Soluciones Ejercicios Tema 4-6

- **Pregunta 1**: Se ha escrito el siguiente manejador con el que pretendemos que la que la IRQ del canal 3 del timer se produzca con una periodicidad de medio segundo. El código que le da servicio se ha omitido para simplificar y sabemos que es la única configurada y habilitada por el programa principal. Completa el código:
    - ldr r0, =STBASE. Cargamos en r0 la dirección de memoria de STBASE para poder usar los registros del Timer.
    - str r1, [r0, #STCS]. Limpiamos el evento de la interrupción del canal 3 del Timer.
    - ldr r1, [r0, #STCLO]. Cargamos el valor del Timer actual para reconfigurar la interrupción.
    - str r1, [r0, #STC3]. Almacenamos el nuevo valor temporal que hará saltar la interrupción en el canal 3 del Timer.

- **Pregunta 2**: Se ha escrito el siguiente manejador para la IRQ del canal 1 del timer, que sabemos es la única configurada y habilitada por el programa principal y cuyo servicio consiste en invertir el estado actual de los leds cada medio segundo:
    - No se vuelve a ejecutar el manejador porque no se ha realizado el CLEAR EVENT de la interrupción IRQ del canal 1 del timer.

- **Pregunta 3**: Sabemos que en el programa principal sólo se ha configurado y habilitado la interrupción IRQ de los dos pulsadores (GPIO3 y 2). Completa su manejador (el código que da servicio a la interrupción de cada botón se ha omitido para simplificar):
    - ldr r2, [r0, #GPEDS0]. Cargamos el estado de GPEDS0 para ver que GPIO ha provocado la interrupción.
    - ands r2, #0b000100. Comprobamos que el motivo de la interrupción ha sido el pulsador del GPIO2.
    - mov r1, #0b00100. Cargamos la máscara para limpiar el evento de la interrupción del GPIO2.
    - str r1, [r0, #GPEDS0]. Limpiamos el evento de la interrupción.
    - ands r3, #0b001000. Comprobamos que el motivo de la interrupción ha sido el pulsador del GPIO3.
    - mov r1, #0b001000. Cargamos la máscara para limpiar el evento de la interrupción del GPIO3.
    - str r1, [r0, #GPEDS0]. Limpiamos el evento de la interrupción.

- **Pregunta 4**: Sabemos que en el programa principal sólo se ha configurado y habilitado la interrupción IRQ de los canales del timer 1 y 3. Completa su manejador, sabiendo que el C1 se reprograma para que vuelva a interrumpir cada 200ms y el C3 cada 100ms (el código que da servicio a la interrupción de cada canal se ha omitido para simplificar):
    - ldr r2, [r0, #SRCS]. Cargamos el estado del Timer para poder comprobar el canal que ha producido la interrupción.
    - ands r2, #0b0010. Comprobamos si el canal 1 ha producido la interrupción.
    - mov r3, #0b0010. Cargamos la máscara para limpiar el evento de la interrupción del canal 1.

    - str r3, [r0, #STCS]. Limpiamos el evento de la interrupción.
    - ldr r3, [r0, #STCLO]. Cargamos el valor actual del Timer.
    - str r3, [r0, #STC1]. Almacenamos el nuevo valor del Timer que provocará la interrupción en el canal 1.
    - ldr r3, [r0 , #STCS]. Cargamos el estado del Timer para poder comprobar el canal que ha producido la interrupción.
    - ands r3, #0b1000. Comprobamos si el canal 3 ha producido la interrupción.
    - mov r3, #0b1000. Cargamos la máscara para limpiar el evento de la interrupción del canal 3.
    - str r3, [r0, #STCS]. Limpiamos el evento de la interrupción.
    - ldr r3, [r0, #STCLO]. Cargamos el valor actual del Timer.
    - str r3, [r0, #STC3]. Almacenamos el nuevo valor del Timer que provocará la interrupción en el canal 3.

- **Pregunta 5**: Cuándo se produce una interrupción en nuestro ARM, ¿qué parte del mecanismo de atención a la misma es responsabilidad del HW (CPU)?
    - Inicializar la tabla de vectores. Incorrecto, esto se realiza en el programa principal.
    - Salvaguardar/restaurar en/de pila los registros de propósito general que va a modificar la RTI. Incorrecto, esto se realiza en el manejador de la interrupción.
    - Dejar a PC apuntando a la posición adecuada de la tabla de vectores. Correcto.
    - Salvaguardar el estado del procesador, es decir, los valores en CPSR y LR. Correcto.
    - Inicializar la/s pila/s para los modos que vayan a utilizarla. Incorrecto, esto se realiza en el programa principal.

- **Pregunta 6**: Para la habilitación/deshabilitación global de las IRQ y FIQ en el ARM hay que escribir en los bits I y F que se localizan en su registro de estado.
    - Verdadero. Hay que escribir un 0 en los bits correspondientes.

- **Pregunta 7**: Queremos cargar en el registro r1 el comando de control que me permite configurar GPIO22 y GPIO27 como salida, comando que enviaremos al puerto GPFSEL2. Señala cuál de las siguientes instrucciones no da error de compilación:
    - mov r1, #b00000000001000000000000001000000. Incorrecto. Aunque sintácticamente el código es correcto. Va a producirse un error de constante no válida puesto que mov solo admite inmediatos de 8 bits. Sin embargo, le estamos pasando uno de 32 bits. En otras ocasiones esto funciona porque los puertos usan un rotador de bits para construir valores más grandes a partir de una secuencia de bits pequeña. Sin embargo, en este caso no se puede construir el valor que le estamos pasando y el ensamblador nos avisa del error.
    - mov r1, =b00000000001000000000000001000000. Incorrecto, con mov debemos usar # para definir un valor inmediato.
    - ldr r1, =b00000000001000000000000001000000. Correcto.

    - ldr r1, #b00000000001000000000000001000000. Incorrecto, con ldr debemos usar = para definir un valor inmediato.

- **Pregunta 8**: Completa el siguiente programa principal para que quede configurada y habilitada la interrupción del pulsador mediante FIQ, que queremos sea servida por la rutina manejadora cuya primera instrucción hemos etiquetado como manejador:
    - ADDEXC 0x1C, manejador. Dado que usamos FIQ, el valor de ADDEXC es 0x1C.
    - mov r0. #0b11010001. Deshabilitamos interrupciones y pasamos al modo FIQ para configurar la pila.
    - mov sp, #0x4000. Inicializamos el puntero de pila.
    - mov r1, #0b10110100. Máscara para configurar la fuente de interrupción.
    - str r1, [r0, # INTFIQCON]. Almacenamos la fuente de interrupción.