Release 2 GFE System Description

1 Capabilities, System and Components

Figure 1 shows a block diagram of the Release 2 GFE hardware. The system in the XCVU9P runs bare-metal code. A host-based gdb debugger connects to the system over the USB/JTAG connector, and a host-based console connects over the USB/UART connector. The Release 2 system can access 1 GB of the 4GB on-board DRAM through the Xilinx DDR controller and PHY.

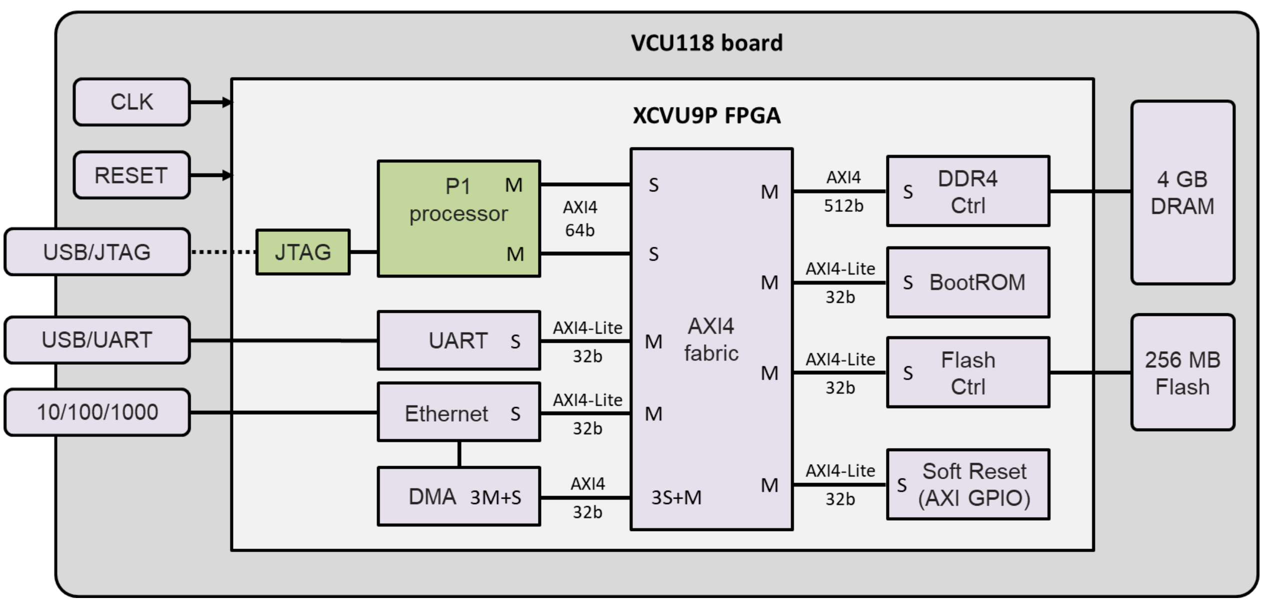

Figure 1: GFE P1/P2 VCU118 FPGA board and device block diagram

The GFE system consists of a Rocket, Piccolo or Flute RISC-V core (see specifications below), a BootROM, Soft Reset and JTAG, UART, Ethernet/DMA, DDR4 and Flash controllers. All components but the P{1,2} processors and the JTAG block (green) are Xilinx IP (purple). The BootROM is a read only memory constructed from a BRAM and AXI BRAM controller. The GFE system is the same whether the processor is Rocket, Piccolo or Flute.

The P{1,2} processors include Debug Modules (not shown) that connect to a custom JTAG block (https://gitlab-ext.galois.com/ssith/gfe/tree/master/jtag) that uses Xilinx BSCAN primitives under the hood. This enables a tap into the FPGA's JTAG system controller to multiplex the P{1,2} JTAG channel onto the external USB/JTAG port (dotted line connection).

The components are connected through an AXI4 fabric with 12 ports. The fabric automatically translates between protocols (AXI4-Lite and AXI4) and channel widths (32, 64, 256, 512 bit).

See section "Xilinx UltraScale+ Implementation" for the Xilinx circuit implementation of this block diagram.

Table 1 lists the Xilinx IP cores used in the GFE system:

Component	Product	Documentation
AXI Interconnect	PG059	https://www.xilinx.com/support/documentation/ip_documentation/axi_interconnect/v2_1/pg059-axi-interconnect.pdf
DDR4 Controller	PG150	https://www.xilinx.com/support/documentation/ip_documentation/ultrascale_memory_ip/v1_4/pg150-ultrascale-memory-ip.pdf
AXI BRAM Controller	PG078	https://www.xilinx.com/support/documentation/ip_documentation/axi_bram_ctrl/v4_0/pg078-axi-bram-ctrl.pdf
AXI UART 16550	PG143	https://www.xilinx.com/support/documentation/ip_documentation/axi_uart16550/v2_0/pg143-axi-uart16550.pdf
AXI 1G/2.5G Ethernet	PG138	https://www.xilinx.com/support/documentation/ip_documentation/axi_ethernet/v7_0/pg138-axi-ethernet.pdf
AXI DMA	PG021	https://www.xilinx.com/support/documentation/ip_documentation/axi_dma/v7_1/pg021_axi_dma.pdf
AXI Quad SPI (Flash ctrl)	PG153	https://www.xilinx.com/support/documentation/ip_documentation/axi_quad_spi/v3_2/pg153-axi-quad-spi.pdf
AXI GPIO Controller	PG144	https://www.xilinx.com/support/documentation/ip_documentation/axi_gpio/v2_0/pg144-axi-gpio.pdf

Table 1: Xilinx IP Cores

2 P1, P2 PROCESSOR SPECIFICATIONS

	P1 Rocket	P1 Piccolo	P2 Rocket	P2 Flute
Instruction set	RV32IMAC		RV64IMAFDC	
Privilege levels	machine, user		machine, user, supervisor (SV39)	
Memory management	No		Yes	
Physical memory protection	No		No	
Compliance tests	https://github.com/riscv/riscv-tests			
	rv32u{i,m,a,c}-p, rv32mi-p		rv64u{i,m,a,f,d,c}-{p,v}, rv64{m,s}i-{p,v}	
Pipeline stages	5	3	5	5
Default GFE clock MHz	83	83	83	83
Max GFE clock MHz	150	83	150	83
Superscalar	No			
Out-of-order	No			
ICache/DCache	4 KB		8 KB	
Multiply-Divide unit	Yes			
Single/double FPU unit	No		Yes	
RISC-V debug module	RISC-V External Debug Support Version 0.13			
	program buffer	system bus	program buffer	system bus
Interrupts (PLIC)	Chapter 8, SiFive U54-MC-RVCoreIP v1p0 – 16 interrupts			
Timer (CLINT)	Chapter 9, SiFive U54-MC-RVCoreIP v1p0			
System bus	64-bit AXI4			
Operating system	FreeRTOS		Linux kernel	
Implementation language	Chisel	BSV	Chisel	BSV

“Default GFE clock MHz” is the current frequency setting for the ACLK signal (see Figures 2 and 3) that clocks all GFE components except the DDR4 controller. ACLK is generated by the clock generator inside the DDR4 controller (for convenience, not necessity). The ACLK frequency can be modified by adjusting the advanced clocking settings of the DDR controller.

“Max GFE clock MHz” is the maximum clock frequency at which the GFE has been tested with Rocket and Piccolo.

For Rocket, it may be possible to run faster than 150 MHz if the rest of the system can keep up (this hasn't been tried). For Piccolo, the GFE clock frequency is currently limited by Piccolo, not the rest of the system.

3 XILINX ULTRASCALE+ DESIGN IMPLEMENTATION

The block diagram in Figure 1 is implemented using Vivado's IP Integrator design flow:
https://www.xilinx.com/content/dam/xilinx/support/documentation/sw_manuals/xilinx2018_3/ug995-vivado-ip-subsystems-tutorial.pdf

The IP Integrator enables quick and easy graphical connections of Xilinx IP that is instantiated and configured within an IP Integrator project session. In addition, the P{1,2} processors are imported into IP Integrator as 3rd party reusable IP blocks that can be connected as easily as Xilinx IP.

The top-level GFE system is a Vivado block design. It consists of three instances (Figure 2): a Xilinx JTAG module (xilinx_jtag_0) and a P{1,2} processor (mkP2_Core_0) that correspond to blocks in Figure 1; and a subsystem module (gfe_subsystem) that instantiates all the other Xilinx modules.

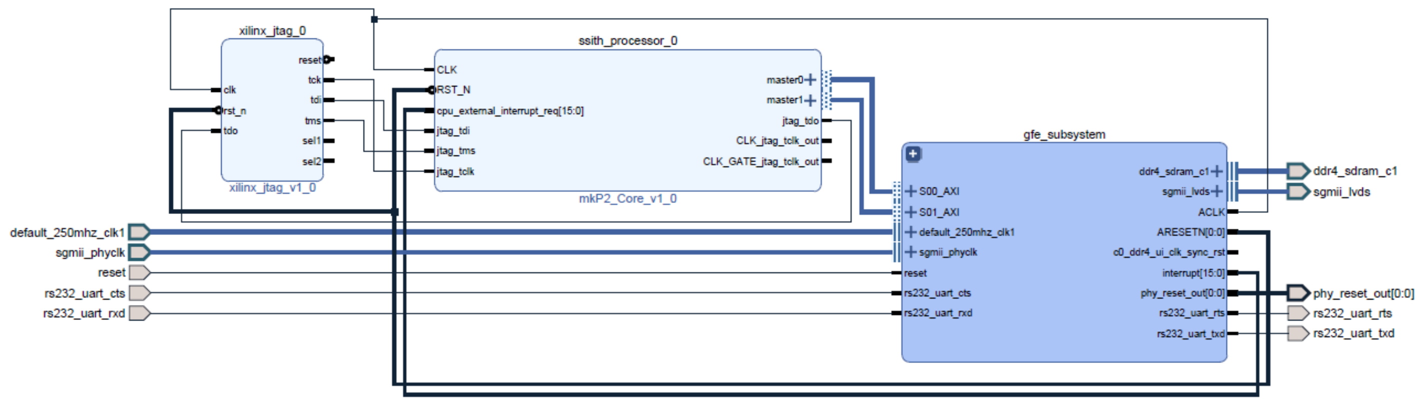

Figure 2: Top-level Vivado block design for Release 2 GFE

The *gfe_subsystem* module is shown in Figure 3. It instantiates all the Xilinx cores that are necessary to implement the fabric, UART, DDR4, Ethernet, Flash, BootROM and GPIO. in Figure 1. In addition, it instantiates two blocks to implement reset.

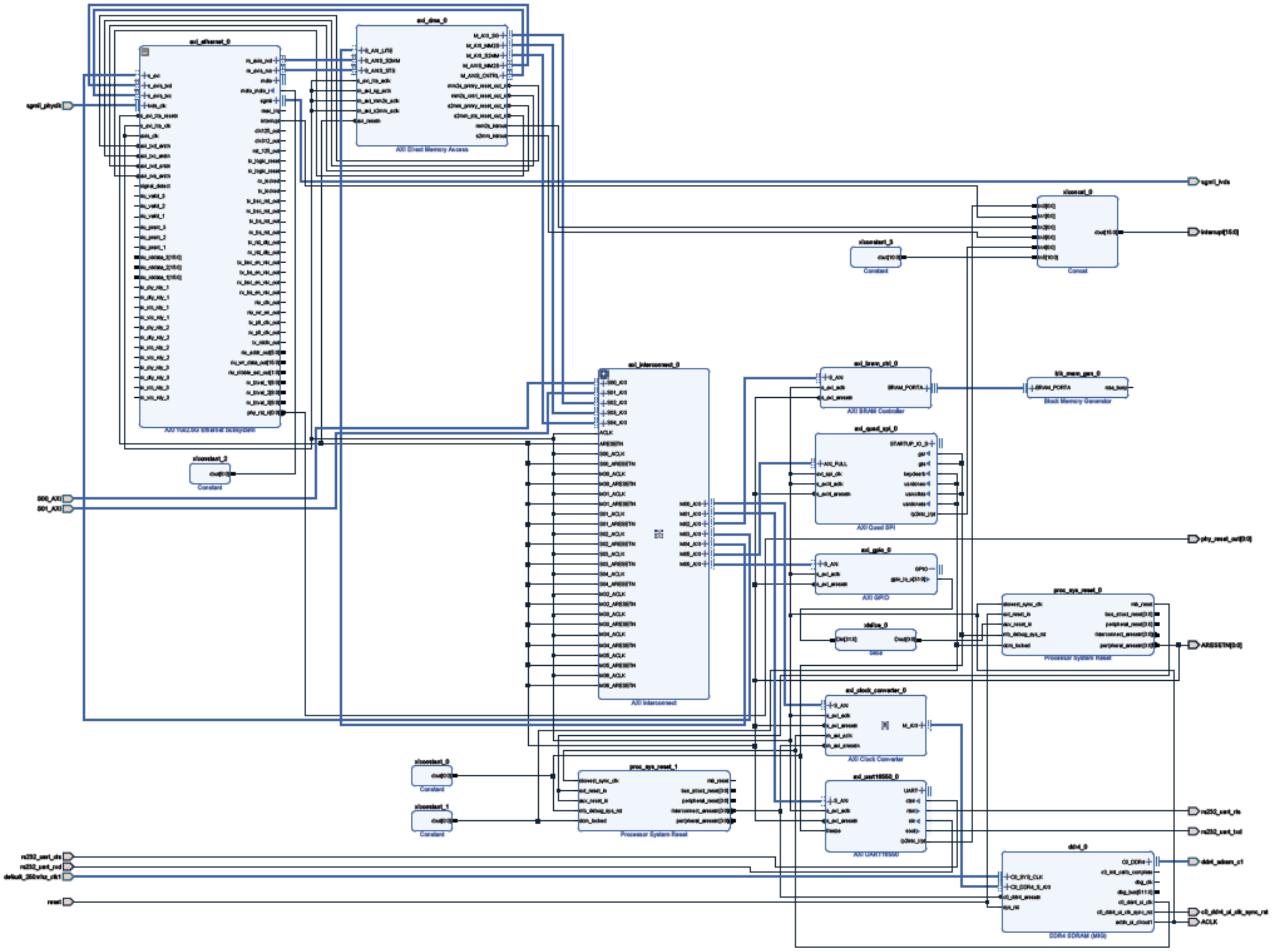

Figure 3: Xilinx block design implementation for *gfe_subsystem*

4 ADDRESS MAPPING

The following table provides the memory map for the memory mapped devices. The CLINT and PLIC are included inside the P{1,2} processors and are therefore not mapped into the AXI fabric.

Device	GFE Instance Name	Slave Interface	Base Name	Offset Address	Range	High Address
CLINT	addressed inside processor			0x0000_0000_1000_0000	64K	0x0000_0000_1000_FFFF
PLIC	addressed inside processor			0x0000_0000_0C00_0000	4M	0x0000_0000_0C3F_FFFF
FLASH	axi_quad_spi	AXI_FULL	MEM0	0x0000_0000_4000_0000	64K	0x0000_0000_4000_FFFF
ETHERNET	axi_ethernet_0	s_axi	Reg0	0x0000_0000_6210_0000	256K	0x0000_0000_6213_FFFF
DMA	axi_dma_0	S_AXI_LITE	Reg	0x0000_0000_6220_0000	64K	0x0000_0000_6220_FFFF
UART	axi_uart16550_0	S_AXI	Reg	0x0000_0000_6230_0000	4K	0x0000_0000_6230_0FFF
GPIO	axi_gpio_0	S_AXI	Reg	0x0000_0000_6FFF_0000	64K	0x0000_0000_6FFF_FFFF
Boot ROM	axi_bram_crtl_0	S_AXI	Mem0	0x0000_0000_7000_0000	4K	0x0000_0000_7000_0FFF
DRAM	dd4r_0	C0_DDR4_S_AXI	C0_DDR4_S_AXI_ADDRESS_BLOCK	0x0000_0000_8000_0000	2G	0x0000_0000_FFFF_FFFF

The memory map for devices connected to the AXI fabric can be viewed in the “Address Editor” tab next to the “Diagram” tab in the Vivado GFE project window.

Addresses 0x8000_0000 to 0xBFFF_FFFF are configured as uncached in the Bluespec and Chisel processors. This designation enables coherent memory between the DMA engine and RISC-V processors. The Linux device tree (bootrom/devicetree.dts) reserves this region for DMA.

Note that the first UART register is at 0x6230_0000, not 0x6230_1000 as the UART documentation would suggest. This is a result of the small address space allocated to UART in the Xilinx interconnect. The interconnect allocates 0x6230_0000 to 0x6230_0fff to the UART, but the UART registers in the Xilinx documentation are defined at 0x6230_100X. Synthesis causes the upper bits to be ignored within the UART block, so the resulting UART registers are located at 0x6230_000X

Our reference code uses the proper addresses (0x6230_000X). Writing to and reading from 0x6230_100X causes undefined behavior (this is processor dependent).

5 INTERRUPT MAPPING

The following table shows the interrupt mapping for cpu_external_interrupt_req[15:0]:

Pin	Connection
0	uart
1	interrupt pin on axi_ethernet
2	mm2s_introut on axi_dma
3	s2mm_introut on axi_dma
4	quad_spi
15:5	tied to zero

The PLIC used by the GFE processors indexes interrupts starting from 1. This means that the interrupt signal connected to cpu_external_interrupt_req[0] corresponds to PLIC interrupt 1. This offset is reflected in the Linux device tree provided in bootrom/devicetree.dts.

6 FREERTOS

The P1 processors boot FreeRTOS 10.2.0.

The Galois FreeRTOS repository is a fork of https://github.com/coldnew/FreeRTOS-mirror which is just a mirror of FreeRTOS SVN, the upstream FreeRTOS. The folder FreeRTOS-mirror/FreeRTOS/Demo/RISC-V_Galois_P1/demo contains applications specific to the P1 and GFE.

See gfe/README.md for instructions on how to run the FreeRTOS demo applications.

7 Linux Kernel

The P2 processors boot the Linux 4.20 kernel.

The kernel configuration is in the repository at gfe/bootmem/linux.config. It indicates which Linux drivers are in use.

The "Linux/Busybox" test exercises the basic kernel boot and initialization of devices. See gfe/README.md for instructions on how to boot Linux.