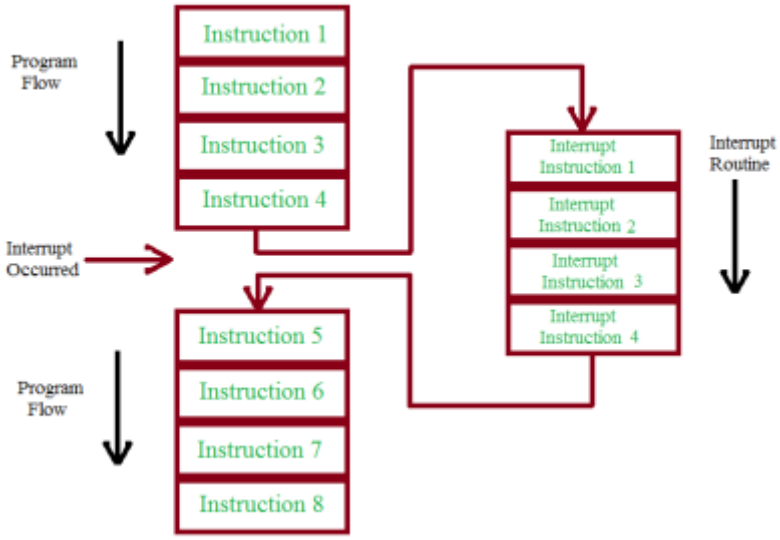
Interrupciones

Introducción

El microprocesador se encuentra en un entorno donde existen otros componentes. La forma de comunicación más usual entre estos componentes y el microprocesador se denomina interrupción. Básicamente, una interrupción es una petición que se hace a la CPU para que detenga temporalmente el trabajo que esté realizando y ejecute una rutina determinada. Por ejemplo, en la siguiente figura se puede ver que el flujo de un programa es interrumpido para ejecutar instrucciones de una interrupción:

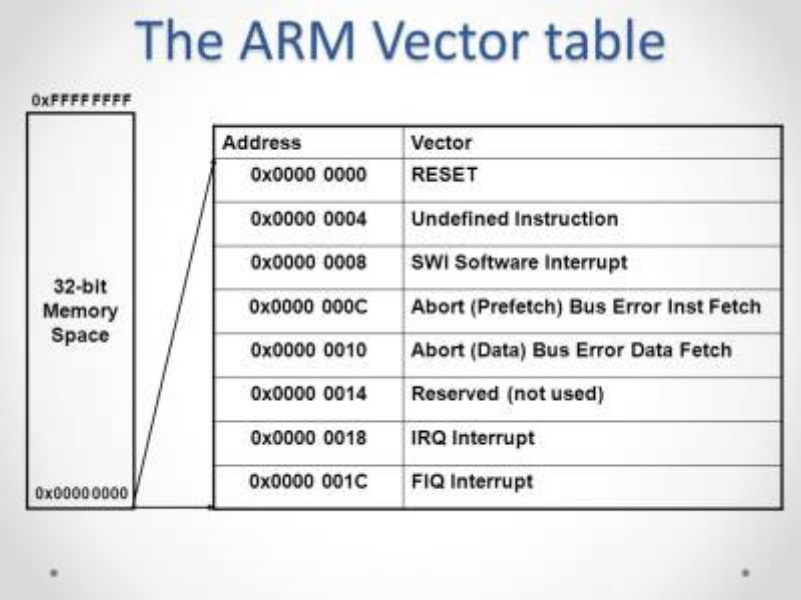


Existen dos tipos de interrupciones: hardware y software. Las interrupciones hardware son aquellas en las que su activación está condicionada por el hardware del sistema, ya sea por:

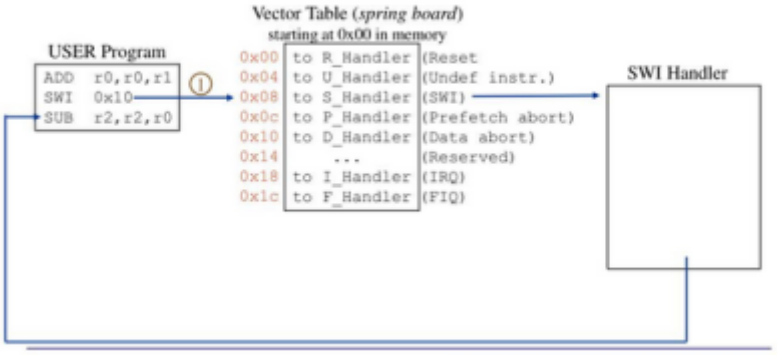
1. excepciones provocadas en la ejecución de alguna instrucción o error grave, o
2. provocadas por la placa base o por cualquier tarjeta implicada en un canal de E/S.

El sistema de interrupciones del ARM

Decimos que las interrupciones del ARM son autovectorizadas. Cada tipo de interrupción lleva asociado un número (que llamamos número de interrupción, NI) que identifica el tipo de servicio a realizar. En total hay 8 tipos de interrupciones. A partir de dicho número se calcula la dirección a la que salta la CPU para atender dicha interrupción. A diferencia de otras arquitecturas donde los vectores contienen las direcciones de las rutinas de tratamiento, en ARM no tenemos direcciones sino instrucciones. Cada vector contiene normalmente un salto a la rutina de tratamiento correspondiente. Dicha rutina se suele llamar RTI (**Rutina de Tratamiento de Interrupción**).



En la arquitectura ARMv6 todos los vectores de interrupción se almacenan en una zona de memoria llamada tabla de vectores de interrupción. Esta tabla comienza en la dirección física 0x00000000 (aunque puede cambiarse por 0xffff0000) y acaba en 0x0000001f y contiene en total 8 vectores de interrupción. Cuando termina de ejecutarse una RTI, el procesador continúa ejecutando la instrucción siguiente a la que se estaba ejecutando cuando se produjo la interrupción. El siguiente grafico muestra un ejemplo con la interrupcion SWI, que se verá en detalle mas adelante.



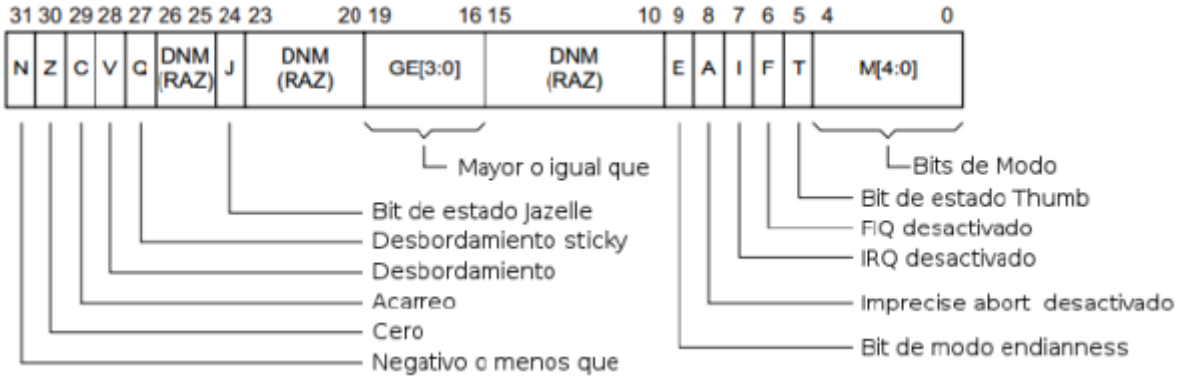
La lista del vector de interrupciones es la siguiente.

Excepción	**Tipo**	**Desplazamiento**	**Modo**
Reset	Interrupción	0x00	Supervisor
Instrucción no definida	Excepción	0x04	Indefinido
Interrupción Software	Int. Software	0x08	Supervisor
Error de prefetch	Excepción	0x0C	Abort
Error de datos	Excepción	0x10	Abort
Reservado	-	0x14	Reservado
IRQ	Interrupción	0x18	IRQ
IRQ	Interrupción	0x1X	IRQ

Vector de Interrupciones

La última columna se refiere al *Modo de operación* forma parte del registro cpsr. Es un estado en el que se encuentra el procesador con una serie de privilegios con respecto a otros modos y que gracias

a ellos podemos construir un sistema operativo con diferentes capas.



Registro CPSR

Cada modo tiene sus propios registros sp, lr y spsr (Saved Program Status Register) de tal forma que no alteramos la pila ni los flags de la secuencia de programa que interrumpimos. Incluso el modo FIQ tiene 5 registros generales propios (desde r8 hasta r12), de esta forma si los empleamos en nuestra rutina de tratamiento no tendremos que salvaguardarlos en pila.

SWI

SWI (Software Interrupt) provoca la excepción SWI

Sintaxis

```
  SWI{<cond>}  <immed_24>
```

Donde:

<cond> es la condicion donde es ejecutada la instrucción

<immed_24> es un valor inmediato de 24-bits y se usa para determinar que servicio del sistema operativo se esta solicitando. Cualquier parametro necesitado por el servicio es pasado por medio de un registro de propósito general

Operaciones principales realizadas por la instrucción:

```
   Si pasa la condicíon(cond)
   Luego
        R14_svc   = dirección de la próxima instrucción despues de SWI
        SPSR_svc  = CPSR
        CPSR[4:0] = 0b10011  /* Entrar a modo Supervisor */
        CPSR[5]   = 0        /* Ejecutar en estado ARM */
        CPSR[7]   = 1        /* Deshabilitar interrupciones*/
        PC    = 0x00000008
```

Ejemplo 1 : Fin de Programa.

```
mov r7, #1  // Indicamos salida del programa
```

```
swi 0           // Llamamos a la interrupción
```

Ejemplo 2 : Escribir en pantalla un mensaje.

```
   .data
   mensaje:
           .ascii "Ingresar texto 4 caracteres!:\n"
   .text
   .global main
   main:
           mov r7, #4       // Indicamos salida por pantalla
           mov r0, #1       // Indicamos que la salida es una cadena
           mov r2, #30      // Tamaño de la cadena
           ldr r1, =mensaje // Ubicación de la cadena
           swi 0            // Llamamos a la interrupción
```

Ejemplo 3 : Leer input del usuario.

```
   .data
   cadena:
           .ascii " "
   .text
   .global main
   main:
           mov r7, #3         // Indicamos lectura por teclado
           mov r0, #0       // Indicamos que el ingreso será una cadena
           mov r2, #4         // Leer #4 caracteres
           ldr r1, =cadena  // Donde se guarda la cadena ingresada
            swi 0         // Llamamos a la interrupción
```

From:
http://wiki.educabit.ar/ - **Wiki Sistemas**

Permanent link:
http://wiki.educabit.ar/doku.php?id=arm_interrupciones

Last update: **2025/09/11 22:48**