AMD

together we advance_

Reading Data from GPIO Input Pins

For GPIO pins configured as inputs, there are two options to monitor the input.

Option 1: Use the gpio.DATA_RO_x register of each bank.

Example: Read the state of all GPIO input pins in bank 0 using the DATA_RO_0 register.

1. **Read Input Bank 0:** Read the gpio.DATA_0 register.

Option 2: Use interrupt logic on input pins (refer to section Interrupt Function).

Example: Configure MIO pin 12 to be triggered as rising edge.

1. **Set the trigger as a rising edge:** Write `1` to gpio.INT_TYPE_0 [12]. Write `1` to gpio.INT_POLARITY_0 [12]. Write `0` to gpio.INT_ANY_0 [12].
2. **Enable interrupt:** Write `1` to gpio.INT_EN_0 [12].
3. **Status of Input pin:** gpio.INT_STAT_0 [12] =`1` implies that an interrupt event occurred.
4. **Disable interrupt**: Write `1` to gpio.INT_DIS_0 [12].

EMIO_MIO_INTR.c 註解

初始化中斷

```
    IntcConfig = XScuGic_LookupConfig(INTC_DEVICE_ID);
//  xil_printf("XScuGic_LookupConfig Successfully  \r\n");
    XScuGic_CfgInitialize(GicInstancePtr, IntcConfig,IntcConfig->CpuBaseAddress);
//  xil_printf("XScuGic_CfgInitialize Successfully  \r\n");
//  xil_printf("================================\r\n");
```

初始化SOC異常判定

```
/*
 * Connect the interrupt controller interrupt handler to the hardware interrupt handling logic in the processor.
 */
Xil_ExceptionInit();
Xil_ExceptionRegisterHandler(XIL_EXCEPTION_ID_INT,(Xil_ExceptionHandler)XScuGic_InterruptHandler, GicInstancePtr);
/* Enable interrupts in the Processor. */
Xil_ExceptionEnableMask(XIL_EXCEPTION_IRQ);
```

EMIO_MIO_INTR.c 註解

連接到中斷處理氣得掛件

```
/*
 * Connect the device driver handler that will be called when an
 * interrupt for the device occurs, the handler defined above performs
 * the specific interrupt processing for the device.
 */
XScuGic_Connect(GicInstancePtr, GpioIntrId,(Xil_ExceptionHandler)IntrHandler,(void *)Gpio);
```

設置的中斷處理掛件

```
void IntrHandler(){
    print("Interrupt Successfully\r\n");
    XGpioPs_WritePin(&Gpio, PS_LED1, Low);
    print("LED is Off\r\n");
    sleep(3);
    XGpioPs_IntrDisablePin(&Gpio, PL_EMIO);
    print("Disable Interrupt Successfully\r\n");
}
```

EMIO_MIO_INTR.c 註解

始能中斷

```
    /* Enable the interrupt for the GPIO device. */
    XScuGic_Enable(GicInstancePtr, GpioIntrId);
```

設置中斷腳位 1,BANK 2,Pin

```
    /* Enable falling edge interrupts for all the pins in GPIO bank. */
//  XGpioPs_SetIntrType(Gpio, GPIO_BANK, 0x00, 0xFFFFFFFF, 0x00);

    /* Enable falling edge interrupts for all the pins in GPIO */
    XGpioPs_SetIntrTypePin(Gpio, PL_EMIO , XGPIOPS_IRQ_TYPE_EDGE_FALLING);
```

設置中斷腳位始能 1,BANK 2,Pin

```
    /* Enable the GPIO interrupts of GPIO Bank. */
//  XGpioPs_IntrEnable(Gpio, GPIO_BANK, (1 << Input_Bank_Pin));

    /* Enable the GPIO interrupts of GPIO. */
    XGpioPs_IntrEnablePin(Gpio,PL_EMIO);
//  print("XGpioPs_IntrEnablePin Successfully\r\n");
```

EMIO_MIO_INTR.c 註解

```
void IntrHandler(){
    print("Interrupt Successfully\r\n");
    XGpioPs_WritePin(&Gpio, PS_LED1, Low);
    print("LED is Off\r\n");
    sleep(3);
    XGpioPs_IntrDisablePin(&Gpio, PL_EMIO);
    print("Disable Interrupt Successfully\r\n");
}
```

中斷啟動後，進入到HANDLER，執行中斷要執行的動作，等到結束後，關閉中斷，跳出原程序。


AMD