

## **Anexo III: Consideraciones de tiempo de ejecución**

## **Introducción**

A continuación se exponen algunos resultados numéricos obtenidos a partir de las simulaciones que se han realizado con los modelos Verilog que forman el sistema POLICOM.

Estos resultados son especialmente interesantes si extrapolamos estas cantidades obtenidas con intención de cuantificar la ejecución de las aplicaciones que soportará POLICOM.

### **Hipótesis generales de la simulación**

- Reloj principal (ASB) → 24MHz (41.66 nsec por ciclo)
- Reloj secundario (APB) → 6 MHz (166.6 nsec por ciclo)
- Todas las especificaciones digitales han sido obtenidas a partir del documento POLICOM DIGITAL SPECIFICATION (24 July 1998)
- El código empleado es ARM (codificación de instrucciones a 32 bits).
- El mapeado de dispositivos así como la configuración de la memoria son los expuestos en el documento.
- Los accesos a memoria se refieren a memoria interna.

## **Algunos de los resultados son:**

### **Accesos a memoria en diferentes condiciones:**

- Tiempo en escribir un tabla en RAM empleando como tamaño de acceso 32 bits (word) → 291 nsec /word aprox. Esta operación dura más de lo que es un ciclo porque al recorrer la tabla estamos también haciendo uso de un contador interno que funciona como índice de la tacla. El incremento de dicho contador y su comparación con un valor final ocupa un tiempo considerable.
- Tiempo en escribir un tabla de 1Kbyte empleando accesos a 32 bits (words) → 74.36 microsec /Kbyte
- Operación de Lectura - comparación - escritura → 500 nsec / Kbyte
- Operación de Lectura - comparación - escritura → 128 microsec / word
- Tiempo en chequear RAM interna: 202.34 microsec. / Kbyte

### **Tiempo de atención de interrupciones:**

Generalmente el tratamiento de interrupciones estará muy presente en la gestión de las comunicaciones de POLICOM. Por esta razón es necesario realizar una serie de estimaciones de tiempo para poder garantizar el funcionamiento correcto de la aplicaciones que soportará POLICOM.

En este sentido, se ha hecho una simulación muy significativa desde el punto de vista de estimación de los tiempos.

#### **Hipótesis de la simulación:**

- UART: 125 Kbps en tx y rx. 8 bits / carácter. En modo loop-back
- Generación de interrupción por cada byte recibido o cuando el byte de transmisión queda libre.
- Hay que tener en cuenta que la UART trasmite un bit se stop.

- Desde el punto de vista software tenemos dos rutinas: una principal encargada de programar el dispositivo y el vector de interrupción, así como habilitar la interrupción de la fuente de interés; y la rutina de atención de interrupción. La rutina de atención de interrupción atenderá las dos causas que previamente se han habilitado en la UART, esto es, la recepción de un byte y transmisor vacío.
- No hay otros manejadores de interrupciones instalados en el sistema.
- Todas las hipótesis generales.

Los resultados obtenidos son:

- T° total entre dos IRQ consecutivas = 88 microsec.
- T° de atención total de la IRQ = 17 microsec.

Esta operación incluye:

Guardar contexto =	2 microsec
Algoritmo de búsqueda de fuente más prioritaria =	10 microsec
Atención de la causa de interrupción =	4 microsec
Restauración de contexto =	1 microsec.

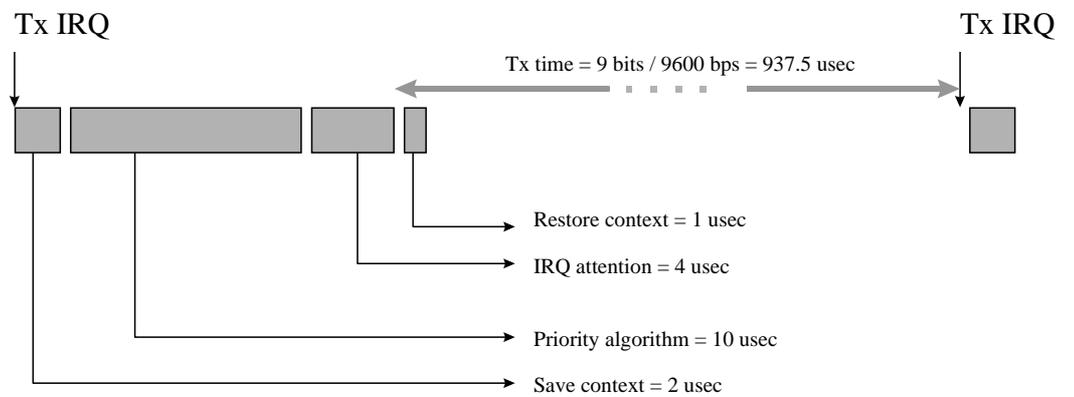
Con los valores anteriores podemos extraer las siguientes conclusiones:

Carga de procesador en atención de interrupción	19.3 %
Tiempo ocioso de procesador	80.7 %

Extrapolando los resultados anteriores a una velocidad de trabajo de 9600 Bps (estamos ahora en un tiempo entre irq de 937.5 microsec.) , podemos deducir que:

Carga de procesador en atención de interrupción	1.78 %
Tiempo ocioso de procesador	98.22 %

En la figura siguiente se aclara en qué se emplea el tiempo cuando el procesador atiende una interrupción según las hipótesis expuestas.



Resulta curioso que las operaciones de restauración y guardar contexto ocupan tiempos diferentes. La razón es que para guardar el contexto es necesario calcular la dirección de retorno de cuando sucedió la interrupción. Para llevar a cabo ese cálculo es necesario introducir una operación aritmética ( resta ) que aumenta el tiempo de ejecución.

## Conclusiones

A partir de los resultados expuestos, parece razonable afirmar que el uso del mecanismo de interrupciones para la gestión de los dispositivos de POLICOM no supone un riesgo de sobrecarga del procesador.

También sería interesante extrapolar estos resultados a situaciones concretas impuestas por el software de protocolos que el sistema va a soportar. Con los valores que se han mostrado anteriormente y las necesidades del software, podrán derivarse estimaciones del rendimiento que el sistema POLICOM ofrece.

Desde el punto de vista de una futura utilización de la arquitectura diseñada para sistemas de transmisión - recepción a velocidades de 1 Mbps, habría que tener en cuenta:

1. La gestión de la interrupciones actual puede resultar un cuello de botella si las velocidades de acceso al dispositivo aumentan. Esto se puede solucionar fácilmente incluyendo en los dispositivos que funcionen a alta velocidad de FIFO's que almacenen temporalmente los datos hasta que sean atendidos. En cualquier caso es recomendable que sea sustituido el actual controlador de interrupciones por uno que gestione de forma 'hardware' las interrupciones. AICIA ya ha diseñado dicho controlador para futuras aplicaciones.
2. Con los cambios que se recomiendan, estaríamos condiciones de asegurar que el futuro sistema puede funcionar con un rendimiento aceptable.
3. Se realizarán simulaciones que tengan por objetivo respaldar el punto anterior.