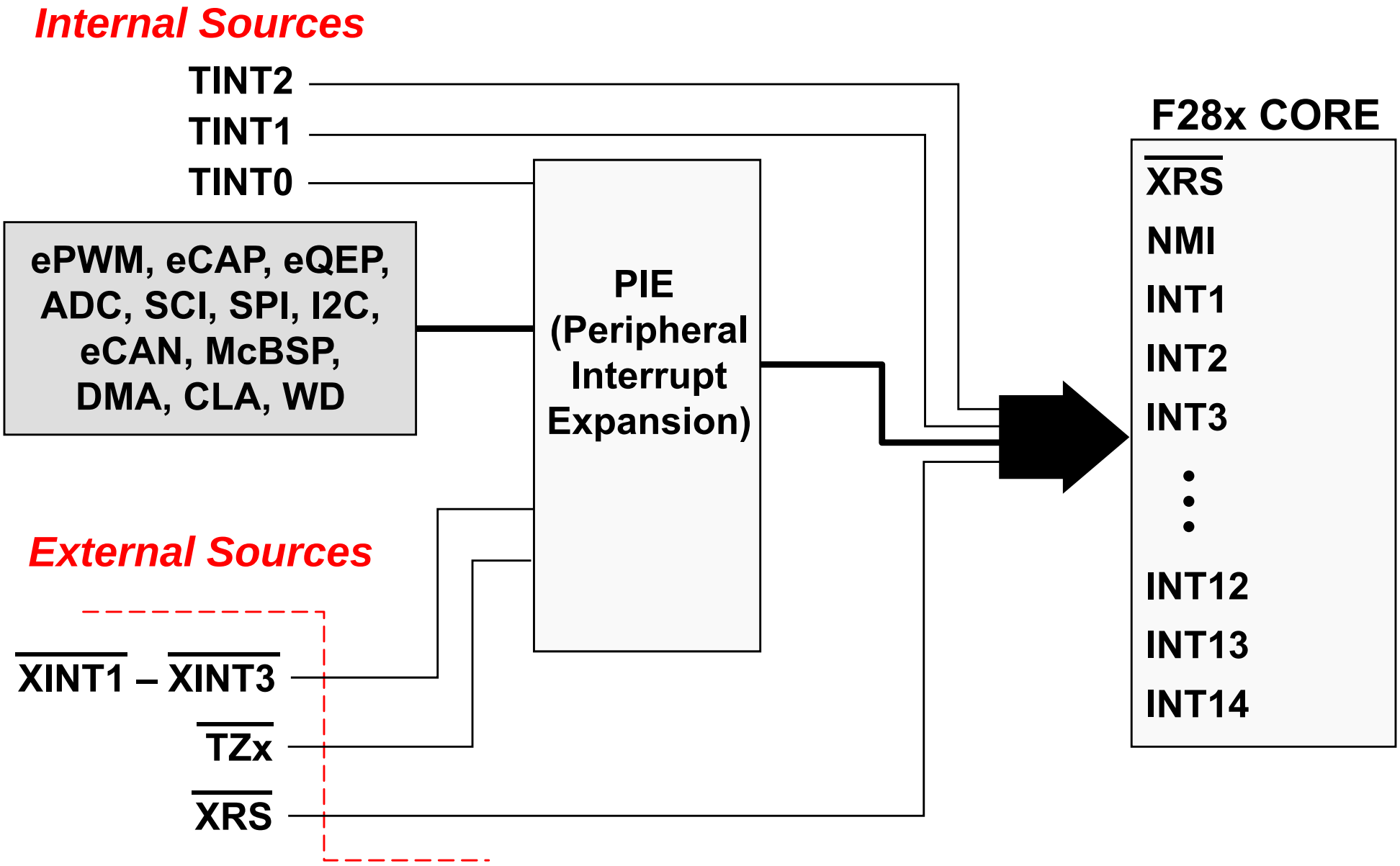
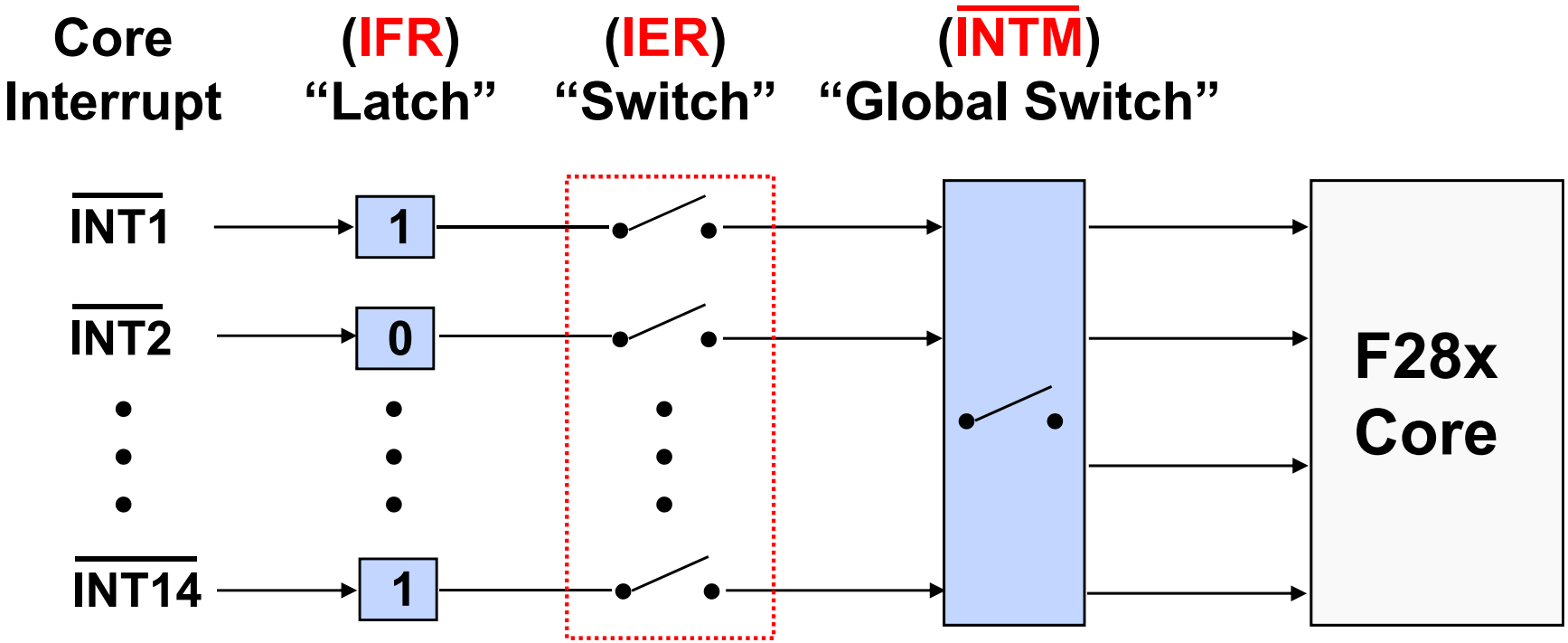
Interrupt Sources



Maskable Interrupt Processing

Conceptual Core Overview



- A valid signal on a specific interrupt line causes the latch to display a "1" in the appropriate bit
- If the individual and global switches are turned "on" the interrupt reaches the core

Interrupt Flag Register (IFR)

15	14	13	12	11	10	9	8
RTOSINT	DLOGINT	INT14	INT13	INT12	INT11	INT10	INT9

7	6	5	4	3	2	1	0
INT8	INT7	INT6	INT5	INT4	INT3	INT2	INT1

Pending : $IFR_{Bit} = 1$

Absent : $IFR_{Bit} = 0$

```
/*** Manual setting/clearing IFR ***/
extern cregister volatile unsigned int IFR;
    IFR  |= 0x0008;              //set INT4 in IFR
    IFR &= 0xFFF7;              //clear INT4 in IFR
```

- Compiler generates atomic instructions (non-interruptible) for setting/clearing IFR
- If interrupt occurs when writing IFR, interrupt has priority
- IFR(bit) cleared when interrupt is acknowledged by CPU
- Register cleared on reset

Interrupt Enable Register (IER)

15	14	13	12	11	10	9	8
RTOSINT	DLOGINT	INT14	INT13	INT12	INT11	INT10	INT9

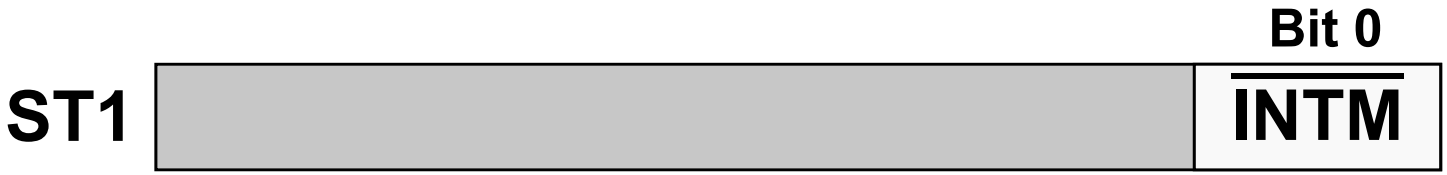
7	6	5	4	3	2	1	0
INT8	INT7	INT6	INT5	INT4	INT3	INT2	INT1

Enable: Set $IER_{Bit} = 1$
Disable: Clear $IER_{Bit} = 0$

```
/*** Interrupt Enable Register ***/
extern cregister volatile unsigned int IER;
    IER  |= 0x0008;              //enable INT4 in IER
    IER &= 0xFFF7;              //disable INT4 in IER
```

- Compiler generates atomic instructions (non-interruptible) for setting/clearing IER
- Register cleared on reset

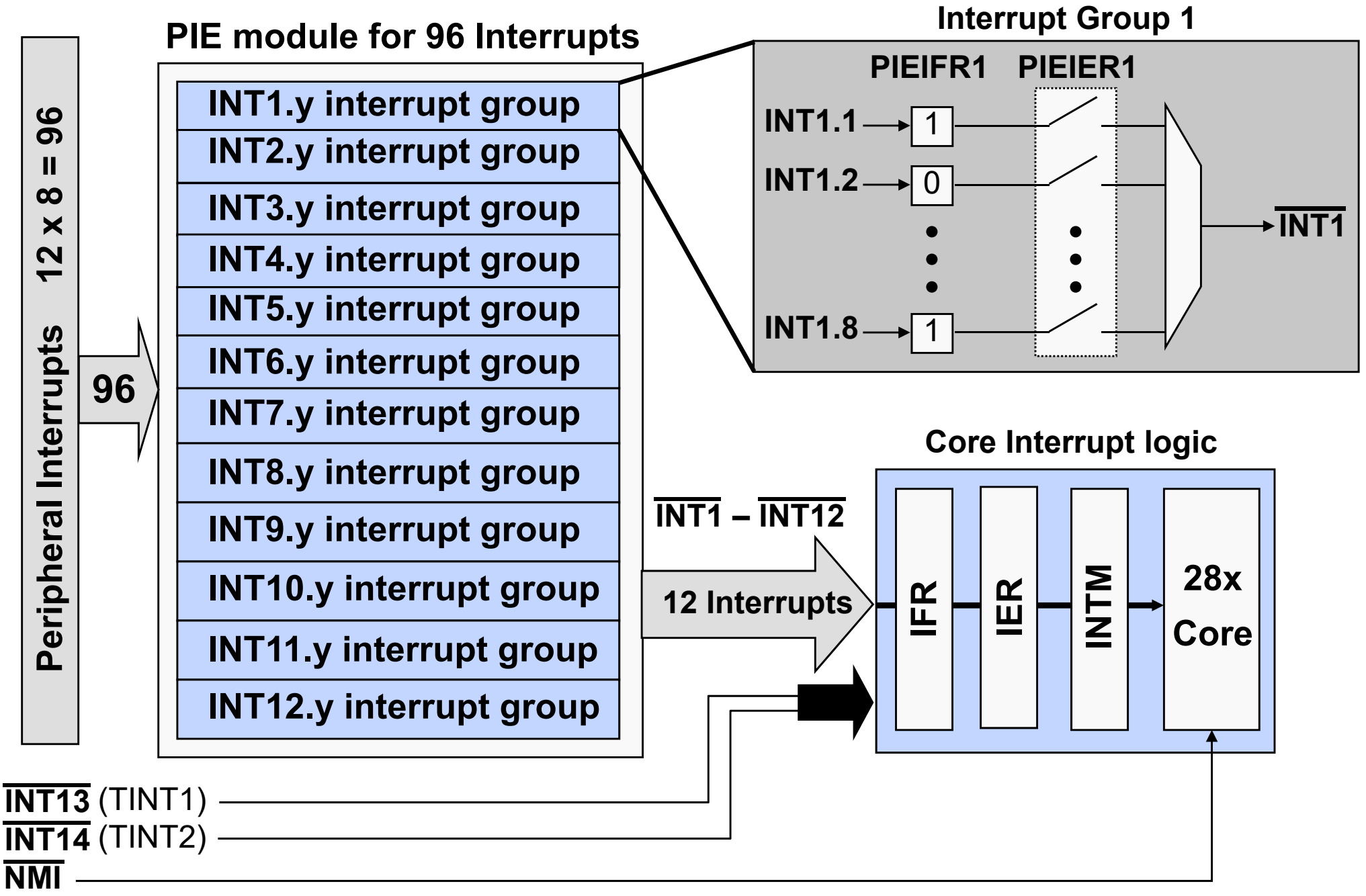
Interrupt Global Mask Bit



- INTM used to globally enable/disable interrupts:
  - Enable: $\overline{\text{INTM}}$ = 0
  - Disable: $\overline{\text{INTM}}$ = 1 (reset value)
- INTM modified from assembly code only:

```
/*** Global Interrupts ***/
    asm(“  CLRC  INTM”);     //enable global interrupts
    asm(“  SETC  INTM”);     //disable global interrupts
```

Peripheral Interrupt Expansion - PIE



F2806x PIE Interrupt Assignment Table

	INTx.8	INTx.7	INTx.6	INTx.5	INTx.4	INTx.3	INTx.2	INTx.1
INT1	WAKEINT	TINT0	ADCINT9	XINT2	XINT1		ADCINT2	ADCINT1
INT2	EPWM8 _TZINT	EPWM7 _TZINT	EPWM6 _TZINT	EPWM5 _TZINT	EPWM4 _TZINT	EPWM3 _TZINT	EPWM2 _TZINT	EPWM1 _TZINT
INT3	EPWM8 _INT	EPWM7 _INT	EPWM6 _INT	EPWM5 _INT	EPWM4 _INT	EPWM3 _INT	EPWM2 _INT	EPWM1 _INT
INT4	HRCAP2 _INT	HRCAP1 _INT				ECAP3 _INT	ECAP2 _INT	ECAP1 _INT
INT5				HRCAP4 _INT	HRCAP3 _INT		EQEP2 _INT	EQEP1 _INT
INT6			MXINTA	MRINTA	SPITX INTB	SPIRX INTB	SPITX INTA	SPIRX INTA
INT7			DINTCH6	DINTCH5	DINTCH4	DINTCH3	DINTCH2	DINTCH1
INT8							I2CINT2A	I2CINT1A
INT9			ECAN1 _INTA	ECAN0 _INTA	SCITX INTB	SCIRX INTB	SCITX INTA	SCIRX INTA
INT10	ADCINT8	ADCINT7	ADCINT6	ADCINT5	ADCINT4	ADCINT3	ADCINT2	ADCINT1
INT11	CLA1 _INT8	CLA1 _INT7	CLA1 _INT6	CLA1 _INT5	CLA1 _INT4	CLA1 _INT3	CLA1 _INT2	CLA1 _INT1
INT12	LUF	LVF						XINT3

PIE Registers

PIEIFRx register (x = 1 to 12)

15 - 8	7	6	5	4	3	2	1	0
reserved	INTx.8	INTx.7	INTx.6	INTx.5	INTx.4	INTx.3	INTx.2	INTx.1

PIEIERx register (x = 1 to 12)

15 - 8	7	6	5	4	3	2	1	0
reserved	INTx.8	INTx.7	INTx.6	INTx.5	INTx.4	INTx.3	INTx.2	INTx.1

PIE Interrupt Acknowledge Register (PIEACK)

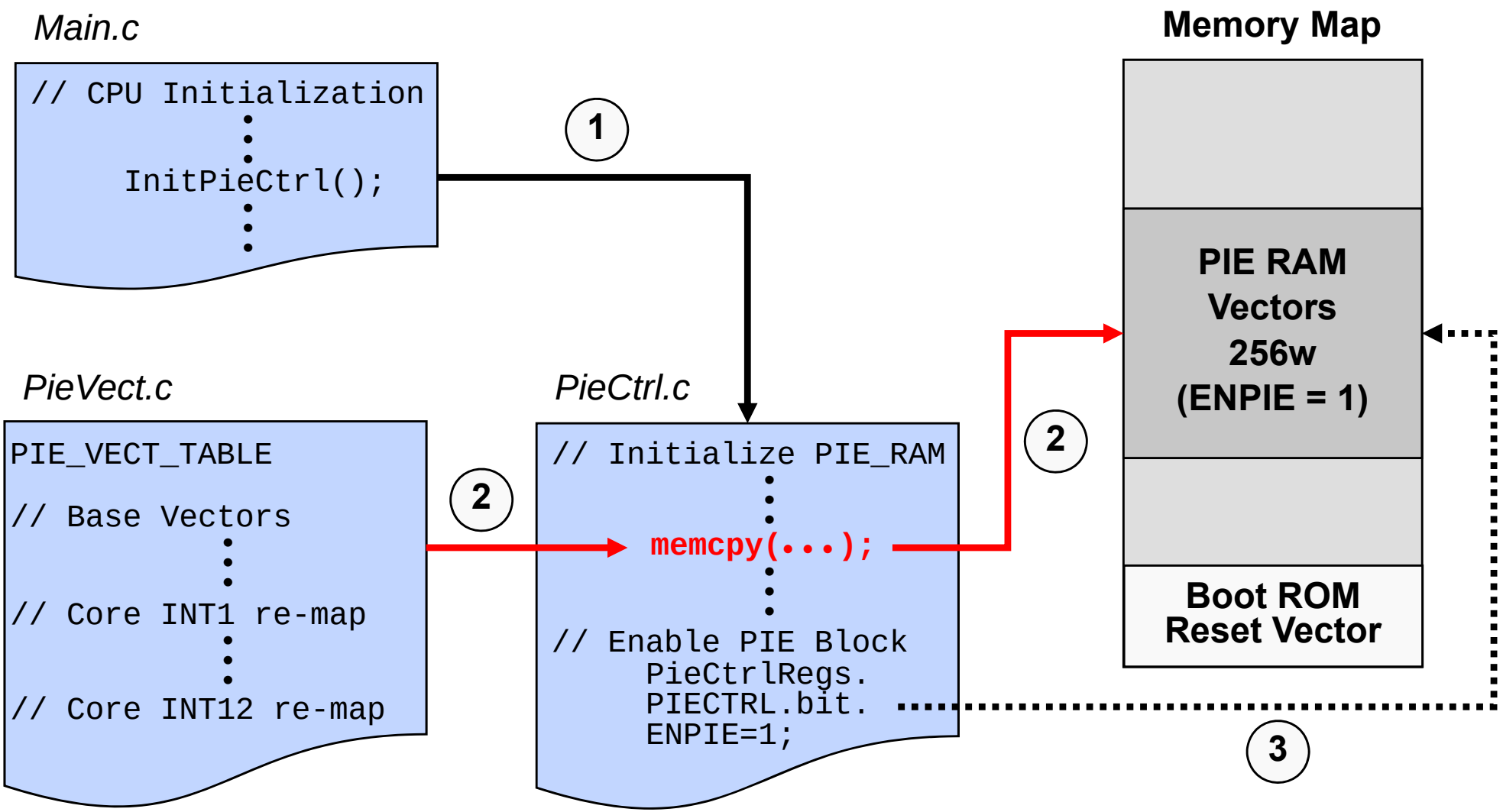
15 - 12	11	10	9	8	7	6	5	4	3	2	1	0
reserved	PIEACKx											

PIECTRL register

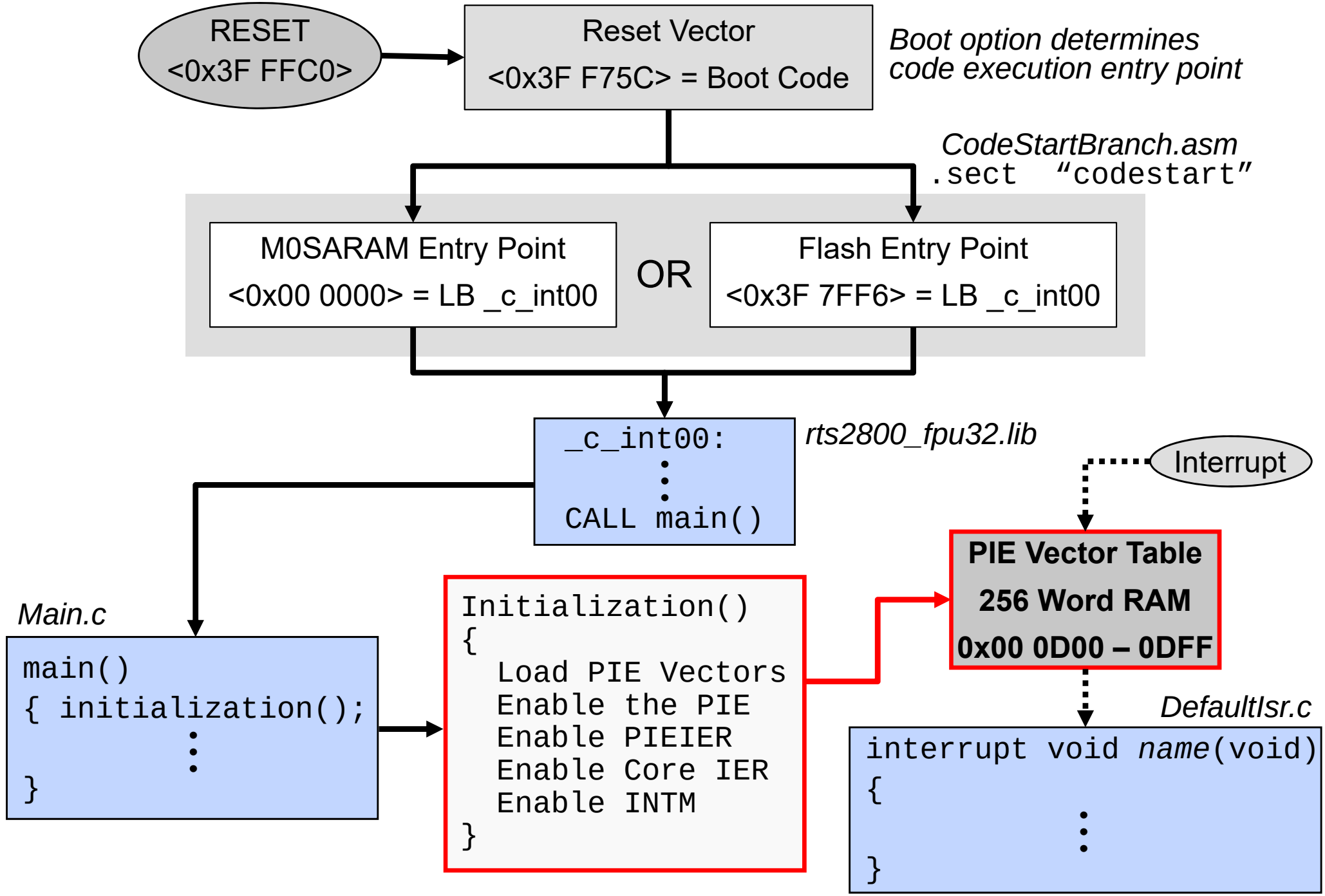
15 - 1	0
PIEVECT	ENPIE

```
#include “F2806x_Device.h”
    PieCtrlRegs.PIEIFR1.bit.INTx4 = 1;    //manually set IFR for XINT1 in PIE group 1
    PieCtrlRegs.PIEIER3.bit.INTx2 = 1;    //enable EPWM2_INT in PIE group 3
    PieCtrlRegs.PIEACK.all = 0x0004;      //acknowledge the PIE group 3
    PieCtrlRegs.PIECTRL.bit.ENPIE = 1;  //enable the PIE
```

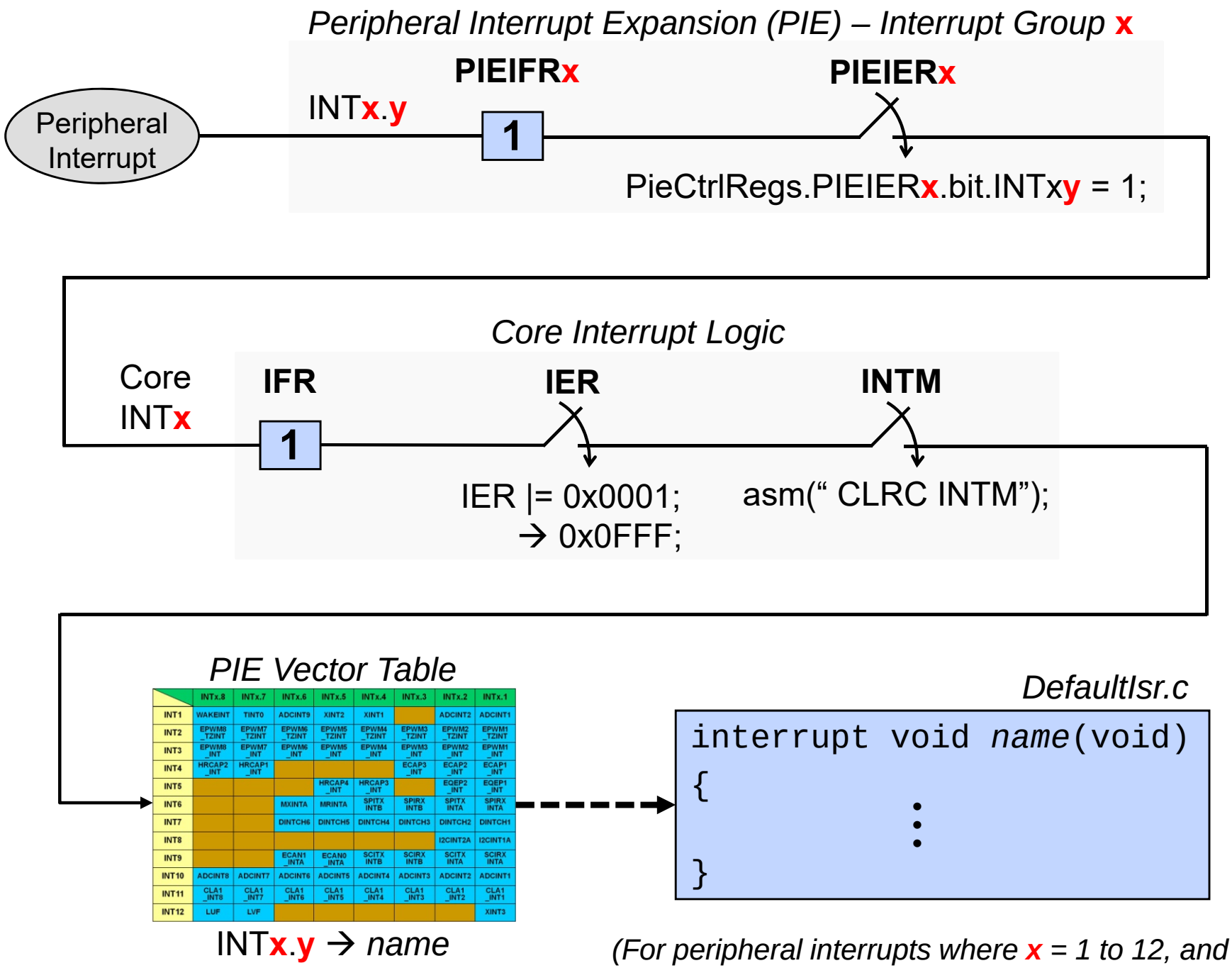
PIE Block Initialization



PIE Initialization Code Flow - Summary



Interrupt Signal Flow – Summary



	INTx.8	INTx.7	INTx.6	INTx.5	INTx.4	INTx.3	INTx.2	INTx.1
INT1	WAKEINT	TINT0	ADCINT9	XINT2	XINT1		ADCINT2	ADCINT1
INT2	EPWM8_TZINT	EPWM7_TZINT	EPWM6_TZINT	EPWM5_TZINT	EPWM4_TZINT	EPWM3_TZINT	EPWM2_TZINT	EPWM1_TZINT
INT3	EPWM8_INT	EPWM7_INT	EPWM6_INT	EPWM5_INT	EPWM4_INT	EPWM3_INT	EPWM2_INT	EPWM1_INT
INT4	HRCAP2_INT	HRCAP1_INT				ECAP3_INT	ECAP2_INT	ECAP1_INT
INT5				HRCAP4_INT	HRCAP3_INT		EQEP2_INT	EQEP1_INT
INT6			MXINTA	MRINTA	SPITXINTB	SPIRXINTB	SPITXINTA	SPIRXINTA
INT7			DINTCH6	DINTCH5	DINTCH4	DINTCH3	DINTCH2	DINTCH1
INT8							I2CINT2A	I2CINT1A
INT9			ECAN1_INTA	ECAN0_INTA	SCITXINTB	SCIRXINTB	SCITXINTA	SCIRXINTA
INT10	ADCINT8	ADCINT7	ADCINT6	ADCINT5	ADCINT4	ADCINT3	ADCINT2	ADCINT1
INT11	CLA1_INT8	CLA1_INT7	CLA1_INT6	CLA1_INT5	CLA1_INT4	CLA1_INT3	CLA1_INT2	CLA1_INT1
INT12	LUF	LVF						XINT3