Taller de Interrupciones

Organización del Computador 1

Verano 2021

El presente taller consiste en extender una micro-arquitectura diseñada sobre el simulador *Logisim* con el fin de soportar interrupciones. Se buscará implementar un controlador de interrupciones y codificar un programa que haga uso de este módulo.

El simulador se puede bajar desde la página `http://www.cburch.com/logisim/` o de los repositorios de Ubuntu. Requiere Java 1.5 o superior. Para ejecutarlo, teclear en una consola `java -jar logisim.jar`.

Procesador `OrgaSmallI`

Se adjuntan como parte de este taller las hojas de detalles del procesador `OrgaSmallI`.

Ejercicios

(1) **Introducción** - Recorrer la máquina y la hoja de datos, y responder:

a) ¿Qué componentes están involucrados en cada paso del ciclo Fetch - Decode - Execute de la máquina OrgaSmallI?

b) ¿En qué parte del ciclo de instrucción (Fetch - Decode - Excecute) deberían detectarse y atenderse las interrupciones?

c) ¿Dónde se almacenan las micro-instrucciones?

d) Explicar cómo podría hallar el micro-programa correspondiente a una instrucción cualquiera.

e) El taller pasado analizamos algunos micro-programas con micro-instrucciones condicionales ("`IF`"). ¿Qué mecanismo implementa la máquina para decidir si ejecutar su rama o no?

(2) **Interrupt Controller**

a) Desarrollar el componente teniendo en cuenta lo siguiente:

- El flag `I` debe encenderse al recibir la señal `STI`.
- El flag `I` debe apagarse al recibir la señal `CLI`.
- Al recibir la señal `INT` se debe almacenar la interrupción de manera asincrónica.
- Al recibir la señal `INTA` se debe dejar de almacenar la interrupción.
- La salida `INTR` debe valer 1 si y sólo si `I` = 1 y hubo un pedido de interrupción.

b) Dentro de la Unidad de Control ¿qué sucede si el cable con una X siempre vale 0? ¿Y si siempre vale 1?

Checkpoint 1

(3) **Micro-instrucciones** - Agregar en el archivo `microCode.ops` las micro-instrucciones necesarias para que:

a) Al momento de realizar un Fetch, en caso de haber una interrupción a atender, se sigan los siguientes pasos:

- guardar el valor de `PC` en la dirección `0xFF`.
- inhabilitar las interrupciones.
- informar que ha sido recibida la interrupción.
- saltar a la rutina de atención de interrupción.

b) Las instrucciones `STI` y `CLI` manejen a `I` como corresponde.

c) La instrucción `IRET` habilite las interrupciones y salte a la dirección almacenada en la posición `0xFF`.

Responder nuevamente a las preguntas del punto *2b)*.

Checkpoint 2

(4) **Ensamblar y correr** - Escribir en un archivo, compilar y cargar el siguiente programa:

```
SET R1, 0x03
SET R2, 0x00
SET R3, rai
STR [0x00], R3
STI

loop:
CMP R1, R2
JZ fin
JMP loop

fin:
CLI

halt:
JMP halt

rai:
DEC R1
IRET
```

a) Antes de correr el programa, identificar el comportamiento esperado.

b) ¿Cuántas interrupciones son necesarias como mínimo para llegar a `halt`?

c) ¿Qué sucede si se reciben varias interrupciones antes del `CMP`?

d) ¿Hay algún problema en recibir una interrupción entre el `CMP` y el `JZ`?

(5) **Programar**

Una empresa ferroviaria adquirió un procesador `OrgaSmallI` para controlar sus trenes de forma segura.

Con el fin de disminuir la velocidad en las curvas y así evitar accidentes, se dispone de un sensor que solicita una interrupción al detectar que se acerca una curva y otra cuando el tren termina de atravesarla.
Además de dicho sensor, se dispone de un dispositivo `Motor` que lee la **dirección 0xF0** periódicamente y, dependiendo del valor obtenido, realiza distintas acciones:

`0x0C` Cambia a velocidad para curva.

`0x0F` Cambia a velocidad máxima.

Para tranquilidad de los altos mandos de la empresa, se nos pide que dichos valores se escriban constantemente en la dirección mencionada.
Por último, nos piden que manejemos la bocina del tren, cuya intensidad controlamos escribiendo en la **dirección 0xF1**. Se nos pide que comience con el valor 2, y cada 32 curvas atravesadas con éxito, incrementemos su intensidad en 1 para compartir nuestra alegría con el pueblo.

Escribir un programa que maneje el tren como corresponde.