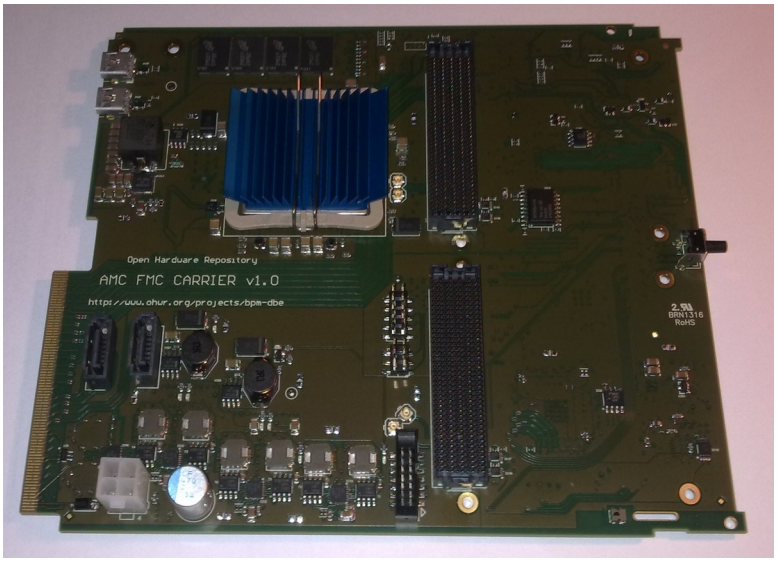
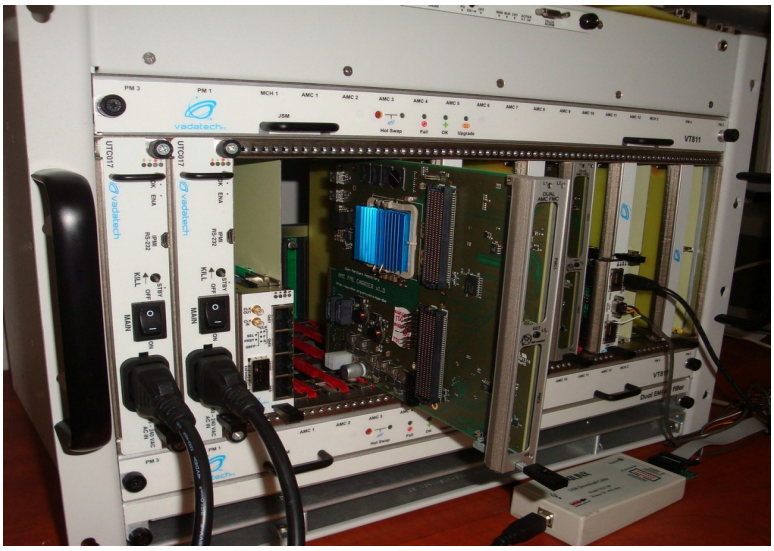
AMC FMC Carrier with Kintex 7 FPGA (AFCK)

Specification

Grzegorz Kasprowicz
Warsaw University of Technology

v0.1





The AMC FMC carrier is partially based on SPEC (supply, WR clocks) design. It was primary designed to support quad, 250MHz, 16 bit ADC FMC boards for the BPM back-end. Among many features, the card has very flexible clock circuit that enables any clock source to be connected to any clock input, including telecom clock, FMC clocks and FPGA.
It is entirely open-hardware, open-source design, available at OHW Repository:
http://www.ohwr.org/projects/afck/wiki

Features:

Programmable resources:

- Xilinx Kintex-7 325T FFG900 FPGA
- MMC: LPC1764FBD100, optionally Atxmega128A1U-AU

Memory

- 2GB (16Gbit) DDR3 SDRAM (32-bit interface), 800MHz (clock)
- SPI Flash for FPGA configuration. Accessible by MMC
- SPI Flash for user data storage
- EEPROM with MAC and unique ID

Connectivity

- 2 high pin count (HPC) slots for 2 single width mezzanines or 1 double width mezzanine
- Mini-USB connected to the MMC processor
- Mini-USB UART connected to FPGA or MMC
- Stand-alone power connector (12V, 3.3V aux)
- SATA connector for Port2, Port3 with possibility of switching to FPGA GTP
- GTP connected to FMC1 (x4), FMC2(x4), FP1(x4), FP2(x4), Port0, Port1, RTM(x8), selected by capacitor placement

Supply

- Power supply for FPGA, memory, FMCs - programmable VADJ 1.8-3.3V (independent for each FMC)
- Monitoring of voltage and current of all FMC buses

Clocking

- Clock distribution circuit compatible with White Rabbit
- Clock crossbar, 16 inputs x 16 outputs.

Other

- Temperature monitoring: FMC1, FMC2, supply, FPGA core, DDR memory
- JTAG multiplexer (SCANSTA) for FMC access, local JTAG port and remote debug/Chipscope via Ethernet

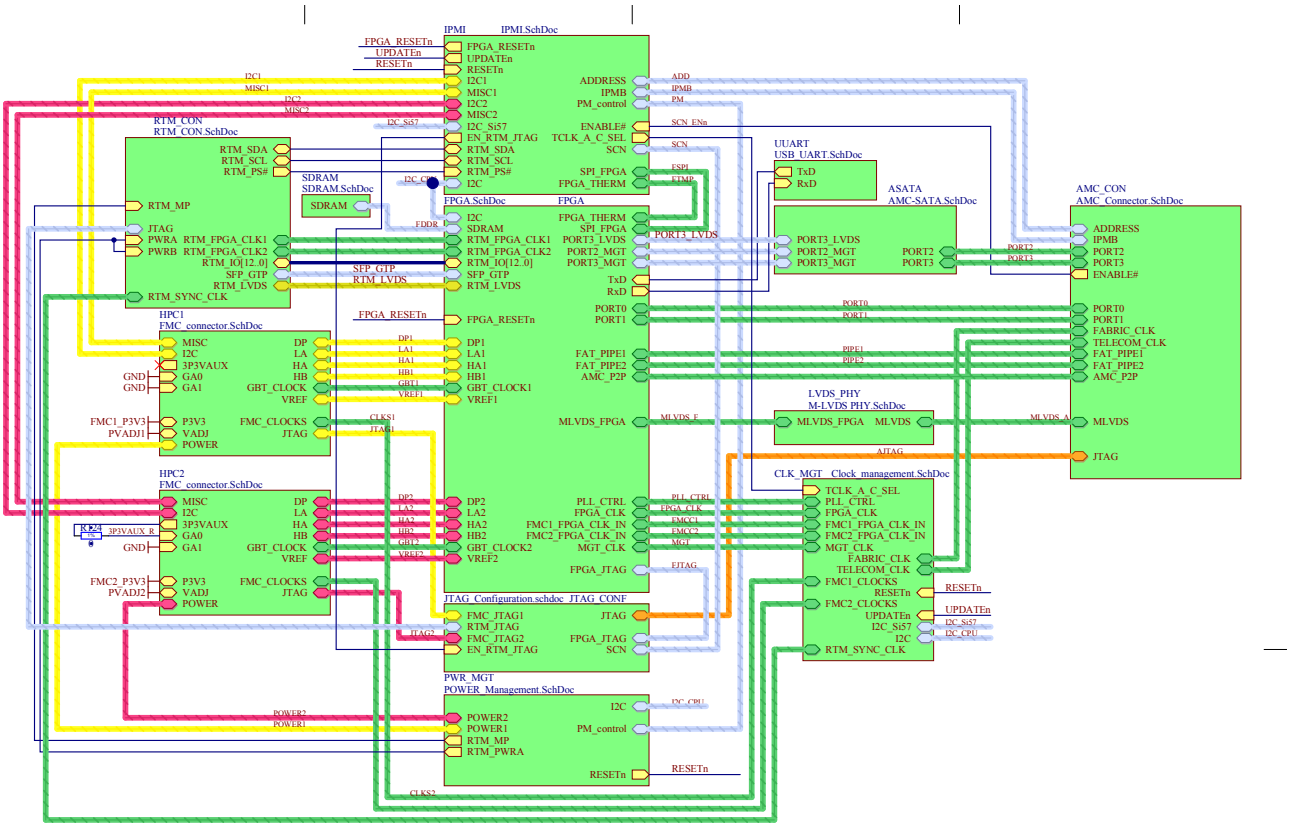


Figure 1: AMC FMC K block schematic

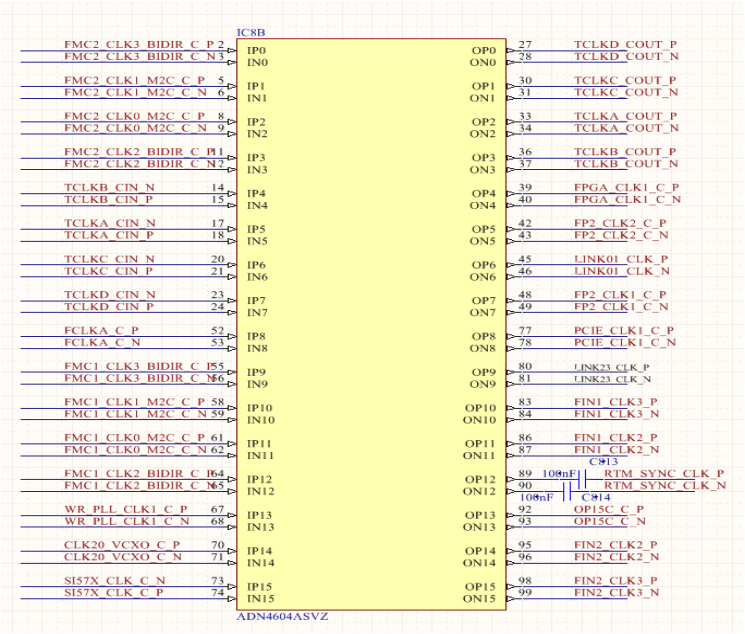


Figure 2: Clock crossbar signals assignment

GTX routing

MGT	ports	Option1	option2	Option3
118	0	FMC1 DP3	RTM6	-
118	1	FMC1 DP2	RTM7	-
118	2	FMC1 DP1	RTM8	-
118	3	FMC1 DP0	RTM5	-
118	clk0	FMC1_GBTCLK1_M2C	LINK23_CLK (clk crossbar)	RTM_CLK1
118	clk1	FMC1_GBTCLK0_M2C	-	-
117	0	FMC2 DP4	RTM1	AMC P2P PORT 15
117	1	FMC2 DP5	RTM2	AMC P2P PORT 14
117	2	FMC2 DP6	RTM3	AMC P2P PORT 13
117	3	FMC2 DP7	RTM4	AMC P2P PORT 12
117	clk0	FMC2_GBTCLK0_M2C	-	RTM_CLK2
117	clk1	FMC2_GBTCLK1_M2C	LINK01_CLK (clk crossbar)	-
116	0	AMC FP2 PORT 8	AMC PORT 3	FMC1_DP7
116	1	AMC FP2 PORT 9	AMC PORT 2	FMC1_DP6
116	2	AMC FP2 PORT 10	AMC PORT1	FMC1_DP5
116	3	AMC FP2 PORT 11	AMC PORT0	FMC1_DP4
116	clk0	FP2_CLK2_C (clk crossbar)	-	-
116	clk1	FP2_CLK1_C (clk crossbar)	-	-
115	0	AMC FP1 PORT 7	-	FMC2_DP3
115	1	AMC FP1 PORT 6	-	FMC2_DP2
115	2	AMC FP1 PORT 5	-	FMC2_DP1
115	3	AMC FP1 PORT 4	-	FMC2_DP0
115	clk0	PCIE_CLK1_C (clk crossbar)	-	-
115	clk1	PCIE_CLK1_C (clk crossbar)	-	-

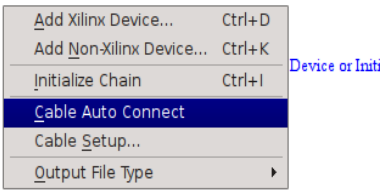
Table 1: MGT connectivity options

1. Configuring the SCANSTA JTAG switch

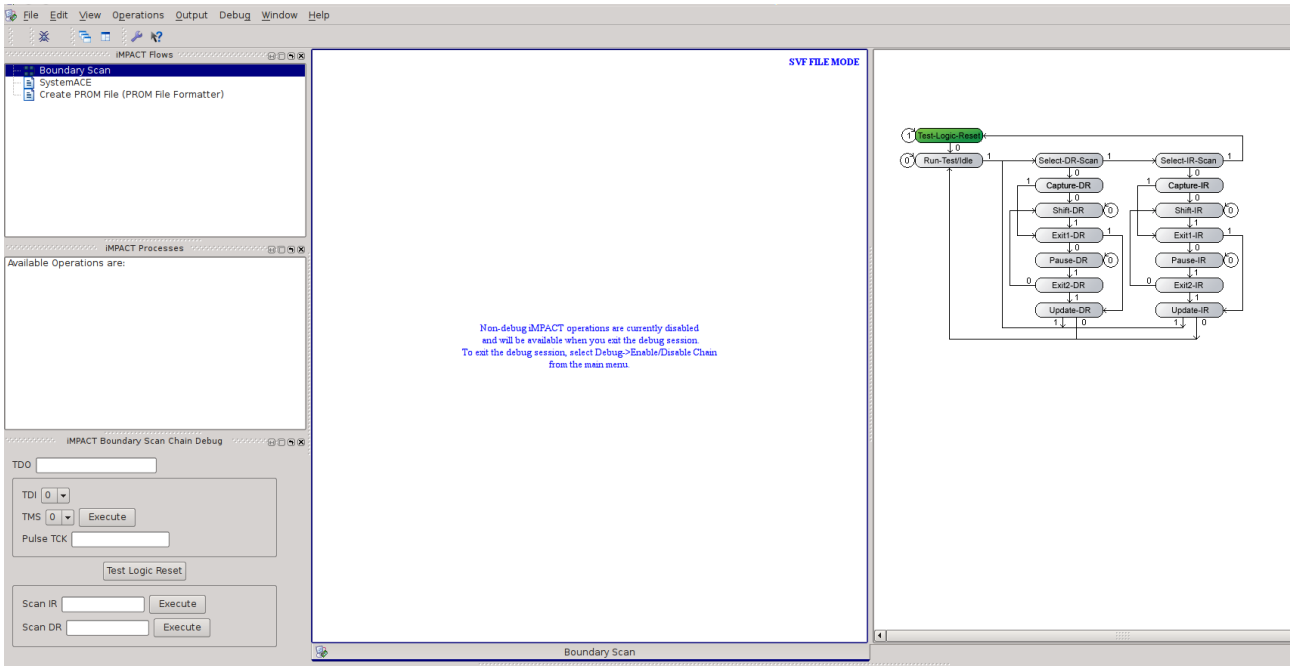
The AFC card features Texas Instruments SCANSTA switch, which is used for multiplexing between JTAG interfaces of main FPGA and FMC cards. Due to it's presence, programming of main FPGA with Xilinx JTAG will fail until SCANSTA is configured in Transparent Bridge mode. It can be done with standard Xilinx JTAG programmer and IMPACT software. The necessary JTAG commands are:

```
SIR 8 TDI (00); ! 00000000 Address ScanBridge
SIR 8 TDI (A0); ! 10100000 Load instruction to enable transparent mode for LSP0
SIR 8 TDI (a5); ! 10100101 Verify SIR
SDR 8 TDI (5a); ! 01011010 Verify SDR
SIR 8 TDI (C3); ! 11000011 Try to load GOTOWAIT in ScanBridge
SDR 8 TDI (5a); ! 01011010 Verify that ScanBridge did not recognize GOTOWAIT
! Now TDIB › lsp0 › TDOB
```

To do so with the IMPACT programmer:
1. Connect all necessary cable, power on AFC board
2. Start IMPACT software and choose 'Boundary Scan' option like for normal programming. Auto Connect cable.



3. Switch to debug mode, from menu bar choose: Debug → Enable/Disable debug chain. Your IMPACT window should look like this:



The 'SIR' commands must be entered in 'Scan IR' text box, while 'SDR' commands in 'Scan DR' text box. After executing last command, TDO output should be '00000000'. You can now proceed to normal programming procedure.
It's also possible to embed these commands at the beginning of SVF file. It should be also possible to have the SCANSTA configuration executed by IMPACT in batch mode.