

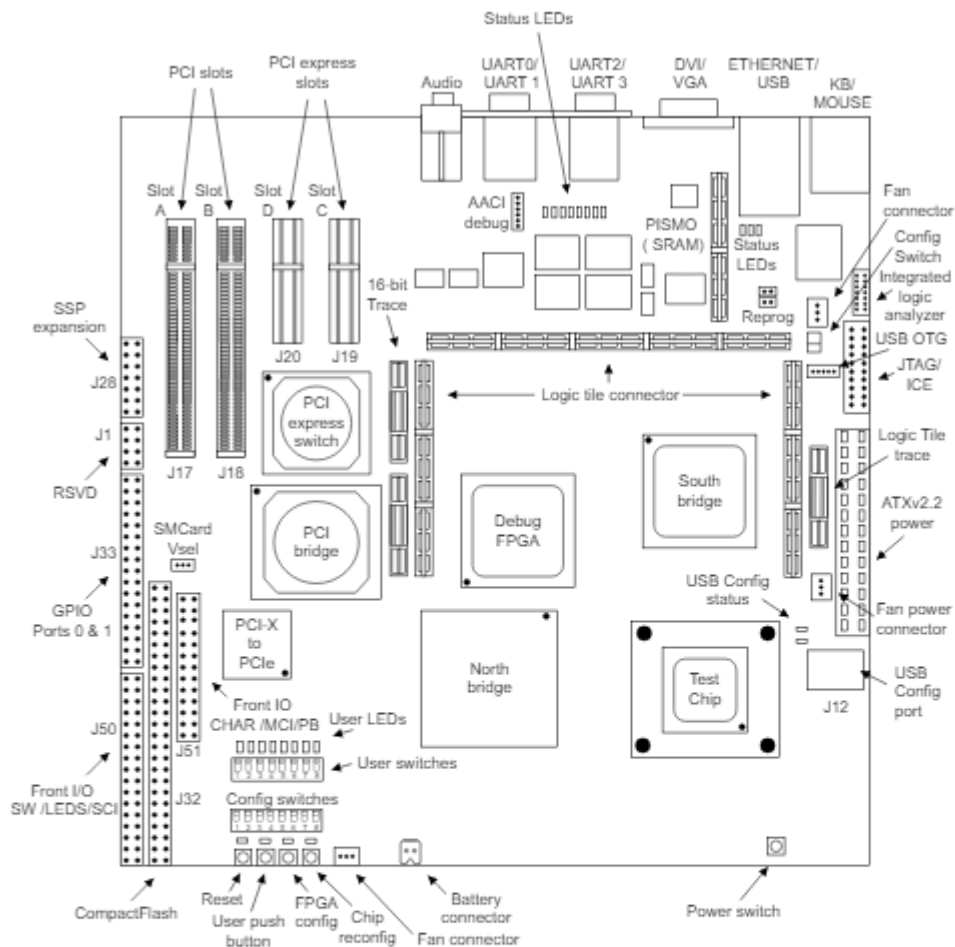
Placa de desarrollo PB-A8 RealView para Cortex A-8

Para poder salir antes al mercado, se desarrolla el hardware y el software en paralelo. En el segundo caso, se usan placas de desarrollo del fabricante del procesador. De esta manera, cuando sale la versión preliminar de la placa que compone el dispositivo que se desea vender, se puede probar con el software producido gracias a la placa de desarrollo.

Vamos a ver una placa de desarrollo producida por la empresa ARM para poder probar software escrito para el procesador Cortex A-8.

Se puede correr una emulación del PB-A8 RealView en la herramienta QEMU. Dicha emulación tiene en cuenta el hardware completo de este sistema.

A continuación, se muestra una imagen de la placa:



El chip de prueba (test chip) es un ASIC (circuito integrado diseñado para un uso particular) que incluye:

- Core Cortex-A8.
- PLL (Phase-Locked Loop) que permite generar frecuencias múltiplo de la frecuencia de entrada para alimentar el clock del core.
- Boundary scan tap, que sirve para observar el interior del circuito integrado (por ejemplo, sus registros) para propósitos de debugging.

Los conectores para logic tiles permiten montar placas encima de la placa principal que agregan funcionalidad.

El circuito integrado denominado Debug FPGA permite acceder al Cortex-A8 usando una plataforma de debugging en una computadora externa. Para eso provee una interfaz JTAG.

Los periféricos que normalmente se encuentran en un chip SoC (System on a Chip) basado en Cortex A-8 aquí se encuentran en dos circuitos integrados: Northbridge y Southbridge.

El Northbridge es un ASIC que incluye:

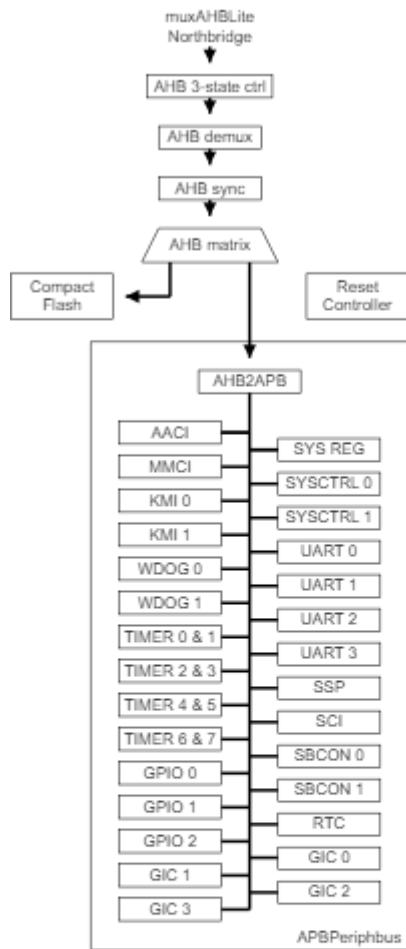
- Interfaz de alta velocidad con el chip de prueba denominado AXI, de 64 bits.
- Controlador de paneles o pantallas LCD.
- Controladores de memoria estática, dinámica y de acceso directo a memoria.
- Puente AXI a PCI para poder acceder a dispositivos que operan con el bus PCI.

El Southbridge es un FPGA cuya configuración se carga desde una memoria flash. Esta memoria flash es un chip distinto a la memoria NOR flash del usuario.

Provee interfaz para códecs de audio, interfaz para SD y eMMC, interfaz para Smart Card, interfaz para teclado y mouse, watchdog, timers, GPIO, controlador de interrupciones GIC, controlador del sistema, puertos serie, interfaz de dos líneas TWI y reloj de tiempo real (RTC).

Las direcciones de los periféricos que se encuentran en el Southbridge y de las memorias que se pueden conectar al Northbridge son fijas y se pueden consultar en el manual de la placa.

Diagrama en bloques del Southbridge:



Generic Interrupt Controllers

Los GIC aceptan interrupciones de periféricos en el Northbridge, el Southbridge, otros periféricos de la placa de desarrollo y el tile site. Los GIC generan interrupciones tipo FIQ e IRQ tanto al Cortex-A8 como al tile site.

El Southbridge tiene cuatro GIC:

GIC 1: genera interrupción nIRQ para el Cortex-A8.

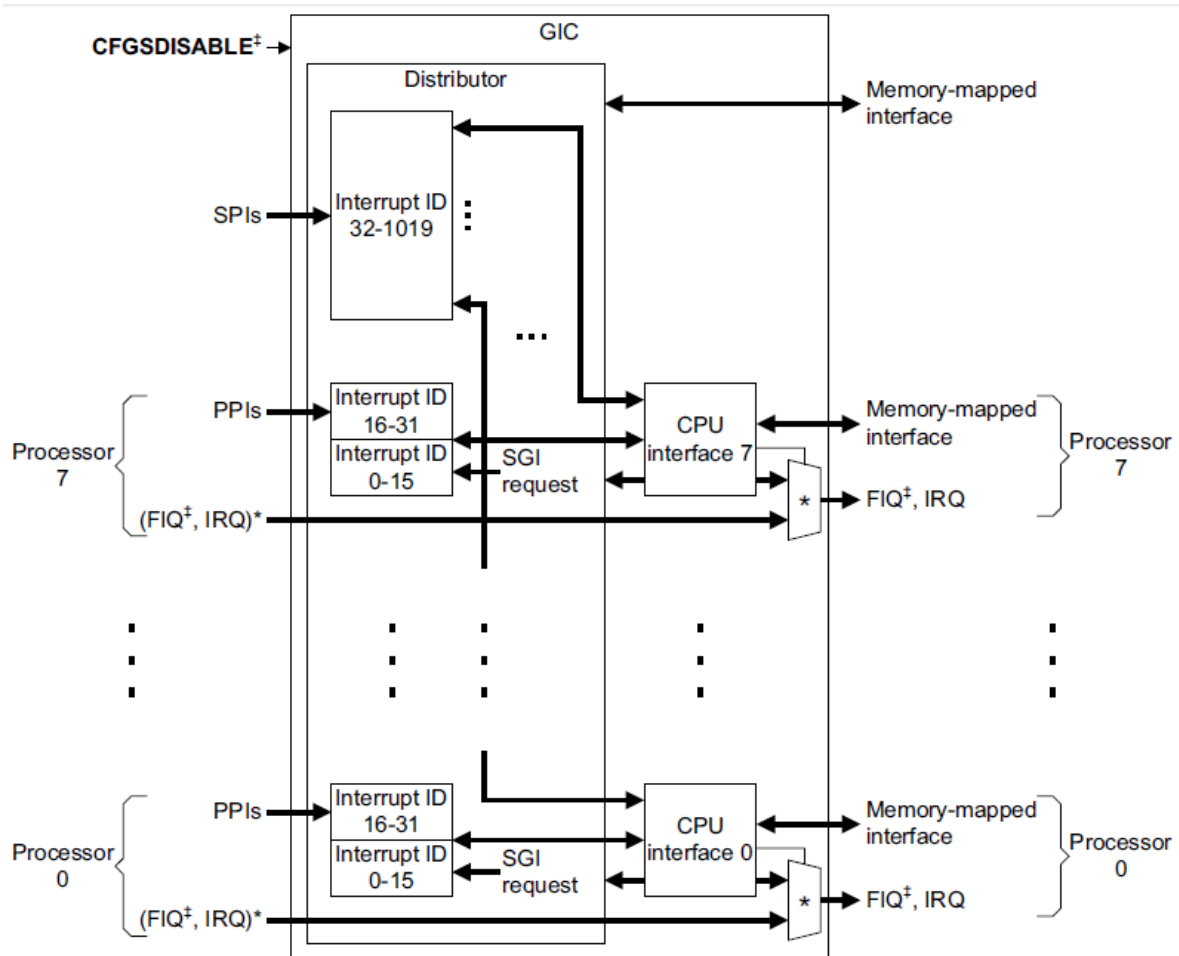
GIC 2: genera interrupción nFIQ para el Cortex-A8.

GIC 3: genera interrupción nIRQ para el tile site.

GIC 4: genera interrupción nFIQ para el tile site.

Nosotros vamos a usar el primer GIC.

Diagrama del GIC:



Si bien el diagrama presenta la arquitectura para múltiples núcleos (processors), en nuestro caso de uso (Cortex-A8) solo basta con reducir el esquema a un solo procesador.

Según la fuente, hay tres tipos principales de interrupciones definidas.

Interrupciones de periféricos compartidos (Shared Peripheral Interrupts):

Identificador de interrupción del 32 al 1019. Son las fuentes de interrupción generadas por los diferentes periféricos del sistema que pueden ser direccionadas (re/compartirse) a cualquiera (uno o más) de los núcleos según la configuración implementada. A modo de ejemplo se podría mencionar el caso de dos temporizadores independientes (TIMER0 y TIMER1) conectados a las líneas SPI y configurándose en el *Distributor* para direccionarse a dos de los núcleos disponibles. En la interrupción del TIMER0, la señal se direcciona al primer procesador disponible (cualquiera de los dos configurados). Si el TIMER1 requiere su atención, cuando aún no finalizó la correspondiente al TIMER0, el *Distributor* la direcciona al restante núcleo configurado para su manejo.

Los periféricos tienen número de interrupciones fijas dentro del GIC, que se numeran entre 32 y 95. Por ejemplo, los timers 0 y 1 comparten el interrupt ID 36, la UART0 usa el ID 44, Ethernet el ID 60, etc.

Interrupciones de periféricos privados (Private Peripheral Interrupts):

Identificador de interrupción del 16 al 31. Son las fuentes de interrupción generadas por los diferentes periféricos del sistema que debe ser direccionadas a un núcleo específico.

Un ejemplo típico de este tipo de interrupción se presenta en el caso los sistemas asimétricos como ser un Cortex A y Cortex M, en donde una interrupción generada por un Watchdog correspondiente al núcleo A solo debe direccionarse al mismo.

Interrupciones generadas por software (Software Generated Interrupts):

Identificador de interrupción del 0 al 15. ARM reserva este rango específicamente para la comunicación entre procesadores. Una SGI puede ser direccionada a uno o más procesadores a través del *Distributor*.

Las Interrupciones 0 a 31 son almacenadas por el *Distributor* para cada *CPU Interface*, es decir, cada procesador las ve de manera diferente y se identifican por el CPUID. Por ejemplo, la PPI 20 podría estar pendiente en CPU0 pero no en CPU1. Mientras que, en el caso de la SPI, será la misma en todos los CPU.

Independientemente de todo lo mencionado, cada núcleo recibe una señal a través de líneas nIRQ o nFIQ para interrumpir la ejecución del programa.

A continuación, describiremos sintéticamente el objetivo de cada submódulo

Distributor:

Es responsable de gestionar las interrupciones en todo el sistema, decide las prioridades entre ellas, el mecanismo de

direccionamiento de las mismas, así como también permite la habilitación de cada una en forma independiente. Solo existe una instancia de este componente y está orientado a presentar las interfaces de interconexión a los periféricos.

CPU Interface

Existe una instancia por núcleo de CPU disponible e interconecta la interfaz provista por el Distributor con el núcleo correspondiente. Cada interfase permite:

- Des/habilitar la señal de IRQ al procesador
- Dar acuse de recibo de la interrupción
- Indicar que el procesamiento de la interrupción se ha finalizado
- Establecer la máscara de prioridades de interrupción
- Definir la [preemption policy](#) del procesador
- Determinar la interrupción de más alta prioridad

Timers

El Southbridge tiene 4 módulos SP804 que tienen dos timers cada uno.

- Timers 0-1: base 0x10011000, interrupt ID #36
- Timers 2-3: base 0x10012000, interrupt ID #37
- Timers 4-5: base 0x10018000, interrupt ID #73
- Timers 6-7: base 0x10019000, interrupt ID #74

Descripción de los registros para cada par de timers (sumar 0x20 a la dirección para acceder a los registros Timer2xxx):

- Base+0x00: Timer1Load: el contador se decrementa y cuando llega a cero, se recarga con este registro. Al escribir este registro, el contador se carga inmediatamente con el nuevo valor.
- Base+0x04: Timer1Value: valor actual del contador.
- Base+0x08: Timer1Control:
 - Bit 7: Habilidad del timer.
 - Bit 6: 0: Modo corrida libre, 1: Modo periódico.
 - Bit 5: Habilidad de interrupción para el timer.
 - Bit 4: Reservado (no modificar su valor).

- Bits 3-2: Prescaler: 00: sin dividir, 01: clock dividido por 16, 10: clock dividido por 256, 11: No usar.
- Bit 1: Contador de 16 bits, 1: Contador de 32 bits.
- Bit 0: Mod recarga, 1: Detención cuando llega a cero.
- Base+0x0C: Timer1IntClr: Al escribir este registro, el módulo reconoce la interrupción enviada.
- Base+0x10: Timer1RIS (Raw Interrupt Status Register): El bit cero indica el estado de la interrupción.
- Base+0x14: Timer1MIS (Masked Interrupt Status Register): Bit 0: AND lógico entre RIS y el bit de habilitación de interrupción. La salida de interrupción del módulo (TIMINTX) sigue este bit.
- Base+0x18: Timer1BGLoad: similar al registro Timer1Load, pero al escribirlo el contador no se recarga con este valor.

El módulo tiene registros adicionales para test que no se explicitan aquí.