

Práctica 5 - Entrada/Salida

Organización del Computador 1

Segundo Cuatrimestre 2020

Ejercicio 1 Una computadora ORGA1 se encuentra conectada a 3 dispositivos de E/S que actúan como sensores. Cada sensor posee un registro de E/S que reporta información climática. Los sensores son:

- Temperatura: mide temperatura en grados celsius (TEMP_STATUS)
- Presión atmosférica: mide presión en hectopascales (ATM_STATUS)
- Intensidad del viento: mide intensidad en metros/segundo (WIND_SPEED)

Las etiquetas MAX_TEMP, MAX_ATM, MAX_WIND_SPEED son constantes de 16 bits.

a) Mapear los registros de E/S a direcciones de E/S de ORGA1 y completar el siguiente código:

```
senzar: CMP [...], MAX_TEMP ;¿se alcanzó la temperatura máxima?
        JG alarma
        CMP [...], MAX_ATM ;¿se alcanzó la presión atmosférica máxima?
        JG alarma
        CMP [...], MAX_WIND_SPEED ;¿se alcanzó la velocidad máxima?
        JG alarma
        JMP senzar
alarma: CALL sonarAlarma ;invocar la rutina de la alarma
        JMP senzar
```

b) Suponiendo que el ciclo de instrucción de cada instrucción del programa tarda t s y los valores máximos nunca se alcanzan ¿cuál es la frecuencia (en Hz) de muestreo (lectura) de los registros de E/S?

c) Suponga que reemplaza el procesador ORGA1 por un procesador ORGA1i que soporta hasta una interrupción. El sensor de temperatura solicita una interrupción si se alcanza la temperatura máxima.

- I. Modificar el programa presentado para aprovechar esta característica de modo tal que la frecuencia de muestreo sea mayor. ¿Cuál es la nueva frecuencia de muestreo en Hz?
- II. Escribir la rutina de atención de la interrupción del sensor de temperatura.

Ejercicio 2 En una computadora ORGA1i conectada a un dispositivo de E/S que la puede interrumpir, se está ejecutando el siguiente programa:

```
main:          MOV R0, 0x8000
                MOV R1, 0xFFFF
                ADD R0, R1
                JVS huboOverflow
noHuboOverflow: ...
                ...
huboOverflow:  ...
```

La rutina de atención de la interrupción es la siguiente:

```
rutinaAtencionInt:  ...
                   ...
                   PUSH R0          ; salvar el valor original de R0
                   PUSH R1          ; salvar el valor original de R1
                   MOV R0, 0x0000
                   MOV R1, 0x0000
                   ADD R0, R1
                   POP R1           ; restaurar el valor original de R1
                   POP R0           ; restaurar el valor original de R0
                   SETI             ; habilitar las interrupciones
                   RET              ; retornar a lo que se estaba ejecutando
```

Suponga que $I=1$ y en la dirección de memoria 0x0000 se encuentra efectivamente la dirección de la etiqueta `rutinaAtencionInt`

- a) ¿La ejecución del programa `main` continuará como se espera si el dispositivo de E/S activa la señal de interrupción durante el ciclo de ejecución de la instrucción `ADD R0, R1`?
- b) ¿Qué cambios realizaría a la rutina de atención de la interrupción para que su ejecución fuese transparente?

Ejercicio 3 El siguiente pseudo-código incompleto describe el ciclo de instrucción de un procesador ORGA1:

```
while (true) {
    Fetch()    // fetch
    Decode()   // decodificación
    Execute()  // ejecución
    if I==1 AND INTR==1 {
        ... Completar
        INTA()
    }
}
```

Completar el pseudo-código de modo tal que describa correctamente las acciones que lleva a cabo el procesador cuando atienda una interrupción.

Ejercicio 4 En una computadora ORGA1 se ha conectado un dispositivo de entrada. Éste posee un registro de E/S (`STATUS`) cuya dirección se *mapea* a la dirección de memoria 0xFFFF0.

La forma de interpretar este registro (de 16 bits: $b_{15} \dots b_0$) es la siguiente:

- El bit b_{15} representa el estado del dispositivo. Dicho bit se encuentra en 1 cuando el dispositivo esté ocupado (es decir, cuando todavía no hay un dato válido para leer), y en 0 cuando esté listo (caso contrario).
- Cuando el estado del dispositivo sea “listo” (es decir, el dato para leer sea válido), $b_7 \dots b_0$ contendrán el dato proporcionado por el dispositivo.

Escribir una rutina en assembler de ORGA1 que, implementando la técnica de *polling*, lea datos válidos del dispositivo tantas veces como se indique en `R0` y los escriba en forma contigua a partir de la dirección de memoria referida por `R1`. Se deberá guardar un dato en cada dirección de memoria, en la parte baja de la misma (es decir, en los bits 7...0), mientras que los bits restantes deberán ser ceros. Por simplicidad, no importa si se lee varias veces el mismo dato o se pierden datos válidos.

Ejercicio 5 Un procesador ORGA1 se encuentra conectado a 2 dispositivos de E/S:

- Un botón que posee un único registro de E/S de sólo lectura (BUTTON_STATUS). De leerse el valor 0x0000 el botón no está siendo presionado, y si se lee 0xFFFF, está siendo presionado. Este registro de E/S se encuentra mapeado a la dirección 0xFFFF0.
- Un display electrónico que posee un registro de E/S de sólo escritura (DISPLAY_DATA). El valor que se escribe en este registro se muestra como un número entre 0 y 65535. Éste se encuentra mapeado a la dirección 0xFFFF1.

Escribir una rutina que controle el procesador de manera que, comenzando con el valor 0, cuente la cantidad de veces que el usuario oprime un botón. Debe tener en cuenta que una pulsación del botón incluye esperar a que se *suelte*.

Ejercicio 6 El instrumental de navegación de un avión posee los siguientes monitores:

Prioridad	Monitor	Dirección RAI
1	Presión del aire en cabina	0xFF00
2	Altura	0xA000
3	Combustible	0xFE00
4	GPS	0x01FF

Cada monitor levanta una señal cuando registra un cambio en el entorno que está midiendo. Un monitor de mayor prioridad debe poder interrumpir la rutina de atención de interrupción de uno de menor.

Suponiendo que la cabina utiliza un procesador INTEL 8086 con un PIC 8259A (ver referencia en el apunte de arquitecturas):

- ¿En qué orden conectaría las entradas del PIC a las salidas de los monitores?
- ¿Cuál debería ser el valor inicial del registro IMR?
- Describe los valores almacenados en el vector de interrupciones.
- Si el monitor de Altura y de GPS levantan la señal de actualización de datos, ¿qué valor tendrá el registro IRR?
- Completar la siguiente rutina de atención:

RAI_MONITOR_ALTURA:

```
; Guardar la máscara
PUSH AX                ; apilar el registro AX
IN AX, IMR              ; copiar el contenido del reg. de E/S IMR en AX
PUSH AX                ; apilar la máscara actual

; Setear la máscara inhibiendo interrupciones de menor prioridad
MOV AX, .... (completar)
OUT IMR, AX             ; copiar el contenido de AX en el reg. de E/S INTMASK

; Habilitar interrupciones
.... (completar)

; Salvar el estado de lo que reste
.... (completar)
```

```

; Obtener la altura nueva
IN AX, 43h          ; copiar el contenido del reg. 43h de E/S en AX
LEA SI, MONITOR_ALTURA ; copiar la constante MONITOR_ALTURA en el reg. SI
MOV [SI], AX        ; copiar el contenido del registro AX en la
                    ; dirección apuntada por el reg. SI

; La altura ya fue actualizada, completar lo que falta para terminar

```

- f) ¿Es necesaria una instrucción especial para indicar el fin de la ejecución de una rutina de atención de interrupción o alcanza con el mismo RET que se utilizar para volver de un CALL?

Ejercicio 7 Se desea diseñar un nuevo modelo del procesador ORGA1i que soporte 2 niveles de interrupciones enmascarables: uno de prioridad alta (HP) y otro de prioridad baja (LP).

El flag I se reemplaza por 2 nuevos flags: HIF y LIF que habilitan y deshabilitan las interrupciones de prioridad alta y baja respectivamente. A su vez, los flags IFH y IFL indican si se solicitó una interrupción de prioridad alta o baja respectivamente.

La dirección de inicio de la rutina de atención de la interrupción HP se encuentra en la dirección de memoria 0x0000 y la de la rutina de atención de la interrupción LP en 0x0001.

- Describir en pseudo-código los pasos que realiza atómicamente el hardware del procesador desde que termina de ejecutar la instrucción actual hasta dejar paso a la rutina de atención de interrupción correspondiente.
- Explique qué instrucciones debe agregar o modificar para completar la arquitectura del nuevo procesador.

Ejercicio 8 Ordenar por prioridad decreciente los pedidos de interrupción de los siguientes dispositivos de E/S. Justificar brevemente.

- | | |
|---------------------|------------------|
| ▪ Reloj del sistema | ▪ Disco rígido |
| ▪ Teclado | ▪ Disco flexible |
| ▪ Puerto serial | ▪ Impresora |

Ejercicios tipo parcial

Ejercicio 9 Se posee un procesador ORGA1i para implementar un sistema de control de la barrera de un estacionamiento. El funcionamiento esperado del sistema es el siguiente:

- El automovilista presiona un botón que activa el motor de apertura de la barrera.
- Una vez que el vehículo cruza la barrera, un sensor especialmente ubicado activa el motor para cerrar la barrera.
- La barrera sólo funciona 12 horas al día desde el momento de encendido del sistema. Si el automovilista presiona el botón fuera del horario de atención, la barrera no se levantará.

Además del microprocesador ORGA1, se cuenta con los siguientes dispositivos de E/S:

- Un **botón** con un registro de E/S (BUTTON_DATA). De leerse el valor 1 de su registro, el botón está siendo presionado por el automovilista, mientras que si se lee el valor 0, no está siendo presionado.

- Un **motor** con un registro de E/S (**MOTOR_CTRL**) para controlar la barrera. Si se escribe el valor 1 en su registro, el motor levantará la barrera, mientras que si se escribe el valor 0, el motor bajará la barrera.
 - Un **sensor** con un registro de E/S (**SENSOR_DATA**). De leerse el valor 1 de su registro, el sensor está captando un automóvil. En cambio, si se lee el valor 0, no hay ningún automóvil en el rango captado por el sensor.
 - Un **clock** que emite un pulso a través de una línea de salida (llamada **CLOCK_OUT**) cada 10 minutos.
- a) Graficar un esquema de conexión del sistema indicando claramente cómo conectaría los registros de los dispositivos a los puertos de E/S de la máquina ORGA1i. No olvide graficar la línea de interrupción.
 - b) Escribir la rutina de atención de interrupciones para controlar el paso de las 12 horas. (**ayuda:** 12 horas equivalen a 720 minutos).
 - c) Escribir (primero en pseudo-código y luego en lenguaje ensamblador) la rutina de control del sistema. Recordar que la barrera sólo debe funcionar dentro del horario establecido.
 - d) Suponiendo que el microprocesador puede ejecutar 750 millones de instrucciones por segundo. ¿Cuánto tardaría en ejecutarse la rutina de atención de interrupciones?

Ejercicio 10 Una importante empresa está diseñando un nuevo modelo de tren que funcione de manera autónoma utilizando para su construcción una máquina ORGA1i y algunos dispositivos de E/S detallados a continuación:

SENSOR_AMBIENTE: informa el estado del entorno permitiéndole al tren decidir la acción a realizar. Cuenta con un registro llamado **STATUS** del que se puede leer 0x0000 si el camino está despejado, permitiéndole al tren avanzar libremente, 0x0001 si se debe avanzar con precaución y 0xFFFF si debe detenerse. Cada vez que este sensor cambia su estado pone en 1 a **INTR** y lo mantiene hasta recibir un pulso a través de **INTA**.

CONTROL_MOTOR: controla el motor del tren. Para que avance a velocidad crucero se debe escribir en el registro **POWER** el valor 0x7FFF, para que circule con precaución, el valor 0x30F0 y para detenerlo, 0xF000.

CUENTA_METROS: cuenta los metros recorridos en el registro **DISTANCIA**. Es posible reiniciar el contador colocando el valor 0xFFFF.

El funcionamiento del tren es sencillo. Siempre debe hacer lo que el estado de **SENSOR_AMBIENTE** le permita. Además se quiere contar la cantidad de kilómetros recorridos por el tren y guardarla en el registro **R0**.

- a) Mapear los registros de E/S e indicar para cada uno si es de lectura, escritura o lectura/escritura.
- b) Realizar el esquema de conexión del sistema, incluyendo todos los dispositivos involucrados y la manera en que están conectados.
- c) Realizar el pseudo-código de la rutina principal y de la RAI que se debe cargar en la máquina ORGA1i para lograr el funcionamiento detallado.
- d) Realizar el código, en lenguaje ensamblador, de la rutina principal y de la RAI.