

Práctica III

Implementación de un microcontrolador de aplicación específica

1. Práctica básica

Protocolo DMA-CPU en recepción basado en *polling*

El enunciado disponible en el Moodle de la asignatura describe con detalle la arquitectura de un sencillo microcontrolador. En esta versión, la recepción de un comando de usuario es avisada al programa ejecutado en la CPU por el módulo DMA, que almacena `0xff` en la posición `NEW_INST` de la RAM. El programa detecta esa condición mediante un lazo de *polling* y procede a la atención del comando recibido.

El objetivo de la práctica básica consiste en la implementación en VHDL del microcontrolador para que sea capaz de ejecutar el programa que interpreta los comandos de usuario, proporcionado en el fichero `ROM.vhd`. El sistema debe probarse enviando a través del RS232 varios comandos de usuario y verificando las respuestas obtenidas y las acciones realizadas sobre los periféricos.

El sistema entregado por los alumnos **deberá ser sintetizable, y funcionar correctamente en una simulación de tipo *post-route***. Es importante recordar la necesidad de incluir el fichero de restricciones `PICtop.ucf` proporcionado antes de realizar la síntesis completa del microcontrolador. El cumplimiento del criterio mencionado permitirá alcanzar **una calificación máxima de 8'5 puntos**, y cualquier desviación reducirá ese valor en función del grado de desarrollo finalmente alcanzado y las competencias demostradas.

Conviene recordar que la síntesis se considera correcta si no produce errores y tampoco *warnings* críticos, principalmente (aunque no exclusivamente) la inferencia de *latches* en señales combinacionales, la existencia de listas de sensibilidad incompletas en procesos y la formación de bucles combinacionales.

El control principal del microcontrolador se ha proporcionado en el fichero `CPU.vhd`, que tendrá que integrarse con el resto de módulos implementados por el alumno, para lo cual será necesario ajustar cuidadosamente los protocolos de comunicación entre ellos.

Posiblemente, el aspecto más crítico de la integración será la comunicación entre CPU y DMA para la transmisión de un comando de usuario (instrucción de tipo 4 y líneas `Send_comm` y `READY` en el DMA). Para facilitar la tarea, aquí se incluyen dos indicaciones que afectan a la implementación de la máquina de estados del DMA:

- La línea `READY` debería ir a '0' **coincidiendo** con la activación de `Send_comm` por parte de la CPU (**no en el siguiente ciclo de reloj**). Para ello, la salida `READY` debe ser de tipo Mealy y tener una relación combinacional con `Send_comm`.
- Terminada la transmisión, el DMA deberá poner `READY` a '1' y **esperar a que la CPU ponga `Send_comm` a '0'** antes de volver al estado de reposo. Esto permitirá asegurar que no se va a reenviar la misma respuesta repetidamente.

En el Moodle se ha incluido un boceto con el análisis en borrador de las distintas posibilidades disponibles para este protocolo, por si algún alumno quiere profundizar.

2. Mejora

Protocolo DMA-CPU en recepción basado en interrupción

Para optar a una calificación de 10 puntos será necesario incluir en el sistema los elementos necesarios para que sea capaz de generar y atender una interrupción.

En el Moodle se ha publicado la presentación de la Práctica III con indicaciones sobre los aspectos a tener en cuenta para modificar el modo en que el DMA realiza el aviso de que se ha terminado la recepción de un comando de usuario.

Para evitar modificaciones en el programa proporcionado, no se cambiará realmente el mecanismo de *polling*, sino que sólo se modificará el procedimiento de escritura de `X"ff"` en la posición `NEW_INST` de la RAM, pasándolo de la máquina de estados del DMA a la rutina de atención de una interrupción lanzada por él. Esto lo hará más cercano a como funciona un sistema real con transferencias por DMA.

A continuación se proporcionan las indicaciones necesarias para incluir el mecanismo de interrupción en el microcontrolador previamente desarrollado. Cada grupo de trabajo puede realizar los ajustes que necesite a su caso particular. El orden de las indicaciones para la implementación no es el único posible.

- Añadir nuevos puertos `int_RQ` a DMA y CPU, salida en el primero y entrada en el segundo, y conectarlos en el fichero *top*.
- En `PIC_pkg.vhd`, añadir al tipo `alu_op` dos nuevos códigos de operación: `op_save` y `op_restore` (incluir si se desea el comentario `-- interrupt handling`).
- También en `PIC_pkg.vhd`, añadir una nueva instrucción de tipo 2:


```
constant RETI : std_logic_vector(5 downto 0) := "01" & X"0";
```
- Añadir en la CPU un nuevo registro de 8 bits, `PC_save`, para salvar el contador de programa mientras se ejecuta la rutina de atención a la interrupción.
- Crear en la CPU un registro de 1 bit, por ejemplo con nombre `int_RQ_FF`, que disponga de `enable` y `reset` (síncronos), y cuya entrada D esté fijada a '1'. El `enable` del registro tiene que estar conectado al puerto `int_RQ`.
- En el DMA, eliminar completamente la escritura de `X"ff"`. En su lugar debe incluirse la activación de la línea `int_RQ`.
- Añadir en la ALU un nuevo registro `ACC_save`, para salvar el valor del acumulador, y también el VHDL para la ejecución de los nuevos códigos de operación: `op_save` (salvar acumulador) y `op_restore` (recuperar acumulador).
- Añadir al final de la ROM la rutina de atención a la interrupción, que deberá incluir instrucciones para: cargar la constante `X"ff"` en el acumulador, salvar el acumulador en la posición `NEW_INST` de la RAM y volver de la interrupción (`RETI`).
- En la CPU, después de comprobar la línea `DMA_RQ`, comprobar también `int_RQ_FF` (de este modo se le da prioridad al DMA). Añadir el VHDL necesario para procesar la interrupción si `int_RQ_FF` está a '1': salvar el acumulador y el contador de programa, saltar a la rutina de atención a la interrupción (probablemente en la dirección `X"e1"` de la ROM) y borrar la petición almacenada en `int_RQ_FF` usando su *reset*.
- Añadir en el lugar apropiado de la CPU el código VHDL para la nueva instrucción `RETI`: recuperar los valores originales del acumulador y el contador de programa.