20201208工作日志

待解决：

1. 总体结构；
2. ux607 的Memory Map细节，地址划分，如何配置；
3. System Bus与PPI的细节：如何与core连接；

1.总体结构：

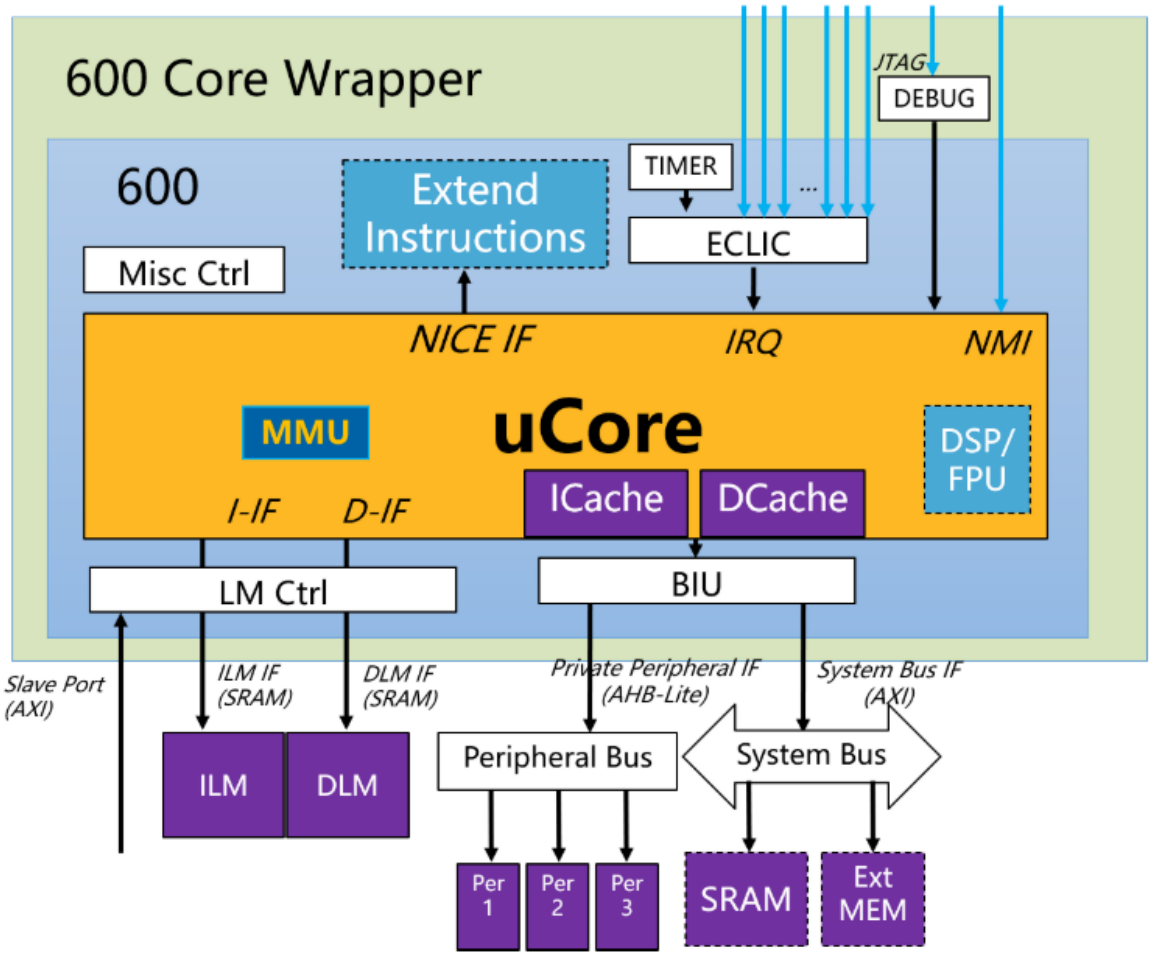


Figure 1-1 The top diagram of 600 Series Core

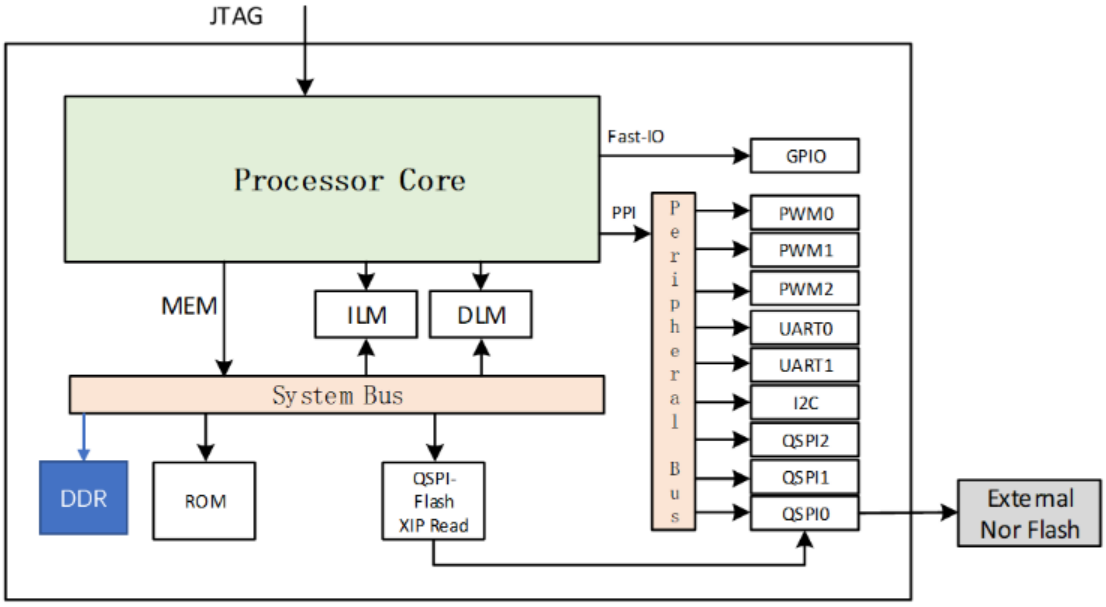
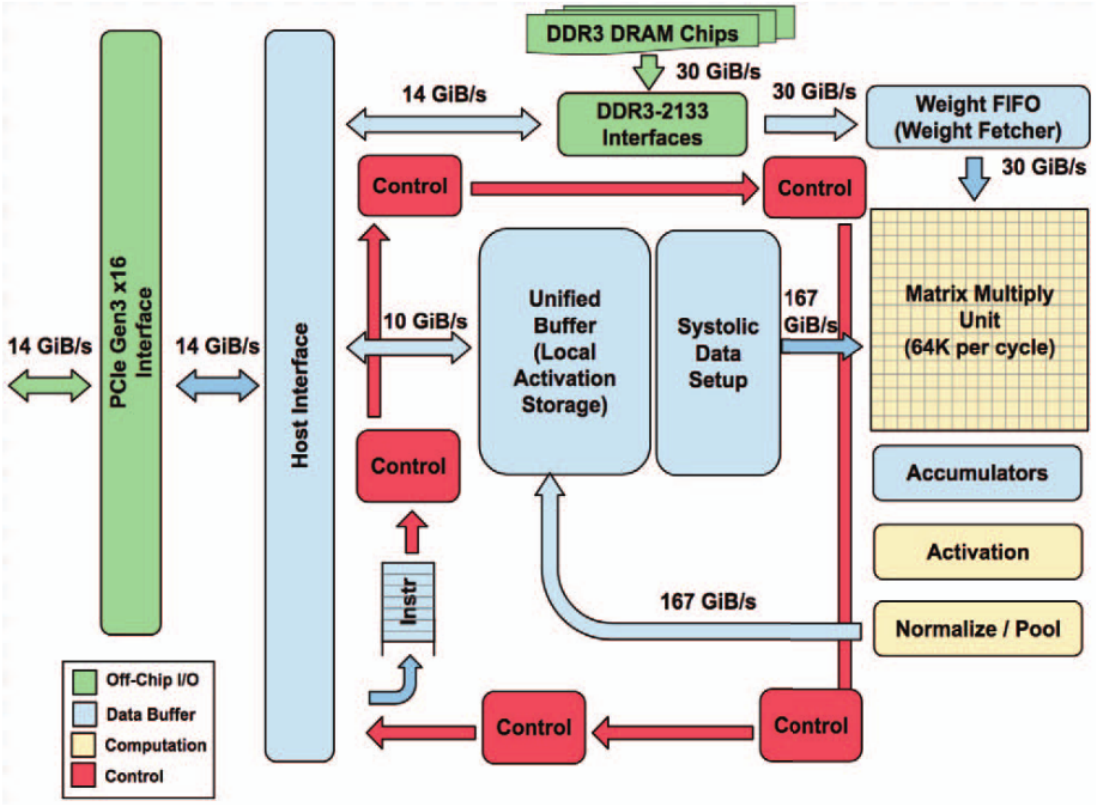


Figure 3-1 Nuclei Evaluation SoC (Hummingbird SoC)

Note: DDR will only exist in UX class cores.



疑问:

1. There is just 1~2 instructions in ROM, after executed the instructions, Core will to jump to ILM (address at 0x8000_0000). That is to say, in SIMULATION environment, the Core will just jump to ILM after reset and start execution from ILM
2.

注意事项:

1. ROM :4kb
2. Soc存储空间：ILM,DLM（空间可配置）、FLASH（外部，可通过QSPI通信）

3. QSPI通过GPIO与外界连接
4. NMI:often used to indicate system-level emergency errors (such as external hardware failures, etc.)
5. 打开宏定义DDR_CONTROLLER,可以添加DDR；
6.

2.地址划分：

(详见Nuclei_Eval_Soc_Intro 5:Address Allocation of SoC)

	Component	Address Spaces	Description
Core Private Peripherals	TIMER	0x0200_0000 ~ 0x0200_0FFF	TIMER Unit address space.
	ECLIC	0x0C00_0000 ~ 0x0C00_FFFF	ECLIC Unit address space.
	DEBUG	0x0000_0000 ~ 0x0000_0FFF	DEBUG Unit address space.
Memory Resource	ILM	0x8000_0000 ~	ILM address space.
	DLM	0x9000_0000 ~	DLM address space.
	ROM	0x0000_1000 ~ 0x0000_1FFF	Internal ROM.
	Off-Chip QSPI0 Flash Read	0x2000_0000 ~ 0x3FFF_FFFF	QSPI0 with XiP mode read-only address space.
Peripherals	GPIO	0x1001_2000 ~ 0x1001_2FFF	GPIO Unit address space.
	UART0	0x1001_3000 ~ 0x1001_3FFF	First UART address space.
	QSPI0	0x1001_4000 ~ 0x1001_4FFF	First QSPI address space.
	PWM0	0x1001_5000 ~ 0x1001_5FFF	First PWM address space.
	UART1	0x1002_3000 ~ 0x1002_3FFF	Second UART address space.
	QSPI1	0x1002_4000 ~ 0x1002_4FFF	Second QSPI address space.
	PWM1	0x1002_5000 ~ 0x1002_5FFF	Second PWM address space.
	QSPI2	0x1003_4000 ~ 0x1003_4FFF	Third QSPI address space.
	PWM2	0x1003_5000 ~ 0x1003_5FFF	Third PWM address space.
	I2C Master	0x1004_2000 ~ 0x1004_2FFF	I2C Master address space.
Default slave	The other space is write-ignored and read-as zero.		

补充：

SRAM起始地址：0xa000_0000;

疑问:

1. peripheral：内部模块的寄存器详细说明文档未有；
2.

地址配置注意事项：

1. CORE的不会去DLM, ECLIC, TIMER, FIO, or PPI取指令；
2. 如果配置了600_CFG _LSU_ACCESS_ILM，ILM和DLM的地址可重叠，反之不可以；
3. ILM和DLM的总地址不能与其他的模块地址重叠；
4. DEBUG TIMER ECLIC FIO PPI等模块的地址不能重叠；

5. 三个PWM：

```
ux607_pwm8_top u_ux607_pwm0_top
ux607_pwm16_top u_ux607_pwm1_top
ux607_pwm16_top u_ux607_pwm2_top
```

6.

3.总线：

(详见Nuclei_600_Series_Databook 3: core interfaces)

PPI总线采用AHB/APB(可在ux607_define.v中配置）：

Signal name	Dir	Bit width	Description
ppi_ahbl_clk_en	Input	1	■ PPI interface clock ratio
ppi_ahbl_htrans	Output	2	■ AHB-Lite protocol HTRANS signal.
ppi_ahbl_hwrite	Output	1	■ AHB-Lite protocol HWRITE signal.
ppi_ahbl_haddr	Output	32	■ AHB-Lite protocol HADDR signal.
ppi_ahbl_hsize	Output	3	■ AHB-Lite protocol HSIZE signal.
ppi_ahbl_hprot	Output	4	■ AHB-Lite protocol HPROT signal.。
ppi_ahbl_hwdata	Output	32	■ AHB-Lite protocol HWDATA signal.
ppi_ahbl_hrdata	Input	32	■ AHB-Lite protocol HRDATA signal.
ppi_ahbl_hresp	Input	2	■ AHB-Lite protocol HRESP signal. ■ Note: support OKAY and ERROR only.
ppi_ahbl_hready	Input	1	■ AHB-Lite protocol HREADY signal.

System Bus采用AXI：

即3.4.2 MEM Interface

Signal name	Dir	Bit width	Description
mem_clk_en	Input	1	MEM interface clock ratio
mem_arvalid	Output	1	AXI protocol ARVALID signal
mem_araddr	Output	64	AXI protocol ARADDR signal
mem_arlen	Output	8	AXI protocol ARLEN signal (0-255)
mem_arsize	Output	3	AXI protocol ARLEN signal (0-7)
mem_arburst	Output	2	AXI protocol ARBURST signal (0-2)
mem_arlock	Output	2	AXI protocol ARLOCK signal
mem_arcache	Output	4	AXI protocol ARCACHE signal
mem_arprot	Output	3	AXI protocol ARPROT signal
mem_arready	Input	1	AXI protocol ARREADY signal
mem_awvalid	Output	1	AXI protocol AWVALID signal
mem_awaddr	Output	64	AXI protocol AWADDR signal
mem_awlen	Output	8	AXI protocol AWLEN signal (0-255)
mem_awsize	Output	3	AXI protocol AWSIZE signal (0-7)
mem_awburst	Output	2	AXI protocol AWBURST signal (0-2)
mem_awlock	Output	2	AXI protocol AWLOCK signal
mem_awcache	Output	4	AXI protocol AWCACHE signal
mem_awprot	Output	3	AXI protocol AWPROT signal
mem_awready	Input	1	AXI protocol AWREADY signal
mem_wvalid	Output	1	AXI protocol WVALID signal
mem_wdata	Output	64	AXI protocol WDATA signal
mem_wlast	Output	1	AXI protocol WLAST signal
mem_wstrb	Output	8	AXI protocol WSTRB signal
mem_wready	Input	1	AXI protocol WREADY signal
mem_rvalid	Output	1	AXI protocol RVALID signal
mem_rdata	Output	64	AXI protocol RDATA signal
mem_rlast	Output	1	AXI protocol RLAST signal
mem_rresp	Output	2	AXI protocol RRESP signal
mem_rready	Input	1	AXI protocol RREADY signal
mem_bvalid	Output	1	AXI protocol BVALID signal
mem_bresp	Output	2	AXI protocol BRESP signal
mem_bready	Input	1	AXI protocol BREADY signal

疑问：

```
1.  AXI总线交易顺序：通过给AXI总线发送的交易分配一个ID标签。
2.   将icb分发与直接将ahb分发对比？；
```

注意事项：

1. axi与ahb总线对具体模块操作时须经过axi2icb与ahbl2icb；

   以pwm为例：

   在ux607_subsys_main.v中line3000:ux607_subsys_perips例化，通过ahb总线与mian层通信；

在ux607_subsys_perips.v中line482：ux607_gnrl_ahbl2icb例化，将ahb总线转化为icb总线，

line1605:ux607_icb1to16_bus例化，将icb总线分发为16条，其中pwm2的icb总现在Line1949;

line2854:ux607_pwm16_top例化将u_ux607_pwm2_top集成进入perips；

在ux607_pwm16_top.v中Line99:ux607_pwm16例化将u_ux607_pwm16集成；此中ux607_pwm16接口中含有a,b,c,d组（a,c为输入；b,d为输出）如下接口（使用了a,d组）：

```
.io_in_0_a_ready
.io_in_0_a_valid
.io_in_0_a_bits_opcode
.io_in_0_a_bits_param
.io_in_0_a_bits_size
.io_in_0_a_bits_source
.io_in_0_a_bits_address
.io_in_0_a_bits_mask
.io_in_0_a_bits_data
```

在ux607_pwm16_core.v中主要实现PWM寄存器（cmp,feed,key）读写与具体功能；

2. PPI可通过配置define选择其他总线；
3. axi总线中还有总线异步，总线分发，总线仲裁