

SoC はじめてガイド  
HPS-FPGA 間のアクセス方法  
(Arria<sup>®</sup> V SoC / Cyclone<sup>®</sup> V SoC 編)

Ver.17

# SoC はじめてガイド

## HPS-FPGA 間のアクセス方法

### (Arria® V SoC / Cyclone® V SoC 編)

## 目次

1. はじめに .....	3
2. HPS-FPGA 間のインタフェース .....	4
3. アドレス・マップ .....	6
4. ハードウェアの設定 .....	8
4-1. HPS コンポーネントの設定 (AXI Bridges / FPGA-to-HPS SDRAM Interface) .....	8
4-1-1. FPGA-to-HPS インタフェース (F2H) .....	9
4-1-2. HPS-to-FPGA インタフェース (H2F) .....	10
4-1-3. Lightweight HPS-to-FPGA インタフェース (LWH2F) .....	11
4-1-4. FPGA-to-HPS SDRAM インタフェース (F2S) .....	12
4-1-5. F2S のポート数 .....	13
4-1-6. SDRAM コントローラの優先度 .....	14
4-2. 他のコンポーネントとの接続 .....	15
4-3. インタフェースの初期化 .....	16
4-3-1. F2H, H2F, LWH2F の初期化 .....	16
4-3-2. F2S の初期化 .....	17
5. ソフトウェアからのアクセス .....	18
5-1. アドレスの考え方 .....	18
5-2. システム・ヘッダ・ファイル .....	20
6. 参考情報 .....	21
改版履歴 .....	22

## 1. はじめに

この「SoC はじめてガイド」シリーズは、インテル® SoC FPGA を使用する初心者ユーザ向けの技術コンテンツです。

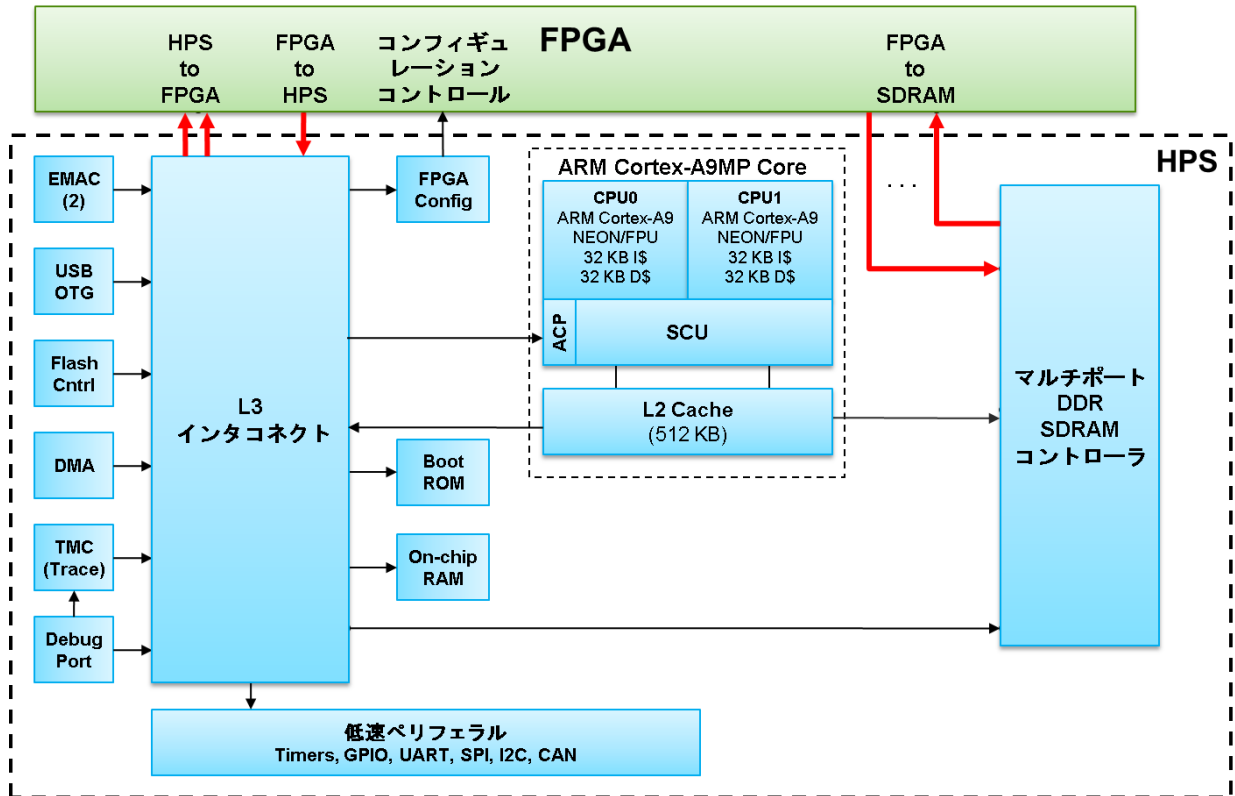
インテル® SoC FPGA は、ARM® Cortex®-A9 MPCore™ プロセッサと FPGA とがデバイス内でバス接続されています。デバイス内で接続されているため、プロセッサと FPGA 間の転送レート不足の解消や、基板上の実装面積の縮小が期待されます。

この資料では、インテル® Arria® V SoC FPGA や Cyclone® V SoC FPGA における、Hard Processor System (HPS) と FPGA 間のアクセス方法を説明します。

なお、この資料内では特に断りが無い場合、Arria® SoC FPGA や Cyclone® V SoC のことをインテル® SoC FPGA と呼ぶことにします。

## 2. HPS-FPGA 間のインタフェース

インテル® SoC FPGA の内部構造は、下図の通りです。図の上方の緑色部分が FPGA ファブリックで、その下方の水色部分を含む破線部分内が HPS です。



インテル® SoC FPGA には、FPGA と HPS との間に 4 つのインタフェースがあります。

HPS 側がバス・マスタとなって FPGA 側にアクセスするインタフェースとしては、最大 128bit 幅の高帯域な HPS-to-FPGA (H2F) インタフェースと、32bit 幅の Lightweight HPS-to-FPGA (LWH2F) インタフェースがあります。H2F は、HPS と FPGA 間の比較的大容量のデータを転送するインタフェースです。LWH2F は、制御信号など比較的低速でのアクセスに適しています。HPS からは、これらのインタフェースは、後述するアドレス・マップ上でメモリ・マップト I/O としてアクセスできます。

FPGA 側がバス・マスタとなって HPS 側にアクセスするインタフェースとしては、最大 128bit の FPGA-to-HPS (F2H) インタフェースがあります。

また、FPGA 側から HPS 側にある SDRAM コントローラに直接アクセスするインタフェースとしては、FPGA-to-SDRAM (F2S) があります。

各インタフェースの概要を次ページの表に示します。

名称	HPS to FPGA Bridge	Lightweight HPS to FPGA Bridge	FPGA to HPS Bridge	FPGA to HPS SDRAM Interface
略称	H2F	LWH2F	F2H	F2S
機能概要	HPS がバス・マスタとなり FPGA 側にアクセスするインタフェース	HPS がバス・マスタとなり FPGA 側にアクセスするインタフェース	FPGA がバス・マスタとなり HPS 側にアクセスするインタフェース	FPGA がバス・マスタとなり HPS 側の SDRAM へ直接アクセスするインタフェース
バス幅	32/64/128bit	32bit	32/64/128bit	32/64/128/256 bit
空間サイズ	960MB	2MB	4GB	4GB
I/F タイプ	AXI3™	AXI3™	AXI3™	AXI3™ / Avalon®-MM
最大ポート数	1	1	1	AXI3™:3 Avalon®-MM:6
主な使用方法	FPGA 側とのデータ転送(広帯域を必要とする大きなサイズのデータを扱う場合)	FPGA 側のコントロール・ステータス・レジスタ(CSR)の読み書き(H2F のデータ・トラフィックに影響されずに低遅延でアクセスしたい場合)	HPS 側とのデータ転送および CSR 読み書き	HPS 側の SDRAM に対する高速アクセス(HPS 内のメイン・スイッチを介さずに SDRAM へアクセスしたい場合)

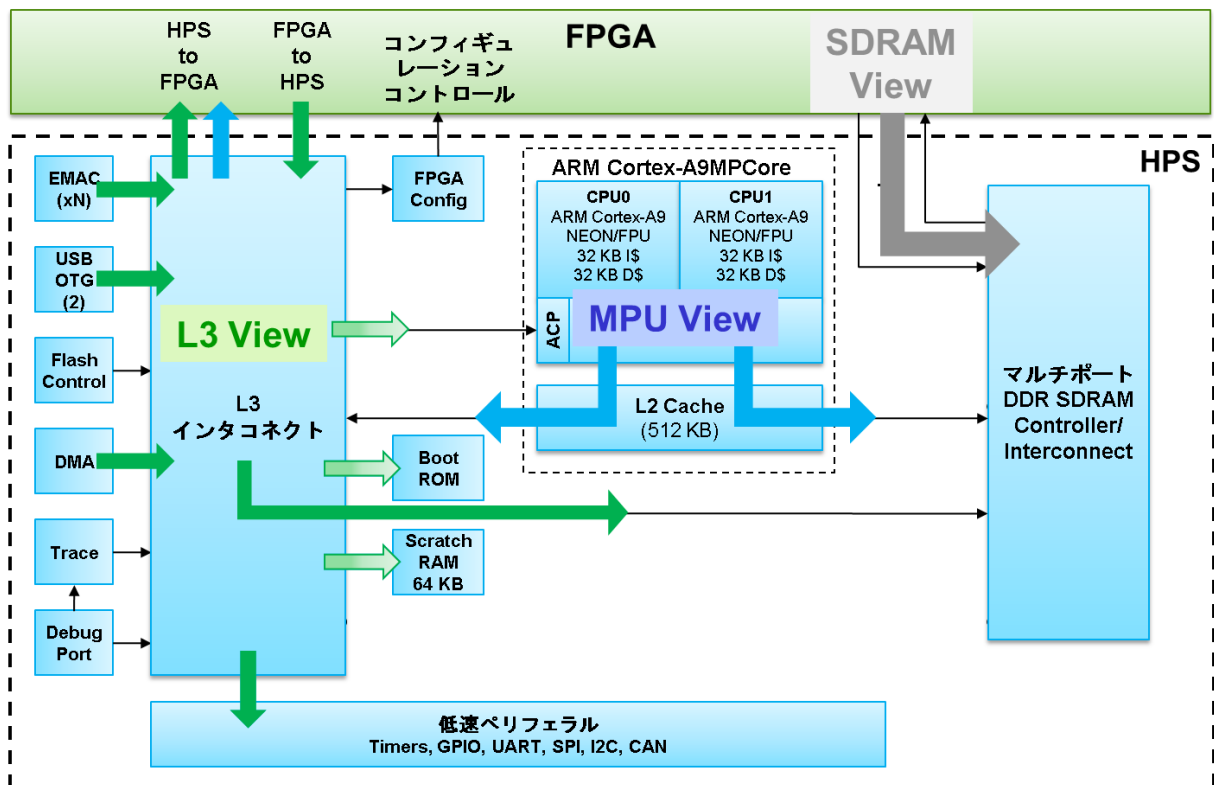
インテル® Qsys システム統合ツールがバス・プロトコルの相互変換を行うためのインタコネクト回路を自動生成するため、Avalon® インタフェースのコンポーネントを I/F タイプが AXI3™ の Bridge/Interface へ接続することも可能です。このため、今までの設計資産を流用することが可能です

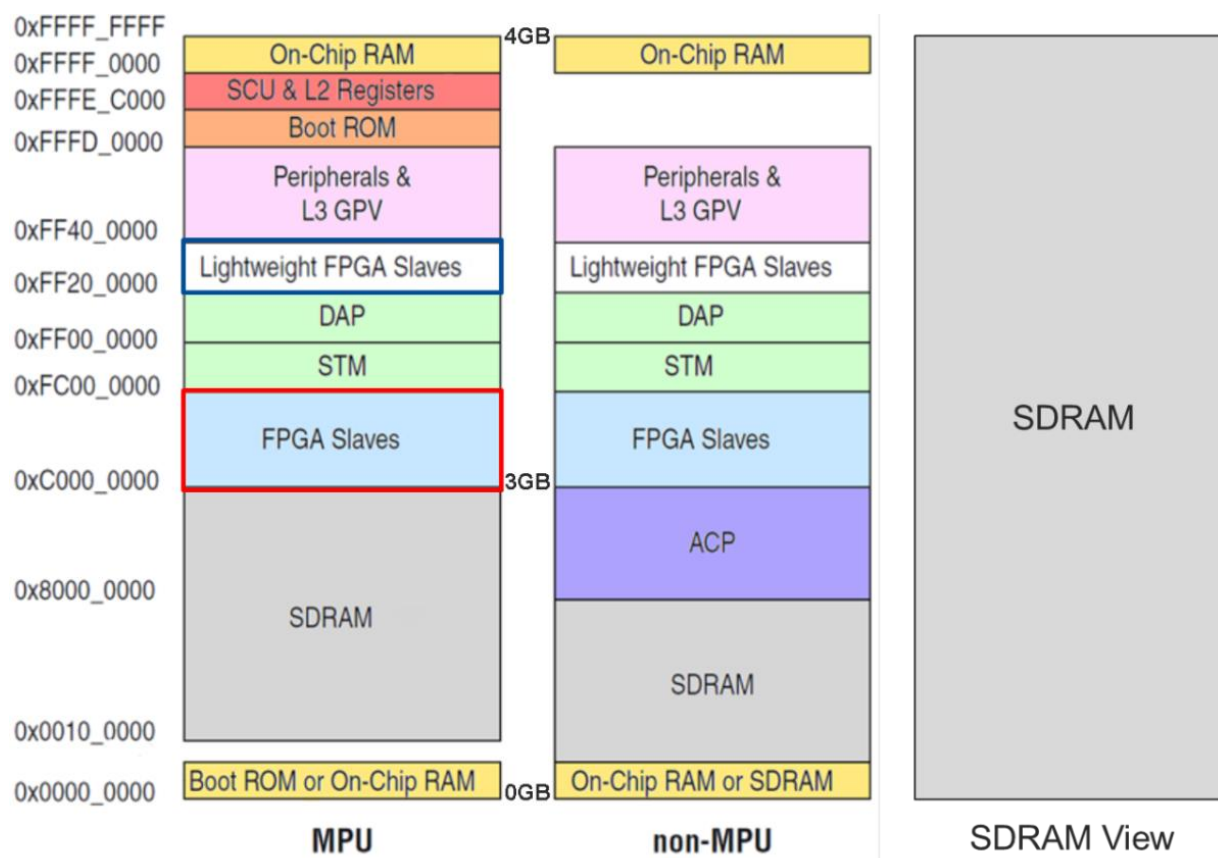
### 3. アドレス・マップ

ARM® プロセッサから見た FPGA は、メモリ・マップト・デバイスとしてアクセスすることができます。アドレス・マップは、「どこからアクセスをするか」によって、以下の 3 種類の概念に分かれます。

- ARM® プロセッサからアクセスする場合に使用する「MPU ビュー」
- L3 インタコネクต์に接続されたマスタから使用する「L3 ビュー」もしくは「non-MPU ビュー」(F2H に接続された FPGA 側のマスタ・デバイスも L3(non-MPU) ビューの概念が適用されます。)
- F2S に接続されたマスタから使用する「SDRAM ビュー」

各ビューの概念を下図に示します。





上図が、3種類の各ビューに対するアドレス・マップとなります。

- MPU ビューと L3(non-MPU)ビューには、H2F、LWH2F アクセス用に固定の領域が割り当てられます。
- HPS 側から H2F ブリッジを介して FPGA にアクセスする際は、0xC000\_0000~ 番地にアクセスします。
- HPS 側から LWH2F ブリッジを介して FPGA にアクセスする際は、0xFF20\_0000~ 番地にアクセスします。

ARM® プロセッサのソフトウェアから FPGA にアクセスする方法は、「5 ソフトウェアからのアクセス」をご参照ください。

L3 ビュー(non-MPU ビュー)の 0x8000\_0000~ にマッピングされているのは、ACP (Accelerator Coherency Port) です。このポートを介してアクセスをすると、L3 に接続されたマスタからとプロセッサとの間でコヒーレントなアクセスを行うことができます。

ACP を介したアクセスは、インテル® FPGA の Web ページにあるテクニカル・リファレンス・マニュアル(使用するデバイス・ファミリ(Arria® V / Cyclone® V)用の Hard Processor System Technical Reference Manual)の ACP に関する記述をご参照ください。

## 4. ハードウェアの設定

ハードウェアの設定は、Qsys によって行います。設定は、「HPS コンポーネントの設定」、「HPS と他のコンポーネントとの接続」および「インタフェースの初期化」の3つが必要です。

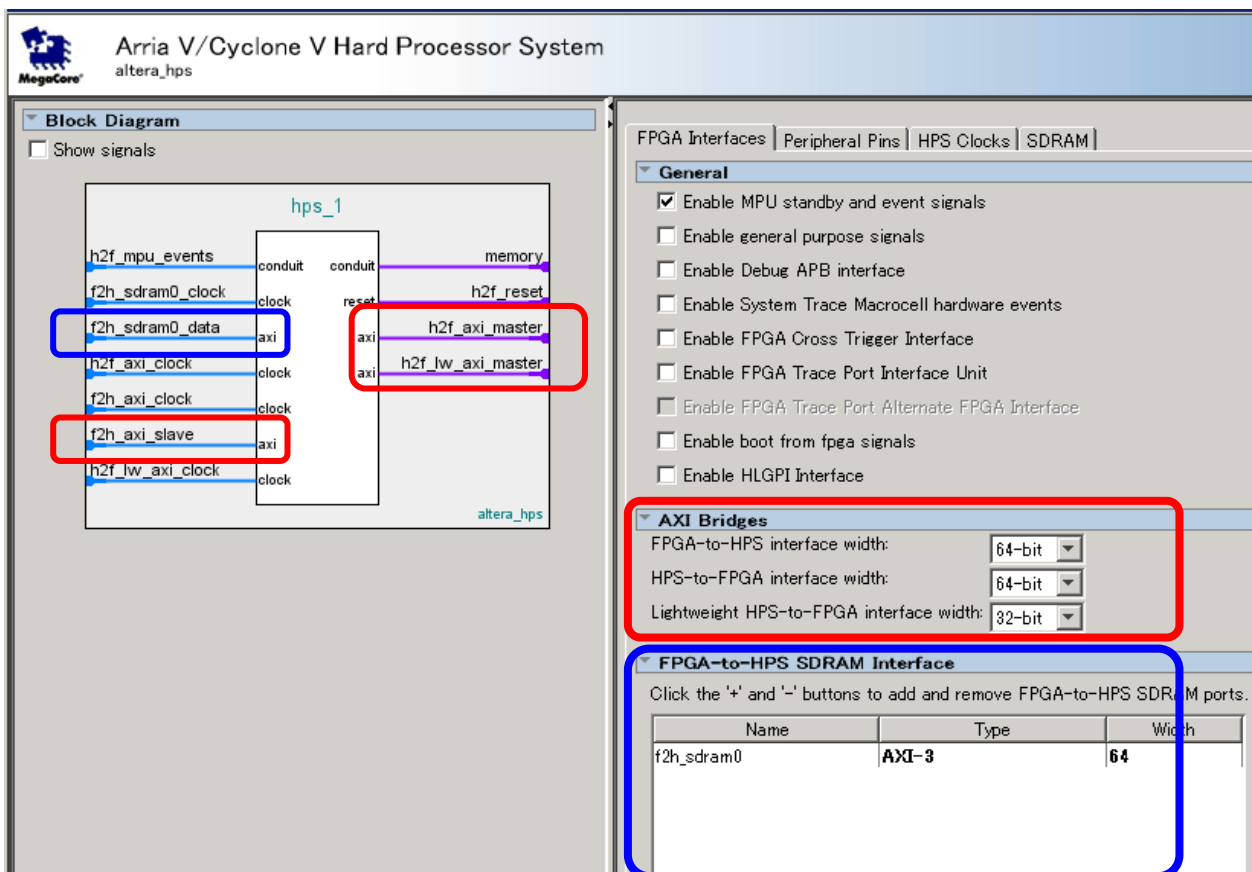
### 4-1. HPS コンポーネントの設定 (AXI Bridges / FPGA-to-HPS SDRAM Interface)

HPS コンポーネントの設定を Qsys の GUI にて行います。

FPGA Interfaces タブの AXI Bridges (下図赤枠) および FPGA-to-HPS SDRAM Interface (下図青枠) が、該当のオプションとなります。

次に、AXI Bridges の設定では、F2H, H2F, LWH2F の各インタフェースに対して、データ・バス幅および使用の有無を選択します。

そして、FPGA-to-HPS SDRAM Interface の設定では、F2S インタフェースで利用するポートを 0 個 (未使用) から条件により最大 6 個まで複数のポートを登録できます。各ポートには、バスの種類、およびデータ・バス幅を選択します。F2S に登録可能なポート数の詳細は「4-1-5. F2S のポート数」をご参照ください。



The screenshot displays the Qsys GUI for the Arria V/Cyclone V Hard Processor System. The left pane shows a block diagram of the hps\_1 component with various interfaces connected. The right pane shows the configuration options for these interfaces.

**AXI Bridges Configuration:**

- FPGA-to-HPS interface width: 64-bit
- HPS-to-FPGA interface width: 64-bit
- Lightweight HPS-to-FPGA interface width: 32-bit

**FPGA-to-HPS SDRAM Interface Configuration:**

Name	Type	Width
f2h_sdrām0	AXI-3	64

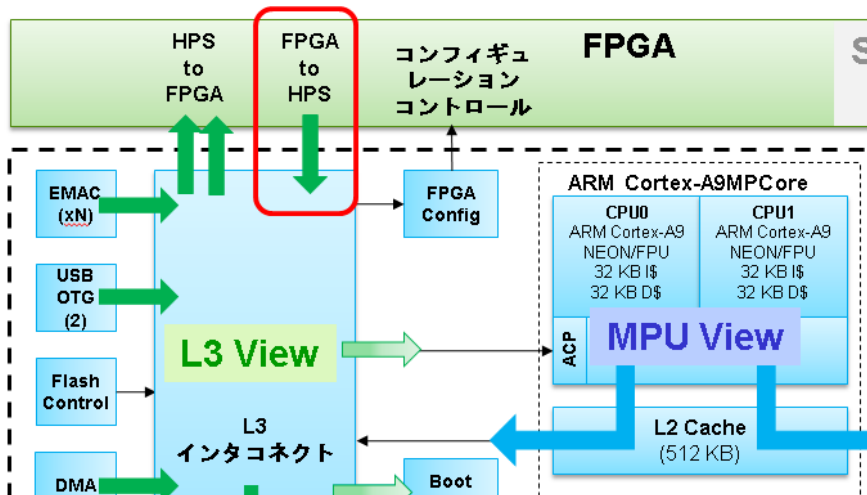
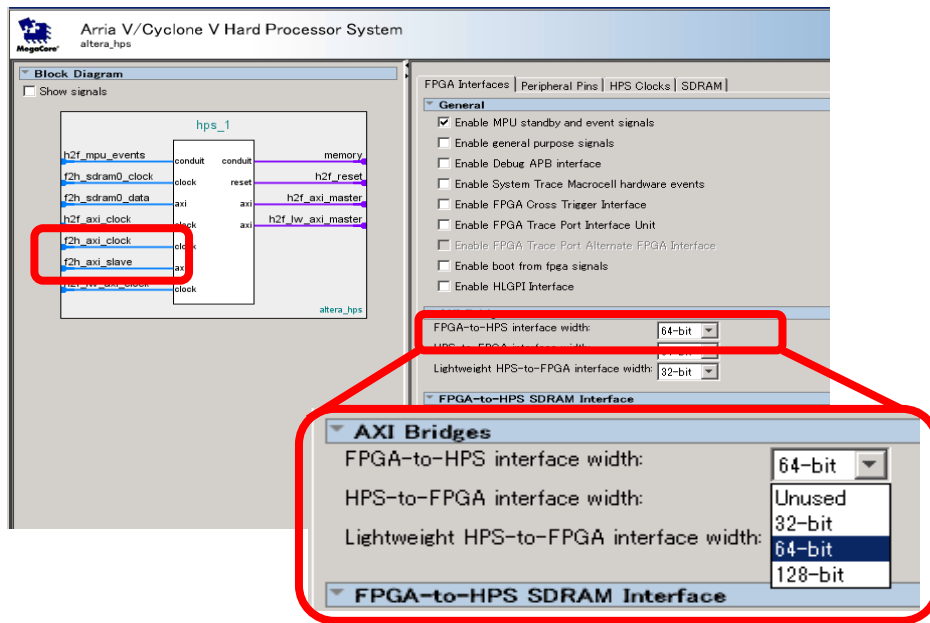


### 4-1-1. FPGA-to-HPS インタフェース (F2H)

FPGA-to-HPS インタフェースは、FPGA 内にあるバス・マスタが、HPS 内のペリフェラルやメモリにアクセスするために使用します。

4GB のアドレス空間を有しています。32bit アドレスをサポートできないバス・マスタは、Qsys の IP カタログにある Address Span Extender (ウィンドウ・ブリッジ) を使用してアドレス拡張することにより、接続できます。

データ・バス幅は、32bit / 64bit / 128bit から選択することが可能です。

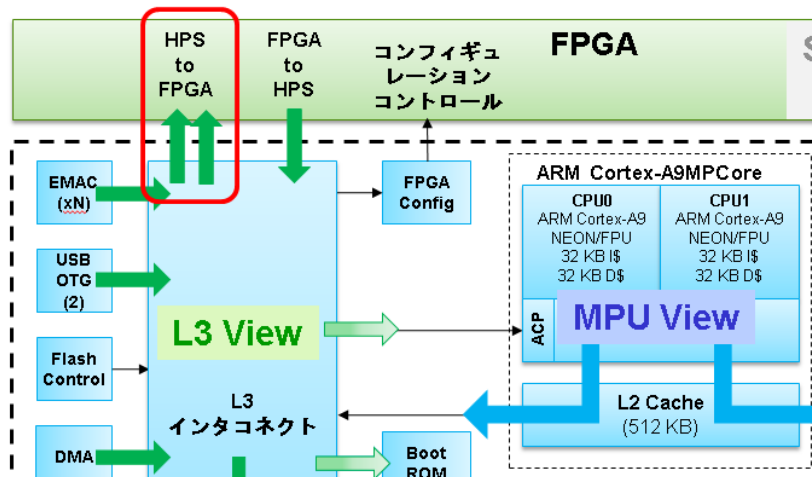
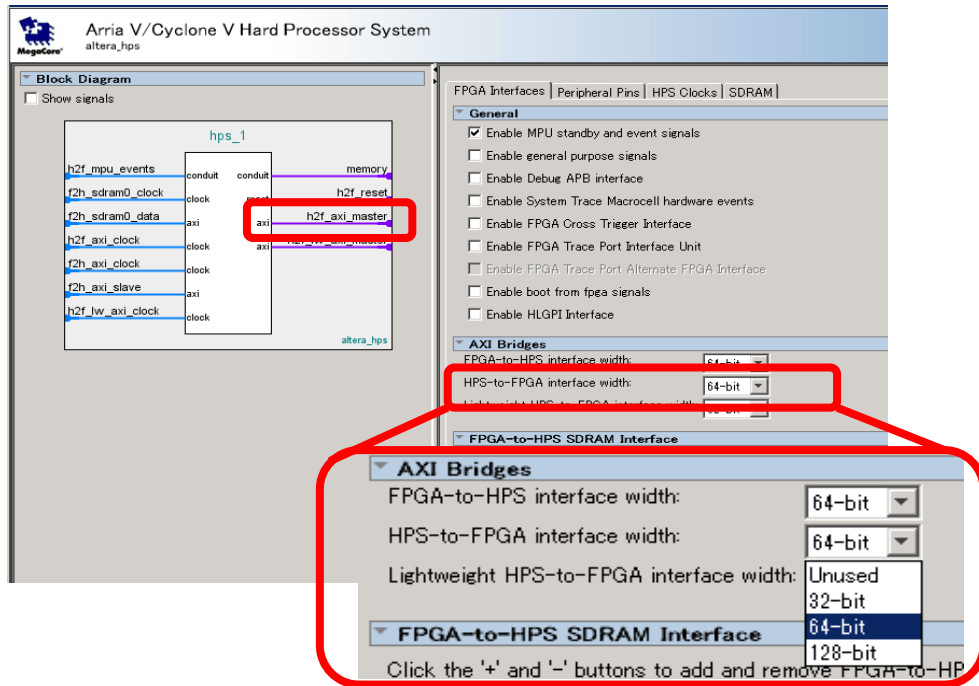


#### 4-1-2. HPS-to-FPGA インタフェース (H2F)

HPS-to-FPGA インタフェースは、HPS 内にある ARM® プロセッサや DMA コントローラ等のバス・マスタが、FPGA 内のペリフェラルにアクセスするために使用します。最大 128bit のデータ・バス幅を選択できるため、大容量のデータ転送に適します。

960MB (0xC000\_0000 ~ 0xFBFF\_FFFF) のアドレス空間を有しています。960MB より大きい空間にアクセスする場合、Qsys の IP カタログにある Address Span Extender (ウィンドウ・ブリッジ) を使用してアドレス拡張することにより、アクセスできます。

データ・バス幅は、32bit / 64bit / 128bit から選択することが可能です。

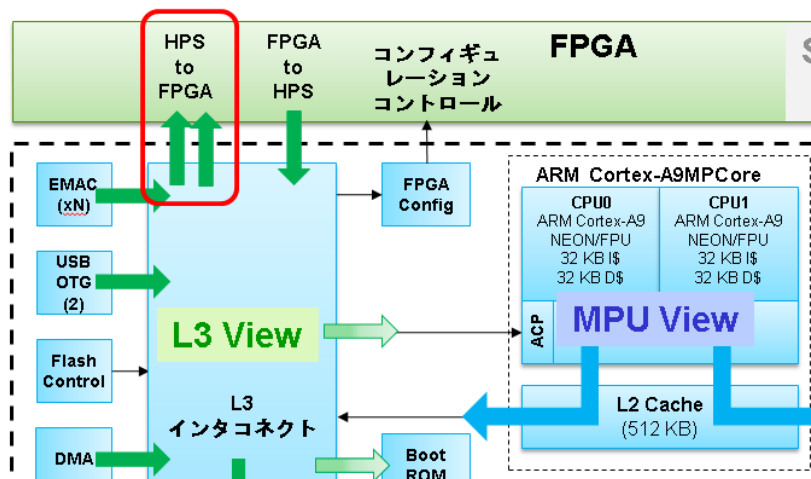
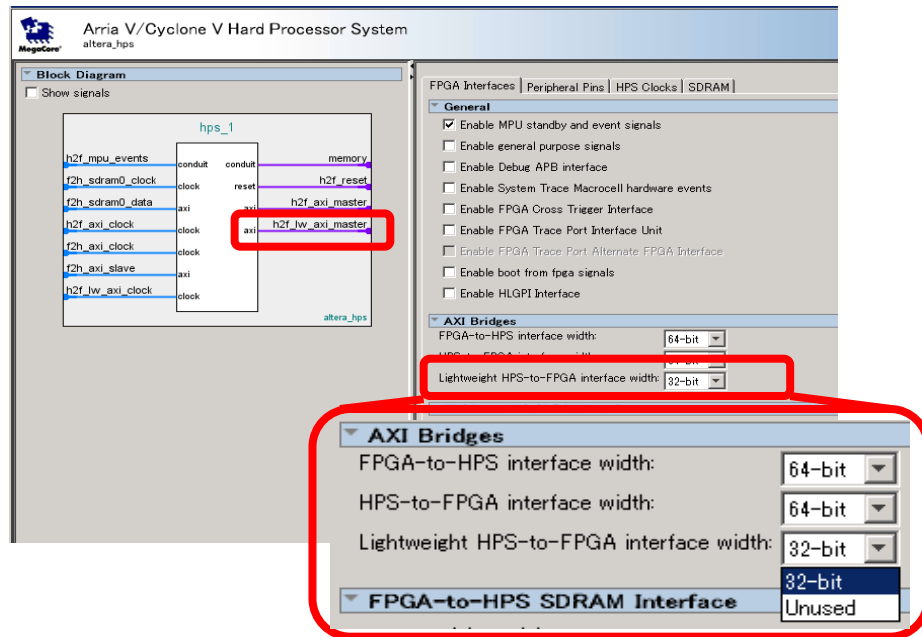


### 4-1-3. Lightweight HPS-to-FPGA インタフェース (LWH2F)

Lightweight HPS-to-FPGA インタフェースは、HPS 内にある ARM® プロセッサや DMA コントローラ等のバス・マスタが、FPGA 内のペリフェラルにアクセスするために使用します。データ・バス幅が 32bit 固定のため、LED 等の各種レジスタ制御用や、状態の確認のためのレジスタアクセスに適します。

2MB (0xFF20\_0000 ~ 0xFF3F\_FFFF) のアドレス空間を有しています。2MB 以上の空間にアクセスする場合、Qsys の IP カタログにある Address Span Extender (ウィンドウ・ブリッジ) を使用してアドレス拡張することにより、アクセスできます。

データ・バス幅は、32bit 幅のみとなります。



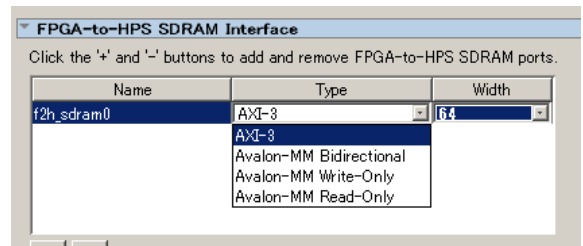
#### 4-1-4. FPGA-to-HPS SDRAM インタフェース (F2S)

FPGA-to-HPS SDRAM インタフェース (F2S) は、FPGA 内のバス・マスタが、HPS 内の L3 インタコネクトを介すことなく、HPS 内の SDRAM にアクセスができるインタフェースです。このポートは、最大 6 ポートまで持つことが可能です。

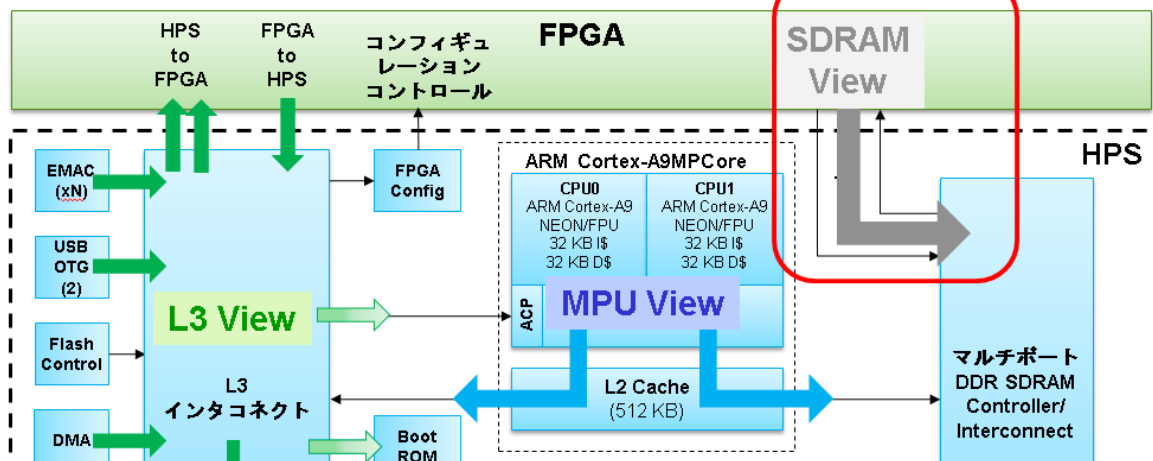
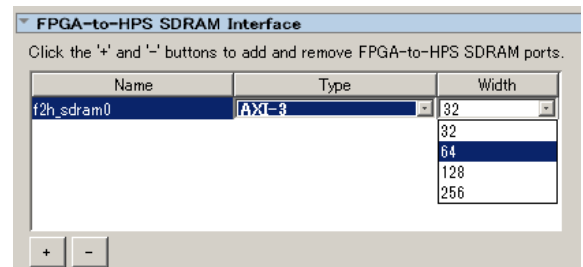
4GB のアドレス空間を有します。アドレス空間は 4GB 存在しますが、メモリの実体がない空間へのアクセスは行わないでください。

インタフェースの種類は、4 種類選択できます。

- AXI-3
- Avalon-MM Bidirectional
- Avalon-MM Write-Only
- Avalon-MM Read-Only



データ・バス幅は、32bit / 64bit / 128bit / 256bit から選択できます。



#### 4-1-5. F2S のポート数

FPGA-to-HPS SDRAM インタフェース (F2S) は、複数のポートを持てることは紹介しました。ここでは、ポート数の考え方について紹介します。

F2S は、コマンド・ポート (6 ポート)、64bit read data ポート (4 ポート)、64bit の write data ポート (4 ポート) にて実装されます。以下の表に示す通り、AXI バスの場合、コマンド・ポートを 2 本、Avalon®-MM バスの場合、コマンド・ポートを 1 ポート使用します。このため、AXI バスの場合、最大 3 ポート、Avalon®-MM バスの場合、最大 6 ポート実装できます。

データ・バス幅で考えると、256bit のバスは最大 1 ポート、128bit のバスは最大 2 ポート、64bit もしくは 32bit のバスの場合は最大 4 ポート実装できます。Avalon®-MM の read only ポートおよび write only ポートを組み合わせることにより、最大 6 ポート実装できることになります。

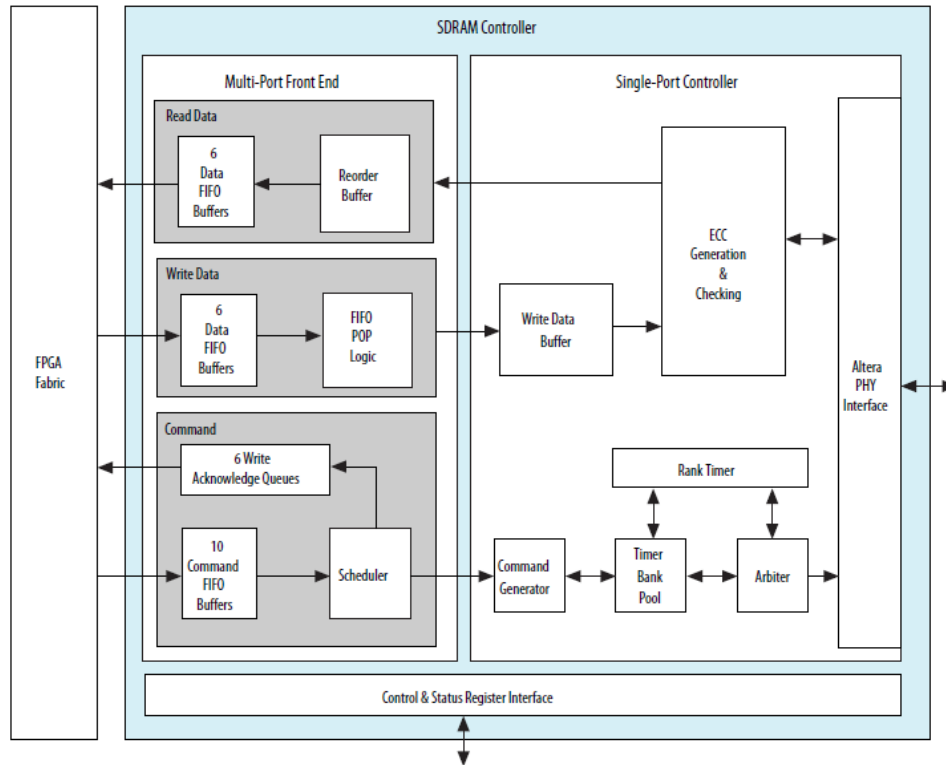
Port Type	Available Number of Ports
Command	6
64-bit read data	4
64-bit write data	4

Bus Protocol	Command Ports	Read Data Ports	Write Data Ports
32- or 64-bit AXI	2	1	1
128-bit AXI	2	2	2
256-bit AXI	2	4	4
32- or 64-bit Avalon-MM	1	1	1
128-bit Avalon-MM	1	2	2
256-bit Avalon-MM	1	4	4
32- or 64-bit Avalon-MM write-only	1	0	1
128-bit Avalon-MM write-only	1	0	2
256-bit Avalon-MM write-only	1	0	4
32- or 64-bit Avalon-MM read-only	1	1	0
128-bit Avalon-MM read-only	1	2	0
256-bit Avalon-MM read-only	1	4	0

#### 4-1-6. SDRAM コントローラの優先度

HPS に実装されている SDRAM コントローラは、複数ポートからのアクセスを可能にするため、アービトレーション・ロジックを実装しています。重みづけのできるラウンド・ロビン方式で実装されています。下表に示すコマンド・ポート単位で重みづけできます。

詳細は、リファレンス・マニュアル (使用するデバイス・ファミリー (Arria® V / Cyclone® V) 用の Hard Processor System Technical Reference Manual) の SDRAM Controller Subsystem の項をご参照ください。



Command Port	Allowed Functions	Data Size
0, 2, 4	FPGA fabric AXI read command ports FPGA fabric Avalon-MM read or write command ports	32-bit to 256-bit data
1, 3, 5	FPGA fabric AXI write command ports FPGA fabric Avalon-MM read or write command ports	
6	L3 AXI read command port	32-bit data
7	MPU AXI read command port	64-bit data
8	L3 AXI write command port	32-bit data
9	MPU AXI write command port	64-bit data

## 4-2. 他のコンポーネントとの接続

HPS と他のコンポーネントとを接続するには、Qsys の GUI 上での設定を行います。HPS の各インターフェースは、Qsys 上でポートとして見ることができます。このポートと接続先のポートを結線することにより、接続できます。

HPS のインターフェースは AXI インターフェースですが、Qsys が自動的に変換するため、Avalon®-MM インターフェースも AXI インターフェースもどちらも接続可能です。

Qsys 内のコンポーネントのアドレスも GUI にて設定可能です。ここで設定したアドレスを元に、ソフトウェアからアクセスを行います。

Connections	Name	Description	Export	Clock	Base
	sysid_qsys	System ID Peripheral		[clk]	
	control_slave	Avalon Memory Mapped Slave	Double-click to export	clk_0	0x0002_0008
	<b>hps_0</b>	Arria V/Cyclone V Hard Processor S...			multiple
	f2h_sdram0_data	Avalon Memory Mapped Slave	Double-click to export	clk_0	0x0000_0000
	h2f_axi_master	AXI Master	Double-click to export	clk_0	
	f2h_axi_slave	AXI Slave	Double-click to export	clk_0	0x0000_0000
	h2f_lw_axi_master	AXI Master	Double-click to export	clk_0	
	<b>hps_only_master</b>	JTAG to Avalon Master Bridge			
	master	Avalon Memory Mapped Master	Double-click to export	clk_0	
	<b>fpga_only_master</b>	JTAG to Avalon Master Bridge			
	master	Avalon Memory Mapped Master	Double-click to export	clk_0	
	<b>f2sdram_only_ma...</b>	JTAG to Avalon Master Bridge			
	master	Avalon Memory Mapped Master	Double-click to export	clk_0	
	<b>mm_bridge_0</b>	Avalon-MM Pipeline Bridge			
	s0	Avalon Memory Mapped Slave	Double-click to export	clk_0	0x0004_0000
	m0	Avalon Memory Mapped Master	Double-click to export	[clk]	
	<b>jtag_uart</b>	JTAG UART			
	avalon_jtag_slave	Avalon Memory Mapped Slave	Double-click to export	clk_0	0x0002_0000
	<b>button_pio</b>	PIO (Parallel I/O)			
	s1	Avalon Memory Mapped Slave	Double-click to export	clk_0	0x0002_00c0
	<b>dipsw_pio</b>	PIO (Parallel I/O)			
	s1	Avalon Memory Mapped Slave	Double-click to export	clk_0	0x0002_0080
	<b>led_pio</b>	PIO (Parallel I/O)			
	s1	Avalon Memory Mapped Slave	Double-click to export	clk_0	0x0002_0040
	<b>onchip_memory2_0</b>	On-Chip Memory (RAM or ROM)			
	s1	Avalon Memory Mapped Slave	Double-click to export	clk_0	0x0000_0000
	<b>ILC</b>	Altera Interrupt Latency Counter			
	avalon_slave	Avalon Memory Mapped Slave	Double-click to export	clk_0	0x0003_0000
	<b>in_system_sources_...</b>	Altera In-System Sources & Probes			

### 4-3. インタフェースの初期化

HPS - FPGA 間のインタフェースを使用するためには、HPS 内にあるレジスタを設定する必要があります。次の条件を満たす場合には、Preloader と呼ばれるインテル提供のブートローダを使用することにより設定されるので、ユーザはあまり気にする必要はありません。

- Preloader 実行前に FPGA のコンフィギュレーションが完了している
- Preloader 実行中に FPGA のコンフィギュレーションを行う

#### 4-3-1. F2H, H2F, LWH2F の初期化

HPS2FPGA (H2F) / LWHP2FPGA (LWH2F) / FPGA2HPS (F2H) ブリッジを利用するには、2 ステップが必要です。

##### ① ブリッジリセットの解除

0xFFD0\_501C に 0x0 をライト (bit0: H2F / bit1: LWH2F / bit2: F2H の各ビットに 0: Reset 解除をセット)

詳細は、以下を参照ください。

brgmodrst レジスタの詳細:

- [Arria® V HPS Memory Map](#)
- [Cyclone® V HPS Memory Map](#)

##### ② L3 インタコネクトの設定

0xFF80\_0000 に 0x18 をライト (bit3: H2F / bit4: LWH2F の各ビットに 1: Visible to L3 Master をセット)

詳細は、以下をご参照ください。

remap レジスタの詳細:

- [Arria® V HPS Memory Map](#)
- [Cyclone® V HPS Memory Map](#)

書き込む値は、デザイン依存になりますので、各レジスタの詳細をご確認の上設定ください。これらの初期化は、後程ご紹介する bridge\_enable\_handoff コマンドで設定することが可能です。



#### 4-3-2. F2S の初期化

Preloader により、SDRAM の初期化が行われていない場合、Bridge へのアクセスができません。また、F2S の設定を変更する場合は、SDRAM が Idle 状態である必要があります。このため、U-Boot に実装されている bridge\_enable\_handoff コマンドを実行することをお勧めします。

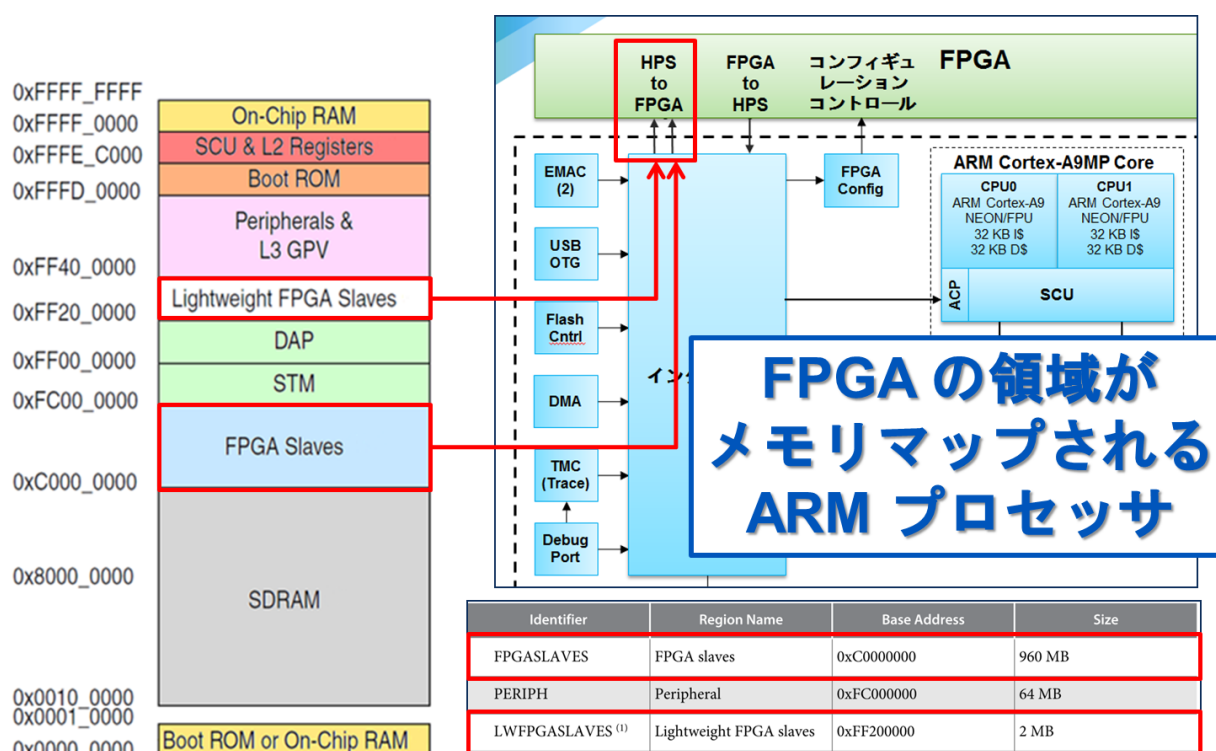
```
run bridge_enable_handoff
```

このコマンドでは、ブリッジのパラメータは、Preloader からの情報を使用します。run bridge\_enable\_handoff に関する詳細情報は、Preloader/U-Boot のソース・ツリーに含まれるドキュメントに記載がありますので、こちらをご参照ください。

ファイル名 : uboot-socfpga/doc/README.SOCFPGA

## 5. ソフトウェアからのアクセス

ARM® Cortex®-A9 MPCore™ プロセッサと FPGA とが、デバイス内でバス接続されています。ARM® から FPGA



内部のペリフェラルは、メモリ・マップト I/O として見るすることができます。

### 5-1. アドレスの考え方

メモリ・マップより、HPS-to-FPGA (H2F) インタフェースのベース・アドレスは 0xC000\_0000、Light Weight HPS-to-FPGA (LWH2F) インタフェースのベース・アドレスは 0xFF20\_0000 であることが分かります。

ARM® から見た FPGA コンポーネントのアドレスは、以下のように計算できます。

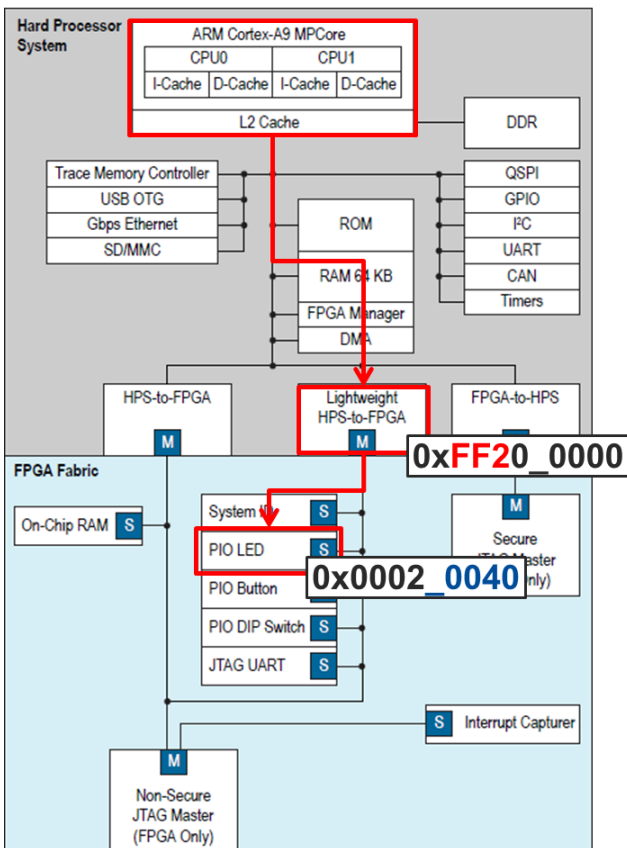
$$\text{FPGA コンポーネントのアドレス} = \text{HPS-FPGA ブリッジのベース・アドレス} + \text{Qsys 上のオフセット・アドレス}$$

例えば、Qsys 上で、LWH2F の 0x00020040 に接続された LED に対してアクセスするには、以下のアドレスにアクセスします。

#### ARM から PIO LED にアクセスする際のアドレス

$$= 0xFF20_0000 (\text{LWH2F base}) + 0x0002_0040 (\text{Qsys 上のオフセット})$$

$$= \underline{\underline{0xFF22\ 0040}}$$



ARM から見た FPGA コンポーネントのアドレス :

HPS-FPGA ブリッジのベース・アドレス  
+ Qsys 上のオフセット・アドレス  
-----  
= FPGA コンポーネントのアドレス

PIO LED にアクセスする場合 :

0xFF20\_0000 (LW\_HPS-to-FPGA)  
+ 0x0002\_0040 (PIO LED Offset)  
-----  
= 0xFF22\_0040 (PIO BaseAddr From ARM)

Qsys 上のアドレス

Connections	Name	Description	Export	Clock	Base
System ID Peripheral	sysid_sys	System ID Peripheral	Double-click to export	clk_0	0x0002_0000
Hard Processor System	hw_sys	Hard Processor System	Double-click to export	clk_0	0x0000_0000
On-Chip RAM	onchip_ram	On-Chip Memory (RAM or ROM)	Double-click to export	clk_0	0x0000_0000
PIO LED	led	PIO LED	Double-click to export	clk_0	0x0002_0040
PIO Button	button	PIO Button	Double-click to export	clk_0	0x0002_0000
PIO DIP Switch	dipsw	PIO DIP Switch	Double-click to export	clk_0	0x0002_0000
JTAG UART	jtag_uart	JTAG UART	Double-click to export	clk_0	0x0002_0000
Interrupt Capturer	intc	Interrupt Capturer	Double-click to export	clk_0	0x0002_0000

```
#include <stdio.h>
#include "social.h"

#define LED_BASE_ADDR (0xFF220040)

int main(int argc, char** argv)
{
    int i;

    printf("Hello from Sodica. \n");

    while(1)
    {
        for(i=0; i < 16; i++){
            alt_write_word(LED_BASE_ADDR,i);
            printf("LED [%0x] \n",i);
        }
    }

    return 0;
}
```

## 5-2. システム・ヘッダ・ファイル

先ほどの例では、ソース・コード上に LED\_PIO のアドレスを定義し、アクセスを行いました。この方法では、ハードウェアの情報が変更される毎に、ソース・コードの変更が必要になる場合があります。

インテル® FPGA では、ハードウェアの情報をソフトウェア担当者に引く次ぐための仕組みを提供しております。その仕組みが、システム・ヘッダ・ファイル生成ツールです。

SoC EDS 17.0 Command Shell 上で、<qsys system>.sopcinfo ファイルのあるディレクトリまで移動して、“sopc-create-header-files” コマンドを実行します。

```
$ socp-create-header-files <qsys subsystem>.sopcinfo
```

「どのマスタから見るか」という観点で、いくつかヘッダ・ファイルが生成されるので、ご利用ください。

ヘッダ・ファイルの例を、以下に示します。

```
/*↓
 * Macros for device 'led_pio', class 'altera_avalon_pio'↓
 * The macros are prefixed with 'LED_PIO_'↓
 * The prefix is the slave descriptor.↓
 */↓
#define LED_PIO_COMPONENT_TYPE altera_avalon_pio↓
#define LED_PIO_COMPONENT_NAME led_pio↓
#define LED_PIO_BASE 0xff280040↓
#define LED_PIO_SPAN 32↓
#define LED_PIO_END 0xff28005f↓
#define LED_PIO_BIT_CLEARING_EDGE_REGISTER 0↓
#define LED_PIO_BIT_MODIFYING_OUTPUT_REGISTER 1↓
#define LED_PIO_CAPTURE 0↓
#define LED_PIO_DATA_WIDTH 4↓
#define LED_PIO_DO_TEST_BENCH_WIRING 0↓
#define LED_PIO_DRIVEN_SIM_VALUE 0↓
#define LED_PIO_EDGE_TYPE NONE↓
#define LED_PIO_FREQ 50000000↓
#define LED_PIO_HAS_IN 1↓
#define LED_PIO_HAS_OUT 1↓
#define LED_PIO_HAS_TRI 0↓
#define LED_PIO_IRQ_TYPE NONE↓
#define LED_PIO_RESET_VALUE 0↓
↓
```

## 6. 参考情報

この資料では、Arria® V SoC FPGA / Cyclone® V SoC FPGA における、HPS と FPGA 間のアクセス方法を紹介しました。この資料を参考に、インテル® SoC FPGA をお使いいただければと思います。

インテル® SoC FPGA には、サンプル・デザインも用意されているので、初めて SoC FPGA を使用される場合は開発キット等でサンプル・デザインをベースに動かしてみると、理解への近道になると思います。

サンプル・デザインは、マクニカオンラインサービスや、インテル® FPGA のページ、および RocketBoards.org で紹介されているので、ぜひご覧ください。

- [マクニカオンラインサービス SoC 関連情報](#)
- [マクニカオンラインサービス SoC 関連 FAQ](#)
- [HPS-to-FPGA Bridges Design Example](#)
- [SoC Design Examples](#)
- [RocketBoards.org](#)

## 改版履歴

Revision	年月	概要
1	2017 年 7 月	初版

### 免責およびご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
2. 本資料は予告なく変更することがあります。
3. 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご一報いただければ幸いです。

株式会社マクニカ アルティマ カンパニー <https://www.alt.macnica.co.jp/> 技術情報サイト アルティマ技術データベース <http://www.altima.jp/members/>

株式会社エルセナ <http://www.elsena.co.jp> 技術情報サイト ETS <https://www.elsena.co.jp/elspear/members/index.cfm>

4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる際は、各メーカー発行の英語版の資料もあわせてご利用ください。