

CrossLink-NX 基板設計時資料



- ・ 本資料は、Lattice社 “CrossLink-NX” の基板設計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、テクニカルノートを参照頂きます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



目次 1/4



1. 電源ピンに関して

- ・1.1. 電源ピンの説明 ... P7
- ・1.2. 電源に関する注意事項 ... P9
 - ・1.2.1. Power On Reset
 - ・1.2.2. 電源シーケンス / 電源ランプレート

2. sysIOに関して

- 2.1. 各sysIOピンに関して ... P11
 - 2.1.1. 各sysIOピンの概要
 - 2.1.2. PCLK, GPLLピンの補足
 - 2.1.3. 内部Pull up/downの抵抗値
 - 2.1.4. 電源起動時のIOの挙動
- 2.2. sysIO BANKに関して ... P21
 - 2.2.1. BANKの構成
 - 2.2.2. BANK毎のStandard IO
 - 2.2.3. Mixed Voltage Mode
- 2.3. 差動信号に関して ... P25
 - 2.3.1. LVDSの注意事項
 - 2.3.2. LVDS Emulate出力の注意事項
 - 2.3.3. SubLVDSの注意事項
 - 2.3.4. Generic Double Data Rate
 - 2.3.5. Analog IF / MIPI IF / PCIe IF
 - 2.3.6. デバック用 IF (Reveal)

4. Radiant Programmerを使用した書き込みについて

- 4.1. Downloadケーブル ... P58
- 4.2. SRAM領域への書き込み ... P59
- 4.3. FPGA経由でのSPI FLASHへの書き込み ... P62
- 4.4. Feature Rowへの書き込み ... P64
- 4.5. サポートしている特殊Boot Mode ... P68
- 4.6. Dual Boot ... P69
- 4.7. Multi Boot ... P71
- 4.8. Ping-Pong Boot ... P73

1. 電源ピンに関して



1.1. 電源ピンの説明

1.1. 電源ピンの説明

	Min	Typ	Max	
V _{CC} 、V _{CC} ELK	0.95	1.00	1.05	コア電源
V _{CC} UX	1.746	1.80	1.89	Bank0,1,2,6,7に対する補助電源
V _{CC} AUX3/4/5	1.746	1.80	1.89	Bank3,4,5に対する補助電源
V _{CC} AUXA	1.746	1.80	1.89	コアに対する補助電源
V _{CC} IO	3.135	3.30	3.465	V _{CC} IO=3.3V(Bank0,1,2,6,7)
	2.375	2.50	2.625	V _{CC} IO=2.5V(Bank0,1,2,6,7)
	1.71	1.80	1.89	V _{CC} IO=1.8V(All Banks)
	1.425	1.50	1.575	V _{CC} IO=1.5V(All Banks)
	1.2825	1.35	1.4175	V _{CC} IO=1.35V(All Banks,)DDDR3L使用時。
	1.14	1.20	1.260	V _{CC} IO=1.2V(All Banks)
	0.950	1.00	1.05	V _{CC} IO = 1.0V(Bank3,4,5)
D-PHY部				
V _{CC} A_DPHY	1.71	1.80	1.89	D-PHYアナログ電源
V _{CC} _DPHY	0.95	1.00	1.05	D-PHYデジタル電源
V _{CC} PLL_DPHY	0.95	1.00	1.05	D-PHY PLL用電源
ADC部				
V _{CC} _ADC_18	1.71	1.80	1.89	ADC用電源
SERDES部				
V _{CC} _HS	0.95	1	1.05	SERDEブロック供給電源
V _{CC} PLL_HS	1.71	1.80	1.89	SERDESブロックPLL供給電源
V _{CC} AUX_HS	1.71	1.80	1.89	SERDESブロック補助電源

- ・同一電圧電源はすべて同じ電圧源から供給してください。
- ・ノイズ対策の為、適切にアイソレーションしてください。

引用 : FPGA-DS-02049-0.81.01 3.2. Recommended Operating Conditions

1. 電源ピンに関して



1.2. 電源に関する注意事項



1.2.2. 電源シーケンス/電源ランプレート

■ Power On Reset

電源の立ち上がり時に以下の条件をすべて満たした後、Power On Reset(POR)が解除され、コンフィグレーションを開始します。

- $V_{CC} > 0.67V \sim 0.74V$
- $V_{CCIO0} > 0.95V \sim 1.05V$
- $V_{CCIO1} > 0.95V \sim 1.05V$
- $V_{CCAUX} > 1.45V \sim 1.61V$

引用 : FPGA-TN-02099-1. 5.1. Power-up Sequence

■ 電源シーケンス

本デバイスに電源に関するシーケンスの規定は基本的にはございません。

引用 : FPGA-DS-02049-0.81.01 3.4. Power up Sequence

ただし、POR後デバイスが起動すると、IOの出力電圧はそのバンクの V_{CCIO} に依存するため、不定電圧の出力を防止するために、POR前に V_{CCIO} を立ち上げることを推奨しております

引用 : FPGA-DS-02049-0.81.01 2.13. sysI/O Buffer

■ 電源ランプレート

各電源ピンへの電圧供給は以下のランプレートを守らなければなりません。

- $0.1 \sim 50V/ms$

引用 : FPGA-DS-02049-0.81.01 3.3. Power Supply Ramp Rates

2. sysIOに関して



2.1. 各sysIOピンに関して

2.1.1. 各sysIOピンの概要 1/4

ピン名	入出力	説明	外部処理
汎用ピン			
P [T/B/L/R] _ [A/B]	I/O	USER IOピン。 TはTop、BはBottom、LはLeft、RはRightのBankを示し、A/Bは差動のペアを示します。 Bottom BankのA/BペアはTrue LVDSの入力・出力に対応しています。 それ以外のBankではEmulated LVDSはA/Bペアで出力のみサポート可能です。	
Dedicatedピン			
JTAG_EN	I	JTAGピンをコントロールするピン。 Low入力の場合はJTAGピンはGPIOに、High入力の場合はJTAGピンはJTAG専用ピンとなります。	4.7k Ω pull-down to GND(Disable時) 1.0k Ω pull-up to Vccio1(Enable時)
ADC_REF[0,1]	I	ADCコンバータそれぞれのADC基準電圧を入力します。	
ADC_DP/N[0,1]	I	ADCへの入力ピンです。	
SD0_RXDP/N	I	PCI-Express用の入力ピン。	
SD0_TXDP/N	O	PCI-Express用の出力ピン。	
SD0_REFCLKP/N	I	PCI-Express用のリファレンスクロック入力ピン。	
SD0_REXT	I	リファレンス抵抗入力ピンで、このピンとSD0_REFRETピンの間に抵抗を接続します。 外部抵抗値に基づいて、オンチップ差動終端インピーダンスを調整するために使用されます。	
SD0_REFRET	I	リファレンスリターン入力ピン。このピンとSD0_REXTピンの間に抵抗を接続します。 VCCPLL_HSにAC結合します。	
D-PHY[0-1]_DP/N[0-3]	I/O	Hard D-PHYへのデータ入出力ピン。	
D-PHY[0-1]_CK/P/N	I	Hard D-PHYへのクロック入力ピン。	
その他のピン			
NC		未接続ピン。	
RESERVED		予約ピン。RESERVEDには何も接続しないでください。	

全ての場合においてP/N表記はP:Positive、N:Negativeの意です。

2.1.1. 各sysIOピンの概要 2/4

NOWHERE,
but HERE.

ピン名	入出力	説明	外部処理
Configuration・Programmingピン			
PRxxx/SDA/USER_SDA(兼用ピン)	I/O	PRxxx : GPIOとして使用。 SDA : コンフィグレーションをI2C/I3C経由で行う場合のSDAピンとして使用。 User_SDA : I2C/I3C IF使用時のSDAピンとして使用。	SDA : 1.0 kΩ to 4.7 kΩ pull-up to VCCIO1
PRxxx/SCL/USER_SCL(兼用ピン)	I/O	PRxxx : GPIOとして使用。 SDA : コンフィグレーションをI2C/I3C経由で行う場合のSCLとして使用。 User_SDA : I2C/I3C IF使用時のSCLとして使用。	SCL : 1.0 kΩ to 4.7 kΩ pull-up to VCCIO1
PRxxx/TDO/SSO(兼用ピン)	I/O	PRxxx : GPIOとして使用。 TDO : JTAG_EN = 1 の時、JTAGピンとして使用。 SSO : Slave SPIモード時の出力ピンとして使用。	TDO : 4.7kΩ pull-up to VCCIO1
PRxxx/TDI/SSI(兼用ピン)	I/O	PRxxx : GPIOとして使用。 TDI : JTAG_EN = 1 の時、JTAGピンとして使用。 SSI : Slave モード時の入力ピンとして使用。	TDI : 4.kpull-up to VCCIO1
PRxxx/TMS/SCSN(兼用ピン)	I/O	PRxxx : GPIOとして使用。 TMS : JTAG_EN = 1 の時、JTAGピンとして使用。 SCSN : Slave SPIモード時のチップセレクトピンとして使用。	TMS : 4.7kΩ pull-up to VCCIO1 SCSN : 4.7kΩ pull-up to VCCIO1
PRxxx/TCK/SCLK(兼用ピン)	I/O	PRxxx : GPIOとして使用。 TCK : JTAG_EN = 1 の時、JTAGピンとして使用。 SCLK : Slave SPIモード時のクロック入力ピンとして使用。	TCK : 2.2kΩ pull-down to GND
PTxxx/MCSNO(兼用ピン)	I/O	PRxxx : GPIOとして使用。 MCSNO : Master SPIモード時のチップセレクトピンとして使用。 Daisy chain構成時に使用。	MCSNO : 4.7kΩ pull-up to VCCIO1
PTxxx/MD3(兼用ピン)	I/O	PTxxx : GPIOとして使用。 MD3 : Master SPIモードのQuadを使用する際に使用。	
PTxxx/MD2(兼用ピン)	I/O	PTxxx : GPIOとして使用。 MD2 : Master SPIモードのQuadを使用する際に使用。	
PTxxx/MSI/MD1(兼用ピン)	I/O	PTxxx : GPIOとして使用。 MSI : Master SPIモードを使用する際の入力として使用。 MD1 : Master SPIモードのQuadを使用する際に使用。	
PTxxx/MSO/MD0(兼用ピン)	I/O	PTxxx : GPIOとして使用。 MSO : Master SPIモードを使用する際出力として使用。 MD0 : Master SPIモードのQuadを使用する際に使用。	
PTxxx/MCSN/PCLKT0_1(兼用ピン)	I/O	PTxxx : GPIOとして使用。 MCSN : Master SPIモードのチップセレクトピンとして使用。 PCLKT0_1 : クロック入力ピンとして使用。	MCSN : 4.7 kΩ pull-up to VCCIO0
PTxxx/MCLK/PCLKT0_0(兼用ピン)	I/O	PTxxx : GPIOとして使用。 MSI : Master SPIモード時のクロックピンとして使用。 PCLKT0_0 : クロック入力ピンとして使用。	MCLK : 1.0 kΩ to GND

2.1.1. 各sysIOピンの概要 3/4

ピン名	入出力	説明	外部処理
Configurationピン (続き)			
PTxxx/PROGRAMN(兼用ピン)	I/O	PTxxx : GPIOとして使用。 PROGRAMN : Lowにアサートされるとコンフィグレーションが開始されます。	PROGRAMN : 4.7 kΩ pull-up to VCCIO0
PTxxx/INITN(兼用ピン)	I/O	PTxxx : GPIOとして使用。 INITN : 双方向オープンドレインピン。コンフィグレーション可能な状態を示します。	INITN : 4.7 kΩ pull-up to VCCIO0
PTxxx/DONE(兼用ピン)	I/O	PTxxx : GPIOとして使用。 DONE : 双方向オープンドレインピン。コンフィグレーションの完了を示します。	DONE : 4.7 kΩ pull-up to VCCIO0
Shared User GPIOピン			
PBxxx/PCLK[T,C][3,4,5]_[0-3] (兼用ピン)	I/O	PRxxx : GPIOとして使用。 PCLK : Primary Clockバスにつながる入力ピン。 T (True)、C (Complement) は差動のP/Nで②シングルエンド入力を使用する場合にはT側を使用します。その際C側は汎用IOとして使用可能です。(記載なし)	
PTxxx/PCLK0_[0-1] (兼用ピン)	I/O	PTxxx : GPIOとして使用。 PCLK : シングルエンドのみ使用可能なPrimary Clockバスにつながる入力ピン。	
PRxxx/PCLK[1,2]_[0-2] (兼用ピン)	I/O	PRxxx : GPIOとして使用。 PCLK : シングルエンドのみ使用可能なPrimary Clockバスにつながる入力ピン。	
PLxxx/PCLK[6,7]_[0,2] (兼用ピン)	I/O	PLxxx : GPIOとして使用。 PCLK : シングルエンドのみ使用可能なPrimary Clockバスにつながる入力ピン。	
PBxxx/LRC_GPLL[T,C]_IN(兼用ピン)	I/O	PBxxx : GPIOとして使用。 LRC_GPLL : Lower Right BankのPLLへのCLK入力ピン。 T (True)、C (Complement) は差動のP/Nで、シングルエンド入力を使用する場合にはT側を使用します。	
PBxxx/LLC_GPLL[T,C]_IN(兼用ピン)	I/O	PLxxx : GPIOとして使用。 ULC_GPLL : Lower Left BankのPLLへのCLK入力ピン。 T (True)、C (Complement) は差動のP/Nで、シングルエンド入力を使用する場合にはT側を使用します。	
PLxxx/ULC_GPLL__IN(兼用ピン)	I/O	PLxxx : GPIOとして使用。 ULC_GPLL : Upper Left BankのPLLへのシングルエンドCLK入力ピン。	
PRxxx/URC_GPLLT_IN(兼用ピン)	I/O	PRxxx : GPIOとして使用。 URC_GPLLT:Upper Right BankのPLLへのシングルエンド入力ピン。	
PBxxx/VREF[3,4,5]_[1_2] (兼用ピン)	I/O	PBxxx : GPIOとして使用。 VREF : DDRメモリ機能の基準電圧ピン。	
PBxxx/ADC_C[P,N]nn(兼用ピン)	I/O	PRxxx : GPIOとして使用。 ADC_C : ADC入力ピン。	

全ての場合においてP/N表記はP:Positive、N : Negativeの意です。

2.1.1. 各sysIOピンの概要 4/4

ピン名	入出力	説明	外部処理
PBxxx/COMP[1-3][P,N] (兼用ピン)	I/O	PBxxx : GPIOとして使用。 COMP : コンパレータ入力ピン。	
PRxxx/SGMII_RX[P,N][0-1] (兼用ピン)	I/O	PRxxx : GPIOとして使用。 SGMII : 差動SGMII入力ピン。	
PLxxx/PCLK[6,7]_[0,2] (兼用ピン)	I/O	PLxxx : GPIOとして使用。 PCLK : シングルエンドのみ使用可能なPrimary Clockバスへ繋がる入力ピン。	
PBxxx/LRC_GPLL[T,C]_IN (兼用ピン)	I/O	PBxxx : GPIOとして使用。 LRC_GPLL : Lower Right BankのPLLへのCLK入力ピン。 T (True) 、C (Complement) は差動のP/Nで、シングルエンド入力を使用する場合にはT側を使用します。	
PBxxx/LLC_GPLL[T,C]_IN (兼用ピン)	I/O	PLxxx : GPIOとして使用。 ULC_GPLL : Lower Left BankのPLLへのCLK入力ピン。 T (True) 、C (Complement) は差動のP/Nで、シングルエンド入力を使用する場合にはT側を使用します。	
PLxxx/ULC_GPLL_IN(兼用ピン)	I/O	PLxxx : GPIOとして使用。 ULC_GPLL : Upper Left BankのPLLへのシングルエンドCLK入力ピン。	
PRxxx/URC_GPLLT_IN(兼用ピン)		PRxxx : GPIOとして使用。 URC_GPLLT: Upper Right BankのPLLへのシングルエンド入力ピン。	

引用 : FPGA-DS-02049-0.81.01 4. Pinout Information

■ 未使用時の各専用ピンの処理

- Analog (ADC)に関する専用ピンは未使用時はopenの状態の問題ありません
またADCに関しては未使用時はADCに対して電源を接続する必要はありません
GNDに関しては未使用時も接続する必要があります
- SerDes (PCIe)に関する専用ピンは未使用時はopenの状態の問題ありません
電源・GNDに関しては未使用時も接続する必要があります
- DPHY (MIPI)に関する専用ピンは未使用時はopenの状態の問題ありません
またDPHYに関しては未使用時はDPHY BANKに対して電源を接続する必要はありません
GNDに関しては未使用時も接続する必要があります

2.1.2. PCLK,GPLLピン補足



- クロック信号は必ずクロック入力ピン(PCLK)から入力してください。
GPIOからも入力可能ですが、内部専用のクロックライン(Primary Clock)に乗せるまでに遅延が発生します。
(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)
- PLLへの入力はスキューを最小限にする為、PLLへのクロック入力ピン(GPLL)を使用することを推奨します。
Primary ClockネットからPLLへ入力することは可能ですが、専用のPLL入力と比較してフィードバック使用時に遅延が発生する場合があります。
- 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)

引用 : FPGA-TN-02095-1.0 16.3. PLL Inputs and Outputs



2.1.3. 内部Pull up/downの抵抗値 1/2

内部Pull Up/Downの抵抗値はそれぞれに流れるリーク電流とVccioの推奨電圧範囲から算出することが可能です。

I_{PU}	I/O Weak Pull-up Resistor Current	$0 \leq V_{IN} \leq 0.7 * V_{CCIO}$	-30	—	-150	μA
I_{PD}	I/O Weak Pull-down Resistor Current	$V_{IL} (max) \leq V_{IN} \leq V_{CCIO}$	30	—	150	μA

引用 : FPGA-DS-02049-0.81.01 3.8. DC Electrical Characteristics

Vccioの推奨電圧

I/O Driver Supply Voltage	$V_{CCIO} = 3.3 V$, Bank 0, Bank 1, Bank 2, Bank 6, Bank 7	3.135	3.30	3.465	V
	$V_{CCIO} = 2.5 V$, Bank 0, Bank 1, Bank 2, Bank 6, Bank 7	2.375	2.50	2.625	V
	$V_{CCIO} = 1.8 V$, All Banks	1.71	1.80	1.89	V
	$V_{CCIO} = 1.5 V$, All Banks	1.425	1.50	1.575	V
	$V_{CCIO} = 1.35 V$, All Banks (For DDR3L Only)	1.2825	1.35	1.4175	V
	$V_{CCIO} = 1.2 V$, All Banks	1.14	1.20	1.26	V
	$V_{CCIO} = 1.0 V$, Bank 3, Bank 4, Bank 5	0.95	1.00	1.05	V

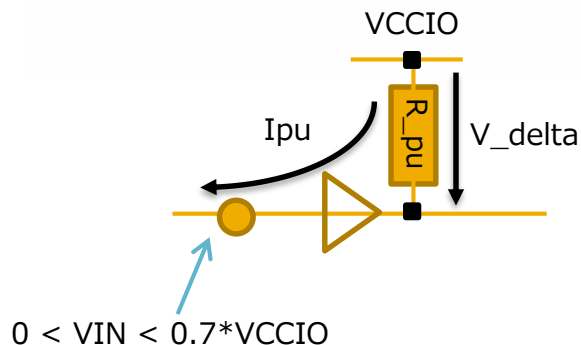
入力電圧レンジ

Input/Output Standard	V_{IL}^1	
	Min (V)	Max (V)
LVTTTL33 LVCMOS33		0.8
LVCMOS25		0.7
LVCMOS18		$0.35 * V_{CCIO}$

引用 : FPGA-DS-02049-0.81.01 3.2. Recommended Operating Conditions,
FPGA-DS-02049-0.81.01 3.11. sysI/O Single-Ended DC Electrical Characteristics

2.1.3. 内部Pull up/downの抵抗値 2/2

実際の計算例) LVCMOS33の場合



内部Pull-Up抵抗 R_{pu} は流れる電流 I_{pu} と電圧降下 V_{Δ} を考慮して計算します。

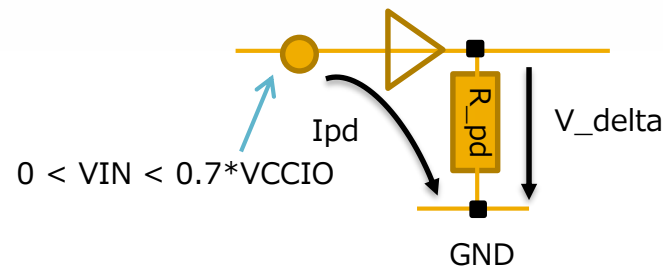
I_{pu} が最大になるのは V_{Δ} が最大の時、即ち $V_{IN}=0V$ のとき

	V_{IN}	V_{Δ}	$I_{pu}(uA)$	$R_{pu}(K\Omega)$
LVCMOS33_max	0	3.465	150	23.1
LVCMOS33_min	0	3.135	150	20.9

I_{pu} が最少になるのは $V_{IN}=0.7 \times V_{CCIO}$

	V_{IN}	V_{Δ}	$I_{pu}(uA)$	$R_{pu}(K\Omega)$
LVCMOS33_max	2.425	1.039	30	34.65
LVCMOS33_min	2.194	0.940	30	31.35

従って、LVCMOS33の場合は内部Pull-Upが有効になっている場合で、外部処理で論理レベルを反対のレベルにする場合、 $R_{pu} = 20.9K\Omega$ と $V_{IL}(MAX) = 0.8V$ を考慮すると外部Pull Down $= 7.16K\Omega$ 以下とします。



内部Pull-down抵抗 R_{pd} は流れる電流 I_{pd} と電圧降下 V_{Δ} を考慮して計算します。

I_{pd} が最大になるのは V_{Δ} が最大の時、 $V_{IN}=V_{CCIO}$ のとき

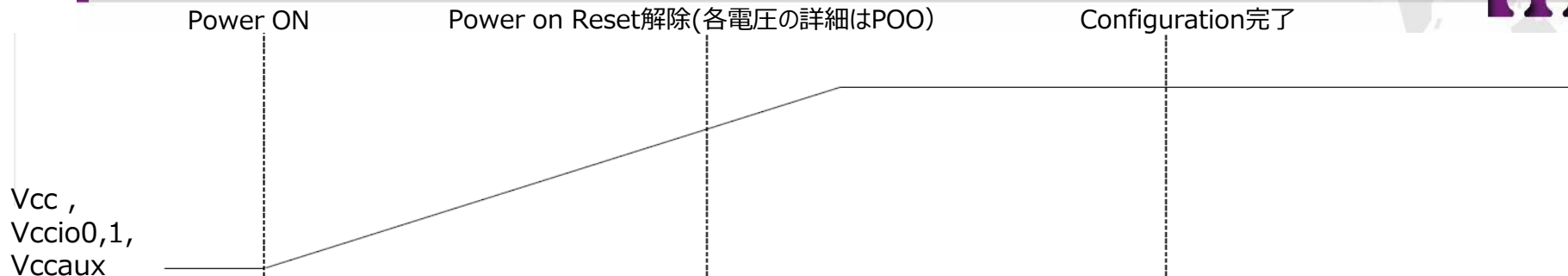
	V_{IN}	V_{Δ}	$I_{pd}(uA)$	$R_{pd}(K\Omega)$
LVCMOS33_max	3.465	3.465	150	23.1
LVCMOS33_min	3.135	3.135	150	20.9

I_{pd} が最少になるのは $V_{IN}=V_{IL}(Max)$ の際

	V_{IN}	V_{Δ}	$I_{pd}(uA)$	$R_{pd}(K\Omega)$
LVCMOS33_max	0.8	2.665	30	26.666
LVCMOS33_min	0.8	2.335	30	26.666

従って、LVCMOS33の場合は内部Pull-Downが有効になっている場合で、外部処理で論理レベルを反対のレベルにする場合、 $R_{pd} = 20.9K\Omega$ と、 $V_{IH}(MIN) = 2.0V$ を考慮すると外部Pull Up $= 11.86K\Omega$ 以下とします。

2.1.4. 電源起動時のIOの挙動に関して 1/2

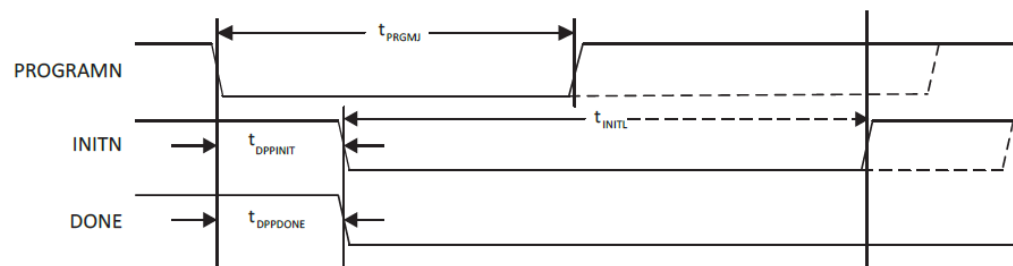


Pin Name	Power up (Default State)	Initialization & Configuration	Wake up & User Mode
汎用IO	Hi-z and internal pull down	Hi-z and internal pull down	開発ツールで設定している状態
JTAG_ENABLE	Hi-z and internal pull down	Hi-z and internal pull down	Hi-z and internal pull down
PROGRAMN	Hi-z and internal pull up	次のページ参照	次のページ参照
INITN	Hi-z and internal pull up	次のページ参照	次のページ参照
DONE	Hi-z and internal pull up	次のページ参照	次のページ参照
MCLK	Hi-z and internal pull up/down(注)	Hi-z and internal pull up/down	Hi-z and internal pull up/down
MCSN	Hi-z and internal pull up	Hi-z and internal pull up	Hi-z and internal pull up
MOSI/MD0	Hi-z and internal pull up	Hi-z and internal pull up	Hi-z and internal pull up
MISO/MD1	Hi-z and internal pull up	Hi-z and internal pull up	Hi-z and internal pull up
MD2	Hi-z and internal pull up	Hi-z and internal pull up	Hi-z and internal pull up
MD3	Hi-z and internal pull up	Hi-z and internal pull up	Hi-z and internal pull up
CSNO/SD0	Hi-z and internal pull up	Hi-z and internal pull up	Hi-z and internal pull up
TCK/SCLK	Hi-z and internal pull down	Hi-z and internal pull down	Hi-z and internal pull down
TMS/SCSN	Hi-z and internal pull up	Hi-z and internal pull up	Hi-z and internal pull up
TDI/SI/SD0	Hi-z and internal pull up	Hi-z and internal pull up	Hi-z and internal pull up
TDO/SO/SD1	Hi-z and internal pull up	Hi-z and internal pull up	Hi-z and internal pull up
SD2/SCL	Hi-z and internal pull up	Hi-z and internal pull up	Hi-z and internal pull up
SD3/SDA	Hi-z and internal pull up	Hi-z and internal pull up	Hi-z and internal pull up

注) Pull up/down設定はControl Register 1内の22bit目、CPOLによりUserで設定が可能です

2.1.4. 電源起動時のIOの挙動に関して 2/2

■ PROGRAMN, INITN, DONEピンの挙動



※各信号のリリースタイミングについては、
3.26. CrossLink-NX sysCONFIG Port Timing Specifications
が更新され次第アップデートします。

PROGRAMN	入力ピン。PROGRAMNに印加される信号がHighからLowに遷移すると、デバイスがコンフィグレーションモードになります。Lowの最小期間は $t_{PROGRAMN}$ として規定がございます。 デバイスのコンフィグレーション中には、PROGRAMNピンをトグルしないでください。 ツール上でPROGRAMN_PORTをDisable設定した場合は、GPIOとなります。
INITN	双方向オープンドレインピン。コンフィグレーション中にLowが出力されると、SRAMの初期化中であることを表し、Highになるとデバイスがコンフィグレーション・データの受け入れ準備ができたことを表します。また、Lowにアサートされた後、 t_{INITN} 期間の間に外部からLowに保持すると、デバイスはコンフィグレーション・モードにはなりません。 コンフィグレーションの開始後は、INITNはビットストリーム・エラーの表示に使用されます。コンフィグレーション中に何らかのエラーが検出されるとINITNピンはLowになり、デバイスがウェイクアップしません。
DONE	双方向オープンドレインピン。デバイスがコンフィグレーションを終了し、ユーザーモードに入ったことを示します。デバイスがコンフィグレーションを開始すると、DONEピンはコンフィグレーション中にLowになり、コンフィグレーションが正常に完了するとHighに遷移します。 外部よりLowに保持することにより、デバイスがユーザーモードに入るのを防ぎます。

引用：FPGA-TN-02099-1.0 4.4. sysCONFIG Pins

FPGA-TN-02099-1.0 3.26. CrossLink-NX sysCONFIG Port Timing Specifications

2. sysIOに関して

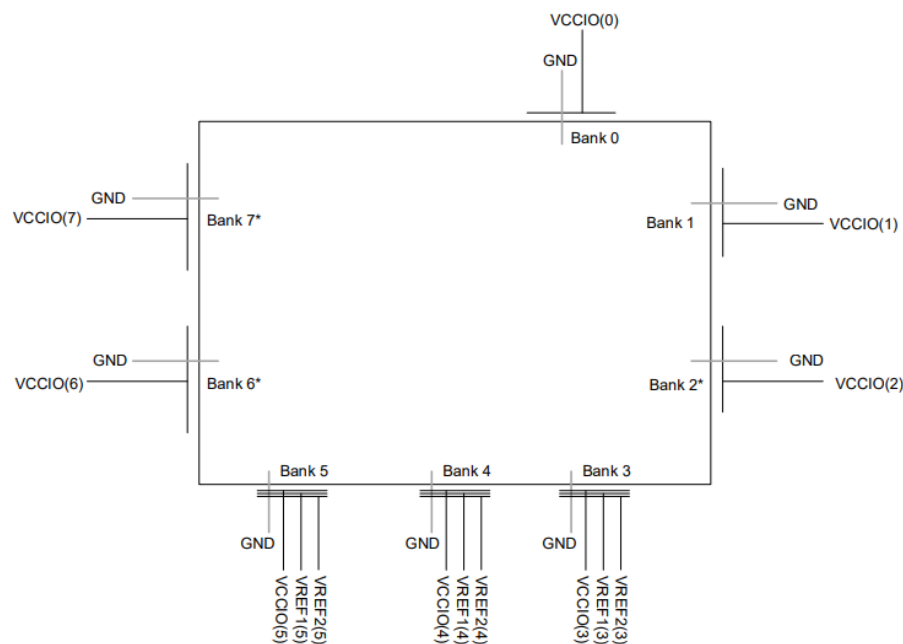


2.2. sysIO BANKに関して

2.2.1. BANKの構成

■ BANK構成の概要

- ・CrossLink-NXはBANK0～7のsysIO BANKで構成されています（17K LUTsの型番ではBANK2, 6, 7は実装なし）
- ・BANK0, 1, 2, 6, 7では1.0V～3.3VのSingle-Ended IOをサポートされています
- ・BANK3, 4, 5ではStaticとDynamicのterminationがあり、Differential IOやDDR memory IFをサポートされています
- ・BANK3, 4, 5ではVREFを保有しており、同じBANKで2つの異なるレベルの入力が可能です
- ・BANK0, 1, 2, 6, 7はHot Socketに対応し、BANK3, 4, 5では非対応です



*Note: Bank not available in LIFCL-17.

Figure 2.27. sysIO Banking

2.2.2. BANK毎のStandard IO



■ 各Standard IOと対応BANKまとめ

Single-Ended I/O Standards			
	INPUT	OUTPUT	Bi-Directional
LVTTL33	BANK 0, 1, 2, 6, 7	BANK 0, 1, 2, 6, 7	BANK 0, 1, 2, 6, 7
LVC MOS33	BANK 0, 1, 2, 6, 7	BANK 0, 1, 2, 6, 7	BANK 0, 1, 2, 6, 7
LVC MOS25	BANK 0, 1, 2, 6, 7	BANK 0, 1, 2, 6, 7	BANK 0, 1, 2, 6, 7
LVC MOS18	BANK 0, 1, 2, 6, 7	BANK 0, 1, 2, 6, 7	BANK 0, 1, 2, 6, 7
LVC MOS15	BANK 0, 1, 2, 6, 7	BANK 0, 1, 2, 6, 7	BANK 0, 1, 2, 6, 7
LVC MOS12	BANK 0, 1, 2, 6, 7	BANK 0, 1, 2, 6, 7	BANK 0, 1, 2, 6, 7
LVC MOS10	BANK 0, 1, 2, 6, 7	—	—
LVC MOS18H	BANK 3, 4, 5	BANK 3, 4, 5	BANK 3, 4, 5
LVC MOS15H	BANK 3, 4, 5	BANK 3, 4, 5	BANK 3, 4, 5
LVC MOS12H	BANK 3, 4, 5	BANK 3, 4, 5	BANK 3, 4, 5
LVC MOS10H	BANK 3, 4, 5	BANK 3, 4, 5	BANK 3, 4, 5
LVC MOS10R	BANK 3, 4, 5	—	BANK 3, 4, 5 *OutputはLVC MOS10H
HTSL15 _ Class I	BANK 3, 4, 5	BANK 3, 4, 5	BANK 3, 4, 5
SSTL15 _ Class I, II	BANK 3, 4, 5	BANK 3, 4, 5	BANK 3, 4, 5
SSTL135 _ Class I, II	BANK 3, 4, 5	BANK 3, 4, 5	BANK 3, 4, 5
HSUL12	BANK 3, 4, 5	BANK 3, 4, 5	BANK 3, 4, 5
Differential I/O Standards			
	INPUT	OUTPUT	Bi-Directional
HTSL15D _ Class I	BANK 3, 4, 5	BANK 3, 4, 5	BANK 3, 4, 5
SSTL18D _ Class I, II	BANK 3, 4, 5	BANK 3, 4, 5	BANK 3, 4, 5
SSTL15D _ Class I, II	BANK 3, 4, 5	BANK 3, 4, 5	BANK 3, 4, 5
SSTL135D _ Class I, II	BANK 3, 4, 5	BANK 3, 4, 5	BANK 3, 4, 5
HSUL12D	BANK 3, 4, 5	BANK 3, 4, 5	BANK 3, 4, 5
LVC MOS33D	—	BANK 3, 4, 5	—
LVC MOS25D	—	BANK 3, 4, 5	—
LVDS	BANK 3, 4, 5	BANK 3, 4, 5	BANK 3, 4, 5
LVDSE	—	BANK 0, 1, 2, 6, 7	—
SUBLVDS	BANK 3, 4, 5	—	—
SUBLVDSE	—	BANK 0, 1, 2, 6, 7	—
SUBLVDSEH	—	BANK 3, 4, 5	—
SLVS	BANK 3, 4, 5	BANK 3, 4, 5	—
MIPI D-PHY	BANK 3, 4, 5	BANK 3, 4, 5	BANK 3, 4, 5

・左記のMIPI D-PHYはuser logicを使用したSoft D-PHYを指し、Hard D-PHYに関しては専用BANKが別途設けられております

引用 : FPGA-DS-02049-0.80

3.10. sysI/O Recommended Operating Conditions

2.2.3. Mixed Voltage Mode

■ 各VCCIO電圧と入力可能なStandard IOまとめ

- ・VCC, VCCAUX, VCCIOの3電源を用いることにより各Bankは以下の通りMixed voltageに対応しています
- ・通常使用する場合と比較しIOのHysteresisおよび消費電力に差分があります（詳細はFPGA-TN-02067を参照）

Wide Range Input Buffers (BANK0, 1, 2, 6, 7)						
VCCIO	LVC MOS10	LVC MOS12	LVC MOS15	LVC MOS18	LVC MOS25	LVC MOS33
1.2	✓	✓	✓			
1.5	✓	✓	✓	✓		
1.8	✓	✓	✓	✓		
2.5	✓	✓	✓	✓	✓	
3.3	✓	✓	✓	✓	✓	✓

High Performance Input Buffers (BANK3, 4, 5)				
VCCIO	LVC MOS10H	LVC MOS12H	LVC MOS15H	LVC MOS18H
1.0	✓			
1.2	✓	✓		
1.5	✓	✓	✓	
1.8	✓	✓	✓	✓

引用 : FPGA-TN-02067-1.0
3. VCCIO Requirement for I/O Standards

2. sysIOに関して



2.3. 差動信号に関して



2.3.1. LVDSの注意事項

■ LVDS入力の注意事項

- ・LVDSの入力バッファはBANK 3, 4, 5でサポートされVCCAUX(=1.8V)によって動作します
(VCCIOに他電圧を接続していてもLVDSの入力は可能)
- ・BANK3, 4, 5において全ての差動ペア間にプログラマブル(ON/OFF可能)な100Ω入力終端を搭載しています

■ LVDS出力の注意事項

- ・LVDSの出力バッファはBANK 3, 4, 5でサポートされVCCIO(=1.8V)によって動作します
(LVDS出力を行うBANKは必然的に1.8Vになるためご注意ください)
- ・BANK3, 4, 5でTRUE LVDS出力を使用する場合は外付け抵抗は不要です

■ LVDS入出力の共通の注意事項

- ・使用するIOは必ず差動ペアを選択する必要があります (T側がTrue、C側がComplementary)
差動ペアはLattice web pageに掲載されているPinlist(csv file)で確認可能です
例: "True_OF_PB4B" と "Comp_OF_PB4A" がペア
- ・差動LVDS入力として扱った場合、内部Pull-upが有効にならないため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、外部での処理が必要です
(フェイルセーフ保護としてVthdがデバイス内に設けられておりますが、Vthdを超えるような差動ノイズを拾うと、レシーバはスイッチングしたり発振し、デバイス内で問題を引き起こす可能性があります)

2.3.2. LVDS Emulate出力の注意事項

■ LVDS Emulate(LVDSE)出力の注意事項

- CrossLink-NXはLVDSE出力をサポートし、基板上に外付け抵抗を設けることで疑似的にLVDS出力を行うことが可能です
- LVDSEはBANK 0, 1, 2, 6, 7でVCCIOは2.5Vでサポートされています（TRUE LVDSはBANK3, 4, 5でサポート）
- 使用するIOは必ず差動ペアを選択する必要があります（T側がTrue、C側がComplementary）
差動ペアはLattice web pageに掲載されているPinlist(csv file)で確認可能です
例：“True_OF_PB4B”と“Comp_OF_PB4A”がペア

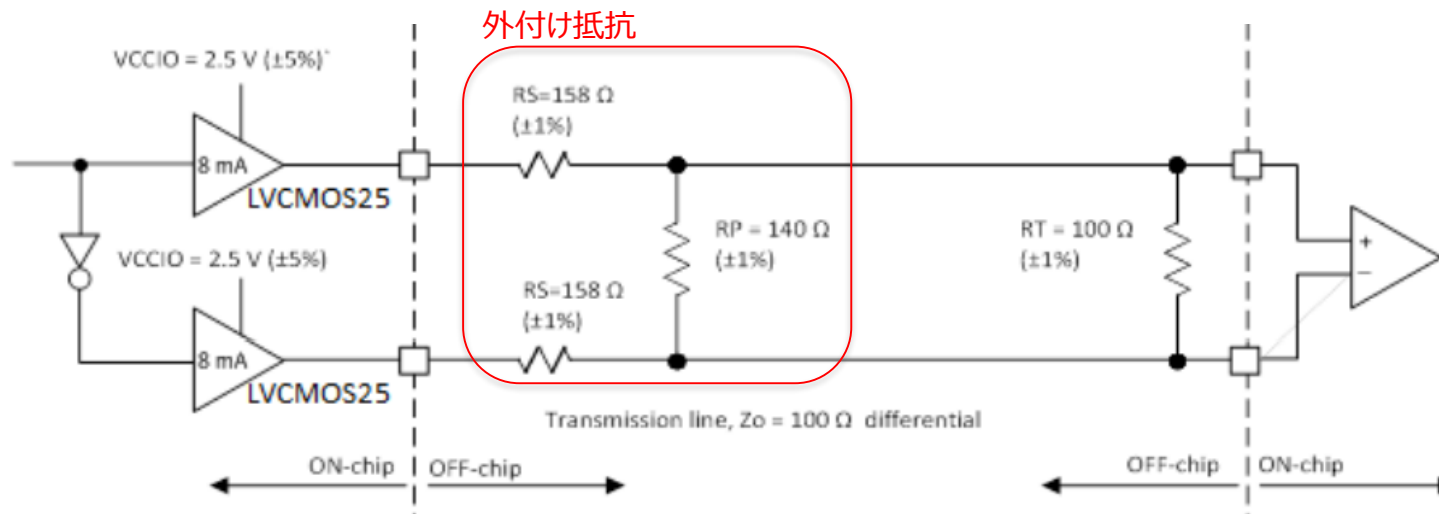


Figure 3.2. LVDS25E Output Termination Example



2.3.3. subLVDSの注意事項

■ subLVDS入力の注意事項

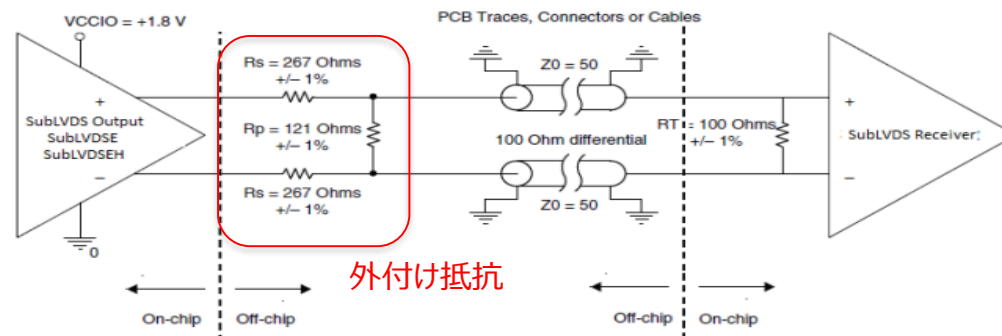
- ・subLVDSの入力バッファはBANK 3, 4, 5でサポートされVCCAUX(=1.8V)によって動作します (VCCIOに他電圧を接続していてもsubLVDSの入力は可能)
- ・BANK3, 4, 5において全ての差動ペア間にプログラマブル(ON/OFF可能)な100Ω入力終端を搭載しています

■ subLVDS出力の注意事項

- ・subLVDS出力は下図の通り基板上に外付け抵抗を設け、VCCIO(=1.8V)によって動作します
- ・BANK 0, 1, 2, 6, 7では"subLVDSE"、BANK 3, 4, 5では"subLVDSEH"でサポートされLVCMOS18/18Hバッファに依存します

■ subLVDS入出力の共通の注意事項

- ・使用するIOは必ず差動ペアを選択する必要があります (T側がTrue、C側がComplementary)
差動ペアはLattice web pageに掲載されているPinlist(csv file)で確認可能です
例: "True_OF_PB4B" と "Comp_OF_PB4A" がペア

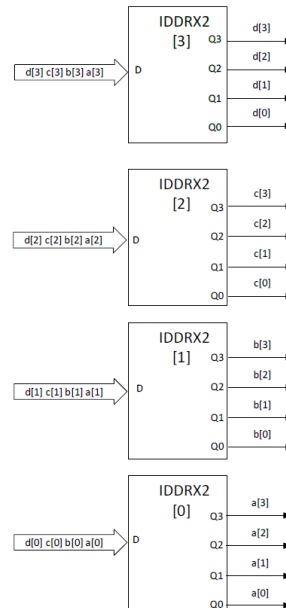


2.3.4. Generic Double Data Rate

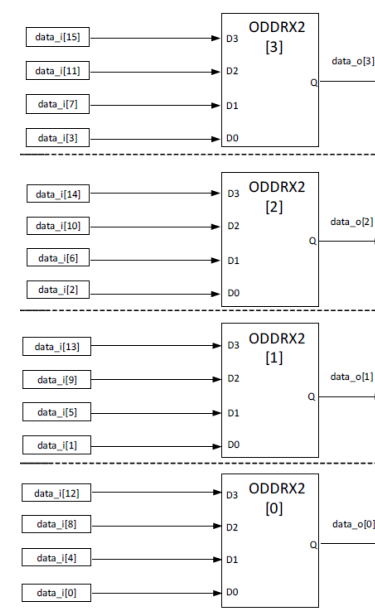
■ Generic Double Data Rate module (GDDR) を使用する場合の注意事項

- X1 (1:2) , X2 (1:4) , X4 (1:8) , X71 (7:1) , X5 (1:10) の変換をGDDR (ハードマクロ) でサポートされています
※入力側はシリパラ変換、出力側はパラシリ変換を行い、動作の例は下図の通り
- IFをLVDS (差動) として使用される場合はBANK 3, 4, 5のみ対応しています
※使用するIOは必ず差動ペアを選択する必要があります (T側がTrue、C側がComplementary)
差動ペアはLattice web pageに掲載されているPinlist(csv file)で確認可能
- IFをLVCMOSとして使用される場合はBANK 0, 1, 2, 6, 7のみ対応しています
- 入力クロックとデータは同じサイドにアサインする必要があります
- クロックはPCLKピンへアサインする必要があります

GDDR Input “X2 (1:4)” の例
• Bus Widthが4の場合



GDDR Output “X2 (4:1)” の例
• Bus Widthが4の場合





2.3.6. デバック用IF (Reveal)

■ Revealを使用する際の注意事項（Reveal：FPGAデバック用の内部信号観測ツール）

- ・RevealはJTAG IFを使用しFPGA内部の信号をPC上で観測することが可能です
- ・FPGAの設計時（試作基板）はRevealを使用するためのJTAG IFを設けることを推奨します

■ SSPIモードでコンフィグレーションを行う場合

- ・CrossLink-NXのJTAGピンはSSPI用のピンと兼用になっているため、以下の2通りの方法でRevealを使用可能です

- 1) SSPI用の回路とJTAG用の回路を基板上でwired orし、コンフィグレーション後にJTAGENピンを制御しJTAGとして使用
- 2) 任意の汎用IOをJTAGとして使用

※任意の汎用IOを使用する場合はRadiantのReveal Inserter上でsoft JTAGを選択することで汎用IOに割り当てが可能

3. コンフィグレーションに関して



3.1. コンフィグレーションモード

3.1.1. 概要

■ Master SPI Modes

SPI Flash Memoryと直接接続し、コンフィグレーションを行います。

メモリとのデータバスは、[Serial/Dual/Quad](#)の3つから選択可能で、最大150MHzの速度に対応可能です。

複数のコンフィグレーションデータをSPI Flashに格納し、ブートする方法は、以下の3種類があります。

- [Dual boot](#)
- [Multi boot](#)
- [Ping-Pong boot](#)

■ Slave SPI Mode

CPU等からSPI経由で接続し、コマンドによりコンフィグレーションを行います。

データバスは、[Serial/Dual/Quad](#)の3つから選択可能で、最大150MHzの速度に対応可能です。

■ Slave I2C/I3C Mode

CPU等からI2C/I3C経由で接続し、コマンドによりコンフィグレーションを行います。

■ JTAG Mode

JTAG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。



3.1.2. モード選択方法 1/2

各コンフィグレーションモードの選択方法を記載します。

■ Master SPI

- デバイス起動後、Master SPI portはデフォルトでenableになっています。
- ユーザーモードでMaster SPI portを維持するためには、RadiantのConstraint EditorにてMaster_spipi_portにSERIAL,DUAL,QUADのいずれかを設定します。

■ Slave SPI

- デバイス起動後、PROGRAMNをLに保持し、Slave SPI portのSCSNをドライブし、Activation key(Table 6.4.)Master側から入力します。アクティベート後はPROGRAMNの状態は無関係です。
- ユーザーモードでSlave SPI portを維持するためには、RadiantのConstraint EditorにてSlave_spi_portにSERIAL,DUAL,QUADのいずれかを設定します。

Table 6.4. Slave SPI Configuration Port Activation Key

Slave Port/ Activation Key	Slave Configuration Port Activation Key	
Slave SPI Port	Dummy Bytes*	32'HA4C6F48A

Note : The number of dummy bytes should be at least 1 only last shifted in 32 bits matters.



3.1.2. モード選択方法 2/2

各コンフィグレーションモードの選択方法を記載します。

■ Slave I2C/I3C

- デバイス起動後、PROGRAMNをLに保持し、Activation key(Table 6.8)をMaster側からSlave addressにWriteします。アクティベート後はPROGRAMNの状態は無関係です。
- ユーザーモードでSlave I2C/I3C portを維持するためには、RadiantのConstraint EditorにてSlave_i2ci3c_portにSERIAL,DUAL,QUADのいずれかを設定します。

Table 6.8. Slave SPI Configuration Port Activation Key

Slave Port/ Activation Key	Slave Configuration Port Activation Key	
Slave I ² C Port	Slave I2C Port Address*	32'HA4C6F48A

*Note : The slave I2C/I3C address could be either 7 bits or 10 bits address

■ JTAG

- JTAG_ENABLE pinがHにドライブされると、デバイスがコンフィグレーションモードかユーザーモードかに関わらず、JTAG port経由でコンフィグレーション可能です。

3. コンフィグレーションに関して



3.2. コンフィグレーション関連ピン



3.2.1. 各コンフィグレーションモードにおける使用ピン

各コンフィグレーションモード中に使用するピンと、コンフィグ中のプル設定は以下の通りです。
各ピンのボード上での外付けプルアップ/ダウン抵抗等の接続に関しては、後述の回路例をご確認ください。

ピン名	コンフィグ中 Pull設定	コンフィグレーションモード			
		JTAG	MSPI	SSPI	I2C/I3C
JTAG_ENABLE	DOWN	1'b1	1'bx	1'b0	1'b0
PROGRAMN	UP	1'b0	1'b1	1'b0	1'b0
INITN	UP	INITN			
DONE	UP	DONE			
MCLK	UP/DOWN	-	MCLK	-	-
MCSN	UP	-	MCSN	-	-
MOSI/MD0	UP	-	MOSI/D0	-	-
MISO/MD1	UP	-	MISO/D1	-	-
MD2	UP	-	D2	-	-
MD3	UP	-	D3	-	-
CSNO/SDO	UP	-	CSNO/SDO	-	-
TCK/SCLK	DOWN	TCK	-	SCLK	-
TMS/SCSN	UP	TMS	-	SCSN	-
TDI/SI/SD1	UP	TDI	-	MOSI/D0	-
TDO/SO/SD1	UP	TDO	-	MISO/D1	-
SD2/SCL	UP	-	-	D2	SCL
SD3/SDA	UP	-	-	D3	SDA

※MCLKの内蔵プルアップ、ダウンは、コントロールレジスタの設定(CPOL)で変更可能です。
詳細はLattice社TN-02099-1-0-CrossLink-NX-sysCONFIG-Usage-Guideをご確認下さい。

3.2.2. 開発ツール上の設定 1/4

NOWHERE,
but **HERE.**

コンフィグレーション関連ピンの設定や、Boot modeの選択などは、開発ツールRadiant上で行います。

RadiantのDevice Constraint EditorのGlobalタブで設定可能な項目について説明します。

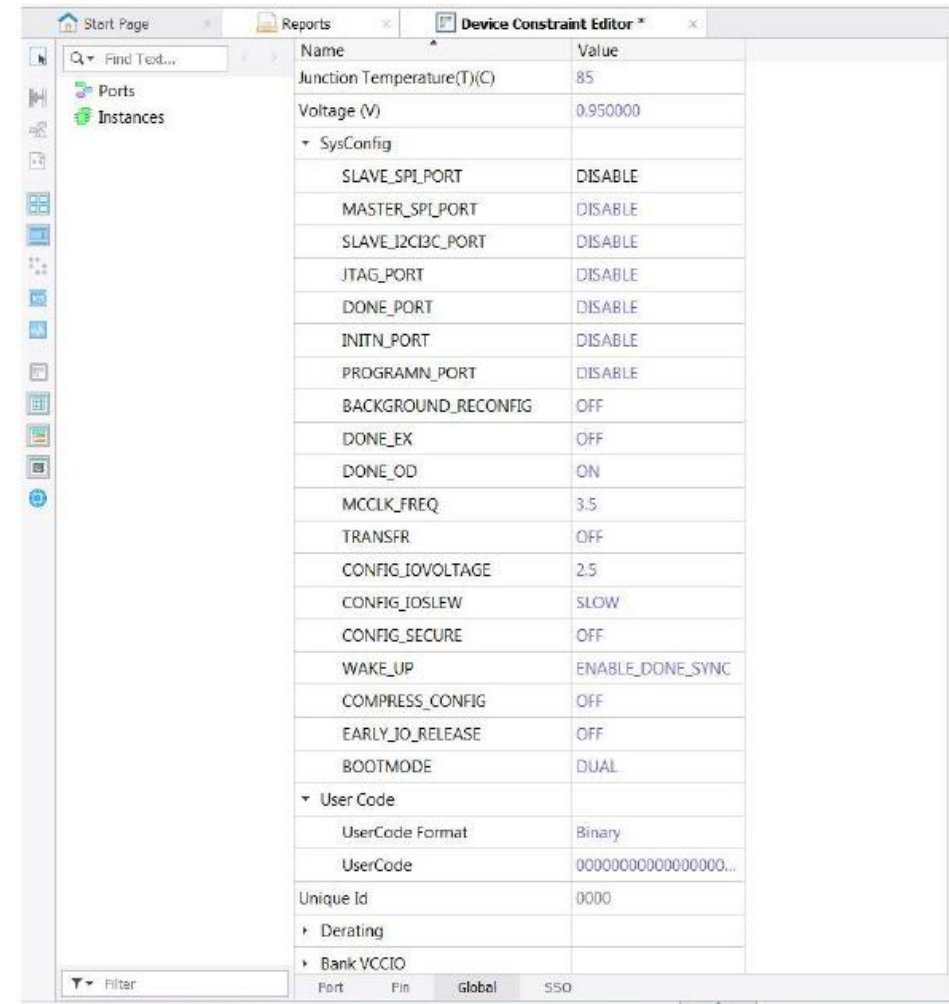


Figure 7.1. sysCONFIG Preferences in Global Tab, Lattice Radiant Device Constraint Editor

3.2.2. 開発ツール上の設定 2/4



sysConfig	設定項目	内容
SLAVE_SPI_PORT	DISABLE (Default) SERIAL DUAL QUAD	SLAVE SPIポートを利用してプログラムまたはリードする際に設定します。 SERIAL/DUAL/QUADに設定した場合、SCLK, SCSN, SI/SD0, SO/SD1, SD2, SD3から、それぞれの対応ピンがコンフィグ後も有効になります。
MASTER_SPI_PORT	DISABLE (Default) SERIAL DUAL QUAD	MASTER SPIポートを利用してプログラムまたはリードする際に設定します。 SERIAL/DUAL/QUADに設定した場合、MCLK, MCSN, MOSI/MD0, MISO/MD1, MD2, MD3から、それぞれの対応ピンがコンフィグ後も有効になります。
SLAVE_I2CI3C_PORT	DISABLE (Default) ENABLE	SLAVE I2CI3Cポートを利用してプログラムまたはリードする際に設定します。 ENABLEに設定した場合、SD2/SCL, SD3/SDAがコンフィグ後も有効になります。
JTAG_PORT	DISABLE (Default) ENABLE	JTAGポートを利用してプログラムまたはリードする際に設定します。ENABLE に設定した場合、TCK, TMS, TDI, TDOがコンフィグ後も有効になります。
DONE_PORT	DISABLE (Default) ENABLE	DONEピンはデフォルトではGPIOとして設定されていますが、コンフィグの際に 必要な場合はENABLEに設定します。
INITIN_PORT	DISABLE (Default) ENABLE	INITNピンはデフォルトではGPIOとして設定されていますが、コンフィグの際に 必要な場合はENABLEに設定します。
PROGRAMN_PORT	DISABLE (Default) ENABLE	PROGRAMNピンはデフォルトではGPIOとして設定されていますが、コンフィグ の際に必要な場合はENABLEに設定します。

3.2.2. 開発ツール上の設定 3/4



sysConfig	設定項目	内容
BACKGROUND_RECONFIG	OFF (Default) ON SRAM_EBR SRAM_ONLY	コンフィグレーション完了するまでに、バックグラウンドで動作させるモードをTransparent アクセスモードと呼びます。そのモード設定と、書き込む内容を設定します。 OFF…コンフィグ完了までユーザロジックは動作しない ON …ユーザロジックを動作させたまま、SRAM、EBR、IPの書き込み可能 SRAM_EBR…ユーザロジックを動作させたまま、SRAM、EBRの書き込み可能 SRAM_ONLY…ユーザロジックを動作させたまま、SRAMの書き込み可能
DONE_EX	OFF (Default) ON	デフォルトではコンフィグ完了しデバイス起動と同期してDONEピンがHになりますが、これを外部信号で制御する場合にONに設定します。 ONの場合、後述のWAKE UPをENABLE_DONE_SYNCに設定する必要があります。 OFFの場合、後述のWAKE UPをDISABLE_DONE_SYNCに設定する必要があります。
DONE_OD	ON (Default) OFF	DONEピンをオープンドレイン出力の設定にします。
MCCLK_Frequency	3.5 (Default) 7.0/14.1/28.1/56.2/90/112.5	外付けSPI Flashからコンフィグする場合のクロック周波数を設定します。 精度は±15%です。
TRANSFR	OFF (Default) ON	TransFR機能の設定をします

3.2.2. 開発ツール上の設定 4/4



sysConfig	設定項目	内容
CONFIG_IOVOLTAGE	2.5(Default) 1.0/1.2/1.5/1.8/2.5/3.3	コンフィグ関連ピンが存在するbank0, 1の電圧を設定します
CONFIG_IOSLEW	SLOW (Default) MEDIUM FAST	コンフィグ関連ピンのSlew Rateを変更できます。基板環境の違いに簡単に適合することが可能です。
CONFIG_SECURE	OFF (Default) ON	ONの場合、sysCONFIGやJTAGポートからのリードバックを防ぎます。
WAKE UP	ENABLE_DONE_SYNC (Default) DISABLE_DONE_SYNC	ENABLE_DONE_SYNC…外部DONEピンと同期してデバイス起動します。このオプションを選択する場合は、DONE_EXをONにする必要があります。 DISABLE_DONE_SYNC…外部DONEピンと同期しません。DONE_EXをOFFにする必要があります。
COMPRESS_CONFIG	OFF (Default) ON	圧縮したBitstreamファイルを生成するかどうか選択します。
EARLY_IO_RELEASE	OFF (Default) ON	Bank1,2,6,7のI/Oに関してEARLY_IO_RELEASEを行うか選択します
BOOTMODE	DUAL (Default) SINGLE NONE	Dual…デュアルブートを実行します。失敗した場合でもデバイスを起動させるためゴールデンブートイメージが呼び出されます。 SINGLE…シングルブートを実行します。失敗した場合、デバイスは未プログラム状態になります。 NONE…Master SPIでのブートではない場合に選択します。Slaveコンフィグレーションポートにてコンフィグされることを待ちます。

3. コンフィグレーションに関して



3.3. コンフィグレーションフロー

3.3. コンフィグレーションフロー 1/4

1 . Power UP

電源が以下の条件を満たすとき、Power On Resetがかかります

- $V_{CC} > 0.67\text{ V} - 0.74\text{ V}$
- $V_{CCAUX} > 1.45\text{ V} - 1.61\text{ V}$
- $V_{CCIO0} > 0.95\text{ V} - 1.05\text{ V}$
- $V_{CCIO1} > 0.95\text{ V} - 1.05\text{ V}$

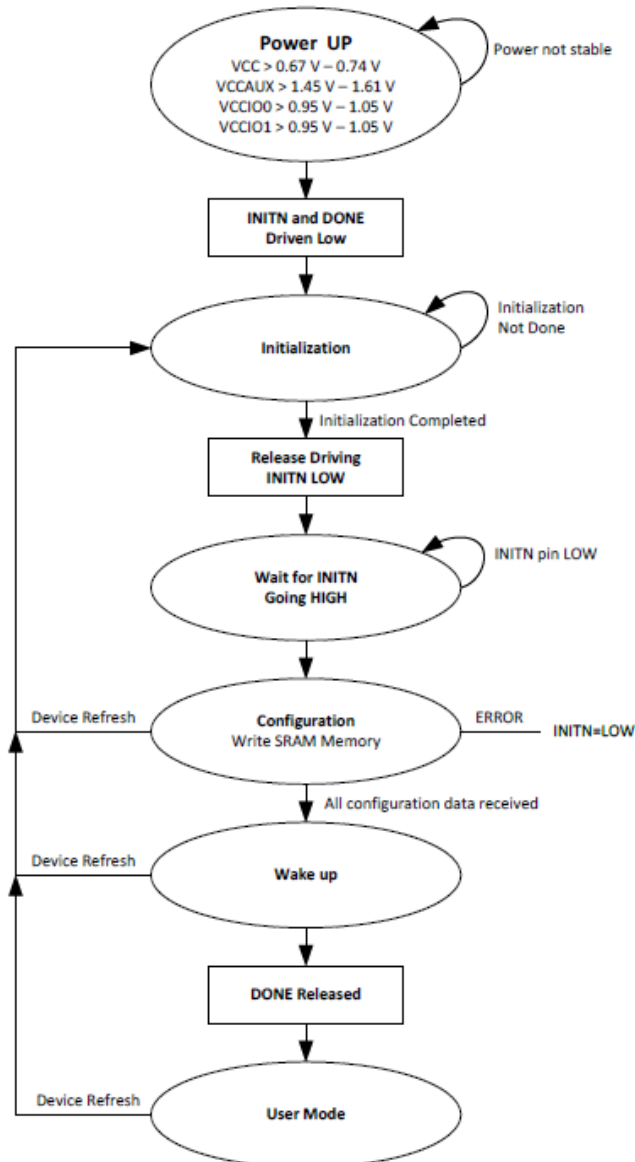


Figure 5.1. Master Port Configuration Flow

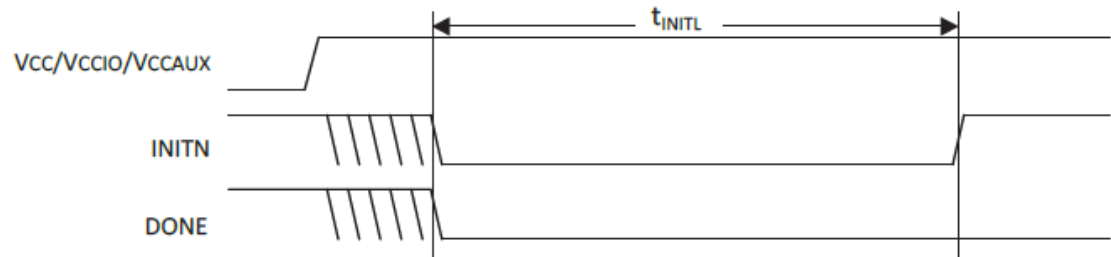


Figure 5.2. Configuration from Power-On-Reset Timing

3.3. コンフィグレーションフロー 2/4

2 . Initialization

Power on reset回路により、INITNとDONEがLowにドライブされた後、デバイスは Initialization ステートに入り、全てのSRAM領域が初期化されます。

下記条件全てが満たされるまで、初期化状態を保ちます。

- t_{INITL}時間経過
- PROGRAMNピンがディアサート
- INITNピンが外部Masterによりアサートされていない

t_{INITL}の期間、FPGA内の全てのSRAM領域がクリアされます。

初期化が完了すると、INITNをリリースし、外付け抵抗によりHにプルアップされます。

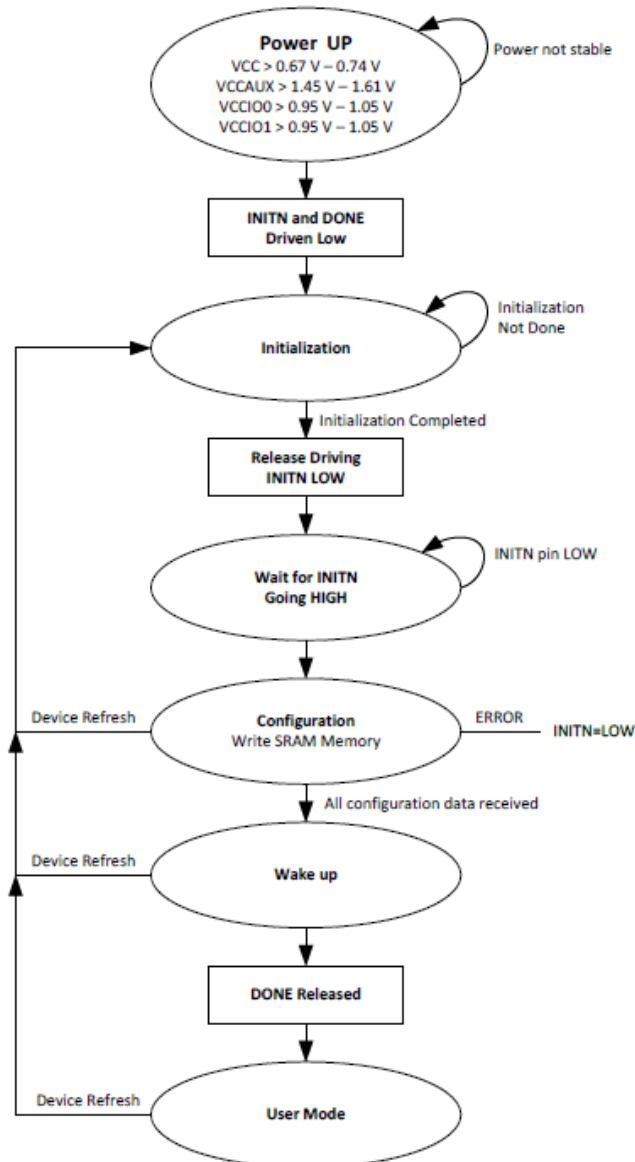


Figure 5.1. Master Port Configuration Flow

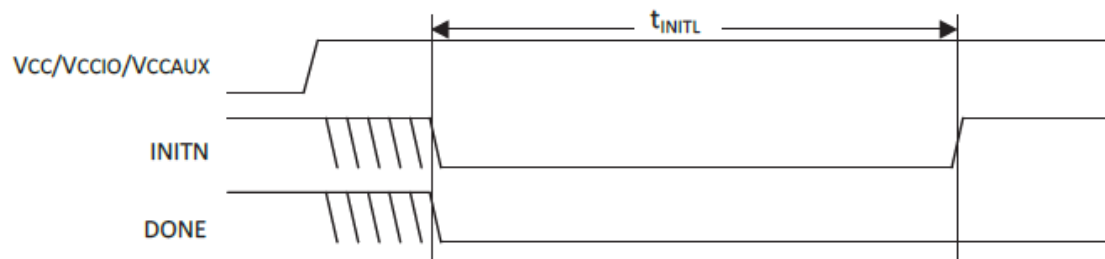


Figure 5.2. Configuration from Power-On-Reset Timing

3.3. コンフィグレーションフロー 3/4



3 . Configuration

初期化が完了後、外付け抵抗によりINITNがHになるとコンフィグレーションが開始します。その後のコンフィグレーションにおいてエラーが発生した際はINITNは再度Lにドライブされます。

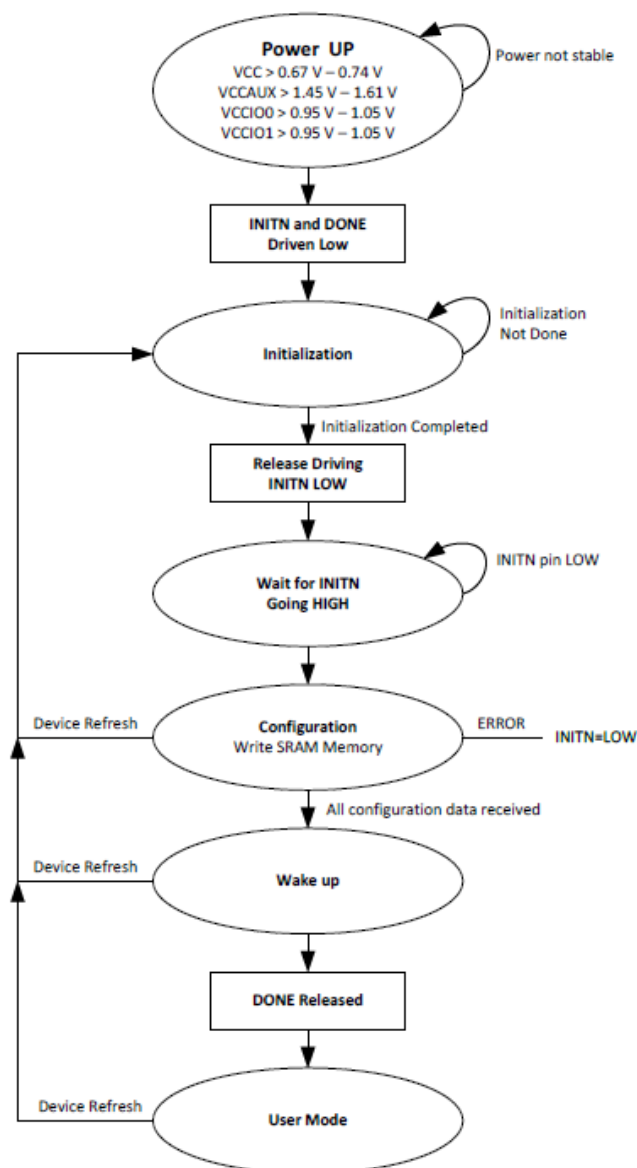


Figure 5.1. Master Port Configuration Flow

3.3. コンフィグレーションフロー 4/4

4 . Wake Up

コンフィグレーションモードからユーザーモードへの移行ステートです。
デバイスがすべてのコンフィグレーションデータを受信すると、内部DONE status bitをアサートし、以下の制御を実行します。

- **Global Set/Reset (GSR)**
アサートされると、すべてのI/O FF, FF, LUT, 分散RAM, EBRが Set/Resetされます
- **Global Output Enable (GOE)**
アサートされると、I/OのHi-z状態を終了し、プログラムされたI/Oの状態となります。
- **Global Write Disable (GWDIS)**
アサートされると、RAMロジックのWrite Enableを無効にし、RAMリソースの破損を防ぎます。
- **External DONE**
DONEピンをリリースし、コンフィグレーションの完了を示します。

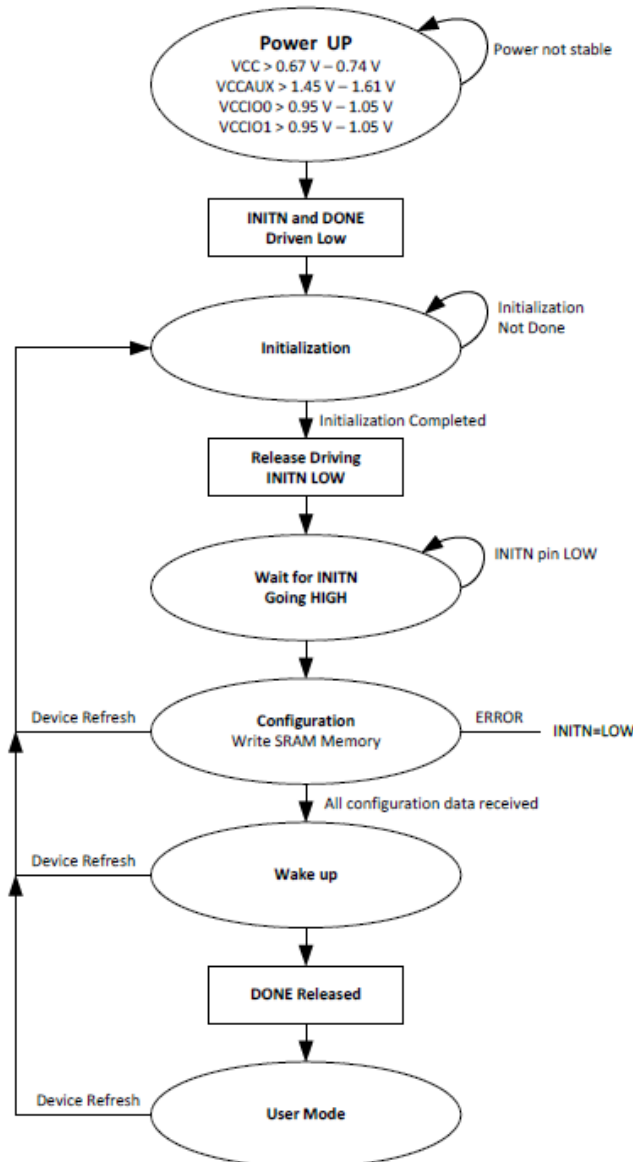
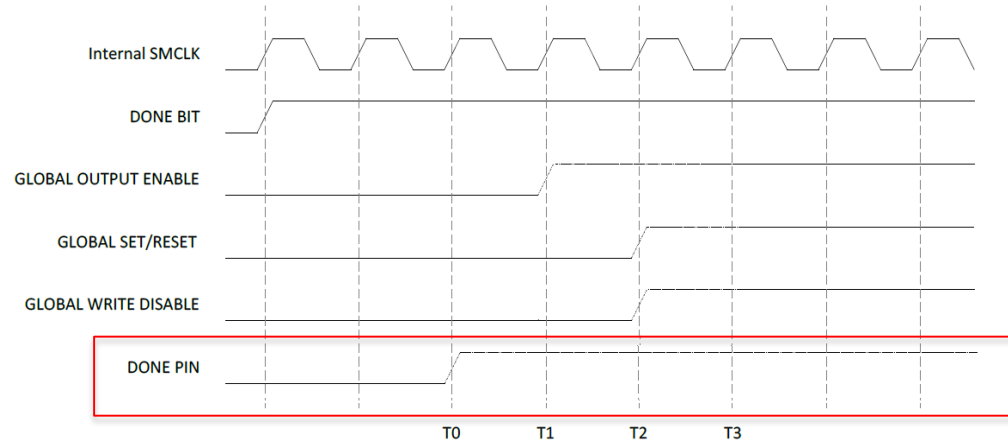


Figure 5.1. Master Port Configuration Flow

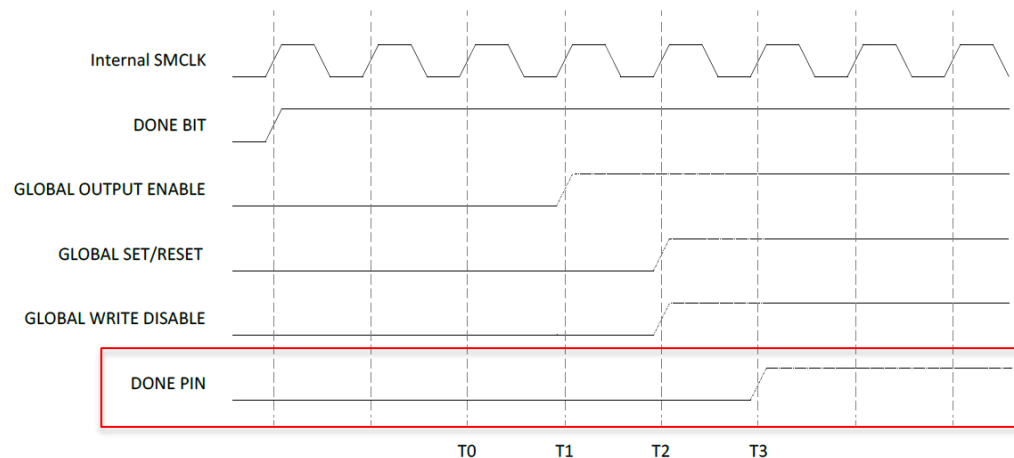
補足 Wake UPにおけるDONEピンの挙動に関して

DONEピンの挙動は、RadiantのGlobal PreferenceにおけるWake Upの設定によって変わります。

- ENABLE_DONE_SYNC (デフォルト)…DONEピンがアサートしてからWake upシーケンス開始



- DISABLE_DONE_SYNC …GSR, GOE, GWDIS後にDONEピンがアサート

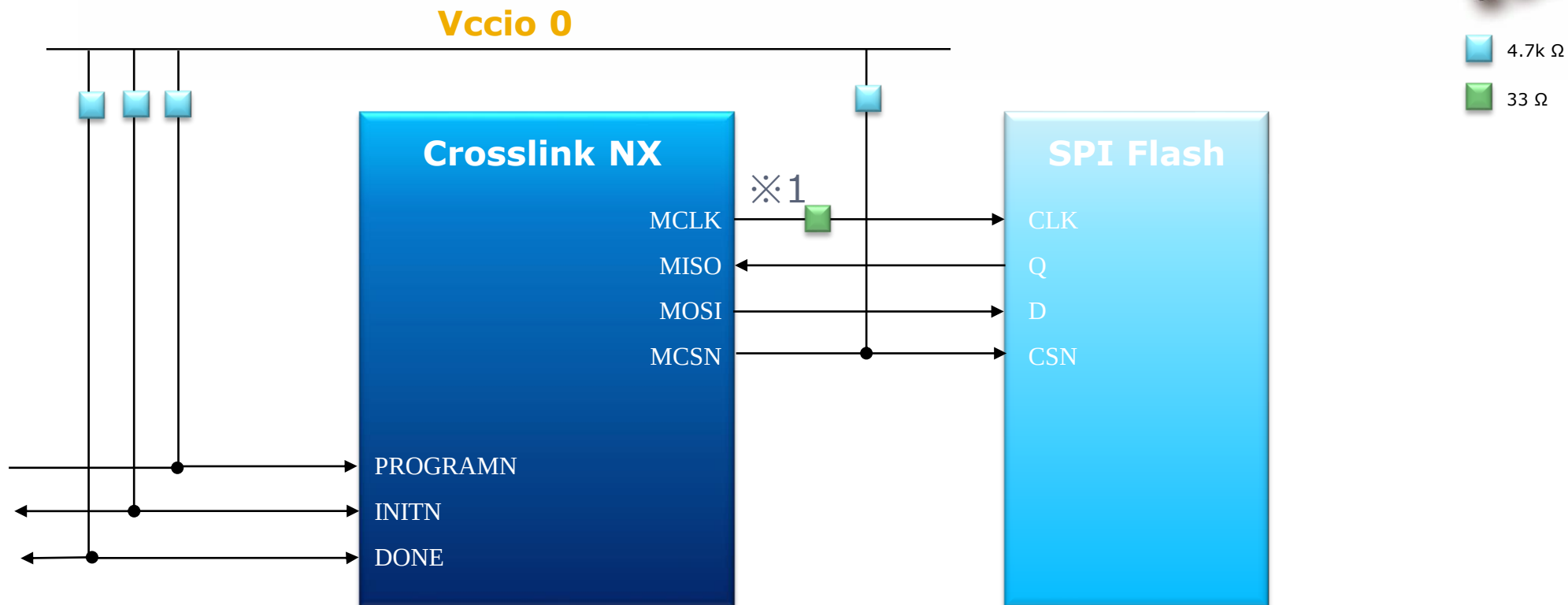


3. コンフィグレーションに関して



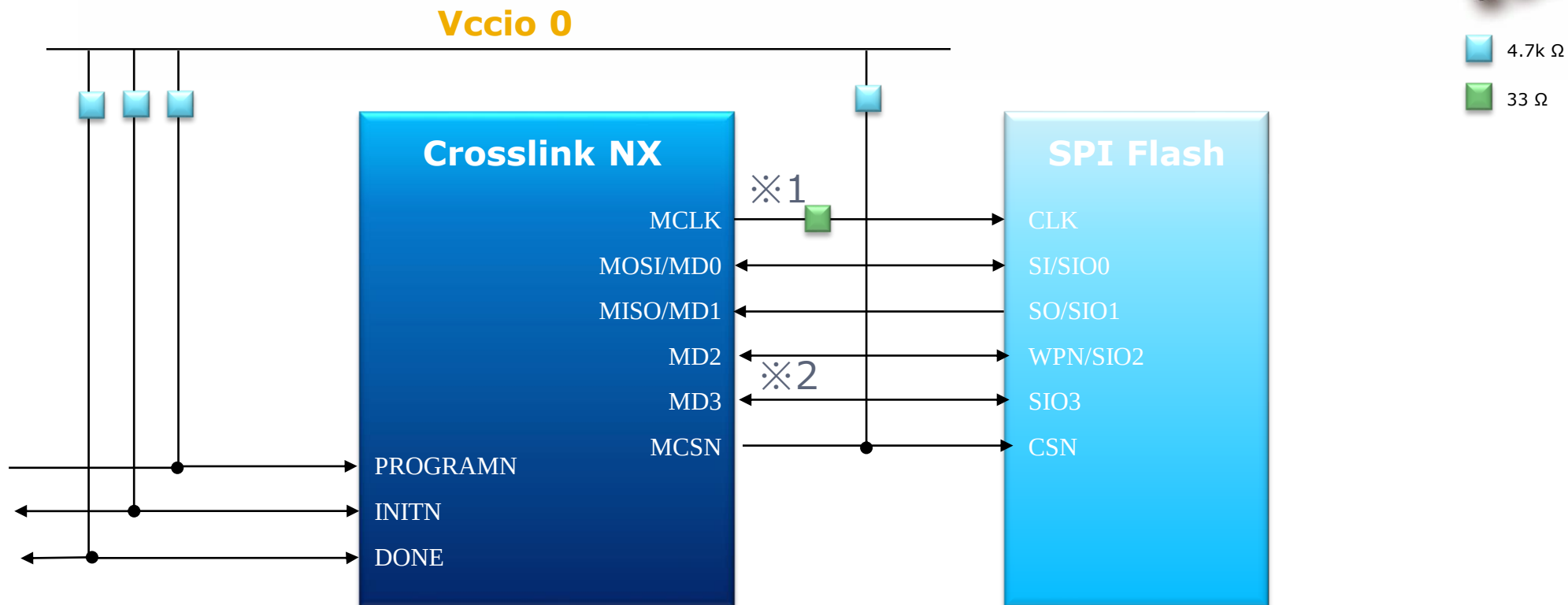
3.4. 回路例

3.4.1 Master SPI Serial



※1 MCLKを高速に設定した場合は、図のようにMCLKに33Ωのダンピング抵抗を挿入することを推奨します。

3.4.2 Master SPI Dual/Quad



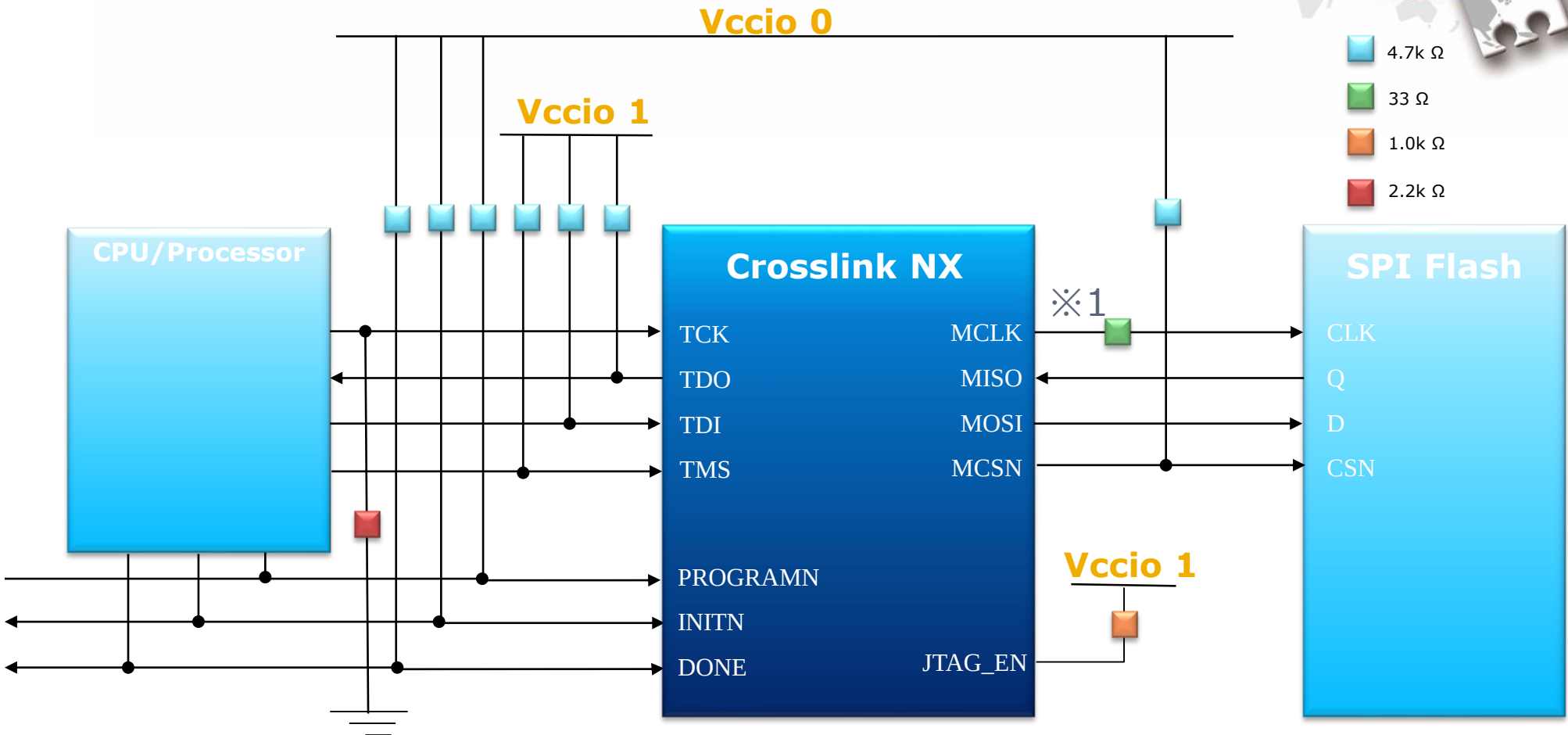
※1 MCLKを高速に設定した場合は、図のようにMCLKに33Ωのダンピング抵抗を挿入することを推奨します

※2 図はQuadの場合を示しております。Dualの場合、MD2⇔WPN/SIO2とMD3⇔SIO3のラインは必要ありません。

3.4.3 JTAG to Master SPI Bridge

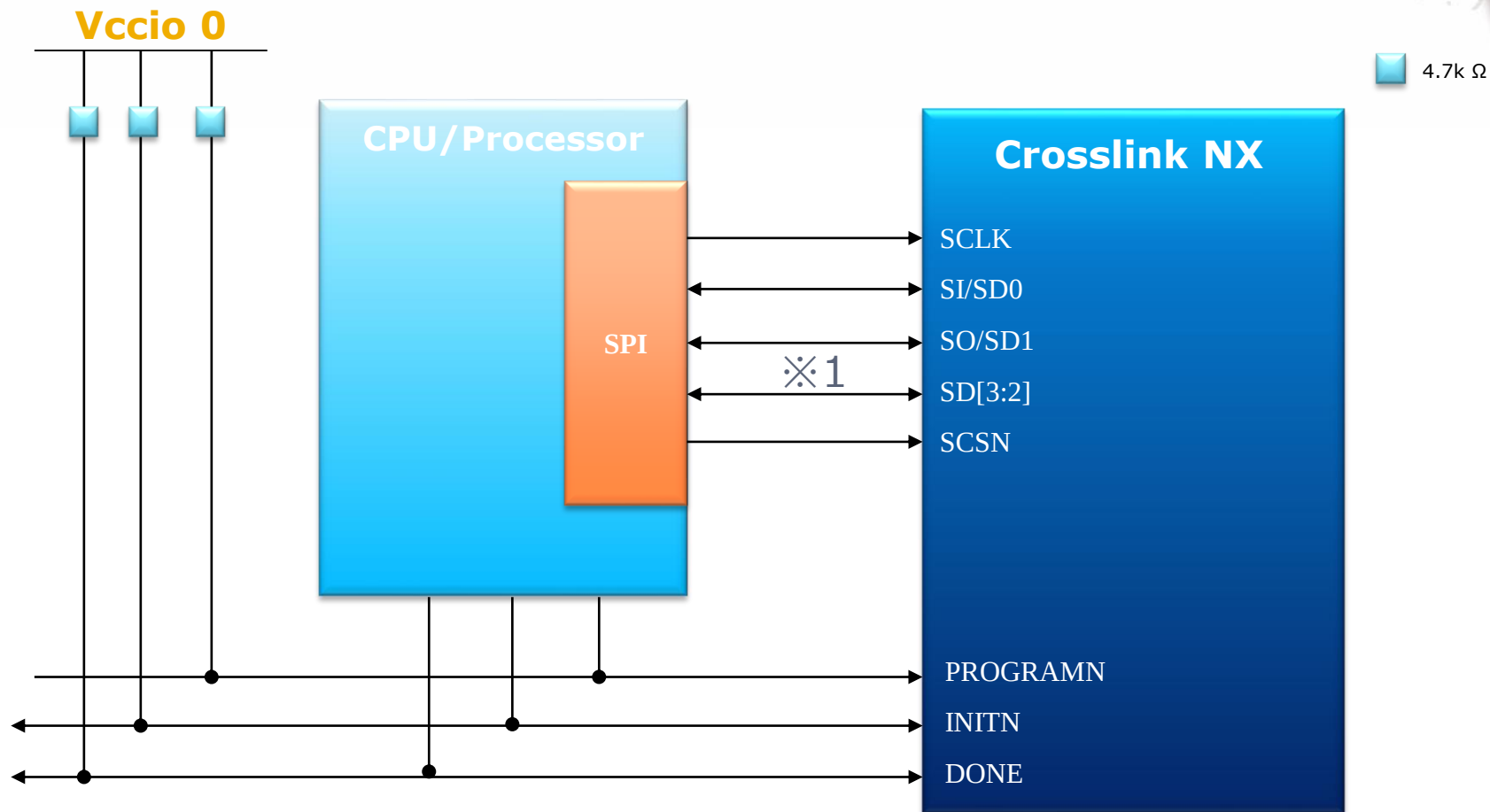


- 4.7k Ω
- 33 Ω
- 1.0k Ω
- 2.2k Ω



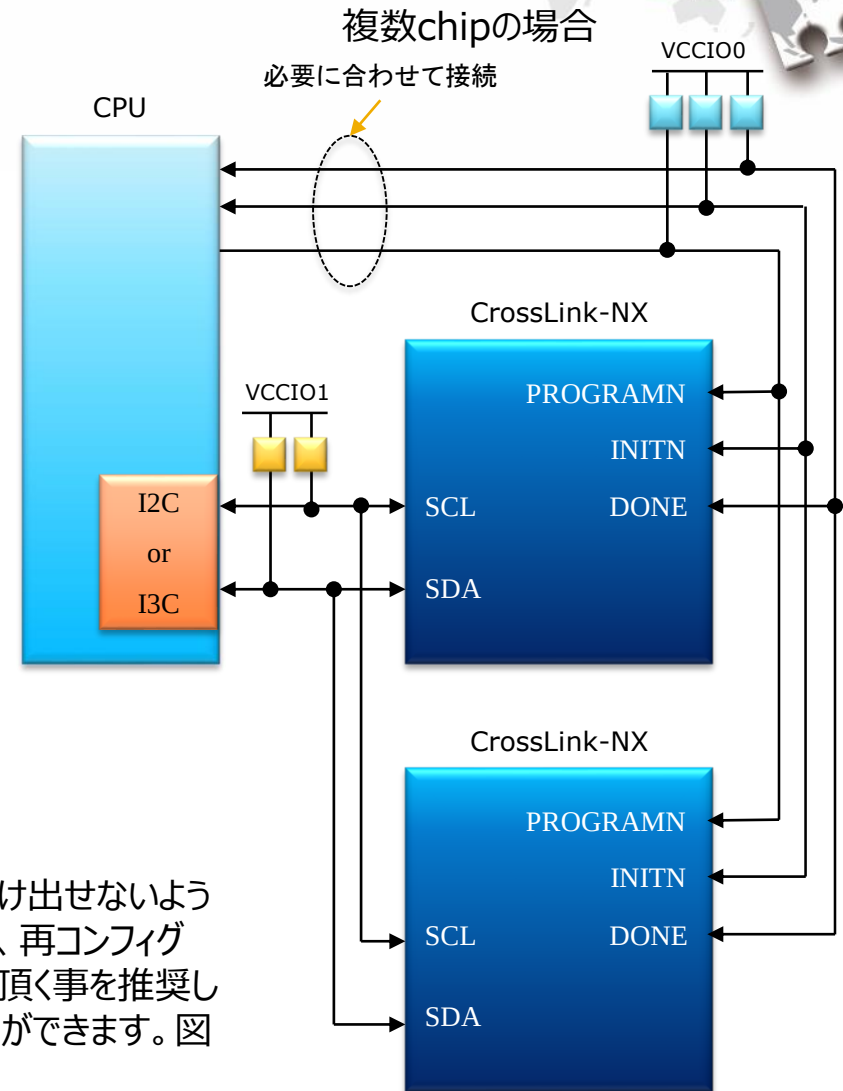
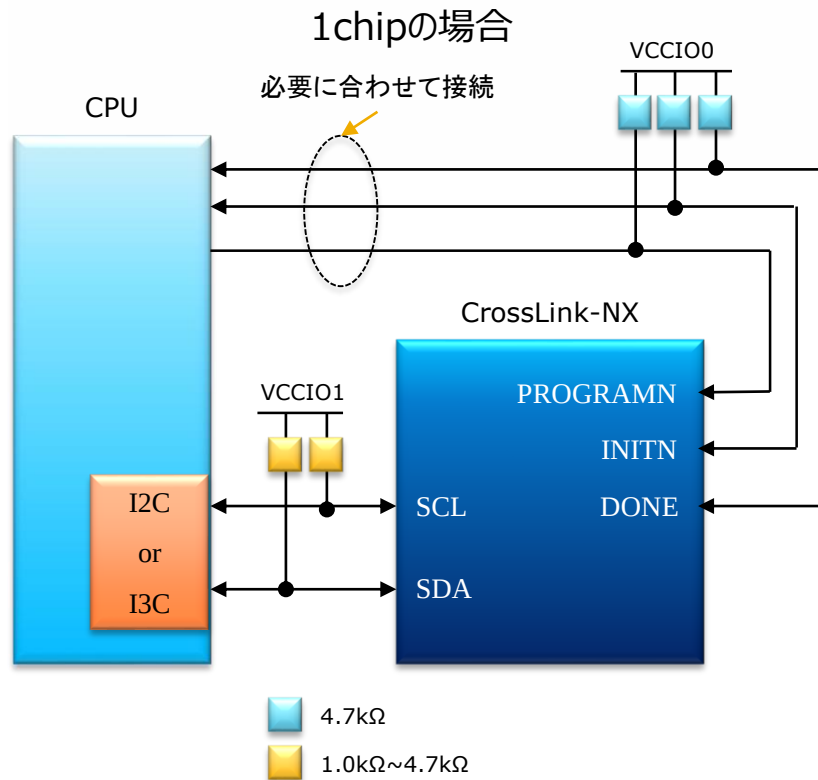
※1 MCLKを高速に設定した場合は、図のようにMCLKに33 Ω のダンピング抵抗を挿入することを推奨します

3.4.4 Slave SPI Serial/Dual/Quad



※1 図はQUADモードの場合を示しています。Dualの場合はSD2,SD3のラインは必要ありません。

3.4.5 Slave I2C/I3C



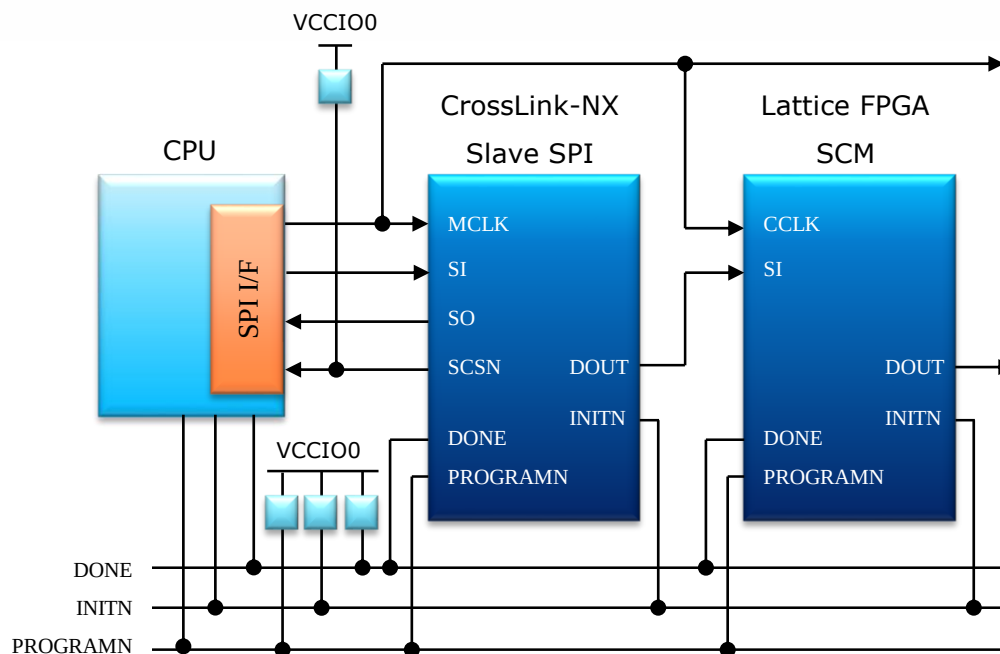
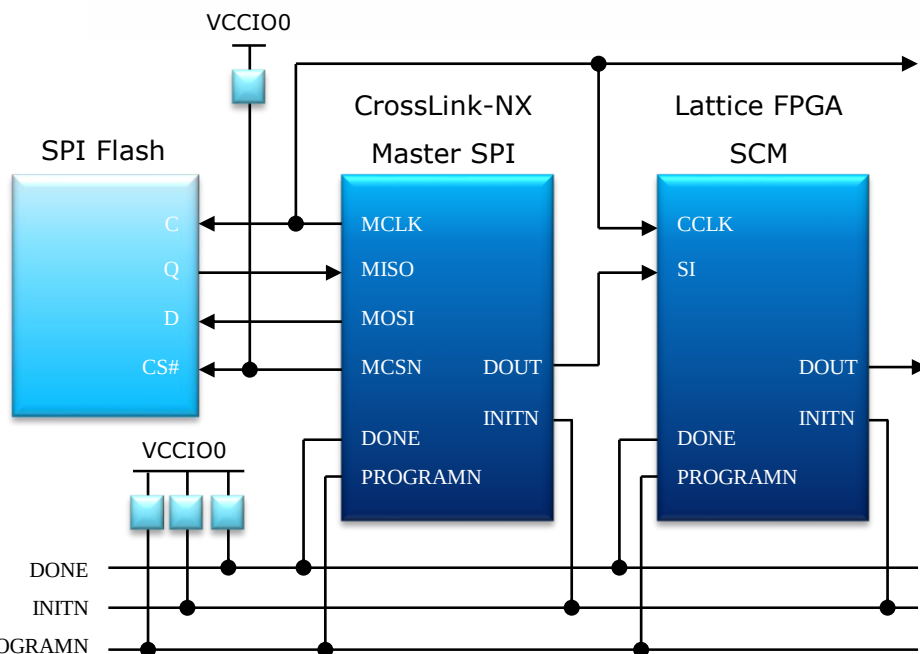
PROGRAMNはコンフィグ失敗時や、コンフィグレーションシーケンスから抜け出せないようなイレギュラーな現象が起こった場合にトグルすることでFPGAを初期化し、再コンフィグレーションを行なうことが可能です。CPUに接続し、制御できるように構成頂く事を推奨します。DONE,INITNに関してはコンフィグレーションの状態を確認することができます。図のようにCPUやLEDなどに接続しておくことでデバック時に有効です。

3.4.6 Daisy Chain

NOWHERE,
but **HERE.**

4.7k Ω

Bypass modeの場合

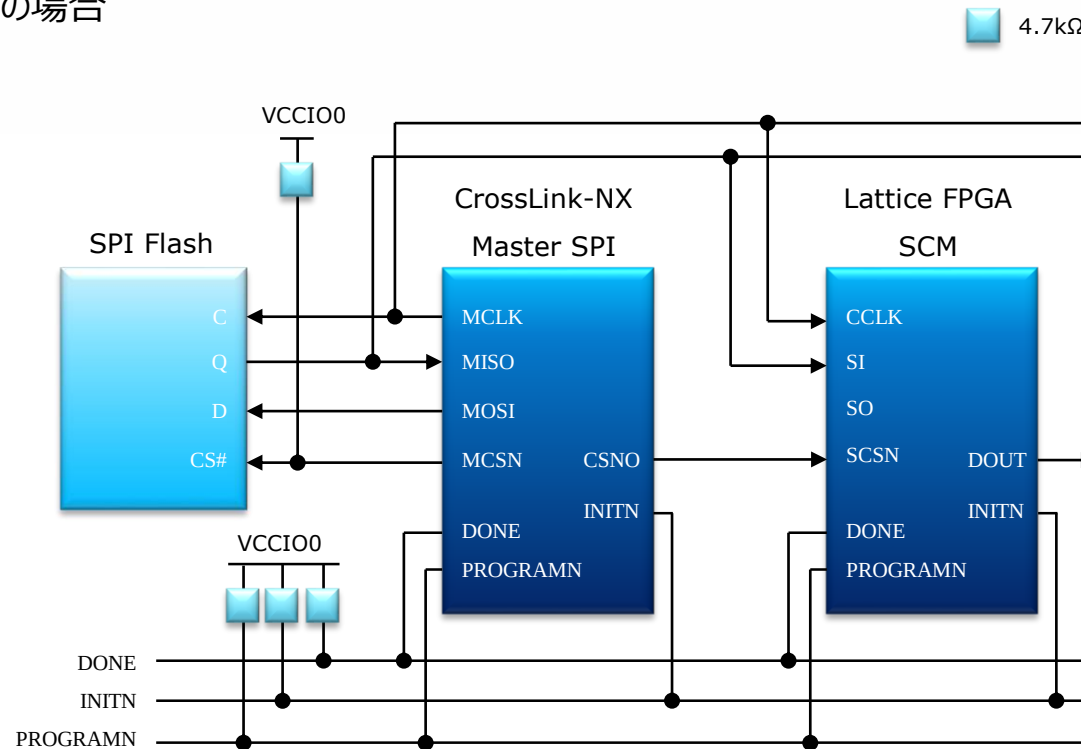


CrossLink-NXはチェーン構成の中の一番最初にのみ配置することができます。

Bypass modeでは、2段目以降のデバイスに対して、CCLK及びSIの2線を接続するだけで構成ができます。(Dual modeやQuad Modeは構成できません)

3.4.7 Daisy Chain

Flow Through modeの場合



CrossLink-NXはチェーン構成の中の一番最初にのみ配置することができます。

Flow Through modeでは、Daisy Chainを構成する各デバイスが対応しているのであれば、データラインの複線化 (Dual Mode、Quad Mode)をサポートします。

これによりコンフィグレーション時間の短縮が図れます。

3. コンフィグレーションに関して



3.5. コンフィグレーションデータ Bit Size

3.5. コンフィグレーションデータサイズ



Table 4.1. Maximum Configuration Bits

Device	Scenario	All Uncompressed	SPI Mode	
		Unencrypted/Encrypted Bitstream Size (Mb)	Recommended SPI Flash Size (Mb)	Dual Boot Recommended SPI Flash Size (Mb)
LIFCL-17	No LRAM, No EBR,	TBD	TBD	TBD
	No LRAM, MAX EBR	TBD	TBD	TBD
	MAX LRAM, No EBR	TBD	TBD	TBD
	MAX LRAM, MAX EBR	TBD	TBD	TBD
LIFCL-40	No LRAM, No EBR,	6.232	8	16
	No LRAM, MAX EBR	7.758	8	16
	MAX LRAM, No EBR	7.281	8	16
	MAX LRAM, MAX EBR	8.807	16	32

Note: Both unencrypted and encrypted bitstreams are the same size. Compression ratio depends on bitstream, so we only provide uncompressed bitstream data.

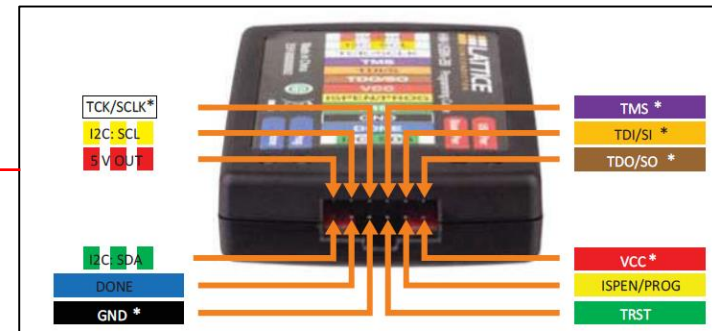
4. Radiant Programmerを使用した書き込みについて



4.1. Downloadケーブル



デバイスへのコンフィグレーションデータのダウンロードには下図のUSB Downloadケーブルを使用します※



USB Download Cable – HW-USBN-2B

コネクタにFlywireケーブルが接続されており、JTAG/SPI/I2Cの各I/Fを兼用できるようになっています。基板側には2.54mm pitchのピンヘッダをご用意いただければFlywireケーブルを接続できます。

Radiant ProgrammerがインストールされたPCと基板とを上記ケーブルで接続します。PCにはドライバーをインストールする必要があります。ドライバのインストールの詳細に関しては、Lattice Radiant Software Installation Guideの“Install and Uninstall Cable Drivers”の項をご参照ください。
http://www.latticesemi.com/view_document?document_id=52751

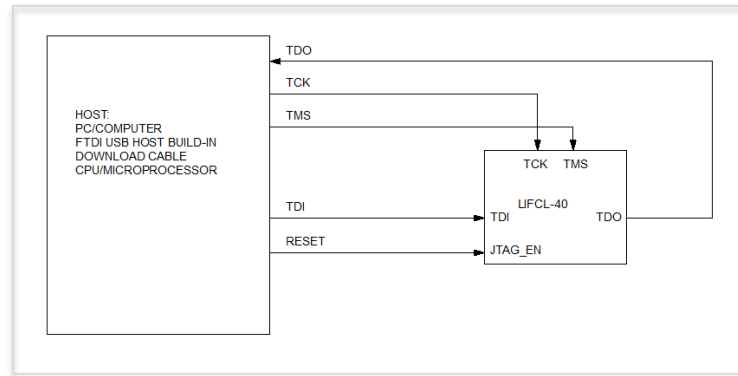
※CrossLink-NXの評価ボードにはUSB DownloadケーブルのBox部とFlywire部に該当する部分が搭載されているため、一般的なMini USB type-B ケーブルで基板⇄PC間を接続します。

4.2. SRAM領域への書き込み 1/3

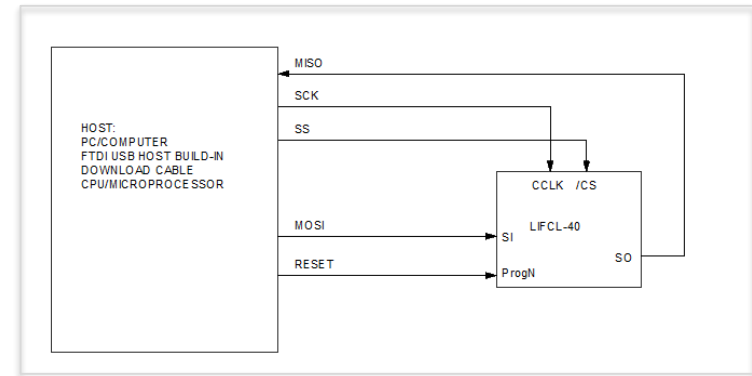


■ Downloadケーブルとデバイス間の接続

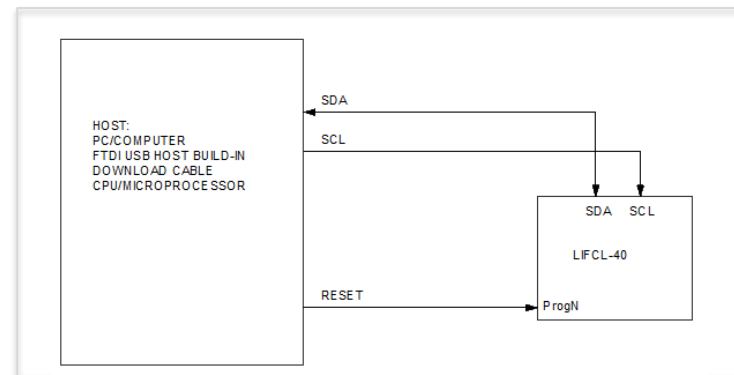
SRAM領域への直接書き込みには**JTAG/SPI/I2C**の3つのI/Fをサポートしています。
Downloadケーブルとデバイス間はそれぞれ以下の図のように接続します。



JTAG I/F使用時



SPI I/F使用時

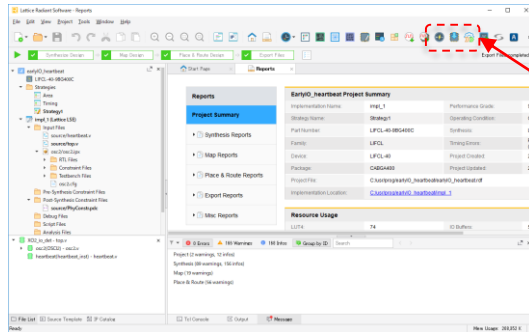


I2C I/F使用時

4.2. SRAM領域への書き込み 2/3

■ SRAM(揮発性)領域に直接Programmingする場合のオペレーション

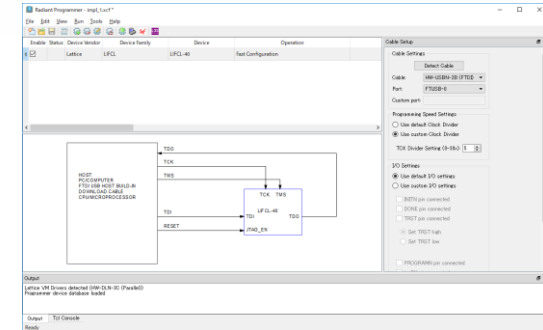
1. RadiantのツールバーからProgrammerを起動



クリック



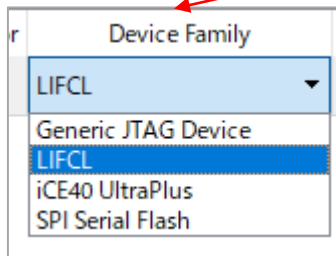
Programmer起動



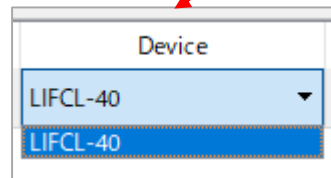
2. Device Family、Device、Operation、書き込みファイルを設定



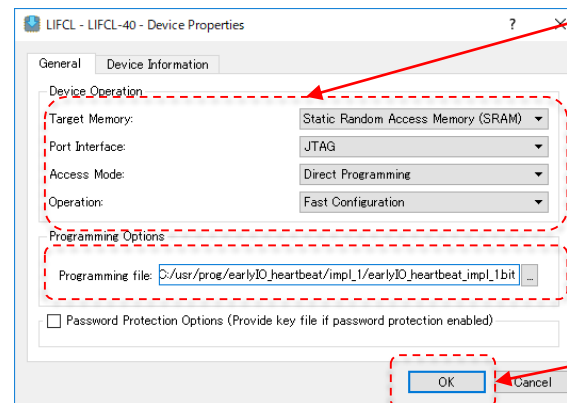
ダブルクリック



① LIFCLを選択



② 使用するデバイスを選択



③ Device Operationを以下に設定

Target Memory:
Static Random Access Memory (SRAM)

Port Interface:
JTAG / Slave SPI / I2C から選択

Access Mode: Direct Programming

Operation: Fast Configuration

④ 書き込むコンフィグファイル(*.bit)を設定

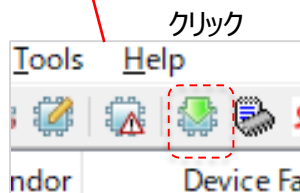
⑤ OKをクリック

4.2. SRAM領域への書き込み 3/3

■ SRAM(揮発性)領域に直接Programmingする場合のオペレーション

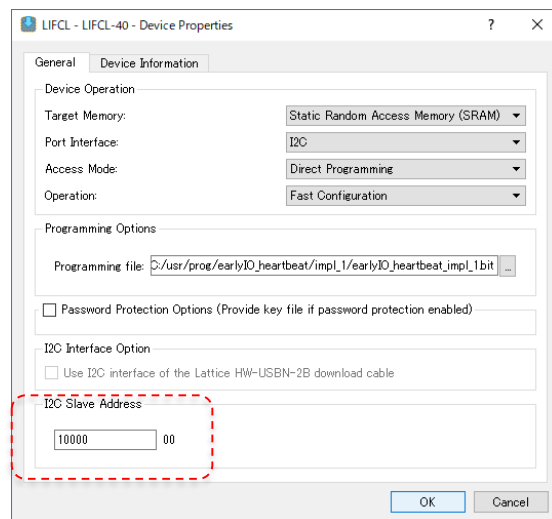
3. Program Deviceをクリックして書き込み実行

	Enable	Status	Device Vendor	Device Family	Device	Operation
1	☒		Lattice	LIFCL	LIFCL-40	Fast Configuration



4. StatusがDoneとなり書き込み完了

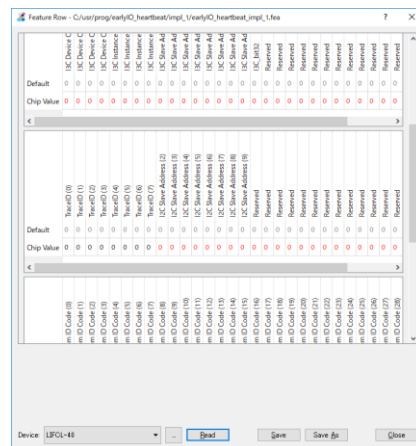
※I2C I/F使用時はDevice PropertiesダイアログにてI2C Slave Addressの設定も行ってください



<DefaultのI2C Slave Address>

7bit addressing: 1000000
10bit addressing: 1111000000

Slave AddressはFeature Rowに定義されており、Radiantに付属している "Programming File Utility" の "Feature Row Editor" で変更可能

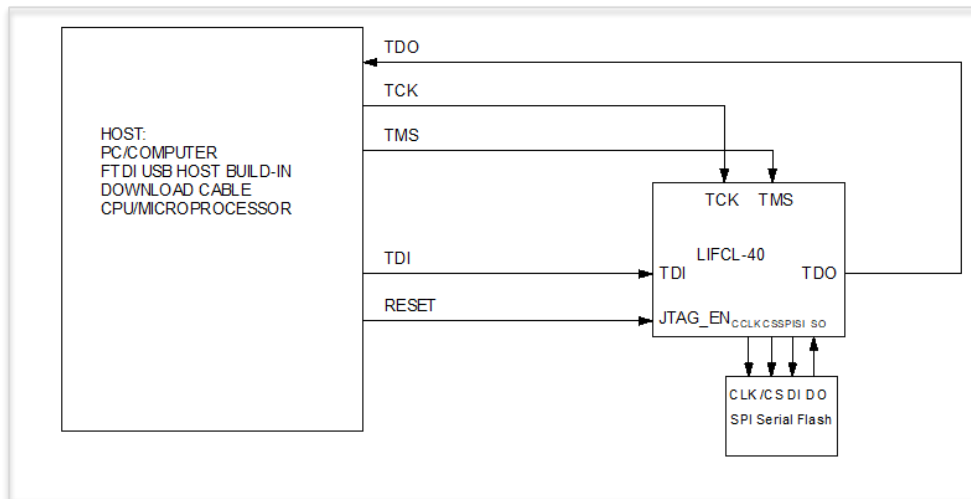


4.3. FPGA経由でのSPI FLASHへの書き込み 1/2

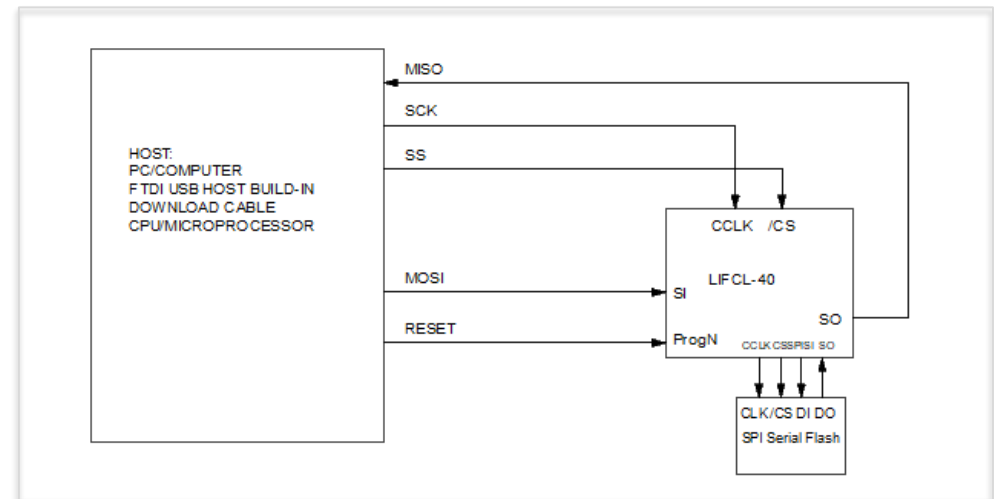


■ Downloadケーブルとデバイス間の接続

FPGAを経由した外部SPI Flashへの書き込みには**JTAG/SPI**の2つのI/Fをサポートしています。
Downloadケーブルとデバイス間およびデバイス⇔SPI Flash間はそれぞれ以下の図のように接続します。



JTAG I/F使用時



SPI I/F使用時

4.3. FPGA経由でのSPI FLASHへの書き込み 2/2

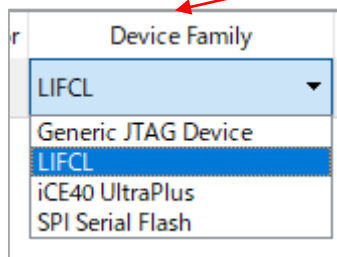
■ CrossLink-NXを経由して外部SPI Flashへ書き込みする場合のオペレーション

1. RadiantのツールバーからProgrammerを起動

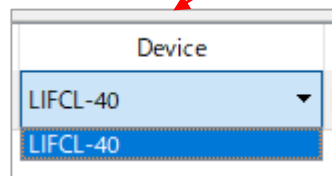


2. Device Family、Device、Operation、書き込みファイルを設定

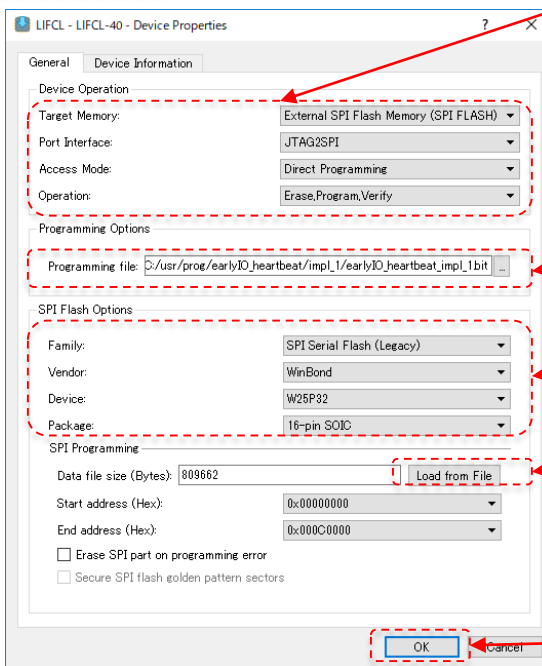
Enable	Status	Device Vendor	Device Family	Device	Operation
1	☒	Lattice	LIFCL	LIFCL-40	Erase, Program, Verify



① LIFCLを選択



② 使用するデバイスを選択



③ Device Operationを以下に設定

Target Memory:
External SPI Flash Memory (SPI Flash)

Port Interface:
JTAG2SPI か SSPI2SPI を選択

Access Mode: Direct Programming

Operation: Erase, Program, Verify

④ 書き込むコンフィグファイル(*.bit)を設定

⑤ 使用するSPI Flashを設定

⑥ Load from Fileをクリックして書き込みファイルサイズを自動算出

⑦ OKをクリック

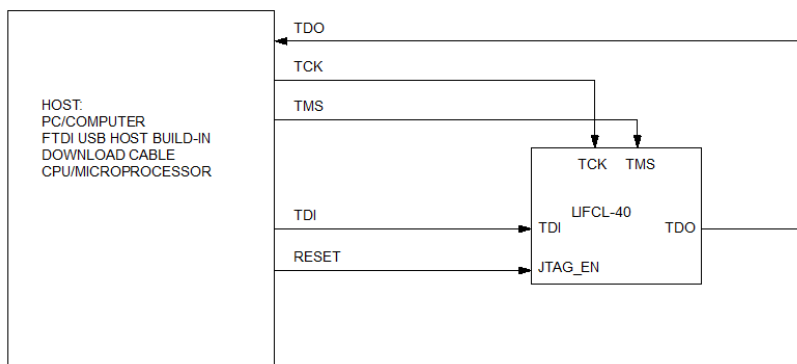
3 以降の手順はSRAM領域への書き込みと同じ。

4.4. Feature Rowへの書き込み 1/4

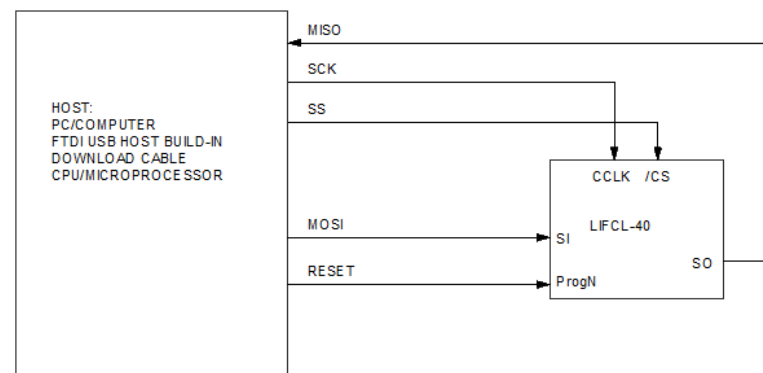


■ Downloadケーブルとデバイス間の接続

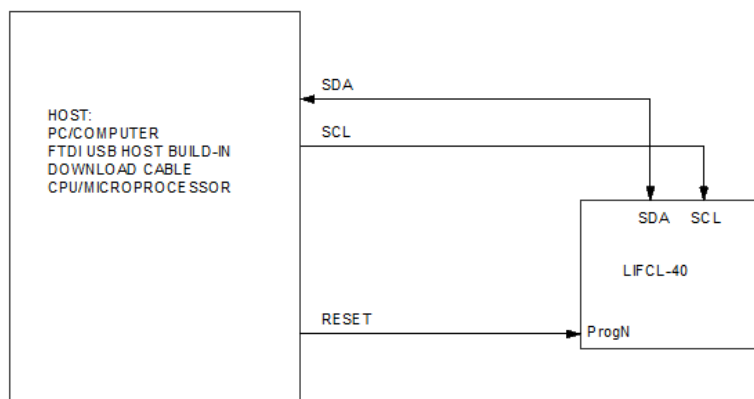
Feature Rowへの書き込みには**JTAG/SPI/I2C**の3つのI/Fをサポートしています。
Downloadケーブルとデバイス間はそれぞれ以下の図のように接続します。



JTAG I/F使用時



SPI I/F使用時



I2C I/F使用時

4.4. Feature Rowへの書き込み 2/4

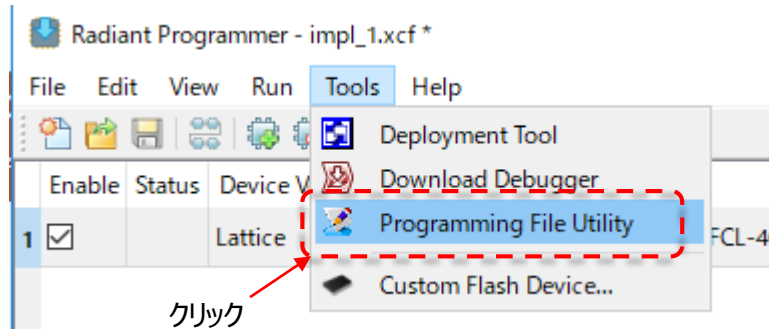


■ Feature Rowファイルの作成・編集

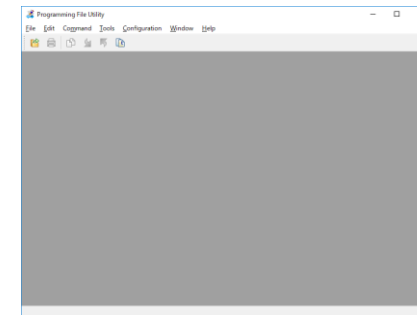
Feature Rowへの書き込む前に、Feature Rowファイル(*.fea)を作成しておく必要があります。Feature RowファイルはRadiantでCrossLink-NXのコンフィグレーションファイル(*.bit)を生成すると同じフォルダ内に生成されます。

Feature Rowファイルを編集したい場合は、Radiant ProgrammerからFeature Row Editorを使用します。

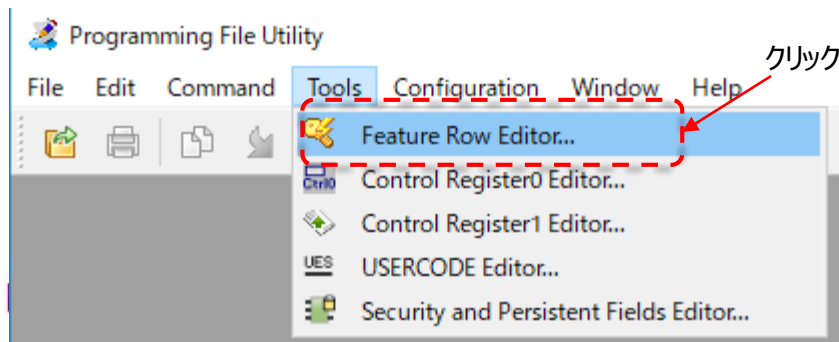
1. ProgrammerからProgramming File Utilityを起動



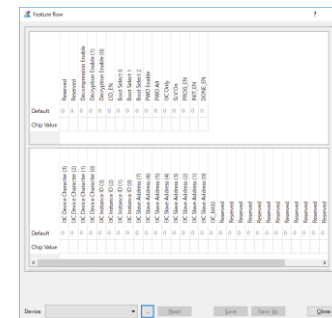
Programming File Utility起動



2. Programming File UtilityからFeature Row Editorを起動

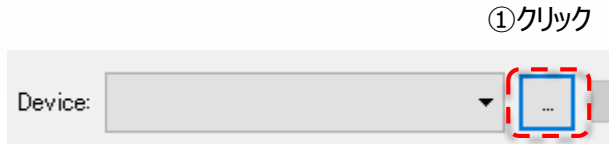
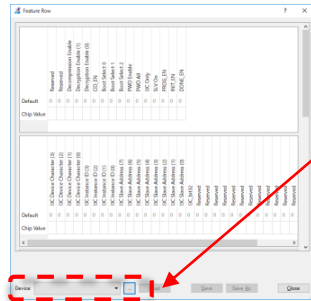


Feature Row Editor起動



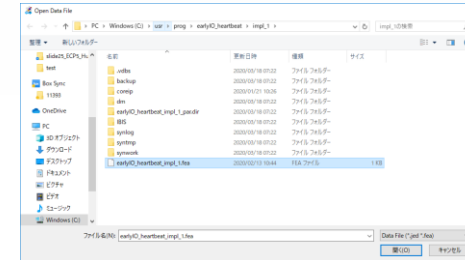
4.4. Feature Rowへの書き込み 3/4

3. Feature Row Editorで内容編集・保存



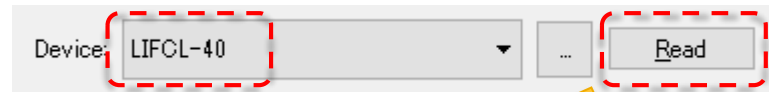
①クリック

②feaファイルを選択して開く

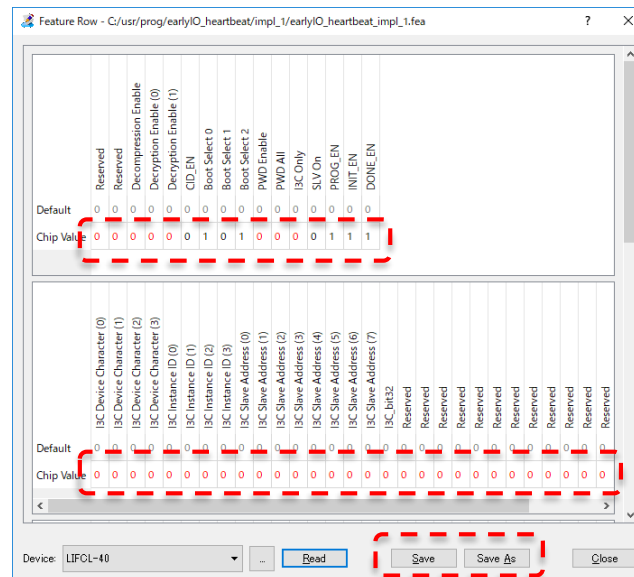


③デバイス名が表示される

④Readをクリック



⑤Feature Rowの値が読み込まれる



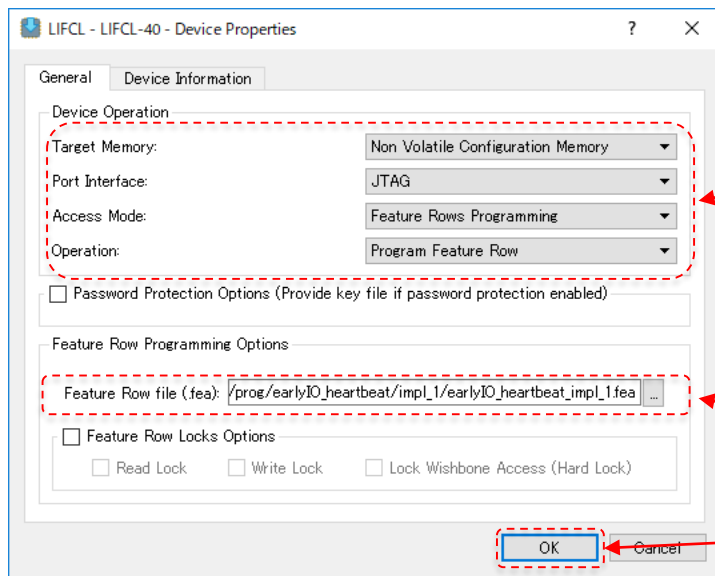
⑥値を編集してSave もしくは
Save Asでfeaファイルを保存

4.4. Feature Rowへの書き込み 4/4

4. ProgrammerでFeature Rowの書き込み

	Enable	Status	Device Vendor	Device Family	Device	Operation
1	☒		Lattice	LIFCL	LIFCL-40	Program Feature Row

①ダブルクリック



②Device Operationを以下に設定

Target Memory:
Non Volatile Configuration Memory

Port Interface:
JTAG / Slave SPI / I2C のいずれかを選択

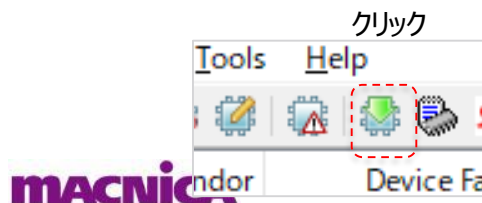
Access Mode: Feature Rows Programming

Operation: Erase, Program, Verify

③書き込むFeature Rowファイル(*.fea)を設定

④OKをクリック

5. Program Deviceをクリックして書き込み実行



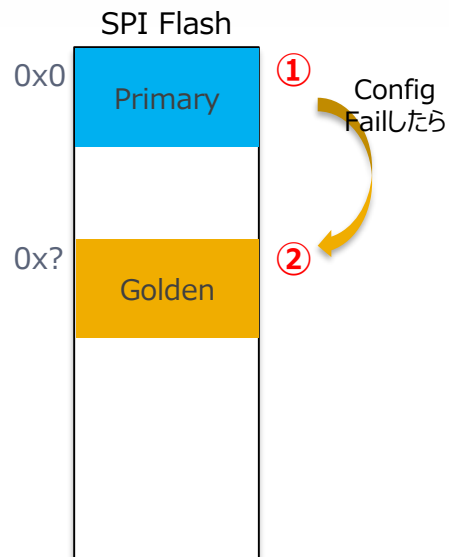
6. StatusがDoneとなり書き込み完了



4.5. サポートしている特殊Boot Mode

■ CrossLink-NXは以下の3つの特殊なBoot Modeをサポートしています

1. Dual Boot

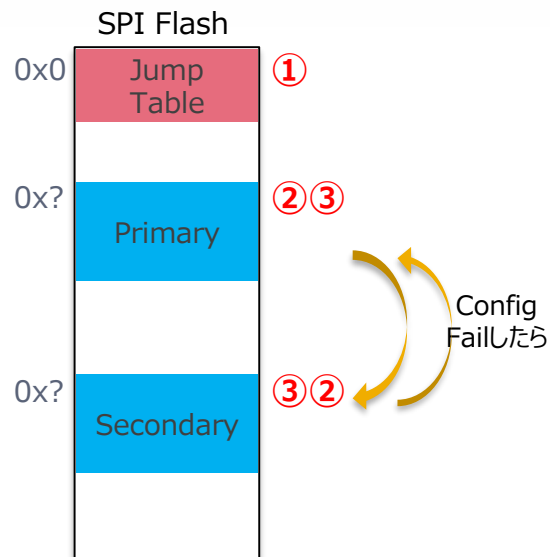


コンフィグデータは2つ

Config Data	保存先Address
Primary Pattern	0x0
Golden Pattern	Userが定義

Primary Patternでのコンフィグに失敗した場合に、自動的にGolden Patternでのコンフィグレーションを行う

2. Ping-Pong Boot



コンフィグデータは2つ

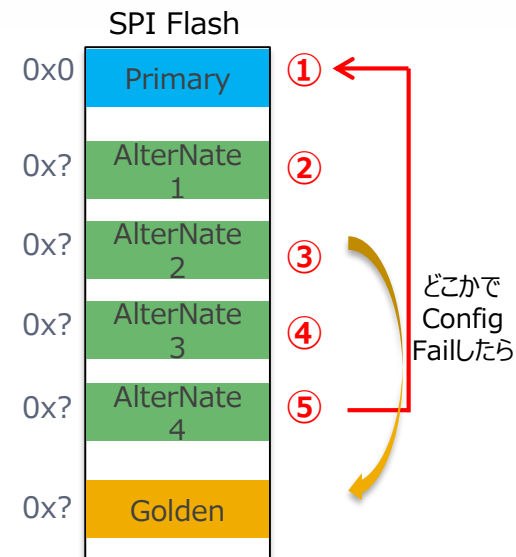
コンフィグデータとは別にJump Tableを定義

Config Data	保存先Address
Jump Table	0x0
Primary Pattern	Userが定義
Secondary Pattern	Userが定義

Jump Tableでどちらのパターンを先にコンフィグするかを定義する。

First Boot Patternに指定されたコンフィグデータでコンフィグし、失敗した場合に、自動的にもう一方のコンフィグデータでコンフィグレーションを行う

3. Multi Boot



コンフィグデータは3つから最大6つまで定義可能

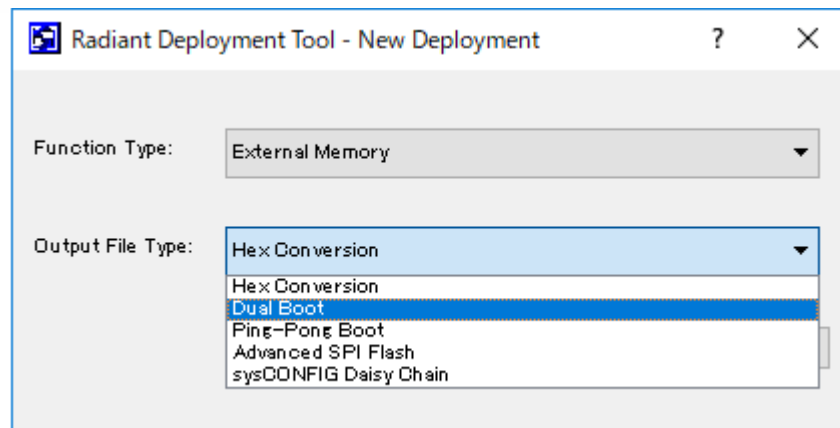
Config Data	保存先Address
Primary Pattern	0x0
Golden Pattern	Userが定義
Alternate Pattern1~4	Userが定義

まずPrimary Patternでコンフィグを行う。
その後、Programnのトグル or Refreshコマンドの発行があれば、Alternate Patternでのコンフィグを行う。
(Alternate Patternは最大4つまで定義可能)
いずれかのデータでのコンフィグに失敗したら自動的にGolden Patternでのコンフィグレーションを行う

4.6. Dual Boot 1/2

■ Dual Boot Fileの生成

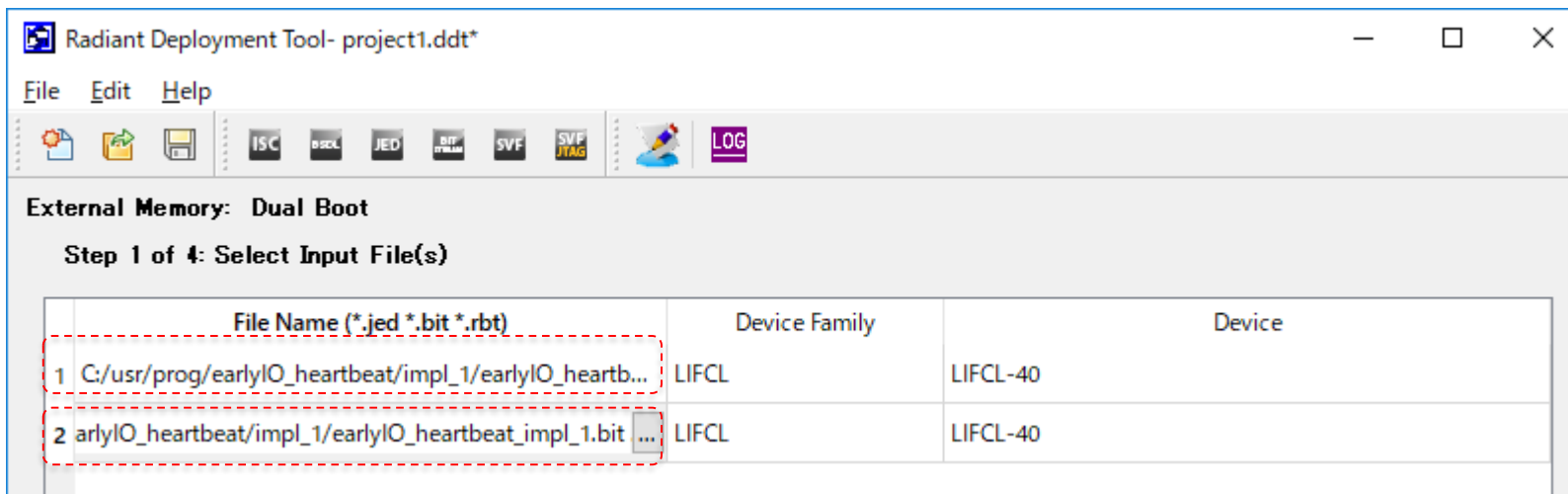
1. ProgrammerのTools→Deployment Toolを選択し、Deployment Toolを起動



Function: **External Memory** を選択

Output File Type: **Dual Boot** を選択

2. 書き込む2種類のbitファイルを選択し、Nextをクリック

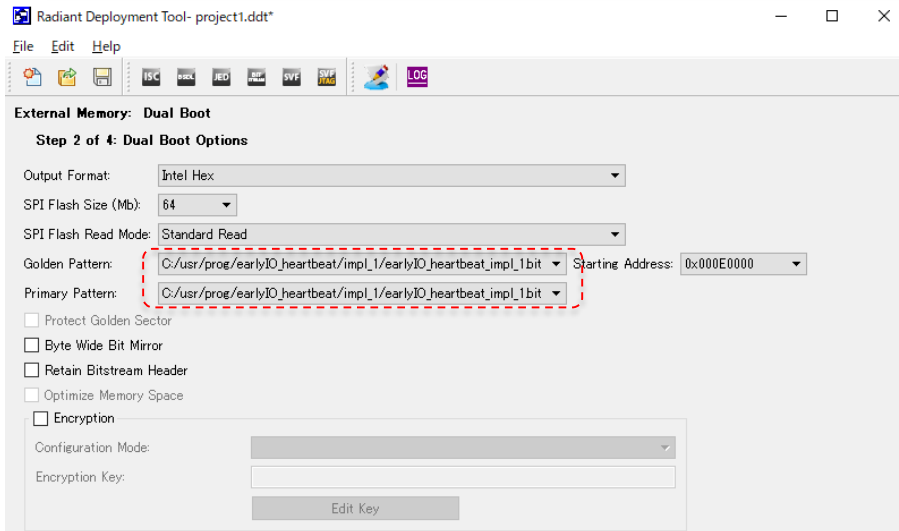


4.6. Dual Boot 2/2

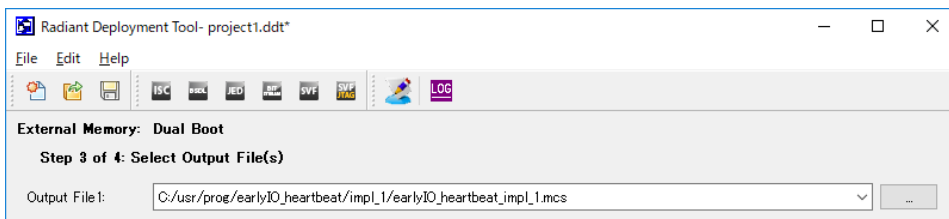
NOWHERE,
but HERE.

■ Dual Boot Fileの生成

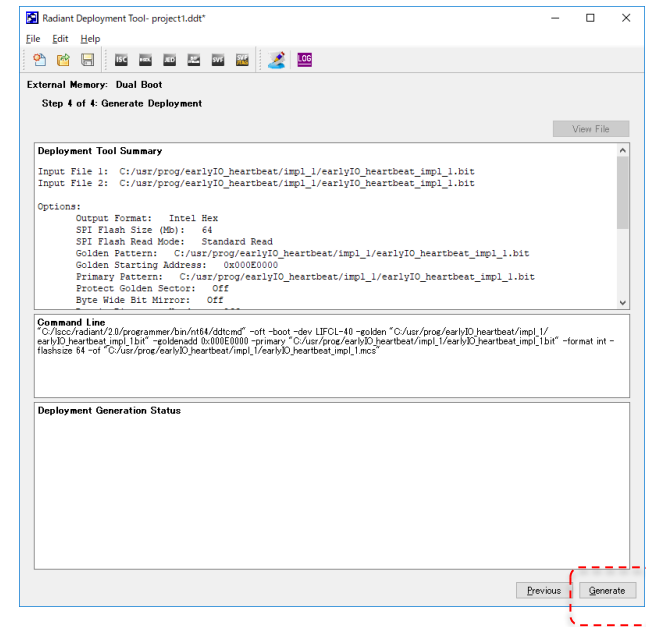
3. Golden / Primary Pattern等の設定を行い、Nextをクリック



4. 生成するファイル(*.mcs)の生成先を指定してNextをクリック



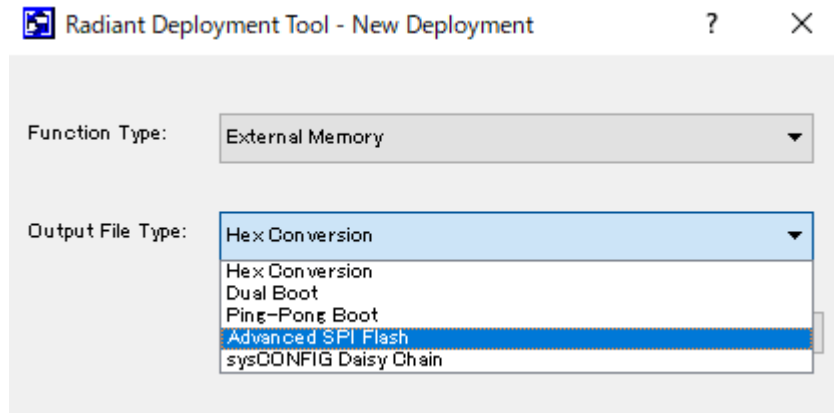
5. Generateをクリックしてファイルを生成



4.7. Multi Boot 1/2

■ Multi Boot Fileの生成

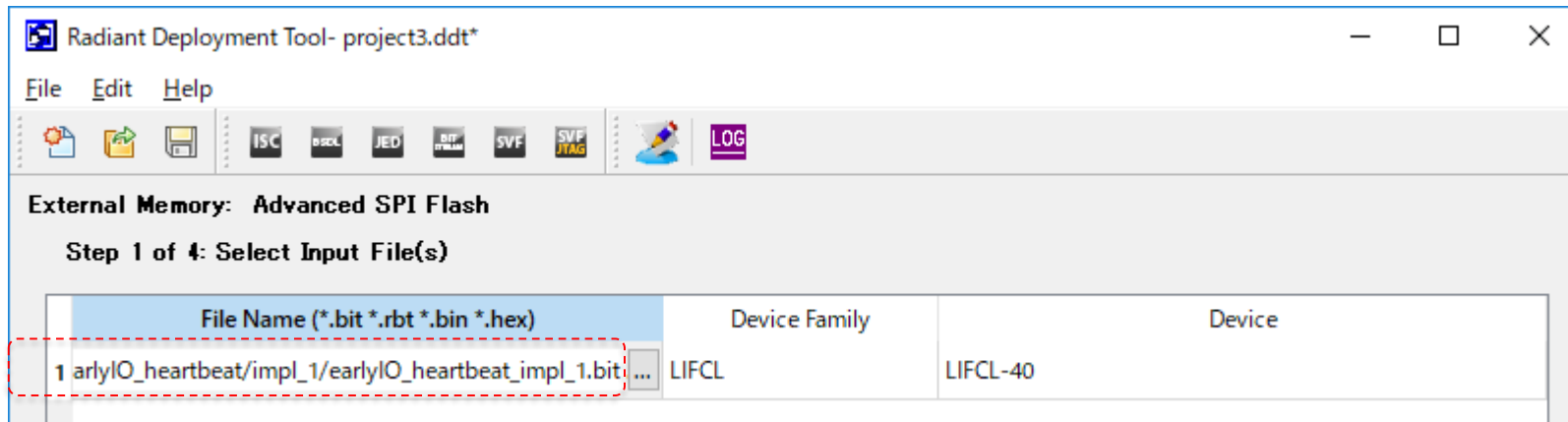
1. ProgrammerのTools→Deployment Toolを選択し、Deployment Toolを起動



Function: **External Memory** を選択

Output File Type: **Advanced SPI Flash** を選択

2. Primary Pattern(最初にコンフィグするデータ)に指定するbitファイルを選択し、Nextをクリック



4.7. Multi Boot 2/2



■ Multi Boot Fileの生成

3. Golden / Alternate Pattern等の設定を行い、Nextをクリック

Radiant Deployment Tool - project5.ddt*

File Edit Help

External Memory: Advanced SPI Flash

Step 2 of 4: Advanced SPI Flash Options

Options User Data Files Multiple Boot

☒ Multiple Boot

Golden Pattern: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.bit

Starting Address: 0x000E0000

☐ Protect Golden Sector

Number of Alternate Patterns: 4

Alternate Pattern 1: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.bit

Starting Address: 0x001C0000

Next Alternate Pattern to Configure: Alternate Pattern 2

Alternate Pattern 2: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.bit

Starting Address: 0x002B0000

Next Alternate Pattern to Configure: Alternate Pattern 3

Alternate Pattern 3: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.bit

Starting Address: 0x003A0000

Next Alternate Pattern to Configure: Alternate Pattern 4

Alternate Pattern 4: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.bit

Starting Address: 0x00490000

Next Alternate Pattern to Configure: Primary Pattern

Previous Next

4. 生成するファイル(*.mcs)の生成先を指定してNextをクリック

Radiant Deployment Tool - project1.ddt*

File Edit Help

External Memory: Dual Boot

Step 3 of 4: Select Output File(s)

Output File: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.mcs

5. Generateをクリックしてファイルを生成

Radiant Deployment Tool - project1.ddt*

File Edit Help

External Memory: Dual Boot

Step 4 of 4: Generate Deployment

View File

Deployment Tool Summary

Input File 1: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.bit

Input File 2: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.bit

Options:

Output Format: Intel Hex

SPI Flash Size (MB): 64

SPI Flash Read Mode: Standard Read

Golden Pattern: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.bit

Golden Starting Address: 0x000E0000

Primary Pattern: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.bit

Protect Golden Sector: Off

Byte Wide Bit Mirror: Off

Command Line

C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.mcs

Deployment Generation Status

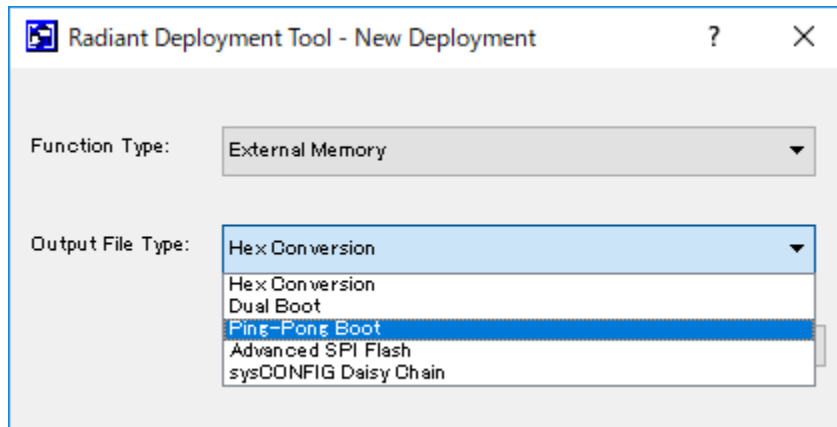
Previous Generate

4.8. Ping-Pong Boot 1/2



■ Ping Pong Boot Fileの生成

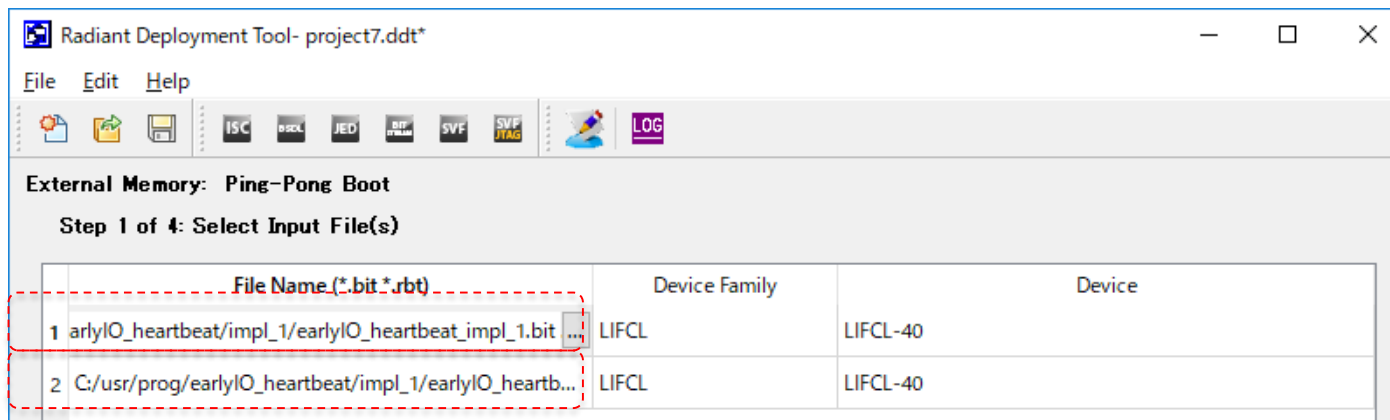
1. ProgrammerのTools→Deployment Toolを選択し、Deployment Toolを起動



Function: **External Memory** を選択

Output File Type: **Ping-Pong Boot** を選択

2. 2つのbitファイルを設定し、Nextをクリック（※ファイルは後でも再設定可能）



4.8. Ping-Pong Boot 2/2

■ Ping-Pong Boot Fileの生成

3. Primary / Secondary Pattern等の設定を行い、Nextをクリック

External Memory: Ping-Pong Boot

Step 2 of 4: Ping-pong Boot Options

Output Format: Intel Hex

SPI Flash Size (Mb): 64

SPI Flash Read Mode: Standard Read

Primary Pattern: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.bit Starting Address: 0x00010000

Secondary Pattern: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.bit Starting Address: 0x000E0000

First Boot Selection: Primary Pattern

☐ Generate Jump Table Only

☐ Byte Wide Bit Mirror

☐ Retain Bitstream Header

各パターンの先頭アドレスの設定

先にコンフィグするパターンの指定

Jump Tableだけを生成したい場合にチェックを付ける

4. 生成するファイル(*.mcs)の生成先を指定してNextをクリック

External Memory: Ping-Pong Boot

Step 3 of 4: Select Output File(s)

Output File: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.mcs

5. Generateをクリックしてファイルを生成

External Memory: Ping-Pong Boot

Step 4 of 4: Generate Deployment

Deployment Tool Summary

Input File 1: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.bit

Input File 2: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.bit

Options:

Output Format: Intel Hex

SPI Flash Size (Mb): 64

SPI Flash Read Mode: Standard Read

Primary Pattern: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.bit

Primary Starting Address: 0x00010000

Secondary Pattern: C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.bit

Secondary Starting Address: 0x000E0000

Byte Wide Bit Mirror: Off

Command Line

C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.mcs -primary C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.mcs -secondary C:/usr/prog/earlyIO_heartbeat/impl_1/earlyIO_heartbeat_impl_1.mcs

Deployment Generation Status

Previous Generate

Revision History



■ Revision History

Rev	Summary	UPDATE Date
Rev 1.0	Initial release	2020/06/23
Rev 1.1	誤記の修正	2020/06/29