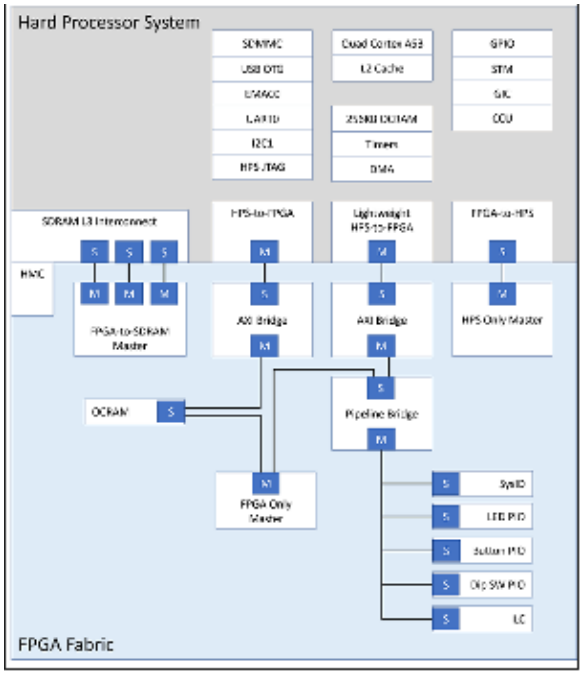
Altera Agilex 7 Issue in Accessing FPGA Fabric from HPS and Loading FPGA Bitfile

Objective: We are trying to access the FPGA fabric from the HPS region through the light weight AXI BUS.



Below is the address map of the region that we are trying to access.

Lightweight HPS-to-FPGA Address Map

The the memory map of system peripherals in the FPGA portion of the SoC as viewed by the MPU (Cortex-A53), which starts at the lightweight HPS-to-FPGA base address of **0xF900_0000**, is listed in the following table.

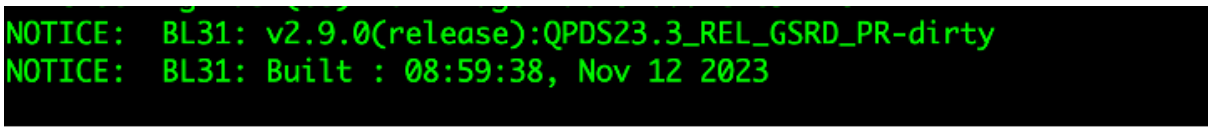
Peripheral	Address Offset	Size (bytes)	Attribute
sysid	0x0000_0000	8	Unique system ID
led_pio	0x0000_1080	16	LED outputs
button_pio	0x0000_1060	16	Push button inputs
dipsw_pio	0x0000_1070	16	DIP switch inputs
ILC	0x0000_1100	256	Interrupt latency counter (ILC)

Environment:

Hardware:

Development Kit Version	Ordering Code	Device Part Number	Starting Serial Number
Intel Agilex® 7 FPGA F-Series Transceiver-SoC Development Kit (Production 2 P-Tiles & E-Tiles)	DK-SI-AGF014EB	AGFB014R24B2E2V (Power Solution 2)	00205001

ATF version:



Uboot version:

GSRD from Rocketboard:

1. SOF
2. Uboot SPL

Issue:

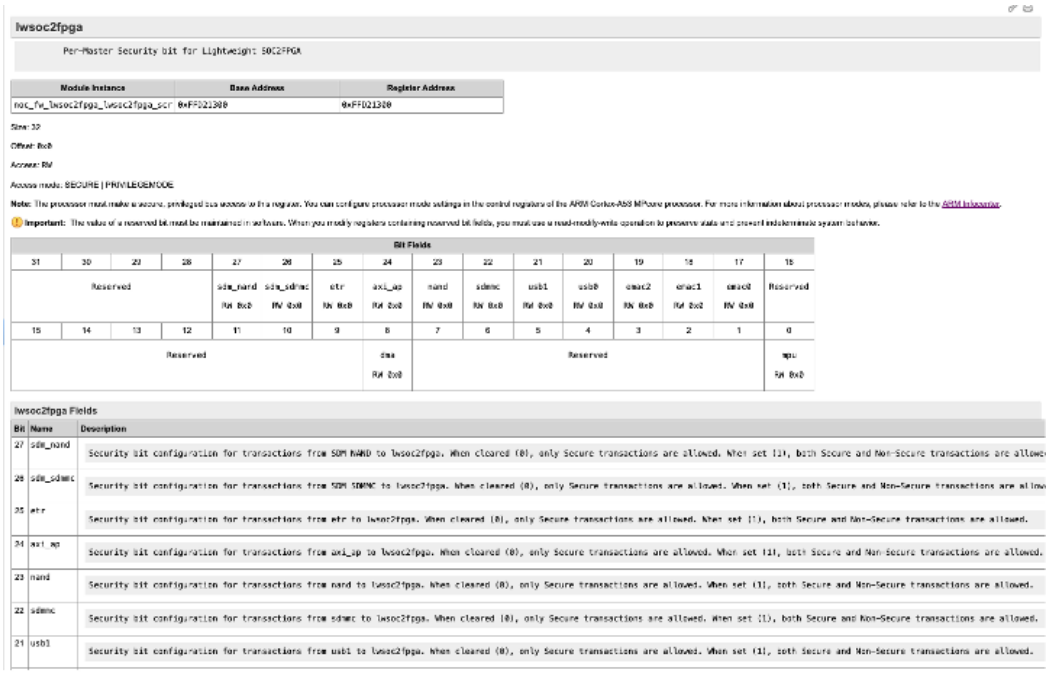
1. When trying to access the FPGA memory region 0xF900_0000 from the Uboot using the "md" command, the FPGA+Uboot crashes.

```
SOCFPGA_AGILEX # md f9000000
"Synchronous Abort" handler, esr 0x96000010
elr: 00000000002971f4 lr : 0000000000297144 (reloc)
elr: 000000007ff8e1f4 lr : 000000007ff8e144
x0 : 0000000000000009 x1 : 000000007faef268
x2 : 00000000fffffffe x3 : 0000000000000020
x4 : 0000000000000000 x5 : 000000007faef260
x6 : 0000000000000030 x7 : 000000007faef1b0
x8 : 0000000000000010 x9 : 0000000000000008
x10: 00000000fffffffd8 x11: 0000000000000010
x12: 000000000001869f x13: 000000007faef4d8
x14: 000000007faef5e0 x15: 0000000000000021
x16: 000000007ff13324 x17: 0000000000000000
x18: 000000007faf4da0 x19: 0000000000000004
x20: 0000000000000004 x21: 0000000000000004
x22: 00000000f9000000 x23: 000000007faef269
x24: 0000000000000000 x25: 000000007faef218
x26: 000000007ffa7a26 x27: 0000000000000008
x28: 0000000000000004 x29: 000000007faef1b0

Code: 2a0403f3 17ffffcb 7100129f 54000181 (b94002c3)
Resetting CPU ...

### ERROR ### Please RESET the board ###
```

2. Wherein if we try to access address map in HPS we are able to read it using the Uboot "md" command, e.g. UART at 0xFFC0_2000

```
SOCFPGA_AGILEX # md 0xffc02000
ffc02000: 00000000 00000000 000000c1 00000003  ................
ffc02010: 00000003 00000000 00000010 00000000  ................
ffc02020: 00000000 00000000 00000000 00000000  ................
ffc02030: 00000000 00000000 00000000 00000000  ................
ffc02040: 00000000 00000000 00000000 00000000  ................
ffc02050: 00000000 00000000 00000000 00000000  ................
ffc02060: 00000000 00000000 00000000 00000000  ................
ffc02070: 00000000 00000000 00000000 00000002  ................
ffc02080: 00000001 00000000 00000000 00000001  ................
ffc02090: 00000000 00000000 00000001 00000000  ................
ffc020a0: 00000000 00000000 00000000 00000000  ................
ffc020b0: 00000000 00000000 00000000 00000000  ................
ffc020c0: 00000000 00000000 00000000 00000000  ................
ffc020d0: 00000000 00000000 00000000 00000000  ................
ffc020e0: 00000000 00000000 00000000 00000000  ................
ffc020f0: 00000000 00083f32 3331352a 44570110  ....2?..*513..WD
```

3. As per the HPS Address Map and Register definition: https://www.intel.com/content/www/us/en/programmable/hps/agilex/hps.html#topic/qfi1561688682142.html



lwsoc2fpga

Per-Master Security bit for Lightweight SOC2FPGA

Module Instance	Base Address	Register Address
noc_fw_lwsoc2fpga_lwsoc2fpga_scr	0xFFD21300	0xFFD21308

Size: 32

Offset: 0x8

Access: RW

Access mode: SECURE | PRIVILEGEMODE

Note: The processor must make a secure, privileged bus access to this register. You can configure processor mode settings in the control registers of the ARM Cortex-A53 MPcore processor. For more information about processor modes, please refer to the ARM Infocenter.

Important: The value of a reserved bit must be maintained in software. When you modify registers containing reserved bit fields, you must use a read-modify-write operation to preserve state and prevent indeterminate system behavior.

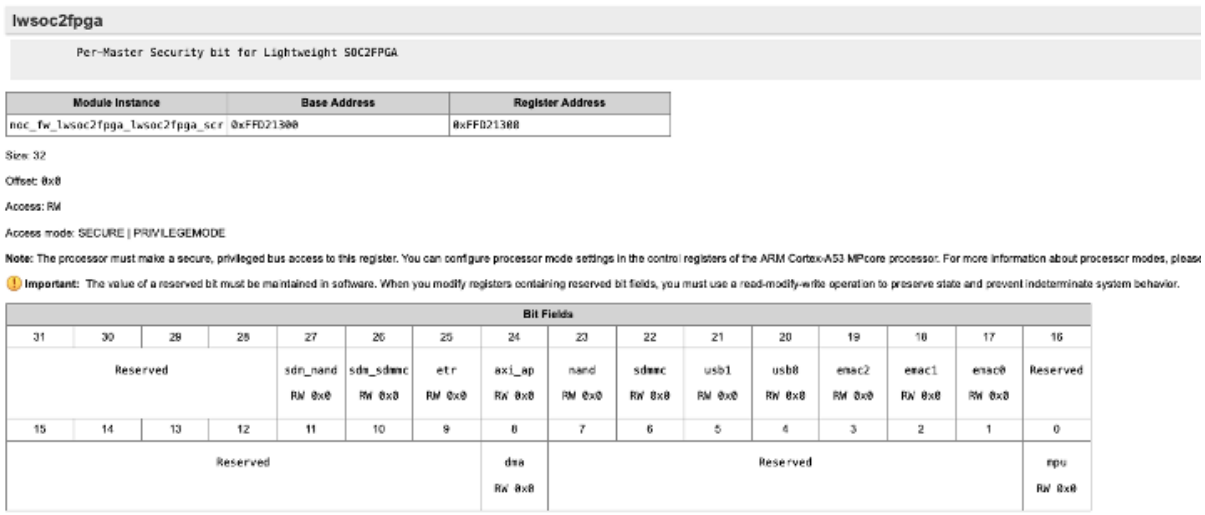
Bit Fields

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				sdm_nand RW 0x0	sdm_sdmmc RW 0x0	etr RW 0x0	axi_ap RW 0x0	nand RW 0x0	sdmmc RW 0x0	usb1 RW 0x0	usb0 RW 0x0	emac2 RW 0x0	emac1 RW 0x0	emac0 RW 0x0	Reserved
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							dma RW 0x0	Reserved							mpu RW 0x0

lwsoc2fpga Fields

Bit	Name	Description
27	sdm_nand	Security bit configuration for transactions from SDM NAND to lwsoc2fpga. When cleared (0), only Secure transactions are allowed. When set (1), both Secure and Non-Secure transactions are allowed.
26	sdm_sdmmc	Security bit configuration for transactions from SDM SDMMC to lwsoc2fpga. When cleared (0), only Secure transactions are allowed. When set (1), both Secure and Non-Secure transactions are allowed.
25	etr	Security bit configuration for transactions from etr to lwsoc2fpga. When cleared (0), only Secure transactions are allowed. When set (1), both Secure and Non-Secure transactions are allowed.
24	axi_ap	Security bit configuration for transactions from axi_ap to lwsoc2fpga. When cleared (0), only Secure transactions are allowed. When set (1), both Secure and Non-Secure transactions are allowed.
23	nand	Security bit configuration for transactions from nand to lwsoc2fpga. When cleared (0), only Secure transactions are allowed. When set (1), both Secure and Non-Secure transactions are allowed.
22	sdmmc	Security bit configuration for transactions from sdmmc to lwsoc2fpga. When cleared (0), only Secure transactions are allowed. When set (1), both Secure and Non-Secure transactions are allowed.
21	usb1	Security bit configuration for transactions from usb1 to lwsoc2fpga. When cleared (0), only Secure transactions are allowed. When set (1), both Secure and Non-Secure transactions are allowed.

The bit 24 in the lwsoc2fpga register should be set:



lwsoc2fpga

Per-Master Security bit for Lightweight SOC2FPGA

Module Instance	Base Address	Register Address
noc_fw_lwsoc2fpga_lwsoc2fpga_scr	0xFFD21300	0xFFD21308

Size: 32

Offset: 0x8

Access: RW

Access mode: SECURE | PRIVILEGEMODE

Note: The processor must make a secure, privileged bus access to this register. You can configure processor mode settings in the control registers of the ARM Cortex-A53 MPcore processor. For more information about processor modes, please

Important: The value of a reserved bit must be maintained in software. When you modify registers containing reserved bit fields, you must use a read-modify-write operation to preserve state and prevent indeterminate system behavior.

Bit Fields

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				sdm_nand RW 0x0	sdm_sdmmc RW 0x0	etr RW 0x0	axi_ap RW 0x0	nand RW 0x0	sdmmc RW 0x0	usb1 RW 0x0	usb0 RW 0x0	emac2 RW 0x0	emac1 RW 0x0	emac0 RW 0x0	Reserved
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							dma RW 0x0	Reserved							mpu RW 0x0

4. From the Uboot arch/arm/dts/socfpga_soc64_u-boot.dtsi file we can confirm this:

```
noc_fw_lwsoc2fpga_lwsoc2fpga_scr@ffd21300 {
        reg = <0xffd21300 0x00000004>;
        /* Disable lightweight soc2fpga security access */
        intel,offset-settings = <0x00000000 0x0ffe0101 0x0ffe0101>;
        u-boot,dm-pre-reloc;
```

};

5. In order to cross check, we enabled the SMC (Secure Monitor Call) in the Uboot and confirmed it.

```
SOCFPGA_AGILEX # smc ${smc_fid_rd} FFD21300
EL = 2
Res:  0 268304641 4291957504 0
```

The value 268304641 is in decimal and is the value contained in the register 0xFFD2_1300.

Expanded the value 268304641 in hexadecimal equivalent = 0x0FFE_0101

6. Few other registers were also not accessible from the Uboot console using SMC command. In order to access then I had to include the registers in the below file:

plat/intel/soc/common/socfpga_sip_svc.c in the function

static int is_out_of_sec_range(uint64_t reg_addr)

as:

```
case(0xFFD21200):   /* SOC2FPGA */
case(0xFFD21300):   /* SOC2FPGA Security Control Registers */
case(0xF9000000):   /* FPGA BRIDGE lwsoc2fpga 2MB */
case(0x00000000):   /* Hard_Memory_Ctrlr_DDRMemoryData_4G */
case(0xFFE00000):   /* OCRAM */
case(0xFFFC1000):   /* GIC */
case(0xF9000000):   /* LWHPS2FPGA SYSID */
```

7. I have confirmed that we are in EL2 in the Uboot console by adding the below code snippet:

```
register uint64_t x0 __asm__ ("x0");
__asm__ ("mrs x0, CurrentEL;" : : : "%x0");
printf("EL = %"PRIu64" \n", x0 >> 2);
```

Output:

EL = 2

But still we are not able to access the address map 0xF900_0000 and few other registers.

Loading the FPGA bitfile using TFTP:

1. tftp ${loadaddr} flash_image.core.rbf -> the .rbf file is the bitfile that is generated using the Quartus Program File Generator

```
ilename 'flash_image.core.rbf'.
oad address: 0x2000000
oading: ##################################################################
        ##########################################
        8.1 MiB/s
one
ytes transferred = 1622016 (18c000 hex)
```

2. fpga load 0 ${loadaddr} ${filesize}

After loading the FPGA bitfile we get the success message:

```
SOCFPGA_AGILEX # fpga load 0 ${loadaddr} ${filesize}
..FPGA reconfiguration OK!
```

3. Bridge enable command doesn't throw any error: bridge enable

```
SOCFPGA_AGILEX # bridge enable
SOCFPGA_AGILEX #
```

Resetting the brgmodrst:

I enabled the SMC command in the Uboot and tried to the steps that you have described:

Without the SMC command we can't access the register 0xFFD1102C and many other registers.

brgmodrst

The BRGMODRST register is used by software to control the bridge module resets. Software explicitly asserts and de-asserts module reset signals by writing bits in the appropriate *MODRST register. It is up to software to ensure module
Software writes a bit to 1 to assert the module reset signal and to 0 to de-assert the module reset signal.
All fields are reset by a cold reset. All fields are also reset by a warm reset if not masked by the corresponding BRGWARMMASK field.
The reset value of all fields is 1. This holds the corresponding module in reset until software is ready to release the module from reset by writing 0 to its field.

Module Instance	Base Address	Register Address
i_rst_mgr_rstmgr	0xFFD11000	0xFFD1102C

Size: 32

Offset: 0x2C

Access: RW

Access mode: PRIVILEGEMODE

Note: The processor must make a privileged bus access to this register. You can configure processor mode settings in the control registers of the ARM Cortex-A53 MPcore processor. For more information about processor modes, please refer to the ARM Infocenter.

Important: The value of a reserved bit must be maintained in software. When you modify registers containing reserved bit fields, you must use a read-modify-write operation to preserve state and prevent indeterminate system behavior.

Bit Fields

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									mpfe RW 0x1	Reserved			fpga2soc RW 0x1	lwhps2fpga RW 0x1	soc2fpga RW 0x1

brgmodrst Fields

Bit	Name	Description	Access	Reset
6	mpfe	Resets logic in the MPFE.	RW	0x1
2	fpga2soc	Resets FPGA2SOC Bridge.	RW	0x1
1	lwhps2fpga	Resets LWHPS2FPGA Bridge.	RW	0x1
0	soc2fpga	Resets SOC2FPGA Bridge.	RW	0x1

```
SOCFPGA_AGILEX # smc ${smc_fid_rd} 0xffd1102c
EL = 2
Res:  0 7 4291891244 0
SOCFPGA_AGILEX # smc ${smc_fid_wr} 0xffd1102c 0x00000000
EL = 2
Res:  0 0 4291891244 0
SOCFPGA_AGILEX # smc ${smc_fid_rd} 0xffd1102c
EL = 2
Res:  0 0 4291891244 0
SOCFPGA_AGILEX #
EL = 2
Res:  0 0 4291891244 0
SOCFPGA_AGILEX # md f9000000
"Synchronous Abort" handler, esr 0x96000010
elr: 00000000002971f4 lr : 0000000000297144 (reloc)
elr: 000000007ff8e1f4 lr : 000000007ff8e144
x0 : 0000000000000009 x1 : 000000007faef268
x2 : 00000000fffffffe x3 : 0000000000000020
x4 : 0000000000000000 x5 : 000000007faef260
x6 : 0000000000000030 x7 : 000000007faef1b0
x8 : 0000000000000010 x9 : 0000000000000008
x10: 00000000ffffffd8 x11: 0000000000000010
x12: 000000000001869f x13: 00000000000000cc
x14: 0000000000000006 x15: 0000000000000021
x16: 000000007ff13324 x17: 0000000000000000
x18: 000000007faf4da0 x19: 0000000000000004
x20: 0000000000000004 x21: 0000000000000004
x22: 00000000f9000000 x23: 000000007faef269
x24: 0000000000000000 x25: 000000007faef218
x26: 000000007ffa7a26 x27: 0000000000000008
x28: 0000000000000004 x29: 000000007faef1b0

Code: 2a0403f3 17ffffcb 7100129f 54000181 (b94002c3)
Resetting CPU ...

### ERROR ### Please RESET the board ###
```

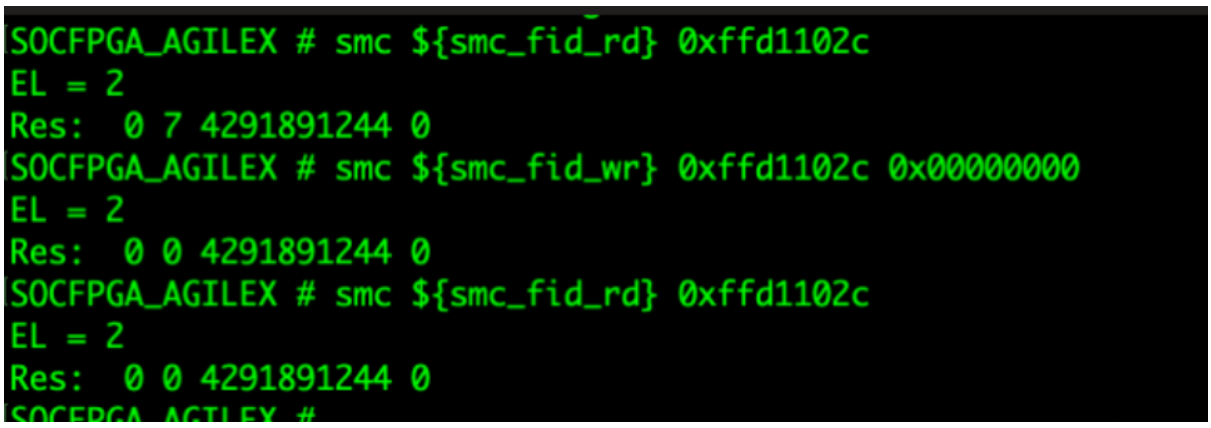
Enabling the AXI bus:
I have enabled the AXI bus command in the Uboot.

The command "AXI bus" doesn't show any AXI bus information.

```
SOCFPGA_AGILEX # axi
axi - AXI sub-system

Usage:
axi bus  - show AXI bus info
axi dev [bus] - show or set current AXI bus to bus number [bus]
axi md size addr [# of objects] - read from AXI device at address [addr] and data width [size] (one of 8, 16, 32)
axi mw size addr value [count] - write data [value] to AXI device at address [addr] and data width [size] (one of 8, 16, 32)

SOCFPGA_AGILEX # axi bus
EL = 2
SOCFPGA_AGILEX #
```

I enabled the SMC command in the Uboot and tried to the steps that you have described:

Without the SMC command we can’t access the register 0xFFD1102C and many other registers.



But, still not able to access the address map 0xF900_0000.

```
SOCFPGA_AGILEX # md f9000000
"Synchronous Abort" handler, esr 0x96000010
elr: 00000000002971f4 lr : 0000000000297144 (reloc)
elr: 000000007ff8e1f4 lr : 000000007ff8e144
x0 : 0000000000000009 x1 : 000000007faef268
x2 : 00000000fffffffe x3 : 0000000000000020
x4 : 0000000000000000 x5 : 000000007faef260
x6 : 0000000000000030 x7 : 000000007faef1b0
x8 : 0000000000000010 x9 : 0000000000000008
x10: 00000000ffffffd8 x11: 0000000000000010
x12: 000000000001869f x13: 00000000000000cc
x14: 0000000000000006 x15: 0000000000000021
x16: 000000007ff13324 x17: 0000000000000000
x18: 000000007faf4da0 x19: 0000000000000004
x20: 0000000000000004 x21: 0000000000000004
x22: 00000000f9000000 x23: 000000007faef269
x24: 0000000000000000 x25: 000000007faef218
x26: 000000007ffa7a26 x27: 0000000000000008
x28: 0000000000000004 x29: 000000007faef1b0

Code: 2a0403f3 17ffffcb 7100129f 54000181 (b94002c3)
Resetting CPU ...
```

I reset the BRGWARMMASK register:

Module Instance	Base Address	Register Address
i_rst_mgr_rstmgr	0xFFD11000	0xFFD1104C

```
SOCFPGA_AGILEX # smc ${smc_fid_rd} 0xffd1104c
EL = 2
Res:  0 7 4291891276 0
SOCFPGA_AGILEX # smc ${smc_fid_wr} 0xffd1104c 5
EL = 2
Res:  0 5 4291891276 0
```

No success.

I modified the Agilex7 default configuration to pull in the .RBF file from the MMC as instructed like in the Rocketboard website:

```
Volume env not found!

** Unable to read env from root:env **
In:    serial0@ffc02000
Out:   serial0@ffc02000
Err:   serial0@ffc02000
Net:
Warning: ethernet@ff800000 (eth0) using random MAC address - 3e:3b:12:75
eth0: ethernet@ff800000
Hit any key to stop autoboot:  0
1617920 bytes read in 77 ms (20 MiB/s)
FPGA not ready. Bridge reset aborted!
..FPGA reconfiguration OK!
SOCFPGA_AGILEX #
```

I just read the register fpga_config:

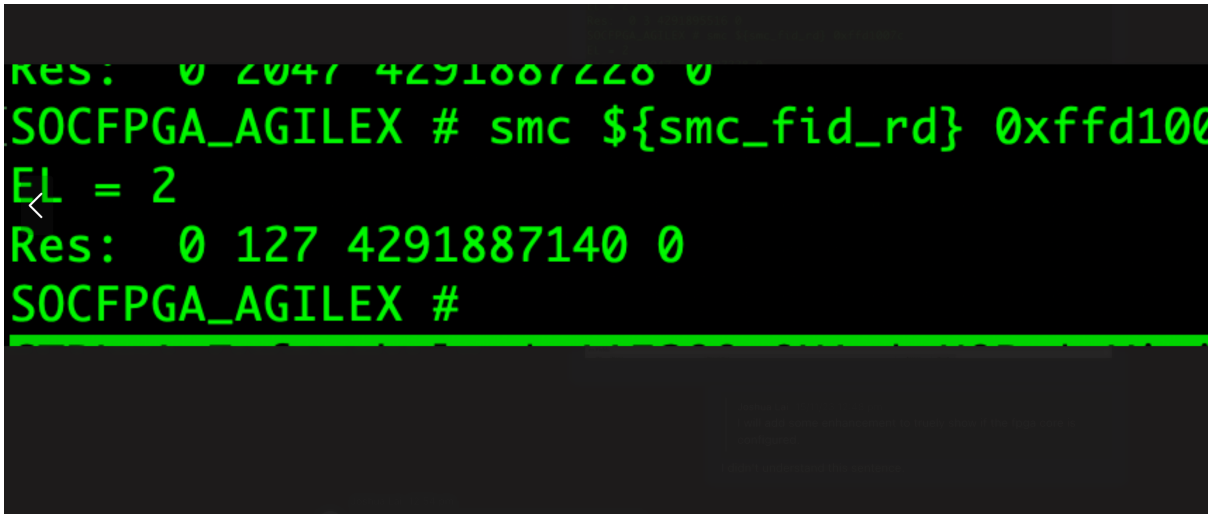
```
1617920 bytes read in 77 ms (20 MiB/s)
FPGA not ready. Bridge reset aborted!
..FPGA reconfiguration OK!
SOCFPGA_AGILEX # smc ${smc_fid_rd} 0xffd120dc
EL = 2
Res:  0 3 4291895516 0
```

Reference:

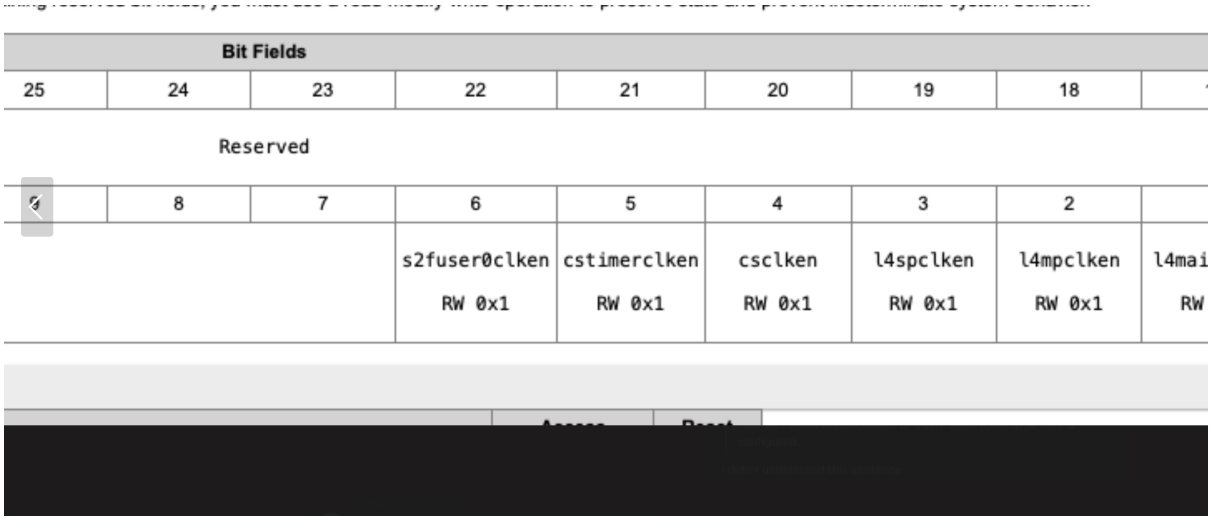
fpga_config Fields

Bit	Name	Description	Access	Reset
1	early_usermode	FGPA configuration complete	RO	0x0
0	fpga_complete	FGPA configuration complete	RO	0x0

The clocks are also enabled:



Reference:



Bit Fields								
25	24	23	22	21	20	19	18	
Reserved								
	8	7	6	5	4	3	2	
			s2fuser0clken RW 0x1	cstimerclken RW 0x1	csclken RW 0x1	l4spclken RW 0x1	l4mpclken RW 0x1	l4mai RW

Question:

1. What wrong I am doing?
2. How can we access the FPGA address map from the HPS using the light weight AXI bus?