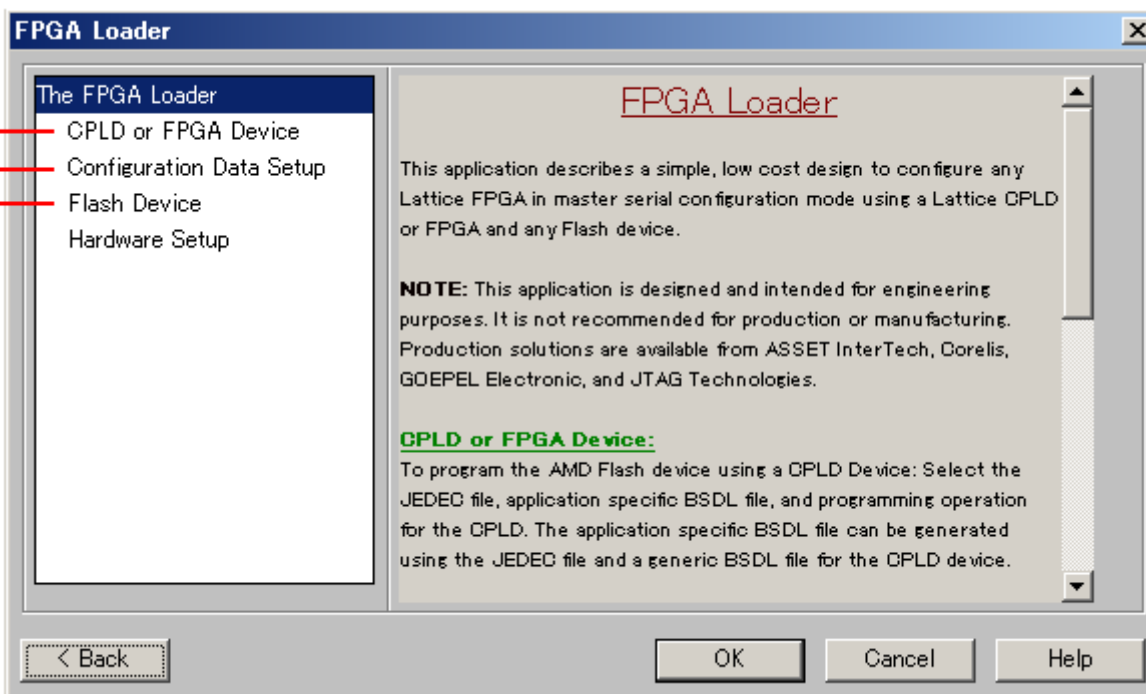
- ・ FPGA Loaderによる書き込み
 - デバイスのJTAGポートを介してSPIフラッシュへのアクセス、もしくはJTAGポートを介さずに直接SPIフラッシュにアクセスを行います。
 - Device InformationのSelect項目からFPGA Loaderを選択します。



デバイスの選択を行います
コンフィグレーションデータの設定を行います
SPIフラッシュROMの選択を行います

デバイスのJTAGポートを介してSPIフラッシュへ書き込みを行う場合はデバイスの選択が必須です。

デバイスのJTAGポートを介さずに直接SPIフラッシュに書きこみを行う場合は、SPIフラッシュROMの選択のみ行ってください。



※Configuration Data Setupのオプションでファイルのマージ(連結)を行う部分がありますが、2007/02/06 時点ではFPGA Loaderによるマージは行えませんので、UFWにて行います。詳細は「3-12 Bitstreamの圧縮、マージについて」の項を参照してください。

・ Bitstreamの圧縮

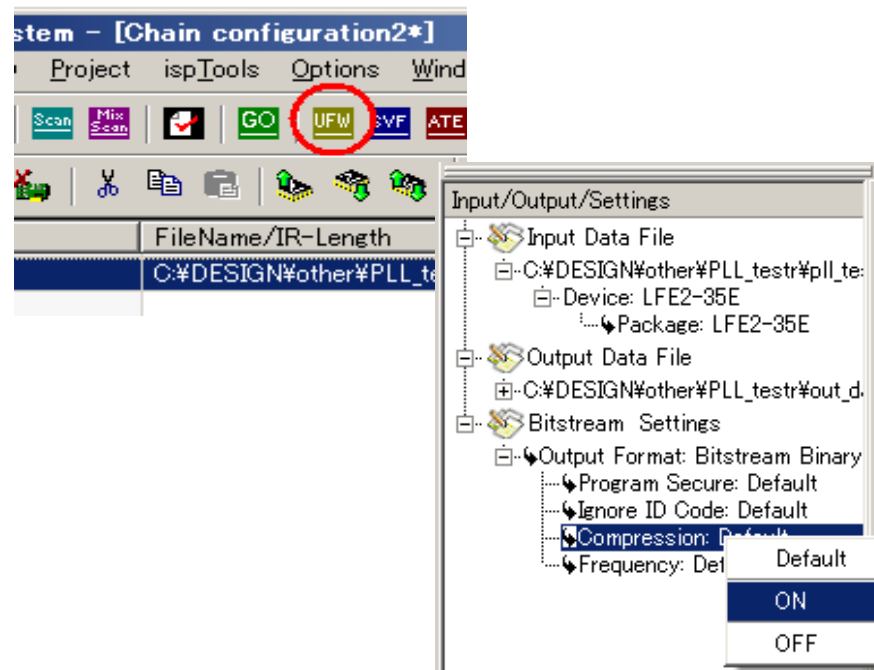
1. ispLEVERで圧縮する

≫ Design PlannerのSpreadsheet ViewでGlobalタブを選択し、COMPRESS_CONFIGをONにしてBitstreamを生成します。

2. UFWで圧縮する

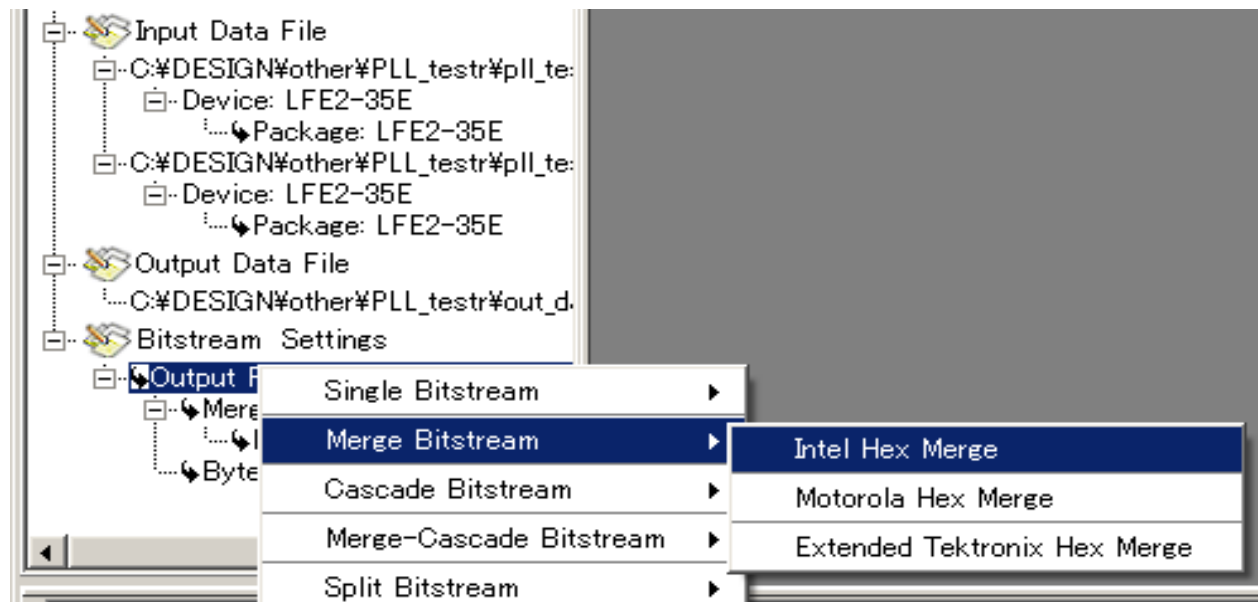
≫ ispVM SystemからUFWを起動します。

Bitstream SettingsでCompressionの部分を右クリックし、ONにしてジェネレートします。

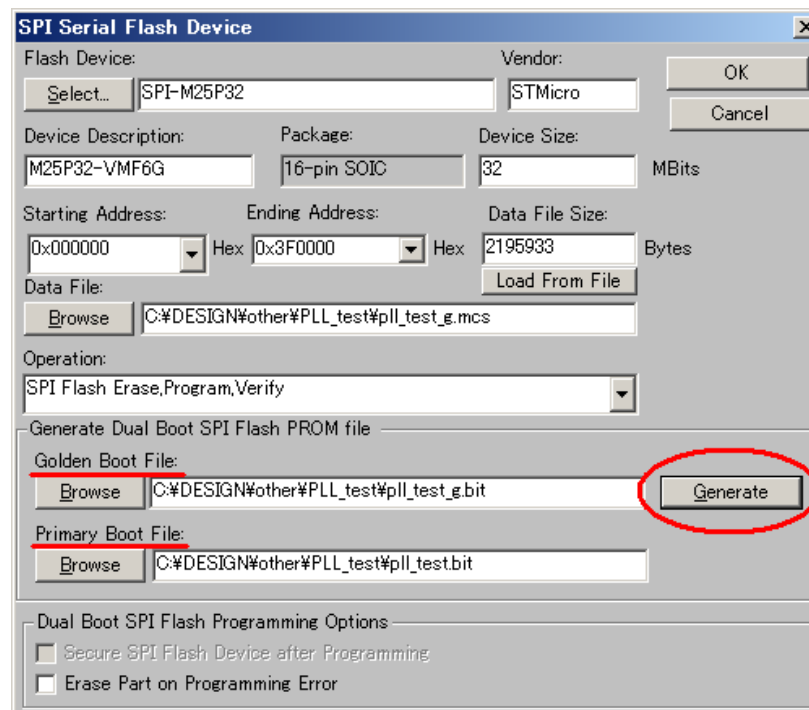


・ Bitstreamのマージ(連結)

- UFWで行います。Bitstream SettingsのOutput Formatの部分を右クリックし、Merge BitstreamからIntel Hex Mergeを選択します。
- Input Data Fileをダブルクリックし、マージするBitstreamを2つ選択し、ジェネレートしてください。
- マージされたファイル(.mcs)が生成されます。



- ・ デュアルブートファイルの生成
 - Device InformationのウィンドウでDual Boot SPI Flash Programmingを選択し、以下の画面を開きます。

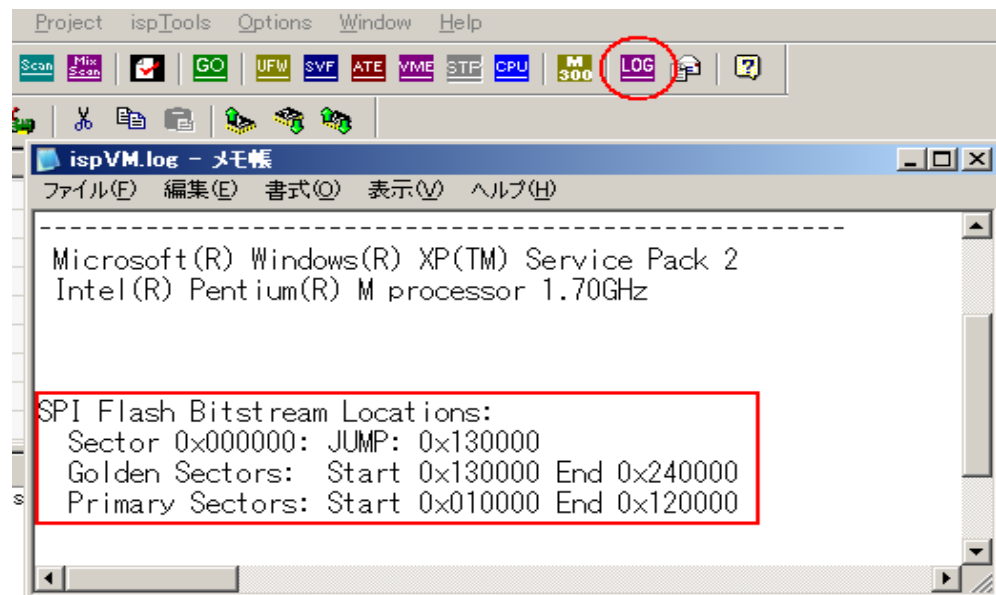


Golden Boot FileとPrimary Boot Fileを選択し、Generateをクリックすると、自動的に2つのbitstreamがマージされたファイルが生成されます。マージされたファイルはData Fileの欄に自動的に入り、データサイズがData File Sizeの欄に表示されます。OKをクリックすると、Device Informationのウィンドウに戻ります。

※デュアルブートモードの注意点

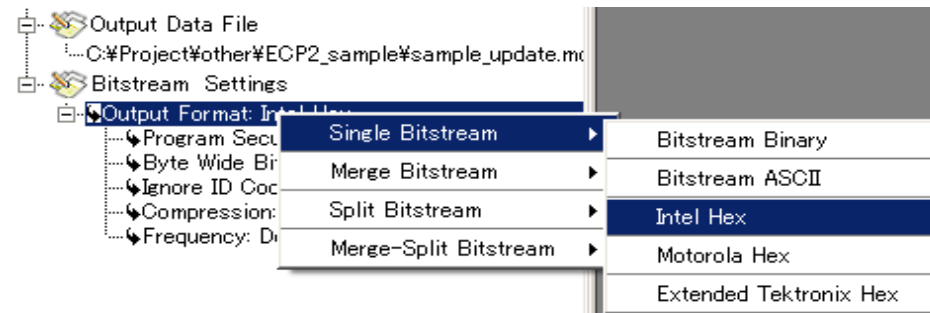
- 前ページで説明した方法はSPIフラッシュROMに初期の書き込みを行う時の方法です。
フィールド上でアップデートする場合は、プライマリデータのみアップデートするようにしてください。
- デュアルブートは1つのSPIフラッシュメモリの中にプライマリセクターとゴールデンセクターの2つの領域を確保します。
それぞれにビットストリームデータを書き込むのでメモリの容量は通常の倍必要になります。
通常サイズのビットストリームを書き込む時に使用するメモリ容量は以下の表及びテクニカルノート TN1108を参照してください。
- デュアルブート用に用意するビットストリームファイルを暗号化する場合は2つのbitstream共に暗号化する必要があります。その際の共通鍵は両方のbitstreamで共通のものを使うようにしてください。
- 電源立上がり時にPROGRAMNピンがトグルされるか、JTAG経由でリフレッシュがかけられた場合に、ECP2MはPrimary Boot Fileを呼び出し、コンフィグレーションが開始されます。
このときにCRCチェックが行われエラーが確認された場合、Golden Boot Fileを呼び出してコンフィグレーションが行われます。Golden Boot Fileにもエラーが確認された場合は、コンフィグレーションは中止されます。(このとき、INITNピンはLowにドライブされます。)

- ・ フィールドアップデートの行い方
 - フィールドでアップデートする際は、プライマリデータのみアップデートします。
 - デュアルブートファイルを生成すると、下図に示すLogファイルにプライマリファイルとゴールデンファイルのスタート/エンドアドレスが表示されます。

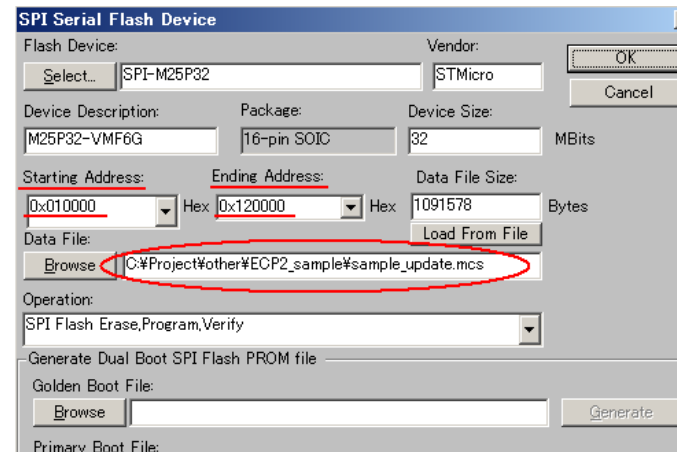


アップデートするプライマリファイルをUFWで圧縮してmcsファイルにします。

※Bitstream Setting ⇒ Single Bitstream ⇒ IntelHexを選択してからCompressionをONにして生成します。



Device InformationのウィンドウでDual Boot SPI Flash Programmingを選択し、Data Fileとしてプライマリデータのmcsファイルを選択し、ログファイルに表示されているプライマリデータのスタートアドレスとエンドアドレスを入力してOKをクリックします。



Revision:	Data:	Author:	Modify:
First Revision 1.0	2009/11/09	Machida, Kajii, Cho	
.rev1	2012/4/12	Takahashi	HP Update Version improved