Michael Daniel Werezak - 20303777

Lab Assignment 2 – Interrupt Shaping

Two interrupt shaping methods were implemented, the Strict Software Scheduler and the Bursty Software Scheduler.

Interrupt Source

As required, the INT0 button was used as the interrupt source to be used to demonstrate the two interrupt shaping methods. In order to properly show the ability of these methods to prevent interrupt overload, the interrupt shaping controlled the inflow of interrupts by enabling and disabling them using NVIC_EnableIRQ(EINT3_IRQn) and NVIC_DisableIRQ(EINT3_IRQn). Both rising and falling edges of the INT0 button were used as a source of interrupts.

In order to make it easier to demonstrate the interrupt shaping, the input from INT0 was debounced using a debounce delay of 60 ms. While this does mean that some interrupts from INT0 never reach the Software Scheduler due to being filtered out by debouncing, it makes the number of interrupts generated per button press more predictable. This may be acceptable for demonstration purposes however in a true production system the debouncing should be implemented in hardware or handled *after* the interrupts have been filtered by interrupt shaping, for obvious reasons.

The debouncing was implemented using the TIM2 timer, however interrupts from this timer were not needed and disabled. Instead, the timer was used as a counter whose value was checked when handling EINT3 interrupts from the INT0 button.

Timers

The functionality was implemented to set timers to fire as either one-shot or at asynchronous intervals. This was done by setting the appropriate bits on the Match Control Register of each timer made available on the LPC1768. For simplicity, only one match channel (MR0) on each timer was used. In total three timers were used for both interrupt shaping implementations as described in the other sections.

Output

Whenever an interrupt was processed, LED-28 was lit for 200 ms. This was accomplished using TIM1 as a one-shot timer. This could have been implemented using the same timer used for debouncing and making use of multiple match channels, however it was done using a separate timer for ease of implementation. As well, the LCD was made to display when EINT3 interrupts were enabled or disabled.

Strict Software Scheduler

The strict software scheduler was implemented by disabling further EINT3 interrupts when handling an interrupt from the INT0 button and then setting a one-shot timer (TIM0) with a duration of 5 s. Once this timer fired, interrupts were re-enabled.

Bursty Software Scheduler

The bursty software scheduler was implemented using a counter to control the number of interrupts processed in the current burst and then using a timer (TIM0) set to fire at asynchronous intervals. At the end each interval the burst counter is reset and interrupts re-enabled if they were disabled during the current interval. When processing an interrupt, the burst counter was incremented once, and if the value of the burst counter reached the max burst size, the EINT3 interrupt was disabled.