

SoC FPGA HPS のアドレス指定方法

ver.14

SoC FPGA HPS のアドレス指定方法

目次

1. はじめに	3
2. HPS と FPGA 間のインタフェース	3
3. HPS アドレス・マップ	4
3-1. HPS アドレス空間	4
3-2. MPU ビュー	5
3-3. Non-MPU ビュー	6
3-4. Qsys デザイン例	7
3-5. HPS から見た Qsys 上のスレーブ・アドレス	8
3-6. FPGA から見た HPS 上のスレーブ・アドレス	8
3-7. L3 インターコネクト・アドレス・リマップ・レジスタ	9
4. 参考資料	9
改版履歴	10

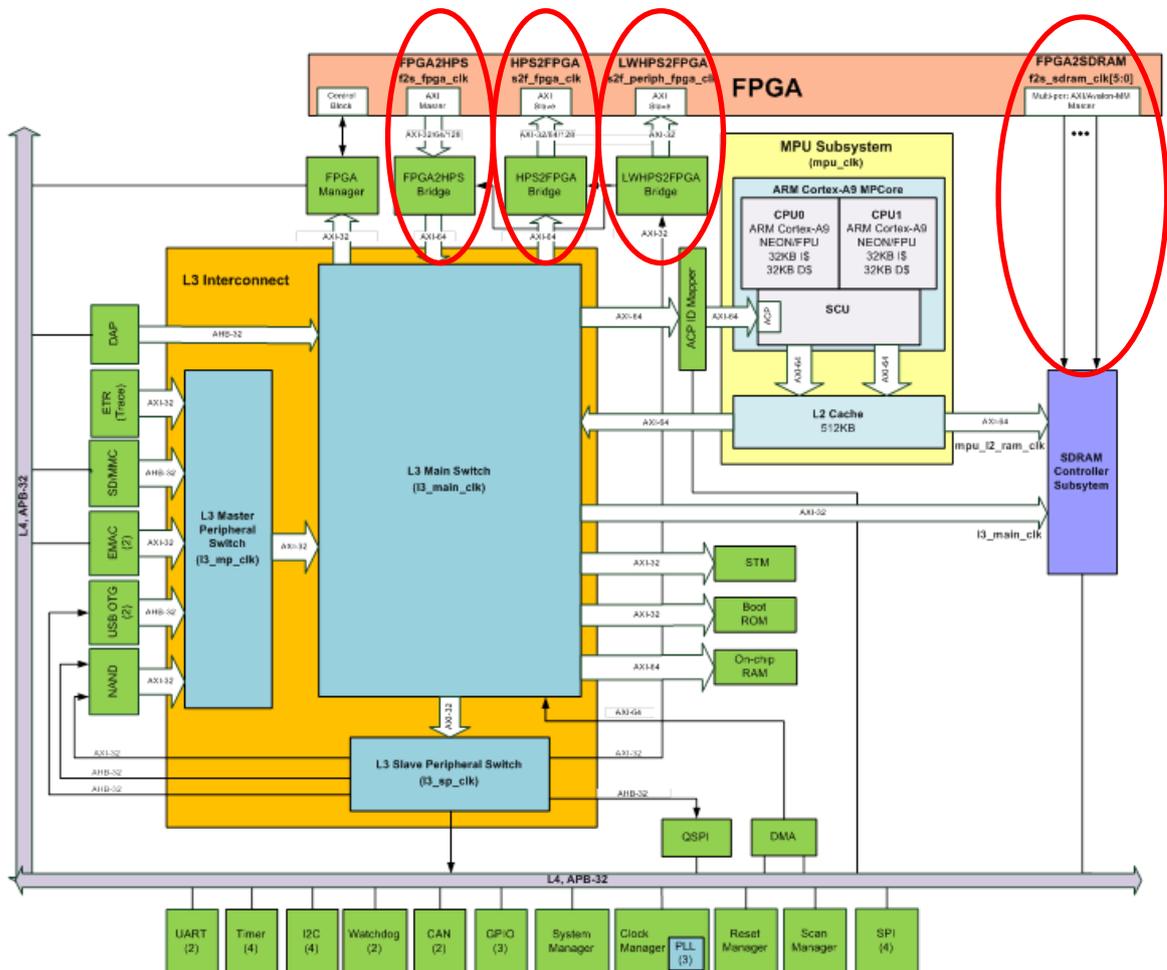
1. はじめに

この資料は、アルテラ社 SoC FPGA デバイスの Hard Processor System（以降、HPS と略）のアドレス・マップと Qsys のアドレス空間との関連について説明した資料です。

2. HPS と FPGA 間のインタフェース

SoC FPGA デバイスでは、Qsys を使用してハードウェアのデザインを行います。HPS 側がマスタとなり FPGA 側のスレーブ・モジュール等にアクセスするためのインタフェースとして、2 つのバスが用意されています。一つは、最大 128 ビット幅の高帯域インタフェースである HPS2FPGA（HPS-to-FPGA）インタフェースです。このバスは、HPS と FPGA 間の比較的大容量のデータのやり取りを行うためのインタフェースとなります。もう一つが、32 ビット幅のインタフェースとして、LWHP2FPGA（Lightweight HPS-to-FPGA）インタフェースです。このバスは、制御信号などの比較的低速でアクセスするデータ伝送のために用意されています。これらのインタフェースは、後述するアドレス・マップの FPGA Slaves と Lightweight FPGA Slaves にそれぞれマッピングされます。さらに、Qsys 内部にも独立したアドレス空間が存在するために、実際にソフトウェア側で指定するアドレス値には考慮が必要です。

逆に、FPGA 側がマスタとなり HPS 側のスレーブ・モジュール等にアクセスするためのインタフェースとして 2 つのバスが用意されています。一つは、最大 128 ビット幅の高帯域インタフェースである FPGA2HPS（FPGA-to-HPS）インタフェースです。もう一つは、FPGA 側から HPS 側の SDRAM Controller に直接アクセスするためのバス・インタフェースとして用意されている FPGA2SDRAM（FPGA-to-SDRAM）インタフェースです。

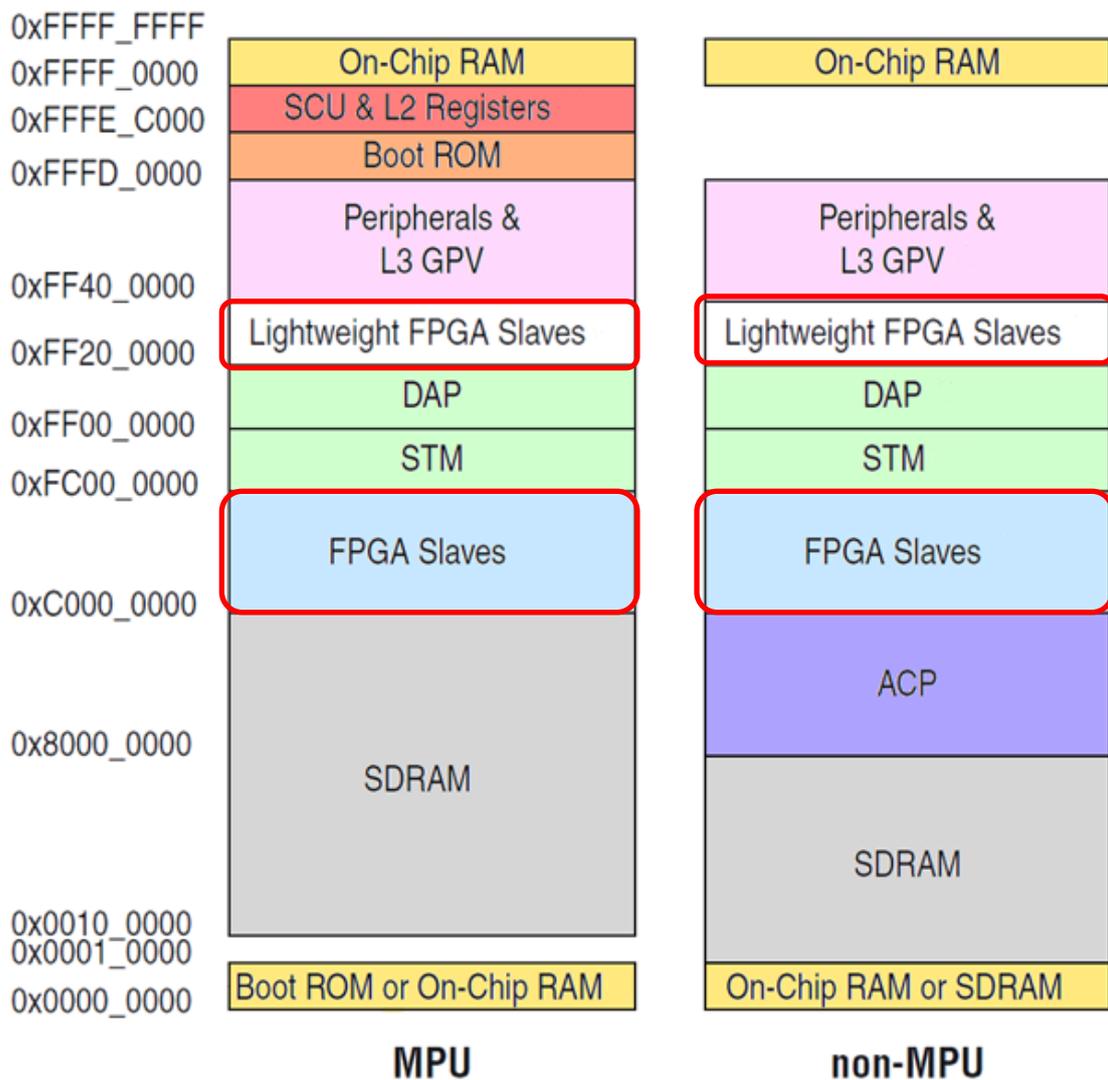


3. HPS アドレス・マップ

3-1. HPS アドレス空間

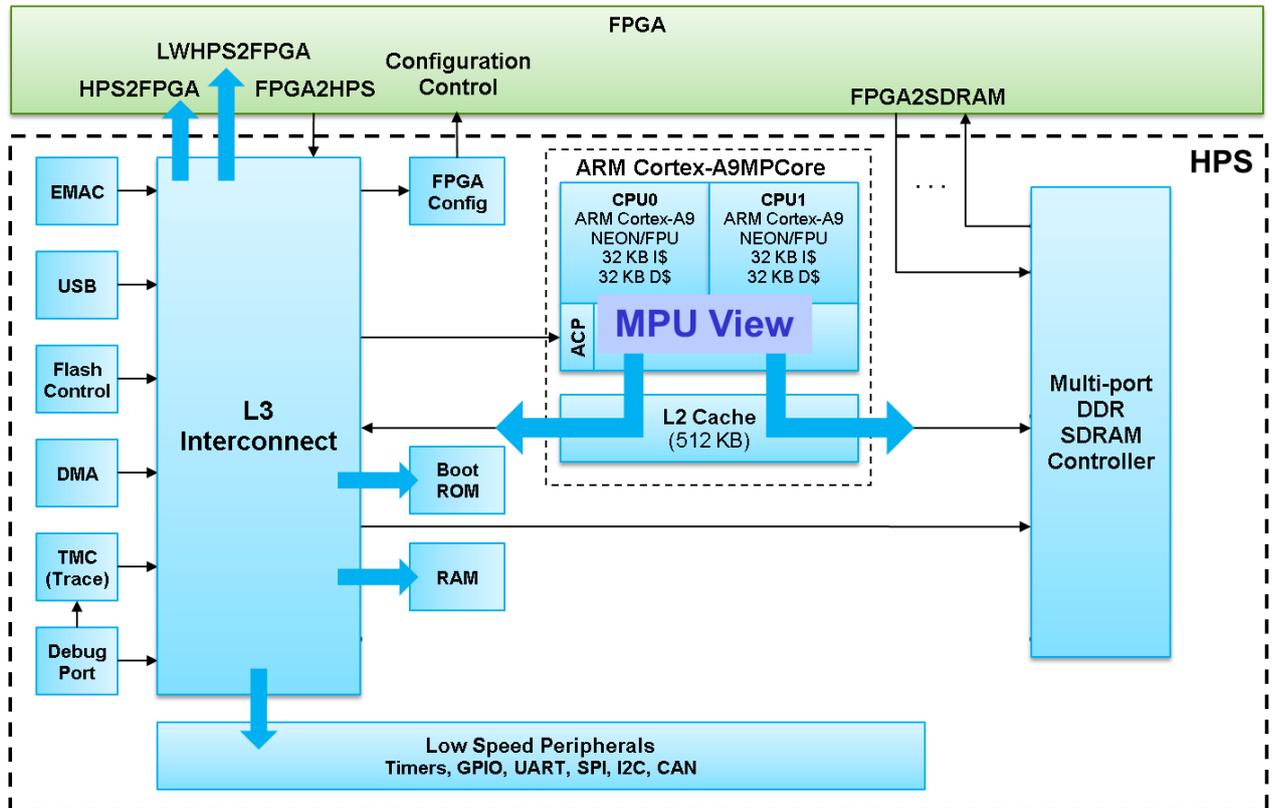
下図が HPS のアドレス空間の定義となります。左側の MPU とある方が、MPU (ARM プロセッサ) から見たアドレス空間となります。右側の Non-MPU とある方が、周辺のペリフェラルがマスタとなり L3 インタコネクト経由でアクセスする際のアドレス空間となります。HPS、または、Non-HPS から FPGA にアクセスするためには、赤枠で囲った FPGA Slaves、Lightweight FPGA Slaves 経由でアクセスする必要があります。

FPGA Slaves のアドレス空間からアクセスされるバスが、HPS2FPGA インタフェースです。Lightweight FPGA Slaves のアドレス空間からアクセスされるバスが、LWHPS2FPGA インタフェースです。



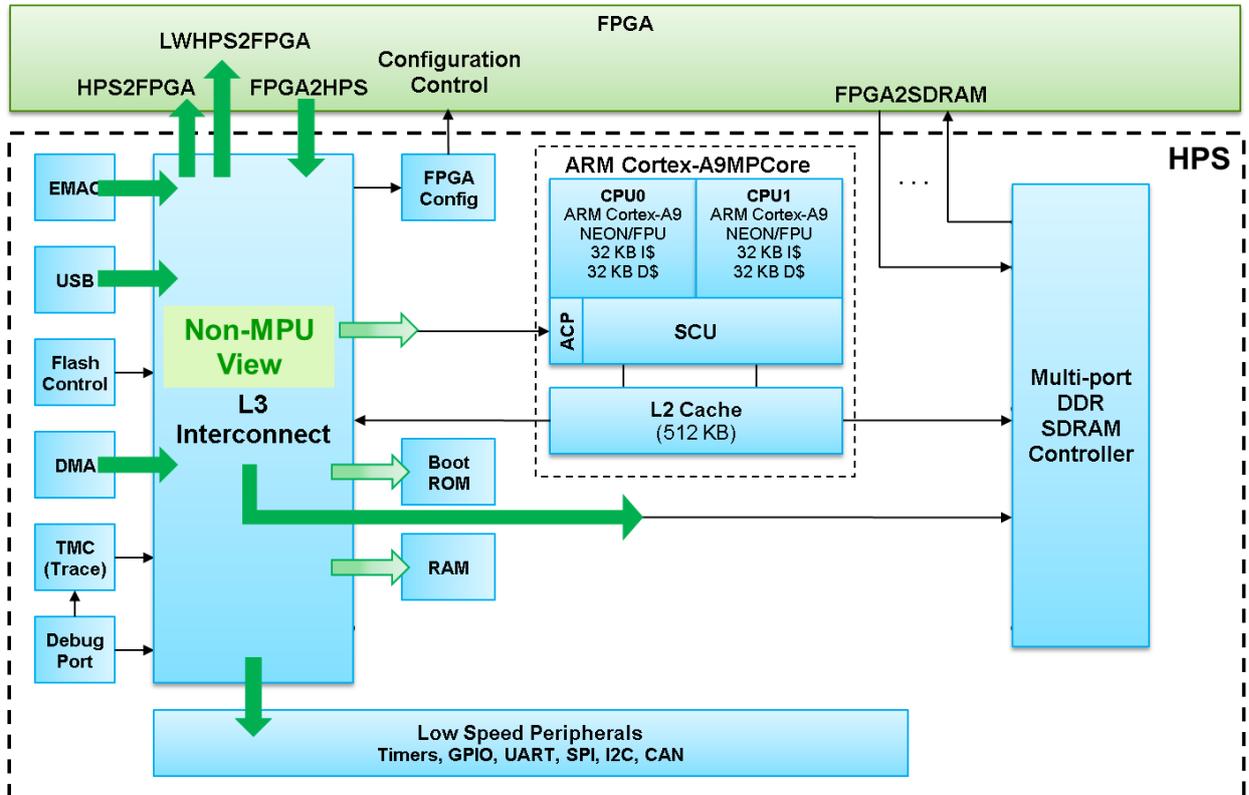
3-2. MPU ビュー

下図の MPU ビューは、前頁の MPU のアドレス空間が MPU からどのように見えるかを水色の矢印で示したものです。例えば、ソフトウェアから HPS2FPGA バスにアクセスする場合には、FPGA Slaves で定義されているアドレス値 0xC000_0000 を指定するアドレスのオフセット値としてアクセスします。同様に、LWHP2FPGA バスにアクセスする場合には、Lightweight FPGA Slaves で定義されているアドレス値 0xFF20_0000 を指定するアドレスのオフセット値としてアクセスします。



3-3. Non-MPU ビュー

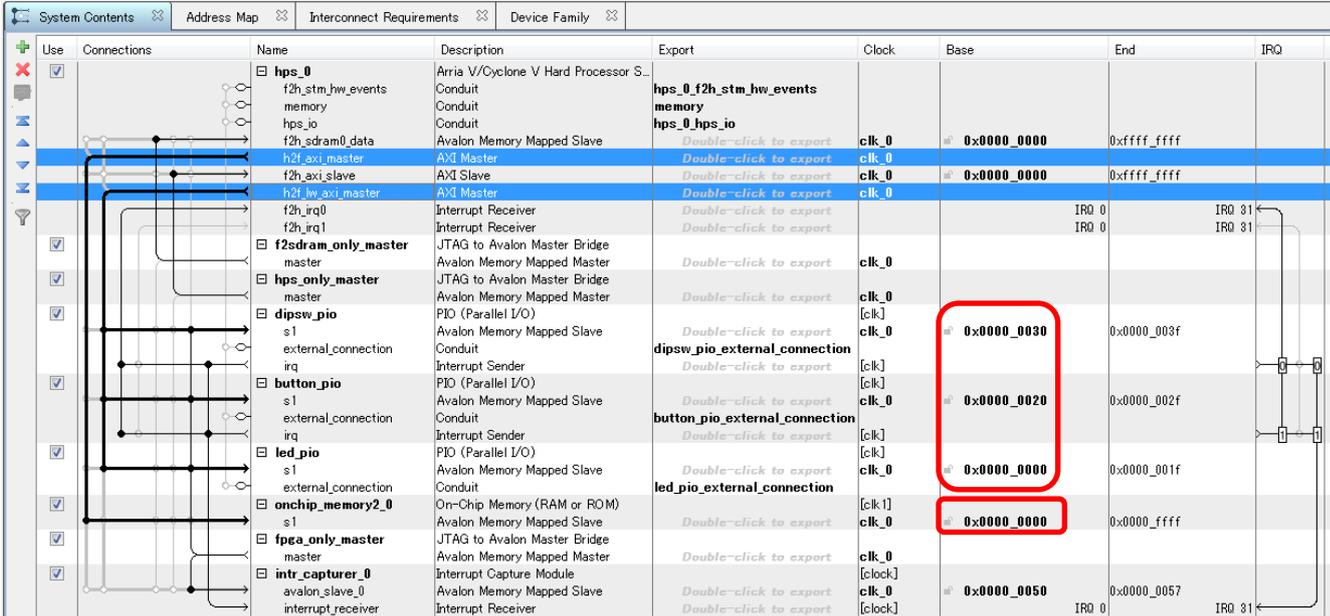
下図の Non-MPU ビューは、前ページの Non-MPU のアドレス空間が Non-MPU (L3 Interconnect) からどのように見えるかを緑色の矢印で示したものです。Non-MPU からのアクセスでは、このように、多少 MPU ビューとは見え方が異なることに注意してください。



3-4. Qsys デザイン例

下図が Qsys デザイン例となります。h2f_axi_master、h2f_lw_axi_master とともに、Qsys 側から見たらマスタ・インタフェースとなっていることに注意してください。h2f_axi_master が FPGA Slaves 経由でアクセスした HPS2FPGA バス・インタフェースとなります。図では、onchip_memory2_0 に接続されています。同様に h2f_lw_axi_master が Lightweight FPGA Slaves 経由でアクセスした LWHPS2FPGA バス・インタフェースとなります。図では、led_pio、dipsw_pio、button_pio にそれぞれ接続されています。

図では、h2f_axi_master に接続されている onchip_memory2_0 のアドレス値が 0x0000_0000 となっています。また、h2f_lw_axi_master に接続されている led_pio が同じく 0x0000_0000 となっています。Qsys ではそれぞれのマスタは独立したアドレス空間を持つため、それぞれのマスタに接続されたスレーブに対してアドレスが重複しているように見えてもこれはエラーではありません。このように、Qsys 内のアドレス表示では、先に説明した HPS アドレス空間の 0xC0000000 や 0xFF200000 が自動的に引き継がれることはなく、0x0000_0000 から始まるアドレスになります(アドレス値は、手動で好きな値に変更することも可能です)。

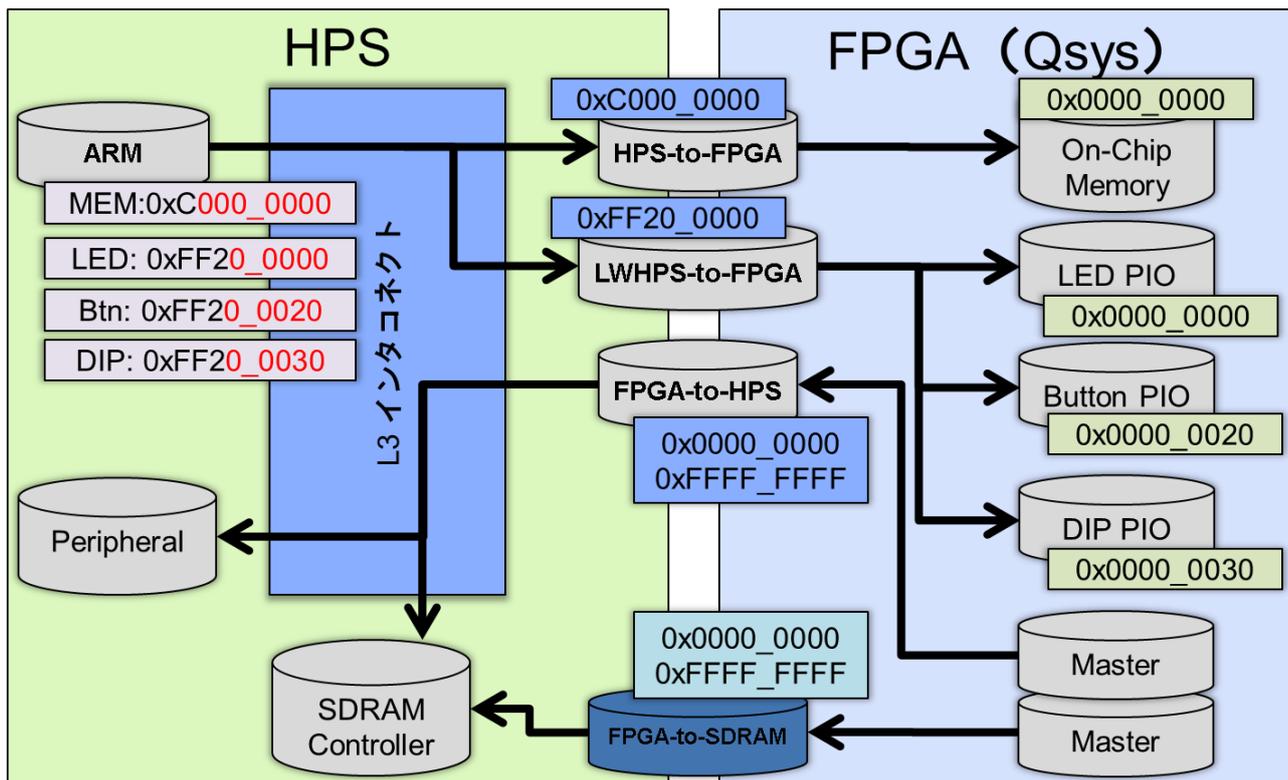


Use	Connections	Name	Description	Export	Clock	Base	End	IRQ
<input checked="" type="checkbox"/>		hps_0	Arria V/Cyclone V Hard Processor S...					
		f2h_stm_hw_events	Conduit	hps_0_f2h_stm_hw_events				
		memory	Conduit	memory				
		hps_io	Conduit	hps_0_hps_io				
		f2h_sdram0_data	Avalon Memory Mapped Slave	Double-click to export	clk_0	# 0x0000_0000	0xffff_ffff	
		h2f_axi_master	AXI Master	Double-click to export	clk_0	# 0x0000_0000	0xffff_ffff	
		f2h_axi_slave	AXI Slave	Double-click to export	clk_0	# 0x0000_0000	0xffff_ffff	
		h2f_lw_axi_master	AXI Master	Double-click to export	clk_0	# 0x0000_0000	0xffff_ffff	
		f2h_irq0	Interrupt Receiver	Double-click to export			IRQ 0	IRQ 31
		f2h_irq1	Interrupt Receiver	Double-click to export			IRQ 0	IRQ 31
<input checked="" type="checkbox"/>		f2sdram_only_master	JTAG to Avalon Master Bridge	Double-click to export	clk_0			
<input checked="" type="checkbox"/>		hps_only_master	Avalon Memory Mapped Master	Double-click to export	clk_0			
<input checked="" type="checkbox"/>		master	JTAG to Avalon Master Bridge	Double-click to export	clk_0			
<input checked="" type="checkbox"/>		dipsw_pio	PIO (Parallel I/O)	Double-click to export	clk_0	# 0x0000_0030	0x0000_003f	
		s1	Avalon Memory Mapped Slave	Double-click to export	clk_0	# 0x0000_0030	0x0000_003f	
		external_connection	Conduit	dipsw_pio_external_connection				
		irq	Interrupt Sender	Double-click to export	[clk]			
<input checked="" type="checkbox"/>		button_pio	PIO (Parallel I/O)	Double-click to export	clk_0	# 0x0000_0020	0x0000_002f	
		s1	Avalon Memory Mapped Slave	Double-click to export	clk_0	# 0x0000_0020	0x0000_002f	
		external_connection	Conduit	button_pio_external_connection				
		irq	Interrupt Sender	Double-click to export	[clk]			
<input checked="" type="checkbox"/>		led_pio	PIO (Parallel I/O)	Double-click to export	clk_0	# 0x0000_0000	0x0000_001f	
		s1	Avalon Memory Mapped Slave	Double-click to export	clk_0	# 0x0000_0000	0x0000_001f	
		external_connection	Conduit	led_pio_external_connection				
<input checked="" type="checkbox"/>		onchip_memory2_0	On-Chip Memory (RAM or ROM)	Double-click to export	clk_1	# 0x0000_0000	0x0000_ffff	
<input checked="" type="checkbox"/>		fpga_only_master	JTAG to Avalon Master Bridge	Double-click to export	clk_0			
<input checked="" type="checkbox"/>		master	Avalon Memory Mapped Master	Double-click to export	clk_0			
<input checked="" type="checkbox"/>		intr_capturer_0	Interrupt Capture Module	Double-click to export	clk_0	# 0x0000_0050	0x0000_0057	
		avalon_slave_0	Avalon Memory Mapped Slave	Double-click to export	clk_0	# 0x0000_0050	0x0000_0057	
		interrupt_receiver	Interrupt Receiver	Double-click to export	[clock]		IRQ 0	IRQ 31

3-5. HPS から見た Qsys 上のスレーブ・アドレス

ここまでで説明してきたように、HPS から Qsys 内にあるスレーブを見たときには、HPS アドレス空間と Qsys 内のスレーブ・アドレスを考慮する必要があります。下図が、前頁の Qsys の構成に基づいたアドレス対応図となります。HPS から見た Qsys スレーブのアドレスは、「L3 のベースアドレス + Qsys のオフセット」の合計値となります。

HPS 内の ARM プロセッサ上のソフトウェアでプログラムを記述する際には、On-Chip Memory にアクセスする場合には、0xC000_0000 を指定する必要があります。同様に、LED_PIO ⇒ 0xFF20_0000、Button_PIO ⇒ 0xFF20_0020、DIP_PIO ⇒ 0xFF20_0030 と指定する必要があります。



3-6. FPGA から見た HPS 上のスレーブ・アドレス

FPGA2HPS バス・インターフェースは、FPGA 側のマスタ・モジュールがこのインターフェース経由でアクセスする際に使用します。図では、FPGA-to-HPS とあるブロックが該当します。マスタ・モジュールからは、0x0000_0000 から 0xFFFF_FFFF の 4GByte のアドレス空間が見えますが、アドレス・マップは、Non-MPU ビューに従います。

FPGA2SDRAM バス・インターフェースについては、FPGA 側のマスタ・モジュールがこのインターフェース経由で HPS 側の SDRAM Controller にアクセスする際に使用します。図では、FPGA-to-SDRAM とあるブロックが該当します。マスタ・モジュールからは、0x0000_0000 から 0xFFFF_FFFF の 4GByte のアドレス空間が見え、そのまま、HPS 側の SDRAM メモリのアドレスに対応します。

3-7. L3 インターコネクト・アドレス・リマップ・レジスタ

HPS 起動後に、ユーザがアドレス・マップを変更するためのレジスタ (0xFF80_0000) が用意されています。下表が該当のレジスタの概要ですが、mpuzero, nonmpuzero レジスタは、0x0 番地のアドレスのマッピングを選択するための設定となります。

hps2fpga, lwhps2fpga 設定は、アドレス空間で定義されている FPGA Slaves、Lightweight FPGA Slaves へのアクセスを有効化するための設定です。起動直後、これらの設定が有効となっていないと、HPS2FPGA、LWHPS2FPGA バス・インタフェース経由でのアクセスは実行できないので注意が必要です。

【remap レジスタ】

Bits	Name	Description
[0]	mpuzero	MPU L3 マスタの 0 番地のアドレス・マップ設定 0 : Boot ROM (デフォルト), 1 : On-chip RAM
[1]	nonmpuzero	Non-MPU L3 マスタの 0 番地のアドレス・マップ設定 0 : SDRAM (デフォルト), 1 : On-chip RAM
[2]	未定義	N/A
[3]	hps2fpga	L3 マスタからの HPS2FPGA バス・インタフェースの有効化設定 0 : 無効化 (デフォルト), 1 : 有効化
[4]	lwhps2fpga	L3 マスタからの LWHPS2FPGA バス・インタフェースの有効化設定 0 : 無効化 (デフォルト), 1 : 有効化
[31:5]	未定義	N/A

4. 参考資料

System Interconnect

http://www.altera.com/literature/hb/cyclone-v/cv_54004.pdf

http://www.altera.com/literature/hb/aria-v/av_54004.pdf

HPS-FPGA Bridges

http://www.altera.com/literature/hb/cyclone-v/cv_54005.pdf

http://www.altera.com/literature/hb/aria-v/av_54005.pdf

Cyclone V SoC HPS Address Map and Register Definitions

<http://www.altera.com/literature/hb/cyclone-v/hps.html>

Arria V SoC HPS Address Map and Register Definitions

<http://www.altera.com/literature/hb/aria-v/hps.html>

改版履歴

Revision	年月	概要
1	2014 年 9 月	新規作成

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。
 株式会社アルティマ ホームページ: <http://www.altima.co.jp> 技術情報サイト EDISON: <https://www.altima.jp/members/index.cfm>
 株式会社エルセナ ホームページ: <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。