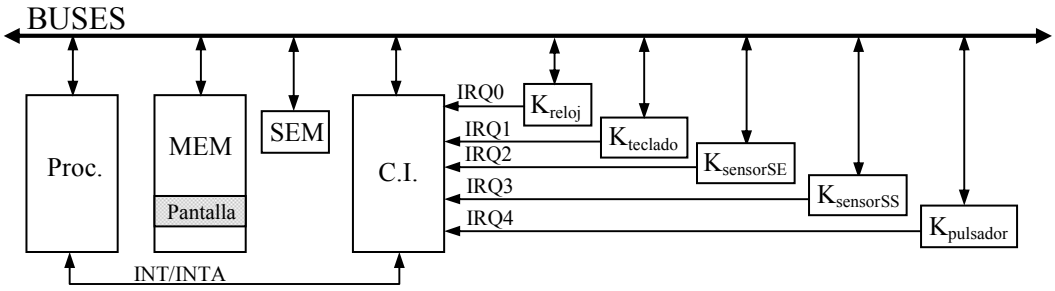
Universidad del País Vasco
Facultad de Informática

Departamento de Arquitectura y Tecnología de Computadores

Arquitectura de Computadores I

Subsistema de entrada/salida 1: Aeropuerto

Se dispone de un sistema de control de la presencia de aves en el área de despegue y aterrizaje de aviones en un aeropuerto. La estructura hardware del sistema es la clásica, como se puede observar en la figura, con la salvedad de que se han añadido dos sensores (SE, SS) y un pulsador, cuyo funcionamiento se detallará posteriormente. Además, se ha de controlar un semáforo, cuyo controlador posee únicamente un registro de control, mapeado en memoria, en la dirección SEM.



Los periféricos que debe controlar el microprocesador son idénticos a los vistos en la asignatura, exceptuando los dos sensores, el pulsador y el semáforo, cuyas características se exponen a continuación:

* **Sensor SE**: Sensor de entrada de aves. Detecta cuándo entra algún ave (una o más) en la zona controlada y genera una petición de interrupción. En su registro de datos se puede leer entonces el número de aves que han entrado en ese instante. Una vez leído el registro de datos, es necesario hacer `strobe` sobre su registro de control.

* **Sensor SS**: Sensor de salida de aves. Detecta cuándo sale algún ave (una o más) de la zona controlada y genera una petición de interrupción. En su registro de datos se puede leer entonces el número de aves que han salido en ese instante. Una vez leído el registro de datos, es necesario hacer `strobe` sobre su registro de control.

* **Semáforo SEM**: Semáforo de aviso a los aviones. Puede estar **Verde**, en cuyo caso significa que no hay aves en la zona controlada y que el avión puede realizar la maniobra correspondiente, o **Rojo**, en caso contrario. Como se ha indicado, posee un registro de control mapeado en memoria en la dirección SEM.

* **Pulsador**: Genera una petición de interrupción al ser pulsado cada vez que el personal encargado de ahuyentar a las aves captura una de ellas.

* **Reloj**: Interrumpe 18 veces por segundo.

* **Teclado**: Cada vez que se pulsa una tecla produce dos peticiones de interrupción: la primera indica que se ha pulsado tecla (MAKE) y la segunda indica que se ha liberado (BREAK). Posee un registro de datos, en el que se puede leer el código de posición ("*scan code*") correspondiente a la última tecla pulsada, y un registro de control, en el que es necesario escribir una secuencia de `strobe` cada vez que se acepta una interrupción. No posee registro de estado.

* **Controlador de interrupciones**: Posee los registros: máscara IMR, de petición de interrupción IRR y de interrupción en servicio ISR.

El **funcionamiento** del sistema es el siguiente. La situación o **estado normal** es que no haya aves en la zona controlada y, por tanto, el semáforo estará **verde**. En cualquier otra situación, el semáforo estará rojo por precaución.

En el momento en que se detecta la entrada de aves, el sistema pasa a un **estado de prealarma**, estado en el que permanecerá mientras haya aves en la zona controlada (por supuesto, las aves pueden salir "de motu propio") y el operador encargado del sistema no indique la urgencia de ahuyentar a las aves por la inminente maniobra de un avión. En este caso de urgencia, el operador pulsará la **tecla A**, lo que hará que el sistema pase a un **estado de alarma**.

En el estado de alarma, además de mantener el semáforo rojo, se genera una alarma sonora (podemos suponer que para ello disponemos de la rutina ya escrita `genera_alarma()`) de aviso al personal encargado de ahuyentar a las aves. Esta "brigada de limpieza" debe atrapar con una red las aves presentes en la zona controlada, apretando el pulsador tantas veces como aves queden atrapadas en dicha red. De nuevo, por supuesto, las aves pueden salir "de motu propio" ahuyentadas por la presencia humana o por el sonido de la alarma.

Cuando por fin la brigada ahuyentadora consiga que se haya vaciado de aves la zona controlada, el sistema pasará a un **estado de post-alarma**, y se deberá desactivar la alarma sonora (supongamos que disponemos de la rutina ya escrita `desactiva_alarma()`). En ese estado de post-alarma, el sistema se mantendrá durante **2 minutos**, por precaución. Si durante este tiempo se detecta de nuevo la entrada de aves, se vuelve al estado de alarma. Si no, pasado ese tiempo sin aves, el sistema vuelve al estado normal.

Se pide: escribir en lenguaje pseudo algorítmico el programa principal y las rutinas de servicio siguientes: sensor SE, sensor SS, pulsador, teclado y reloj. Para simplificar el trabajo, conviene representar gráficamente el funcionamiento del sistema mediante un autómata de estados.