

Dept. of Computer Science and Engineering

EECS3215 – Embedded Systems

Interrupts Lab 7

Objectives:

To learn how to use the interrupt capabilities of the KL43Z.

Brief description

Your code will be running a foreground code that performs some function (toggle the green LED every 2 seconds). When interrupted by a switch connected to one of the inputs, the program will toggle the red LED.

Detailed description

The foreground program toggle the green LED every two seconds, that means you have to:

- Check which pin is connected to the green LED? Not difficult you did it before
- Configure that pin to be an output and general GPIO pin
- In a loop, toggle it every 2 seconds.

Then you have to configure the interrupt. That means initialize the interrupt to allow interrupt on pin C3 (connected to the push button).

It is a good idea to disable the interrupts before you configure them, then enable them again, you can use `_disable_irq()` to disable all interrupts.

Ports C and D are used for one interrupt, any interrupt on any pin in C or D trigger the same Interrupt Service Routine (ISR). The name of this ISR is

```
void PORTC_PORTD_IRQHandler(void);
```

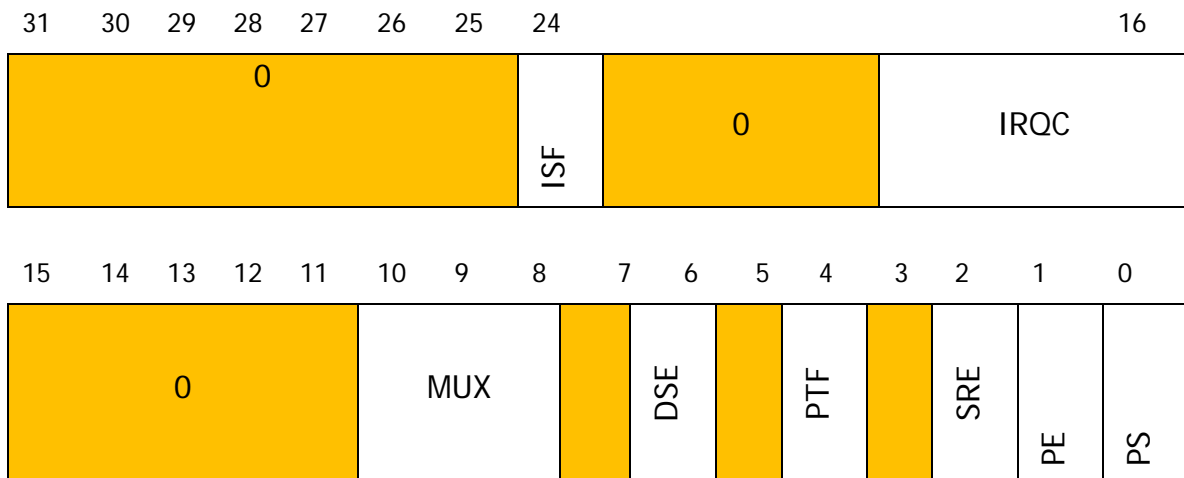
There are some functions that helps you in disabling and enabling the interrupts, these are

NVIC_DisableIRQ(Interrupt name), disable the ISR in the NVIC table, the corresponding ISR will not run, before configuring PORTC_PORTD_IRQn interrupt, disable it.

Then the interrupt is configured using PORTC->PCR[3],

The KL43Z has one ADC (ADC0) with 16 channels. In order to configure and use the ADC, you have to setup some registers, her are these registers

PORTC->PCR[n] or PORTx_PCRn



ISF if configured for interrupt, 0 -> interrupt not detected, 1-> detected
writing 1 clears the flag

IRQC Interrupt configuration, check the manual **first disable then set it to 1010**

0000 -> interrupt disabled

...

1010 -> ISF flag and interrupt on falling edge

...

MUX 000 -> disabled (analog)

001 -> alternate 1 GPIO **set to 1**

rest is chip specific

PE Pull Enable **set to 1**

PS Pull Select **set to 1**

Note that interrupts can only be detected by checking the ISFR register of the specific port, writing 1 to the associated pin clears the flag.

Then to complete the configuration of the port you need to

Set priority `NVIC_SetPriority(interrupt name)`

Clear pending requests `NVIC_ClearpendingIRQ(interrupt name)`

Enable that ISR `NVIC_EnableIRQ(interrupt name)`

The interrupt handler checks that the interrupt came from PTC3, by checking bit 3 of `PORTC->ISFR`, or the ISF bit in `PCR[3]` as explained above. Finally, you have to write 1 to the flag in order to clear it.