

Gestión de excepciones

POLICOM incorpora un eficiente y flexible mecanismo de gestión de excepciones, fruto del compromiso entre la arquitectura del procesador ARM7TDMI, tanto software como hardware, las necesidades del propio hardware específico que integra POLCIOM, y la facilidad de programación de este sistema.

En POLICOM se entenderá que está ocurriendo una excepción cuando determinados eventos, externos o internos a POLICOM, provoquen la alteración de la normal ejecución que en ese momento esté llevando a cabo en el procesador. Así pues la gestión de este tipo de situaciones está íntimamente relacionada con la forma con que el procesador de POLICOM trata las excepciones.

El tratamiento de Excepciones en el ARM7TDMI

Durante la ejecución normal de un programa, el contador de programa va incrementando secuencialmente por el espacio de direcciones, con saltos a subrutinas con o sin retorno. Pero dicha ejecución normal puede ser alterada por ciertos eventos, externos o internos, que son atendidos por el procesador a costa de abandonar temporalmente la ejecución que se encontraba en curso. A estos eventos que tienen la propiedad de alterar el funcionamiento del procesador se les denominan *excepciones*.

Las excepciones provocan el cambio de modo de operación del procesador.

Los eventos que provocan excepción en el procesador ARM7TDMI son los siguientes:

- **Reset:** dicha excepción es provocada cuando el pin de reset del ARM es activado. En ese instante el procesador pasa a modo supervisor y ejecuta la instrucción localizada en la primera posición memoria.
- **Undefined Instruction:** se da cuando el procesador no reconoce la instrucción que acaba de cargar. En el caso de POLICOM, si ocurre dicha excepción se ejecutará automáticamente la rutina de RESET, esto es, no existe manejador específico para cada tipo de excepción.
- **Software Interrupt:** Se trata de un mecanismo por el cual se permite a un programa ejecutándose bajo modo usuario entrar en modo supervisor, y con ello poder acceder a operaciones reservadas a este modo. Las interrupciones software ocurren cuando el procesador ejecuta una instrucción 'SWI', previamente programada por el usuario. Su utilización está relacionada con tareas propias de la gestión del sistema.
- **Prefetch Abort:** esta excepción ocurre cuando el procesador intenta acceder a una dirección que el controlador de memoria entiende que es ilegal (no existe o es reservada). Este mecanismo no ha sido implementado en POLICOM ya que los controladores de memoria no realizan funciones de protección de zonas de memoria o similares.
- **Data abort:** es provocada cuando el tipo de dato al que se quiere acceder (bytes de 8 bits, half-word de 16 bits, word de 32 bits) es incompatible con

la dirección especificada.(por ejemplo un word de dirección impar).
POLICOM no responde ante un evento de esta naturaleza.

- **IRQ:** se da cuando el pin de IRQ del procesador es activado.
- **FIQ:** se da cuando el pin de FIQ del procesador es activado.

De los siete tipos de eventos que pueden causar excepción en el procesador ARM7TDMI, POLICOM responde ante cuatro de ellos: reset, interrupción software, IRQ y FIQ. Si cualquiera de estos eventos se produjera, el procesador cambiará al modo característico de la excepción, ejecutando la el Manejador de Excepción (EH) correspondiente.

Ante la ejecución de una instrucción no definida, POLICOM responde como si de un reset software se tratara, esto es, volviendo a ejecutar la primera instrucción codificada en ROM.

POLICOM no responde a los eventos anteriormente denominados como '*prefetch abort*' y '*data abort*'.

Los eventos que en POLICOM provocan excepción desencadenarán la ejecución de los denominados Manejadores de Excepción (EH). Existen tantos EH como eventos atiende POLICOM, esto es, un total de cuatro. En cualquier caso, cada EH tendrá que realizar de manera genérica las siguientes tareas:

- Guardar el contexto (registro de estado, dirección de retorno, y registros de propósito general) empleando la pila del modo correspondiente. Para ello el procesador ya tiene unos registros en los que guarda automáticamente el estado en que se encontraba así como la dirección última que ejecutó antes de que ocurriera la excepción.
- Operaciones relacionadas con la naturaleza propia de cada evento
- Restaurar el contexto de la pila

Los Manejadores de Excepción que incorpora POLICOM se encuentran codificados en ROM interna. No pueden, por lo tanto ser alterados. Estas rutinas se encuentran en posiciones de memoria fijas, cuya dirección es proporcionada por la Tabla de Vectores de Excepción (EVT).

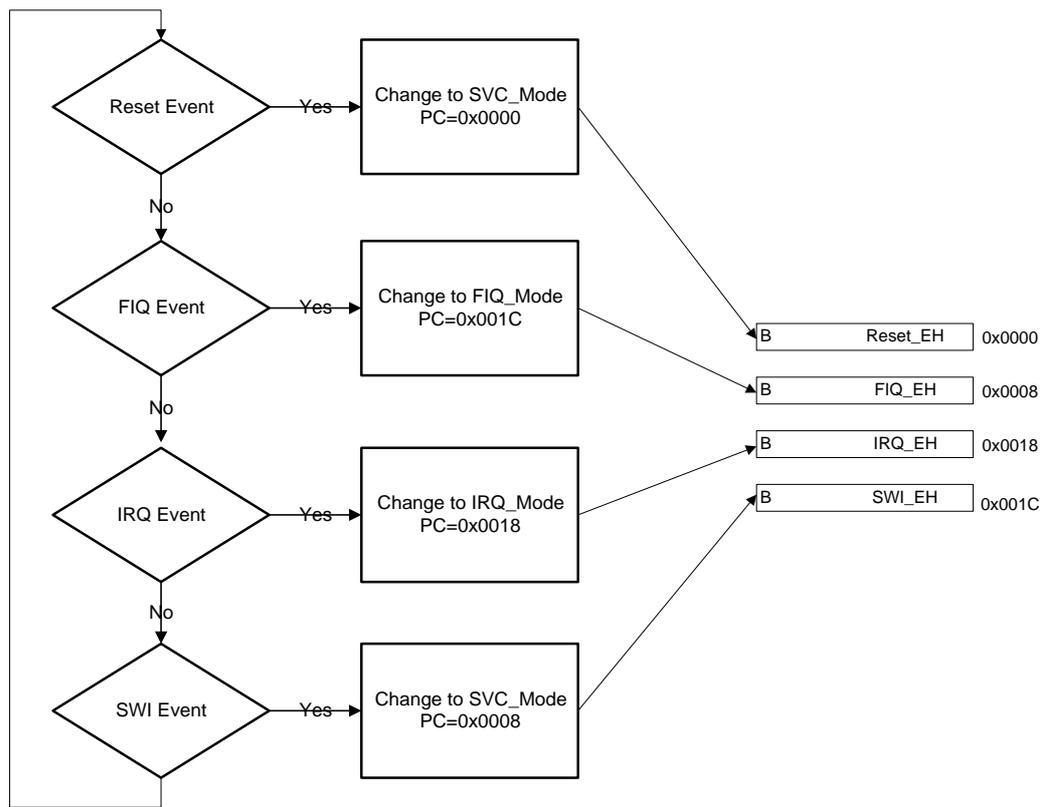
La Tabla de Vectores de Excepción (EVT)

La gestión de excepciones en POLICOM se lleva a cabo mediante la Tabla de Vectores de Excepción (EVT).

La EVT consiste en un área reservada de 32 bytes (1 word por excepción permitida en ARM) localizado al inicio del mapa de direcciones (correspondiente con ROM interna). Cada posición de la tabla, de tamaño word, corresponde a cada uno de los eventos descritos anteriormente, de tal manera que cuando ocurre alguno de los eventos, el procesador pasa a ejecutar la instrucción contenida en dicha posición. En cada una de estas posiciones existirá una instrucción de salto al manejador de excepción (EH) correspondiente. La siguiente figura muestra cómo se codifica la EVT en ensamblador ARM para el caso particular de Policom.

```
-----initial exception vector table EVT (BEGIN)-----  
    B    Reset_Handler  
    B    Reset_Handler  
    B    SWI_Handler  
    B    Reset_Handler  
    B    Reset_Handler  
    NOP                ; Reserved vector  
    B    IRQ_Handler  
    B    FIQ_Handler  
-----initial exception vector table EVT (END)-----
```

En el siguiente diagrama se aclara cómo POLICOM responde ante una determinada excepción.



En el momento de provocarse uno de los eventos señalados anteriormente, y siguiendo siempre la prioridad impuesta, el contador de programa pasará a contener uno de los valores de la EVT. La EVT por su parte lanzará la ejecución de la EH, realizando para ello un salto incondicional a la localización de dicho EH.

La Excepción de Reset

La excepción de Reset ocurre cada vez que se producen alguno de los siguientes eventos:

- el pin externo de reset de POLICOM ha alcanzado el valor '1',
- el WatchDog interno de POLICOM ha ordenado el reset al expirar el tiempo límite programado,
- y por último, se ha intentado ejecutar una instrucción que no reconoce la arquitectura 4T de ARM.

Esta excepción será atendida por la Reset_EH. Esta rutina se encuentra en ROM y es por lo tanto inaccesible para el programador. Su principal función es forzar al sistema Policom a un estado conocido. De esta manera, el Reset Exception Handler contiene las primeras instrucciones que el sistema va a procesar.

Al realizar un RESET al sistema, el bus de direcciones sacará la dirección 0x000000. En dicha posición de memoria de la EVT se encuentra una instrucción de salto a la dirección base del manejador de reset. De esta manera pasamos la ejecución al manejador.

El estado del POLICOM en el instante del reset es el siguiente:

- Modo del procesador: Supervisor
- Modo de ejecución ARM (ejecutará instrucciones del set de ARM).
- Bus de direcciones a 0x000000
- Interrupciones del tipo IRQ inhabilitadas.
- Interrupciones del tipo FIQ inhabilitadas.
- Todos los periféricos han sido iniciados a su valor de reset por defecto.

La rutina realizará las siguientes tareas de forma secuencial (ver diagrama de flujo correspondiente):

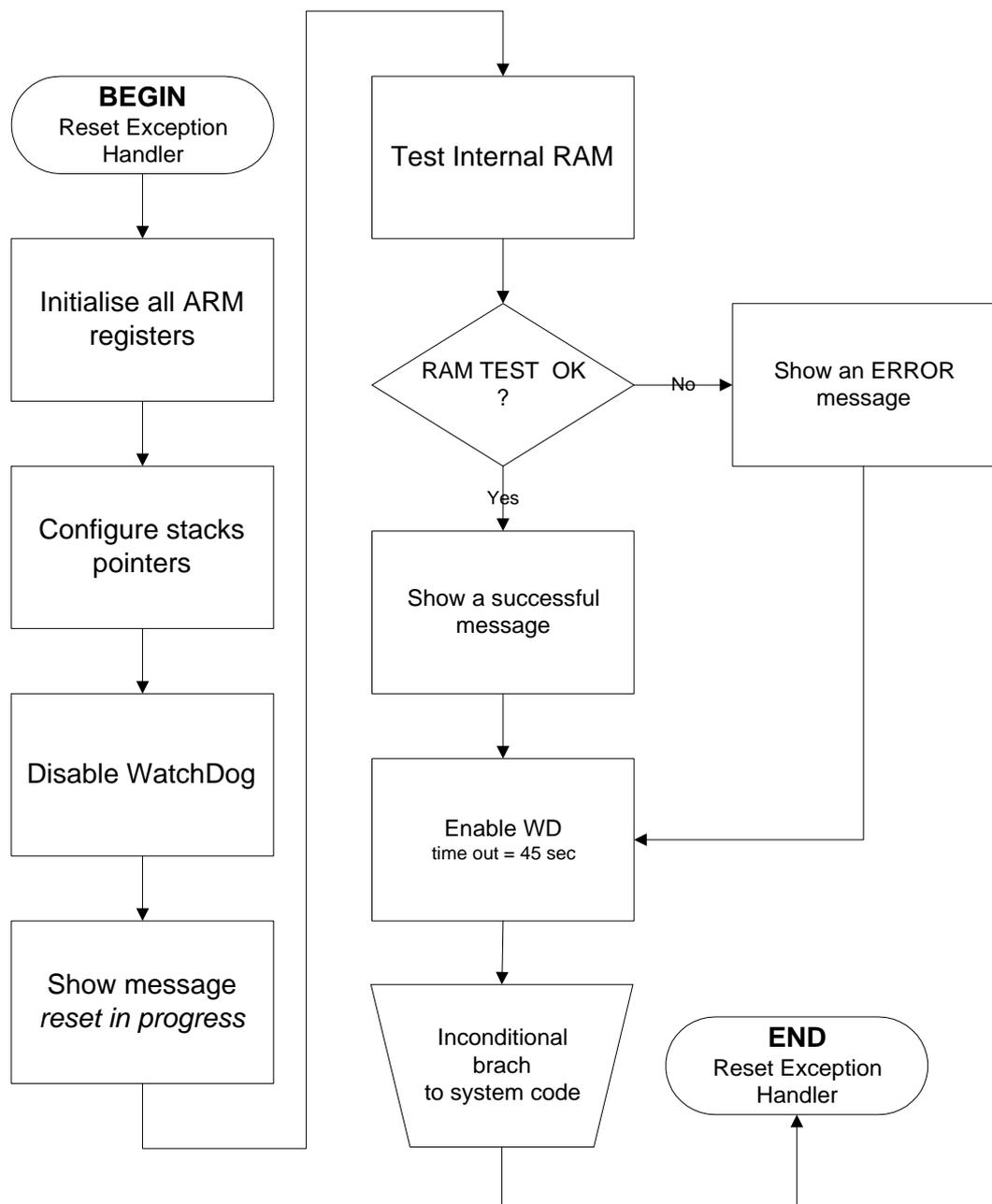
- Inicialá al valor cero todos los registros internos del ARM7TDMI. Esto es imprescindible para evitar partir de un estado desconocido de los registros del microprocesador.
- Configurar la direcci3n de la pila para el modo IRQ. Para ello tendremos que entrar en modo IRQ, cargar el SP (registro n3mero 13) con la direcci3n de memoria correspondiente a la pila para interrupciones y volver a modo supervisor.
- Una vez en modo supervisor tambi3n se inicialá el valor del SP de dicho modo con el valor correspondiente para la pila de modo supervisor. Esta operaci3n se realiza sin tener que salir del modo supervisor.
- Inhabilitar el watchdog para evitar un posible reset espont3neo debido a un valor inicial del watchdog.
- Chequeo de toda la RAM interna. Para ello se procederá a escribir en cada word de memoria RAM interna un contador que comenzará con el valor cero y se ir3 incrementando conforme vaya recorriendo la memoria hasta el 3ltimo word de memoria. Una vez escrita toda la memoria, se procede a leer el contenido de cada word, comprobando que su contenido sea el correcto, tal y como se ha ido escribiendo. Por 3ltimo limpiamos el contenido del word procesado. Durante todo este proceso de comprobaci3n de la memoria interna RAM, se mostrar3 por el puerto paralelo un mensaje indicando que se est3 chequeando la RAM.
- Chequeo de otros dispositivos que en su momento se consideren necesarios.
- Habilitar el watchdog con un time out suficientemente alto.
- Saltar a c3digo de sistema.

Al final de la ejecuci3n de la rutina de reset, POLICOM se encontrar3 en el siguiente estado:

- Modo supervisor (SVC)
- IRQ inhabilitada en ARM, y todas las causas enmascaradas en el AIC.

- RAM completa a 0x000000 (es decir, todas las zonas se encuentran a este valor).
- Punteros de pila con el valor inicial según el mapa de memoria de POLICOM.
- El puerto mostrará el mensaje correspondiente al resultado del test de la RAM interna.
- Modo de ejecución ARM.

Diagrama de flujo.



Como puede verse en el diagrama de flujo, se mostrará durante el proceso de inicialización del sistema unos mensajes que tienen como misión proporcionar al usuario cierta información de cómo ha resultado el proceso. EL dispositivo empleado para mostrar los mensajes dependerá de la aplicación final. En simulación se ha utilizado el BPP (puerto paralelo).

Las Excepciones IRQ y FIQ

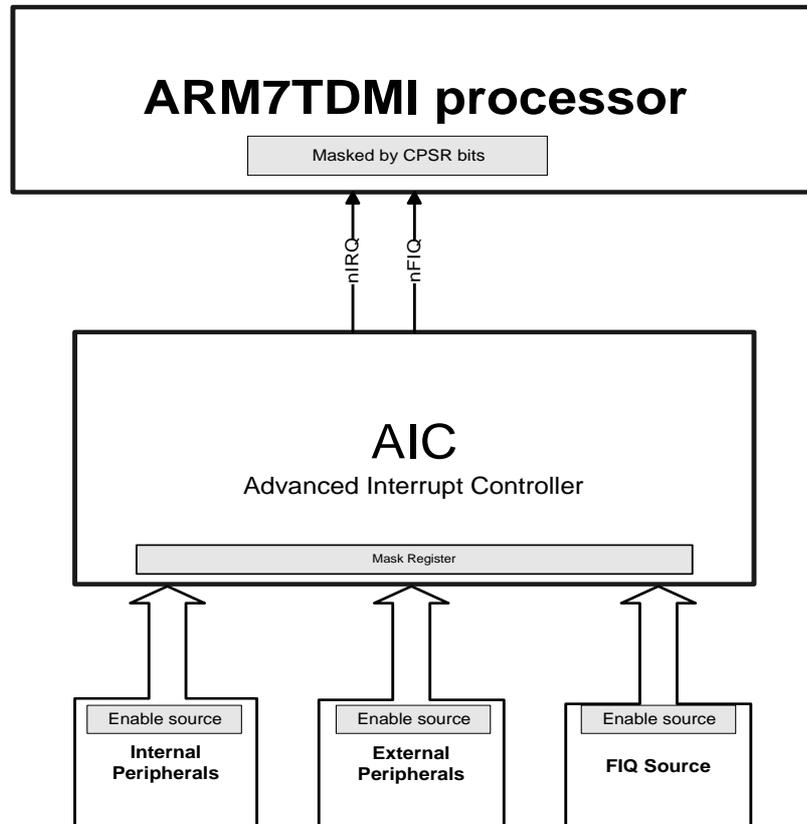
Como se ha explicado anteriormente las excepciones en general son provocadas por eventos, tanto internos como externos al procesador. Para el caso particular de las IRQ's y FIQ, lo que provoca la excepción es un evento externo al procesador consistente en la activación de las líneas IRQ y FIQ respectivamente. Estos tipos de excepción están orientados a que sean dispositivos externos los que sean capaces de provocar excepción.

A las excepciones provocadas por dispositivos externos al procesador, tanto internos a POLICOM como externos a POLICOM, empleando las líneas IRQ o FIQ, las denominaremos genéricamente como *interrupciones hardware*. Este mecanismo de atención de los periféricos permite la gestión óptima de los mismos.

Las líneas IRQ y FIQ del procesador son gestionadas desde el punto de vista físico por un dispositivo hardware integrado en POLICOM denominado 'Advanced Interrupt Controller'. El AIC estará conectado a todos los dispositivos internos o externos a POLICOM que puedan provocar interrupción. Su misión es ofrecerle al programador la posibilidad de configurar de manera flexible la gestión de las líneas de interrupción que finalmente llegan al procesador. Ver figura.

Desde el punto de vista del software, el tratamiento de este tipo de excepción implica la existencia de tres rutinas que serán explicadas con detalle:

- Los Manejadores de Excepción (IRQ_EH y FIQ_EH).
- La Rutina de Configuración de la Interrupción (DeviceX_ICR), que permite al programador configurar que un determinado dispositivo emplee este mecanismo de interrupciones.
- Las Rutinas de Atención de Interrupción (IAR), encargadas de atender la causa puntual que provocó la interrupción.



IRQ's y FIQ son tratadas en POLICOM de manera similar, si bien hay dos diferencias que las caracterizan: por un lado, IRQ y FIQ tienen diferentes manejadores de interrupción (EH), y por otro, la FIQ es más prioritaria que las IRQ's.

Respuesta de POLICOM ante una interrupción IRQ

Niveles de la respuesta

El esquema de la figura muestra la respuesta de POLICOM ante una excepción del tipo IRQ. Dicha respuesta puede desglosarse en tres niveles de atención según los recursos software o hardware que se utilicen:

- Nivel 1: activación de la línea de nIRQ del ARM. Son procedimientos puramente hardware.
- Nivel 2: ejecución del manejador de excepciones IRQ_EH. Se trata de software codificado en ROM y por los tanto inalterable por el programador.

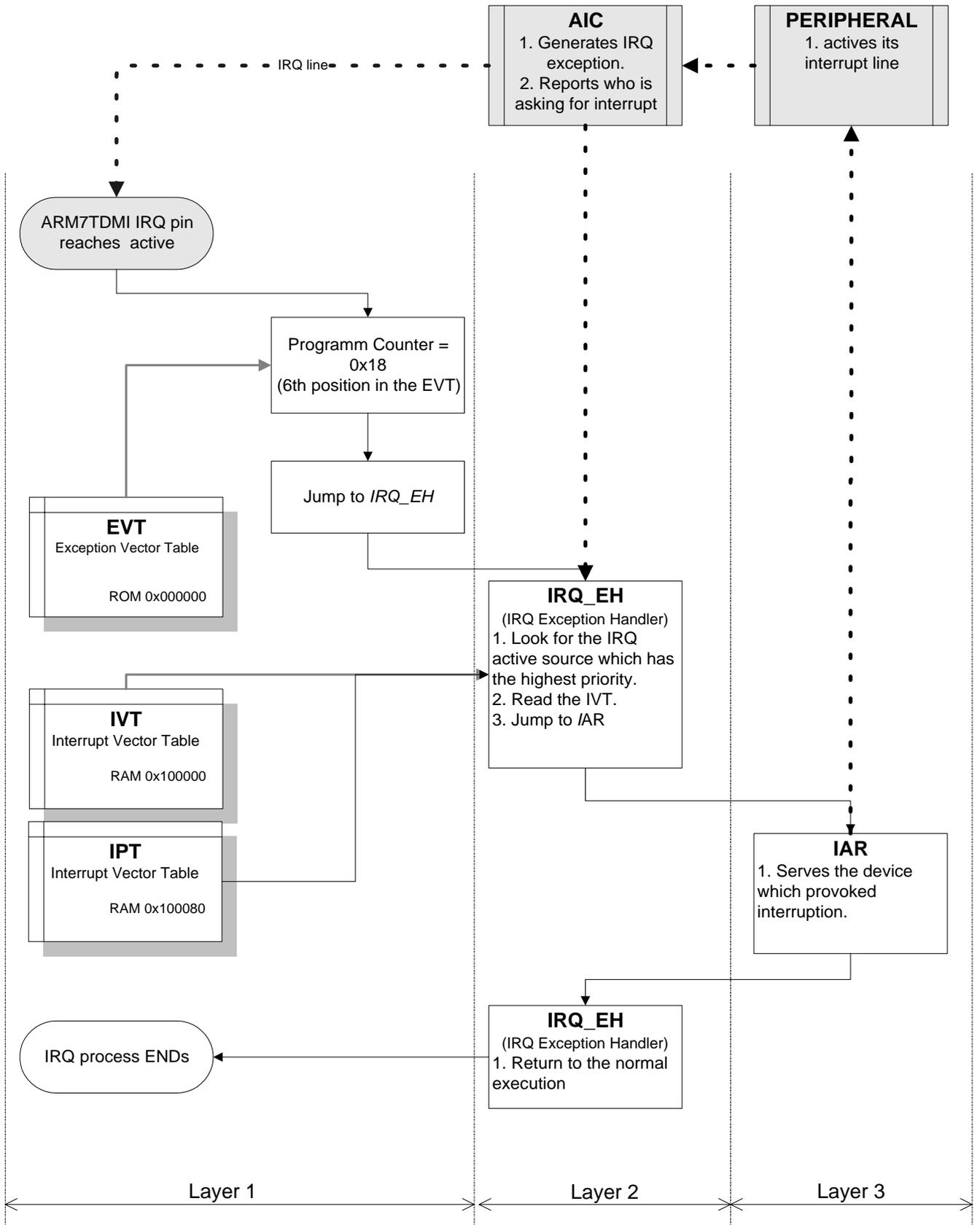
- Nivel 3: atención software del dispositivo que provocó la interrupción.

La primera de las etapas consiste en la respuesta que el propio procesador hace cuando su línea de IRQ se activa como consecuencia de que existe algún periférico que pide a través del AIC que le atienda el procesador. En ese momento el procesador pasará a estar en modo IRQ, colocando la dirección de la Tabla de Vectores de Excepción (EVT) correspondiente a esa excepción (0x18). A partir de este momento la pila empleada por defecto es la pila IRQ. El procesador accederá a la EVT para cargar la instrucción que se encuentra codificada en la posición 0x18 de la EVT. En esa posición se encontrará con una instrucción de salto incondicional al Manejador de Excepción IRQ, IRQ_EH, cuya dirección es fija.

Posteriormente pasamos a ejecutar el IRQ_EH. Se trata de una rutina codificada en ROM interna, y por lo tanto inaccesible al programador, cuya misión es detectar cuál es el dispositivo más prioritario que se encuentra pidiendo al procesador ser interrumpido gracias a la información que le ofrecen el AIC (hardware) y la IPT (Tabla de Prioridades de Interrupción). Una vez se conoce quién tiene que ser atendido, se consulta la Tabla de Vectores de Interrupción (IVT) para conocer cuál es la dirección de la rutina que finalmente servirá la petición de interrupción. A esta rutina se le denominará Rutina de Atención de Interrupción (DeviceX_IAR).

Por último se procederá a la ejecución de la rutina DeviceX_IAR, encargada de atender el dispositivo etiquetado como 'DeviceX' y que estaba pidiendo interrupción. El código de dicha rutina deberá ser el adecuado para atender a cada dispositivo. Su dirección es dinámica, y se conoce a través de la IVT, previamente programada.

Como se puede ver, el sistema de gestión de interrupciones de POLICOM es doblemente vectorizado y priorizado por software. Es decir, existe una primera vectorización estática a través de la EVT, y otra vectorización dinámica a través de la IVT. La priorización se realiza mediante un algoritmo que se ejecuta en la EH y permite conocer cuál es la fuente de interrupción activa más prioritaria, a partir de unos valores de prioridad que han sido previamente programados por el usuario en la IPT.



El IRQ_EH

En la figura se muestra el diagrama de flujo del Manejador de Excepciones IRQ, IRQ_EH, que POLICOM contiene codificado en ROM interna.

El IRQ_EH se encarga de:

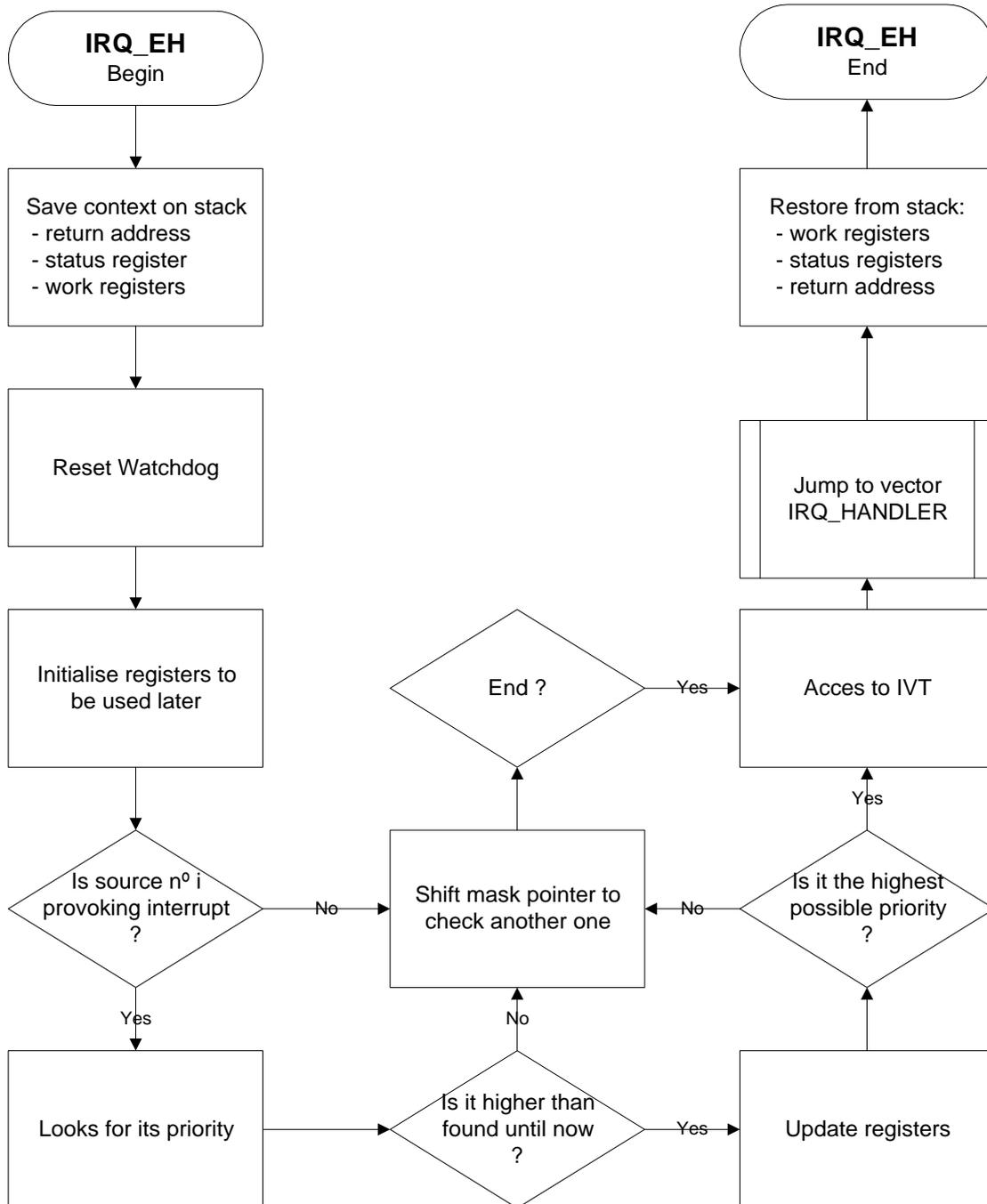
- guarda el contexto en la pila, incluyendo la dirección de retorno de cuando se produjo la interrupción.
- busca entre las líneas de interrupción activas cuál es la más prioritaria, leyendo la prioridad previamente programada de la Tabla de Prioridades de Interrupción (IPT) residente en RAM interna
- leer el vector de interrupción IRQ correspondiente a la interrupción más prioritaria de las activas consultando para ello la IVT.
- saltar a la rutina de atención de interrupción apuntada por el vector anteriormente leído.
- Restaurar el contexto.

Es importante resaltar que cuando se está atendiendo una interrupción no es posible que el procesador atienda a otra, ya que el manejador se encarga de inhabilitar las interrupciones mientras sirve a una. Es decir, no se permite anidamiento.

En cualquier caso, las propiedades de anidamiento de interrupciones de Policom dependerán del código finalmente implementado con Manejador de IRQ, no del hardware asociado.

El estado del procesador en el momento de la ejecución es el siguiente:

- Modo de procesador IRQ.
- Ejecución en set de instrucciones ARM.
- El SPSR contiene el estado del procesador antes de que ocurriera la IRQ.
- IRQ no están habilitadas.



La Tabla de Vectores de Interrupción (IVT).

Esta tabla contiene las direcciones de inicio de las Rutinas de Atención de Interrupción (IAR), también llamados vectores de interrupción, asociados a cada una de las fuentes de interrupción del tipo IRQ.

La tabla comienza en la dirección más baja de la RAM interna. En dicha posición estará situado el vector número 0, correspondiente a la rutina de atención de interrupción de la fuente cuya etiqueta de dispositivo (DeviceX) es 0.

Cada Vector de Interrupción ocupará 4 bytes.

Será el usuario el que programe adecuadamente los valores de esta tabla, y el manejador de IRQ (IRQ_EH) el encargado de pasar el control del procesador a la rutina de atención de interrupción apuntada por el vector correspondiente.

Para ver información detallada ver *Mapa de Memoria Interna*.

La Tabla de Prioridades de Interrupción (IPT).

Está compuesta por 32 bytes, uno por cada fuente de interrupción del tipo IRQ. Cada elemento de la tabla contiene un número de 0 a 7 que indica la prioridad de la IRQ que se está produciéndose, siendo la cero la más prioritaria.

El sistema de prioridades por el que se ha optado en POLICOM no permite anidar interrupciones, es decir, cuando una rutina de atención de interrupción IRQ está en curso, el procesador no atiende ninguna otra interrupción ya que el manejador de interrupciones IRQ ha enmascarado en el procesador la señal IRQ. Esta situación se mantendrá hasta que finalice la ejecución de la rutina de atención de la interrupción actualmente activa. En este momento sí podremos atender una nueva causa de interrupción.

Se entiende que las interrupciones están priorizadas en el sentido que cuando hay varias causas de interrupción activas y esperando ser atendidas, el procesador atenderá aquella que tenga mayor prioridad.

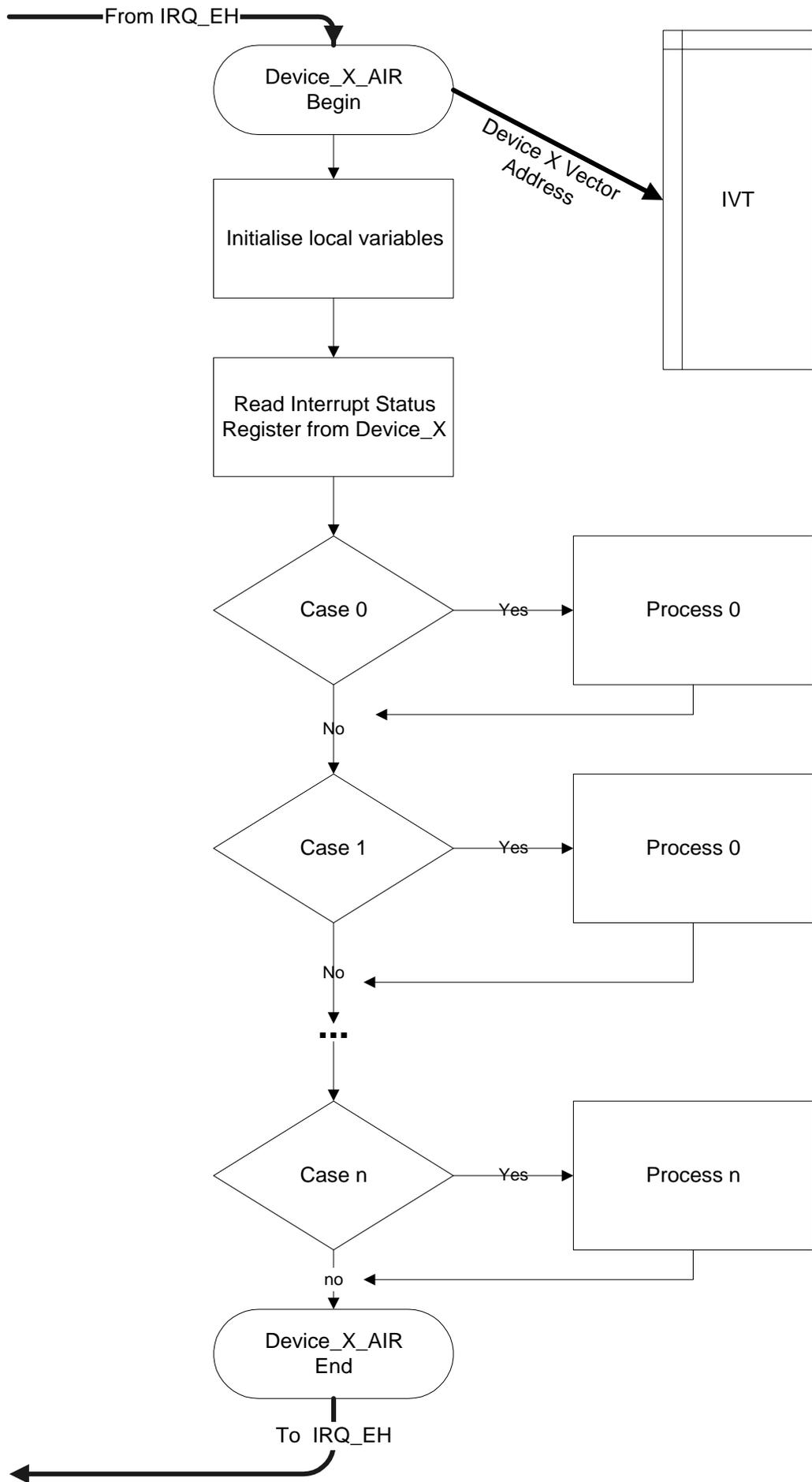
Las Rutinas de Atención de Interrupción (IAR)

La Rutina de Atención de Interrupción (IAR) tiene por objetivo atender el periférico que está provocando interrupción. Así pues su código dependerá del dispositivo que en cada momento deba ser atendido.

Generalmente las rutinas de atención de interrupción eliminan la causa que provocó la interrupción. El esquema a seguir es el mostrado en la figura.

Como puede verse en la figura, una vez que se ha pasado el control a la IAR, esta leerá un registro de estado de interrupciones del dispositivo en cuestión (Device _X) y procederá a chequear las diversas causas por las cuales el dispositivo ha podido solicitar interrupción. Para cada fuente de interrupción activa, la rutina deberá ejecutar un proceso cuya misión es eliminar, mediante las acciones necesarias sobre el dispositivo, la causa de interrupción. Una vez finalizado este procedimiento de chequeo-procesado, se devuelve el control al IRQ_EH.

Se recomienda ver como ejemplo la rutina de atención de interrupción presentada en el modo de operación de la UART.



Instalación de Interrupciones. Las ICR

Como se vio al comienzo del capítulo, existen tres rutinas que intervienen en la gestión de las interrupciones:

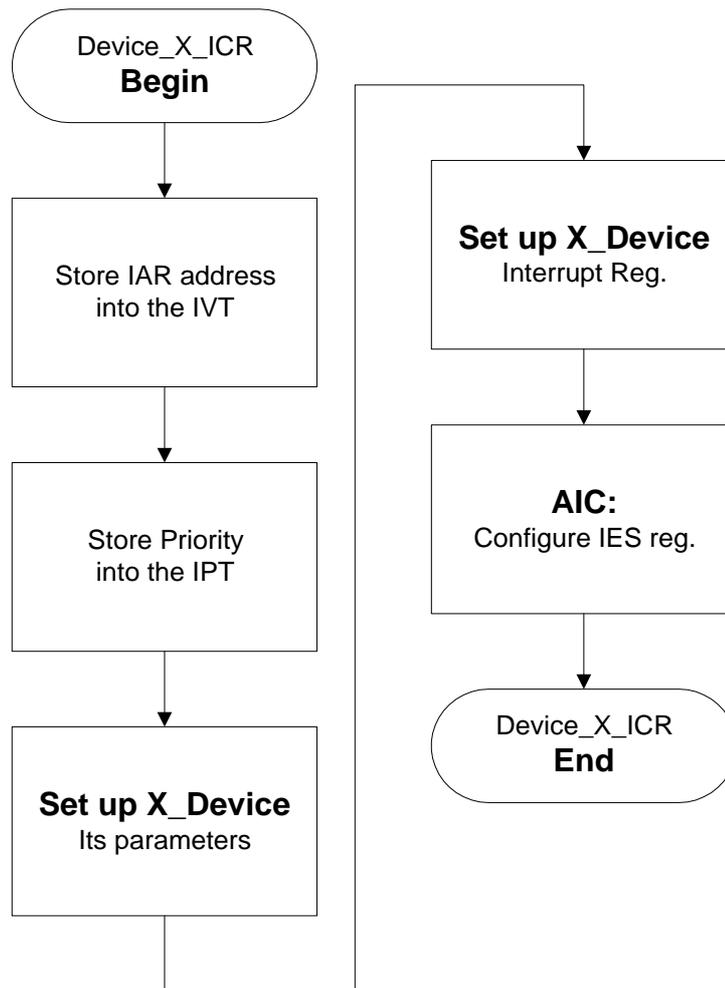
- El Manejador de Interrupciones(IRQ_EH y FIQ_EH), codificado en ROM y por lo tanto inaccesible al programador.
- Las Rutinas de Atención de Interrupción (Device_x_IAR), programadas por el usuario para atender a los dispositivos que requieren ser atendidos.
- La Rutina de Instalación de Interrupción (Device_X_ICR), cuya misión es proporcionar al sistema de gestión de interrupciones de POLICOM la información necesaria para poder llevar a cabo dicha gestión de manera adecuada.

Las Rutinas de Instalación de Interrupción (ICR) deberán realizar las siguientes operaciones sobre POLICOM:

- Configurar el dispositivo concreto para que puede generar la línea de interrupción dedicada. Generalmente los dispositivos disponen de un registro donde se indica que el dispositivo queda autorizado para poder activar la línea de interrupción ante causas prefijadas previamente.
- Cargar el vector de la rutina de atención de interrupción (IAR) en la Tabla de Vectores de Interrupción (IVT). Dicha rutina atenderá al dispositivo cuando éste solicite interrupción.
- Cargar la prioridad de la que gozará la interrupción en la Tabla de Prioridades de Interrupción.
- Habilitar la fuente de interrupción en cuestión en el Controlador Hardware de Interrupciones (AIC).

Es posible que para algunas aplicaciones sea necesario inhabilitar las IRQ en ARM para evitar inestabilidad mientras se configura la ICR. Esta tarea siempre se realizará en modo supervisor, y teniendo en cuenta que durante ese tiempo el procesador no atenderá causa alguna de interrupción.

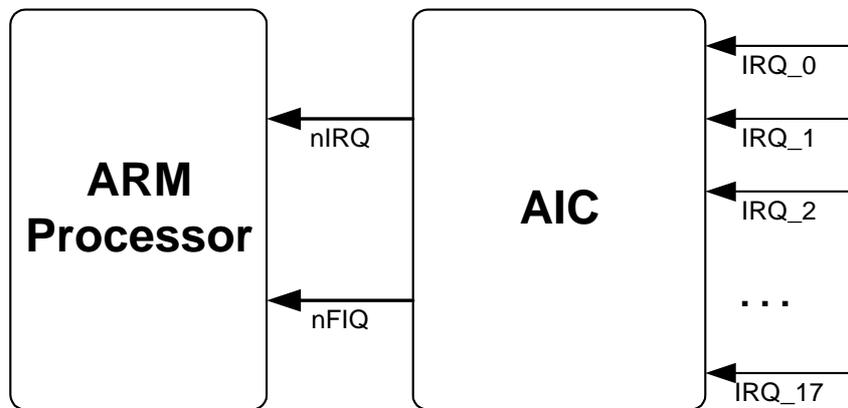
El orden a seguir queda especificado en el siguiente diagrama de flujo:



Programación de Controlador Hardware de Interrupciones

Introducción

Desde el punto de vista del programador el AIC es un dispositivo integrado en POLICOM para facilitar la gestión de interrupciones de los distintos dispositivos que también integra POLICOM y que pueden provocar interrupción. EL AIC controlará tanto la línea IRQ como la FIQ que llegan al procesador, como se muestra en la figura. Lo hará de forma programable, esto es, el usuario podrá controlar cada una de las líneas de interrupción programando adecuadamente el AIC.



Numeración de las Interrupciones

Las diferentes causas de interrupción existentes en el sistema se distinguen mediante un identificador de interrupción (IID) según la siguiente tabla:

Identificador de Interrupción	Dispositivo que provoca interrupción
0	Reservado a FIQ
1	Interrupción Programada
2	Reservado
3	Reservado
4	Temporizador 1 (Timer 1)
5	Temporizador 2 (Timer 2)
6	Reservado
7	Reservado
8	UART
9	Reservado
A	Puerto paralelo (BPP)
B	Convertidor Analógico Digital (ADC).
C	Modem SBM
D	Puerto serie de alta velocidad (SPI)
E	Externa nº 1
F	Externa nº 2
10	Externa nº 3
11	Externa nº 4

Es importante destacar que esta numeración no sólo se emplea para distinguir las diferentes causas de interrupción en el AIC, sino que también

sirve para indexar las tablas de vectores y prioridades de interrupción descritas anteriormente (IVT y IPT respectivamente).

Registros Programables del AIC

Los registros del AIC se encuentran mapeados a partir de la dirección base 41_0000. Básicamente existen tres tipos de registros:

- de habilitación de interrupción, encargados de autorizar que la activación de la fuente de interrupción se propague hasta el procesador.
- de inhabilitación, que cortan dicha posibilidad.
- y de estado de las interrupciones, que informan en cada momento de cuáles son las fuentes de interrupción activas, tanto de las que se encuentran habilitadas como de las que no lo están.

Si bien el AIC trata conjuntamente a las interrupciones del tipo IRQ y a la del tipo FIQ, al nivel de registros quedan bien diferenciados en dos grupos independientes de registros.

La tabla siguiente describe los registros internos del AIC. En ella se especifica la dirección del registro, su nombre, descripción, tipo de acceso, el tipo de dato ARM y su valor por defecto.

Dirección (hex.)	Nombre	Descripción	Acceso	Dato	Valor inicial
410000	aic_IRS	Interrupt Request Status. Informa qué fuentes de las que está habitadas se encuentran activas. Aic_IRS [i] = ‘0’ inactiva o inhabilitada. ‘1’ activa y habilitada.	Lectura	W	0
410004	aic_ISS	Interrupt Source Status. Informa de las fuentes de interrupción que se encuentran activas. Aic_ISS [i] = ‘0’ inactiva. ‘1’ activa.	Lectura	W	0
410008	aic_IER	Interrupt Enable Register Informa de cuales son las fuentes de interrupción IRQ que se encuenra habilitadas. aic_IER= ‘0’ → Fuente i habilitada. ‘1’ → Fuente i inhabilita	Lectura	W	0
	aic_IES	Interrupt Enable Set Habilita individualmente cada un de las fuentes de interrupción IRQ. aic_IER [i] = ‘0’ → no afecta. ‘1’ → Habilita fuente i.	Escritura	W	-
41000C	aic_IEC	Interrupt Enable Clear Inhabilita individualmente cada una de las fuentes de IRQ. aic_IEC [i] = ‘0’ → no afecta. ‘1’ → Inhabilita fuente i.	Escritura	W	-
410010	aic_SIT	Software Interrupt Toggle aic_SIT [i] = ‘0’ → desactiva int. Programada ‘1’ → activa int. Programada	Escritura	-	
410100	aic_FSS	FIQ Status Source Informa si la línea FIQ se encuentra activa y habilitada.	Lectura	1	0
410104	aic_FRS	FIQ Request Status Informa si la línea FIQ se encuentra activa.	Lectura	1	0
410108	aic_FER	FIQ Enable Register Informa si la línea FIQ puede o no provocar excepción. ‘0’ → Fuente FIQ inhabilitada. ‘1’ → Fuente FIQ habilita	Escritura	1	0
	aic_FES	FIQ Enable Set aic_FES [0] = ‘0’ → no afecta. ‘1’ → Habilita fuente FIQ.	Escritura	1	0
41010C	aic_FEC	FIQ Enable Clear aic_FEC [0] = ‘0’ → no afecta. ‘1’ → Inhabilita fuente FIQ.	Escritura	1	0

Cuadro resumen

Objeto	Identificador	Descripción	Dirección	Acceso
Manejador de Excepción del tipo IRQ	IRQ_EH	Rutina que se ejecuta cada vez que un dispositivo quiere ser atendido vía IRQ. Pasa el control a la IAR según su prioridad programada en la IPT.	En la EVT	No accesible
Manejador de Excepción del tipo FIQ	FIQ_EH	Rutina que se ejecuta al activarse la línea FIQ (externa a POLCOM). Pasa el control a la FIQ_IAR.	En la EVT	No accesible
Tabla de Vectores de Excepción	EVT	Tabla codificada en ROM. Vectoriza todas las excepciones del ARM.	ROM 0x000000	No accesible
Tabla de Vectores de Interrupción	IVT	Tabla codificada en RAM, indexada al IID y que proporciona las direcciones de las IAR.	RAM 0x100000 a la 0x10005F	Programable
Identificador de Interrupción	IID	Número que identifica a cada una de las interrupciones.	No procede	No procede
Tabla de Prioridades de Interrupción	IPT	Tabla residente en RAM que alberga las prioridades previamente programadas por el usuario.	RAM 0x100060 a la 0x100077	Programable
Rutina de Atención de Interrupción del dispositivo Device_x	Device_x_IAR	Rutina que atenderá al dispositivo que solicitó interrupción al procesador. Eliminará la causa que provocó la interrupción.	IVT	
Rutina de Configuración de Interrupción	Device_x_ICR	Rutina que configurará todos los parámetros necesarios para que el sistema responda adecuadamente ante solicitudes de interrupción del dispositivo implicado.	Indiferente	

La Excepción SWI

Las excepciones del tipo SWI son las únicas provocadas por eventos internos al procesador ARM. De forma genérica denominaremos *Interrupciones software* a este tipo de excepciones.

Generalmente en todos los sistemas que incluyen procesadores existe un esquema de tratamiento de este tipo de interrupciones, muy empleado para tareas relacionadas con la gestión del propio sistema (sistema operativo, etc).

Policom provee del hardware y software necesario para que el programador pueda emplear este recurso ampliamente utilizado en programación de bajo nivel.

Desde el punto de vista hardware, el único subsistema que interviene en el tratamiento de las interrupciones software es el propio procesador ARM7TDMI, que al igual que ocurre con el resto de excepciones vistas hasta ahora, altera su normal funcionamiento de la misma forma que lo hacía con los otros tipos de excepción.

En lo que se refiere al software, el sistema de interrupciones software funciona a partir de tres rutinas, de características análogas a las utilizadas en la gestión de las interrupciones hardware:

- El Manejador de Interrupciones Software SWI_EH
- La Rutina de Configuración de Interrupción Software (SWICR)
- La Rutina de Atención de Interrupción Software (SWIAR)

El esquema de tratamiento de SWI

La Excepción SWI se provoca cuando el procesador ejecuta una instrucción del tipo:

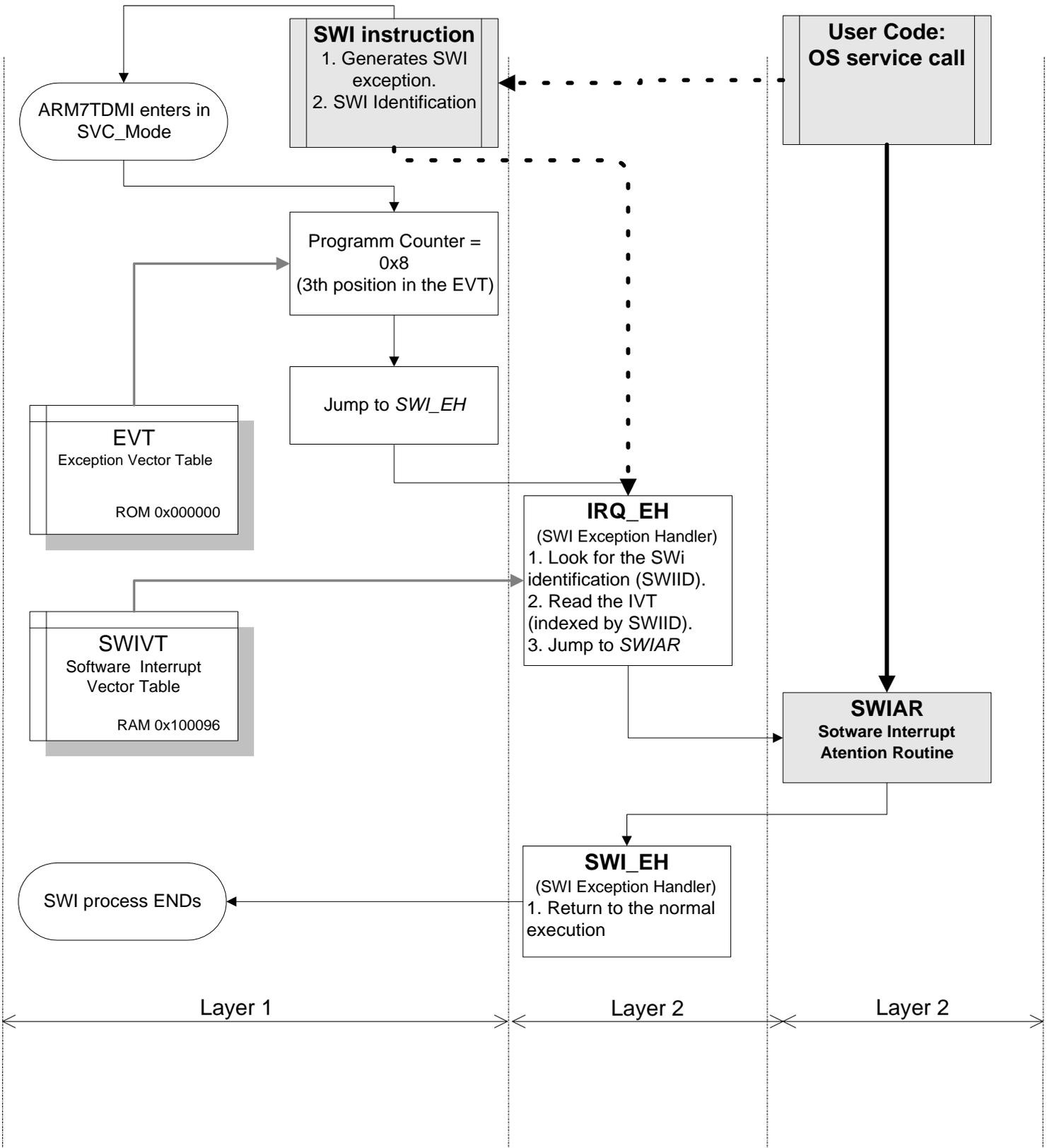
<i>SWI identificador (SWIID)</i>

La ejecución de dicha instrucción provoca los siguientes cambios en el procesador:

- Modo Supervisor.
- Excepciones del tipo IRQ inhabilitadas en el ARM.
- PC = 0x8 (posición 3ª de la EVT).

Como puede ver en la figura, una vez que el procesador detecta una operación de SWI, pasa a buscar en la Tabla de Vectores de Excepción la dirección del Manejador de este tipo de excepción. En la posición tercera (dirección 0x8) de la tabla se encuentra codificada la instrucción de salto a la SWI_EH, cuya ejecución proporcionará la entrada en la rutina SWI_EH. Durante la ejecución de dicha rutina, el procesador obtendrá el código de identificación de la interrupción software, denominado SWIID, a partir del cual consultará la Tabla de Vectores de Interrupción Software (SWIVT) donde podrá obtener la dirección de la Rutina de Atención de Interrupción Software (SWIAR) previamente programada por el usuario.

Una vez la SWIAR realiza las operaciones para las que ha sido programada, el control pasará al Manejador de SWI para abandonar finalmente la atención de la Interrupción Software.



El tratamiento de las Interrupciones Software en Policom es, como puede verse en la figura anterior, vectorizado a través de la Tabla de Vectores de Interrupción, SWIVT, que está indexada por el Identificador de Interrupción Software (SWIID). Esto permite al usuario la utilización de los servicios del software del sistema de manera transparente, esto es, sin necesidad de conocer en qué posición de memoria se encuentran dichos servicios. Es más, el usuario accederá a dichos servicios mediante el Identificador de Interrupción Software, el SWIID, sin necesidad de conocer nada más sobre el software que está invocando.

La tabla de vectores SWI (SWIVT)

Al igual que los vectores de interrupción IRQ, los vectores de interrupción software proporcionan la dirección dónde se encuentra la rutina de atención de interrupción software, denominada SWIAR.

La SWIVT se encuentra residente en RAM interna de POLICOM.

Será el manejador de interrupciones software, SWI_EH, el encargado de leer el vector correspondiente según el identificador de interrupción software, y saltar a la rutina correspondiente. EL programador deberá haber colocado en esta tabla el vector adecuado antes de hacer uso de la interrupción software correspondiente.

La estructura de la tabla sigue el esquema de la tabla de vectores de interrupción IRQ. De esta manera, la dirección base (la más baja de la tabla) corresponde con el identificador de interrupción software número '0'. Cada entrada enésima de la tabla consta de un word donde el usuario deberá cargar la dirección de la enésima rutina de atención de interrupción software.

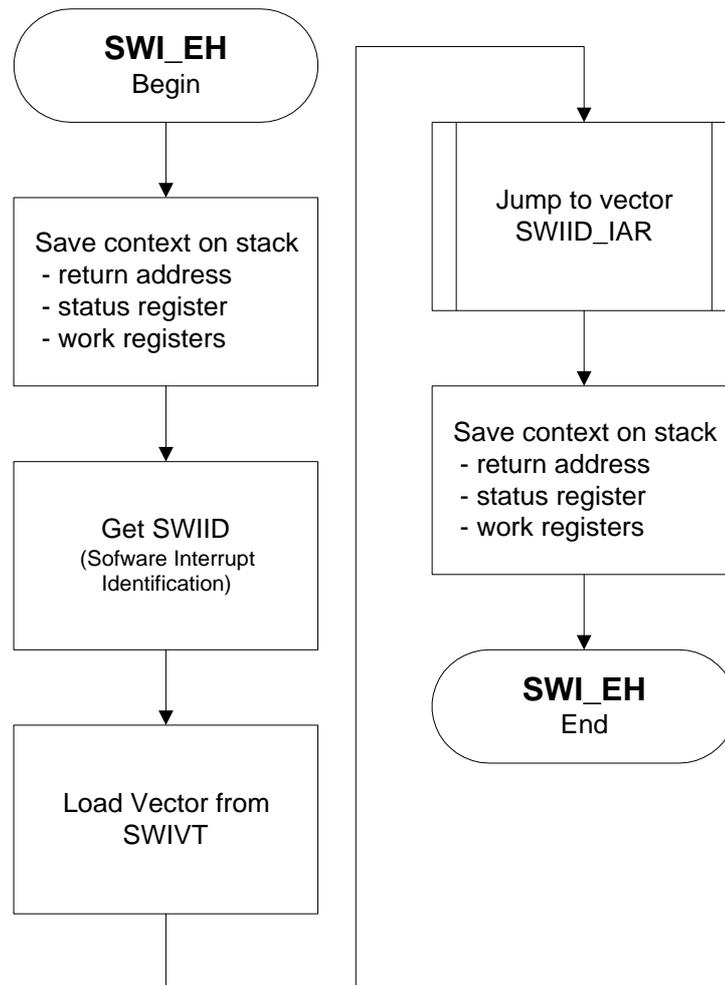
Manejador de interrupciones software

El manejador de interrupción software, SWI_EH, se ejecutará cuando el procesador se encuentre con la instrucción SWI, y sus funciones son:

- guardar el contexto

- obtener el SWI_ID
- leer el vector correspondiente a ese SWI_ID de la tabla de vectores de interrupción software (en RAM y previamente programada).
- saltar a la rutina apuntada por el vector correspondiente
- restaurar contexto

Su diagrama de flujo se muestra a continuación.



SWIAR y SWICR

Las rutinas de Atención y de Configuración de Interrupción Software, programadas por el usuario, siguen el esquema sus análogas en las interrupciones hardware.

Cuadro resumen

Objeto	Identificador	Descripción	Dirección	Acceso al código
Manejador de Excepción del tipo SWI	SWI_EH	Rutina que se ejecuta cada vez que el programador realiza una llamada a una interrupción software.	En la EVT	No accesible
Tabla de Vectores de Excepción	EVT	Tabla codificada en ROM. Vectoriza todas las excepciones del ARM.	ROM 0x000000	No accesible
Tabla de Vectores de Interrupción Software	SWIVT	Tabla codificada en RAM, indexada al SWIID y que proporciona las direcciones de las SWIAR.	RAM 0x100120	Abierto al programador. Cada dirección se codifica en 32 bits
Identificador de Interrupción Software	SWIID	Número que identifica a cada una de las interrupciones.	No procede	No procede
Rutina de Atención de Interrupción Software.	N_SWIAR	Rutina de atención a la SWI etiquetada con N	SWIVT	
Rutina de Configuración de Interrupción	N_SWICR	Rutina que configurará todos los parámetros necesarios para que el sistema responda adecuadamente ante solicitudes de interrupción software etiquetada como N.	Indiferente	

