

Excepciones y llamadas al sistema

Departamento de Automática

Índice

- **Modos de operación**
 - Conmutación entre modos
- **Unidad de protección de memoria**
 - Registros
 - Modos de región
- **Excepciones**
 - Marco de excepción
- **Programación de llamadas al sistema**
 - Manejador de llamadas al sistema
- **Ejercicios propuestos**

<p>Modos de operación Unidad de protección de memoria Excepciones Programación de llamadas al sistema Ejercicios propuestos</p>	<p>Descripción general Comutación entre modos</p>
<p>Modos de operación Descripción general</p>	
<ul style="list-style-type: none">● El procesador implementa dos modos de operación:<ul style="list-style-type: none">○ Modo supervisor<ul style="list-style-type: none">■ Permite la ejecución del juego completo de instrucciones○ Modo usuario<ul style="list-style-type: none">■ No permite la ejecución de instrucciones privilegiadas (STI, CLI, IN, OUT, etc.)■ Si se intenta ejecutar una instrucción privilegiada en modo usuario se produce un error (excepción)● El sistema siempre comienza en modo supervisor● La CPU proporciona instrucciones para conmutar de modo	
<p>Sistemas Operativos</p>	<p>Excepciones y llamadas al sistema 3 / 15</p>

Las excepciones se describen en una transparencia posterior.

Conmutación entre modos

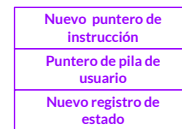
Instrucción SRET

- Conmuta la CPU de **modo supervisor a modo usuario**
- Conmuta a la pila de modo usuario
- Modifica el puntero de instrucción
- El nuevo valor de los registros lo obtiene del marco de supervisor



- El **marco de supervisor** (*supervisor frame*) es una estructura que contiene:
 - El nuevo valor del puntero de instrucción después de la conmutación
 - El valor que tendrá el puntero de pila de modo usuario
 - El valor que tendrá el registro de estado después de la conmutación
- El marco se almacena en **la pila de supervisor**

SP →



Conmutación entre modos

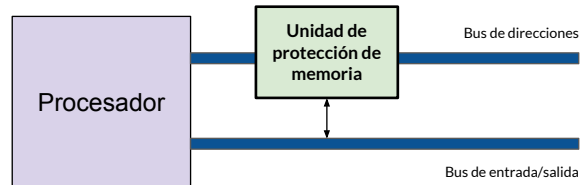
Instrucción SVC

- Conmuta la CPU de **modo usuario a modo supervisor**
 - Sólo se puede ejecutar en modo usuario
- Cuando se ejecuta, la CPU realiza las siguientes operaciones:
 1. Conmuta a la pila de modo supervisor
 2. Almacena en la pila el marco de supervisor:
 - Dirección de la **siguiente instrucción** a SVC
 - Puntero de pila de modo usuario
 - Valor previo del registro de estado
 3. Modifica el puntero de instrucción **¡Punto de entrada conocido y fijo!**



Unidad de protección de memoria

Descripción general



- El sistema proporciona una unidad de protección de memoria (*memory protection unit*):
 - Permite bloquear los accesos a memoria en modo escritura dentro (o fuera) de una región de memoria
 - El acceso se puede controlar para los dos modos de operación (usuario y supervisor)
 - Implementa dos registros: MEMPTSTART y MEMPTEND

Las excepciones se describen en una transparencia posterior.

Registros de la unidad de protección de memoria

Memory Protection Unit Start Register (MEMPTSTART)

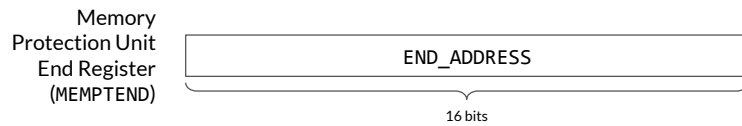


El registro MEMPTSTART establece la dirección inicial de la región de memoria y el modo de protección:

- Active. Protección de memoria: 1 ⇒ activada, 0 ⇒ desactivada
 - Block. Modo de región: 1 ⇒ bloque, 0 ⇒ segmento
 - User. Escritura en modo usuario: 1 ⇒ permitida, 0 ⇒ no permitida
 - Supervisor. Escritura en modo supervisor: 1 ⇒ permitida, 0 ⇒ no permitida
- La dirección de comienzo de la región será **START_ADDRESS * 16**

Registros de la unidad de protección de memoria

Memory Protection Unit End Register (MEMPTEND)

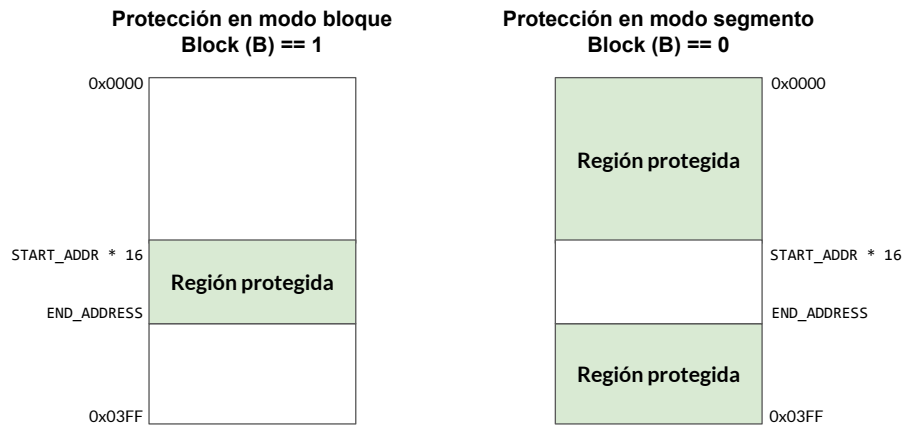


El registro MEMPTEND establece la dirección final de la región de memoria

Región de memoria = [START_ADDRESS * 16, END_ADDRESS]

¡No se debe modificar el registro MEMPTEND con la región activada!

Modos de región



Excepciones

Descripción general

- Una excepción es un error provocado durante la ejecución de una instrucción
- El simulador define seis tipos de excepciones:

Nº	Nombre	Descripción
0	DIVIDE_BY_ZERO	El procesador ejecuta una instrucción que resulta en una división por cero
1	INSTRUCTION_FETCH_ERROR	Error al obtener de memoria la siguiente instrucción a ejecutar
2	MEMORY_ACCESS_ERROR	El procesador ejecuta una instrucción que accede a una posición de memoria no válida
3	UNKNOWN_OPCODE	El procesador intenta ejecutar una instrucción con un código de operación inválido
4	ILLEGAL_INSTRUCTION	El procesador intenta ejecutar una instrucción privilegiada en modo usuario
5	STACK_ACCESS_ERROR	Error al acceder a la memoria correspondiente a la pila

Excepciones

Descripción general

- Las excepciones se pueden tratar **sólo si se producen en modo usuario**
 - Si se producen en modo supervisor, el procesador pasa a modo fallo (*fault mode*)
- Cuando se produce una excepción en modo usuario, el procesador realiza las siguientes operaciones:
 - Conmuta a modo supervisor
 - Guarda el marco de excepción (*exception frame*)
 - Modifica el registro puntero de instrucción

Instruction
Pointer

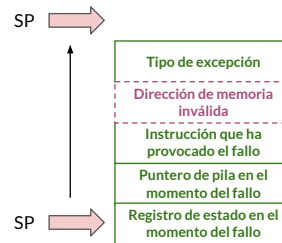
Valor previo



0x0009

Marco de excepción

- El marco de excepción contiene la información necesaria poder **identificar el fallo que ha ocurrido**:
 - El tipo (número) de excepción
 - La dirección de memoria cuyo acceso ha provocado el fallo (sólo si la excepción es de tipo MEMORY_ACCESS_ERROR)
 - La instrucción que ha provocado el fallo
 - El valor del puntero de pila de usuario en el momento del fallo
 - El valor del registro de estado en el momento del fallo
- El marco **se almacena en la pila de supervisor**



Programación de llamadas al sistema

- En un sistema convencional, las aplicaciones se ejecutan en modo usuario y el núcleo del SO en modo supervisor
- Las aplicaciones solicitan servicios al SO mediante llamadas al sistema:
 - Permiten ejecutar código privilegiado en nombre de la aplicación
 - Se implementan utilizando una instrucción específica (TRAP)
 - En el simulador, la instrucción de TRAP es SVC
 - El acceso al código del núcleo se realiza de forma controlada
 - Una vez ejecutado el servicio, el núcleo retorna a la aplicación

Programación de llamadas al sistema

- El manejador de llamadas al sistema es la rutina que se ejecuta cuando se produce el TRAP
- Un mismo manejador puede proporcionar múltiples servicios, cada uno con un número variable de parámetros
 - El manejador establece un mecanismo para indicar el tipo de servicio y la forma en que se pasan los parámetros:
 - Registros, pila, posiciones fijas de memoria, etc.
 - Las aplicaciones tienen que invocar las llamadas siguiendo las directrices del manejador

Comentar paso a paso el Ejemplo 5 (Sample 5). Explicar la directiva ORG.

Añadir algún tipo de manejo de una excepción sobre la marcha y provocarla.

Ejercicios propuestos

Añadir al programa del Ejemplo 5 las siguiente llamada al sistema:

```
print_pixel
```

Permite colorear un píxel determinado del display visual

- **Parámetros:**
 - **Registro A:** número de la llamada al sistema (2)
 - **Registro BH:** coordenada X del píxel a colorear
 - **Registro BL:** coordenada Y del píxel a colorear
 - **Registro CL:** color del píxel

Se deben implementar el código del sistema y el wrapper de usuario.