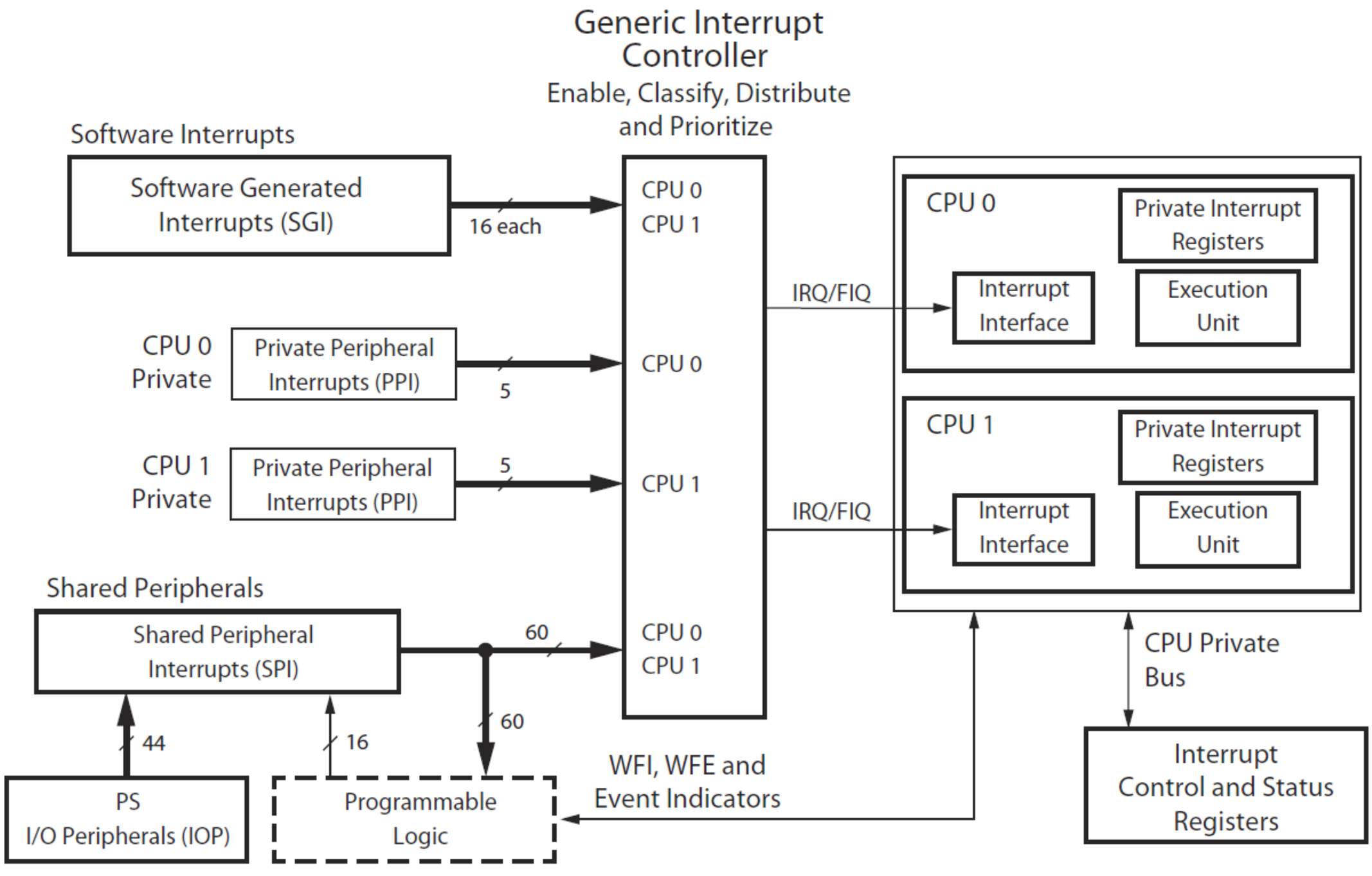
嵌入式体系结构基于ZYNQ
第四讲

ZYNQ-SoC

Zynq中断系统

- Zynq中断概述
- 中断硬件
- 中断编程模式

ZYNQ-SoC 中断概述



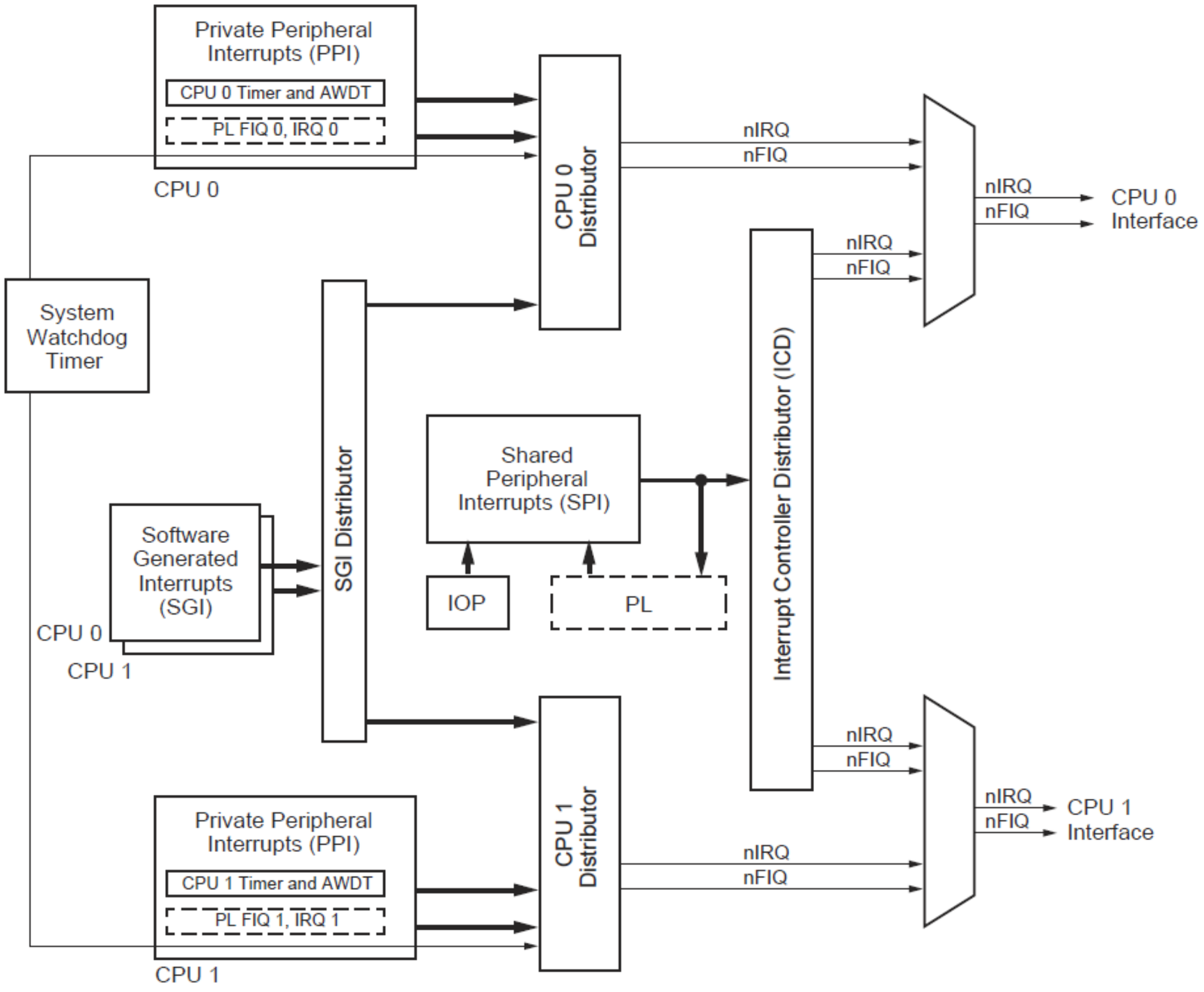
中断概述

- 私有、共享和软件中断
- 通用中断控制器功能
- 中断优先级和处理

私有、共享和软件中断

- 全局定时器、私有看门狗、私有定时器、PL部分产生的FIG/IRQ（PPI）
- 对通用中断控制器寄存器读写而产生的中断（SGI）
- 位于PS和PL的I/O及内存控制器产生的中断（SPI）

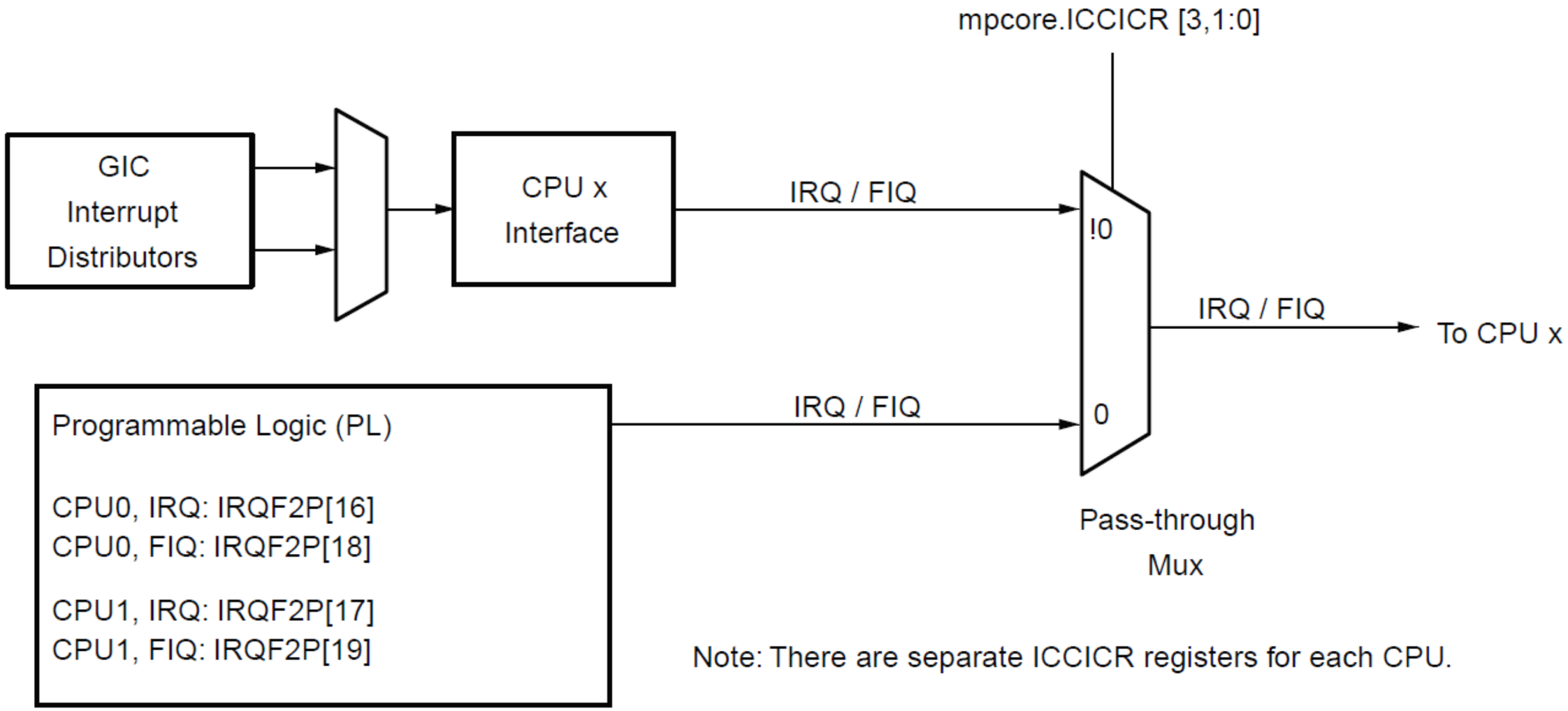
ZYNQ-SoC 通用中断控制器GIC



通用中断控制器GIC

- 允许、禁止中断和管理中断优先级
- 基于ARM的GIC V1.0
- 所有CPU均通过私有总线访问GIC

ZYNQ-SoC 中断信号传递



中断信号传递

FIQEn (ICCICR[3])	SecureS (ICCICR[0])	SecureNS (ICCICR[1])	IRQ to CPU x	FIQ to CPU x
0	0	0	pass through	pass through
0	0	1	driven by GIC	pass through
0	1	0	driven by GIC	pass through
0	1	1	driven by GIC	pass through
1	0	0	pass through	pass through
1	0	1	driven by GIC	pass through
1	1	0	pass through	driven by GIC
1	1	1	driven by GIC	driven by GIC

软件生成中断（SGI）

- 通过写相关寄存器产生
- 中断CPU自身，其它CPU或者所有CPU
- 每个CPU有自己的SGI寄存器组

软件生成中断（SGI）

IRQ ID#	Name	SGI#	Type	Description
0	Software 0	0	Rising edge	A set of 16 interrupt sources that are private to each CPU that can be routed to up to 16 common interrupt destinations where each destination can be one or more CPUs.
1	Software 1	1	Rising edge	
~	…	~	…	
15	Software 15	15	Rising edge	

ZYNQ-SoC CPU私有外设中断（PPI）

IRQ ID#	Name	PPI#	Type	Description
26:16	Reserved	~	~	Reserved
27	Global Timer	0	Rising edge	Global timer
28	nFIQ	1	Active Low level (active High at PS-PL interface)	Fast interrupt signal from the PL: CPU0: IRQF2P[18] CPU1: IRQF2P[19]
29	CPU Private Timer	2	Rising edge	Interrupt from private CPU timer
30	AWDT{0, 1}	3	Rising edge	Private watchdog timer for each CPU
31	nIRQ	4	Active Low level (active High at PS-PL interface)	Interrupt signal from the PL: CPU0: IRQF2P[16] CPU1: IRQF2P[17]

ZYNQ-SoC 共享外设中断（SPI）

Source	Interrupt Name	IRQ ID#	Status Bits (mpcore Registers)	Required Type	PS-PL Signal Name	I/O
APU	CPU 1, 0 (L2, TLB, BTAC)	33:32	spi_status_0[1:0]	Rising edge	~	~
	L2 Cache	34	spi_status_0[2]	High level	~	~
	OCM	35	spi_status_0[3]	High level	~	~
Reserved	~	36	spi_status_0[3]	~	~	~
PMU	PMU [1,0]	38, 37	spi_status_0[6:5]	High level	~	~
XADC	XADC	39	spi_status_0[7]	High level	~	~
DevC	DevC	40	spi_status_0[8]	High level	~	~
SWDT	SWDT	41	spi_status_0[9]	Rising edge	~	~
Timer	TTC 0	44:42	spi_status_0[12:10]	High level	~	~
DMAC	DMAC Abort	45	spi_status_0[13]	High level	IRQP2F[28]	Output
	DMAC [3:0]	49:46	spi_status_0[17:14]	High level	IRQP2F[23:20]	Output
Memory	SMC	50	spi_status_0[18]	High level	IRQP2F[19]	Output
	Quad SPI	51	spi_status_0[19]	High level	IRQP2F[18]	Output
Reserved	~	~	~	Always driven Low	IRQP2F[17]	Output
IOP	GPIO	52	spi_status_0[20]	High level	IRQP2F[16]	Output
	USB 0	53	spi_status_0[21]	High level	IRQP2F[15]	Output
	Ethernet 0	54	spi_status_0[22]	High level	IRQP2F[14]	Output
	Ethernet 0 Wake-up	55	spi_status_0[23]	Rising edge	IRQP2F[13]	Output
	SDIO 0	56	spi_status_0[24]	High level	IRQP2F[12]	Output
	I2C 0	57	spi_status_0[25]	High level	IRQP2F[11]	Output
	SPI 0	58	spi_status_0[26]	High level	IRQP2F[10]	Output
	UART 0	59	spi_status_0[27]	High level	IRQP2F[9]	Output
	CAN 0	60	spi_status_0[28]	High level	IRQP2F[8]	Output

ZYNQ-SoC 共享外设中断（SPI）

Source	Interrupt Name	IRQ ID#	Status Bits (mpcore Registers)	Required Type	PS-PL Signal Name	I/O
PL	PL [2:0]	63:61	spi_status_0[31:29]	Rising edge/ High level	IRQF2P[2:0]	Input
	PL [7:3]	68:64	spi_status_1[4:0]	Rising edge/ High level	IRQF2P[7:3]	Input
Timer	TTC 1	71:69	spi_status_1[7:5]	High level	~	~
DMAC	DMAC[7:4]	75:72	spi_status_1[11:8]	High level	IRQP2F[27:24]	Output
IOP	USB 1	76	spi_status_1[12]	High level	IRQP2F[7]	Output
	Ethernet 1	77	spi_status_1[13]	High level	IRQP2F[6]	Output
	Ethernet 1 Wake-up	78	spi_status_1[14]	Rising edge	IRQP2F[5]	Output
	SDIO 1	79	spi_status_1[15]	High level	IRQP2F[4]	Output
	I2C 1	80	spi_status_1[16]	High level	IRQP2F[3]	Output
	SPI 1	81	spi_status_1[17]	High level	IRQP2F[2]	Output
	UART 1	82	spi_status_1[18]	High level	IRQP2F[1]	Output
	CAN 1	83	spi_status_1[19]	High level	IRQP2F[0]	Output
PL	PL [15:8]	91:84	spi_status_1[27:20]	Rising edge/ High level	IRQF2P[15:8]	Input
SCU	Parity	92	spi_status_1[28]	Rising edge	~	~
Reserved	~	95:93	spi_status_1[31:29]	~	~	~

中断编程模式

- 中断优先级
- 中断处理
- ARM编程相关
  - GIC寄存器访问
  - 分发器和CPU接口
  - GIC安全扩展的影响
  - PU接口寄存器
  - 保存和存储控制器状态
- 延迟中断和安全扩展
  - GIC中只有IRQ会抢占
  - FIQ采用安全模式

思考题