**Interconexión con Buses**

De acuerdo con Das, Palesi, Kim, Pratim Pande (2024), el concepto de la interconexión se ha vuelto más relevante que nunca. Según ellos, el reto más importante de la arquitectura de computadoras modernas ya no es la capacidad de cálculo, sino la comunicación entre los componentes, lo que hace de la interconexión un punto clave. Las interconexiones a escala de chip y de paquete son los elementos más relevantes que determinan el rendimiento, la eficiencia energética y la escalabilidad de un sistema. En el artículo se muestra cómo este papel central de la interconexión se convierte en un patrón común en todo tipo de arquitecturas como en las de propósito general (CPU, GPU), el dominio específico (las que se utilizan en IA) y las arquitecturas de sistemas de computación cuántica. Para abordar el creciente aumento de la necesidad de comunicación, el artículo presenta las tecnologías avanzadas como los cables metálicos (la base de los enlaces cableados), la nanofotónica de alta densidad de ancho de banda y baja latencia, y los sistemas inalámbricos como las principales soluciones para los retos futuros [1].

1. **Estructuras de buses**

Según el artículo denominado “Security of Electrical, Optical and Wireless On-Chip Interconnects: A Survey” (2023) antes de que surgieran las Network-on-Chip (NoC), la topología principal para establecer la comunicación dentro de los System-on-Chip (SoC) era la arquitectura por buses. En este sentido, el artículo establece que la arquitectura de buses es sencilla, de bajo costo y opera con un conjunto de roles muy bien definidos. Aquellos tienen que ver con un maestro el cual inicia las operaciones, un esclavo que da respuestas a su solicitud, un árbitro que se encarga de controlar el uso del bus y un decodificador que envía las señales al dispositivo correspondiente. Según este estudio, aunque esta arquitectura por buses resulta ser muy útil para SoCs pequeños, a medida que los sistemas se extienden se convierten en un retraso en su comunicación. También se presentan ejemplos de estas arquitecturas, como el IBM's CoreConnect y la AMBA de ARM, así como sus vulnerabilidades frente a los ataques de canal lateral de energía [2].

1. **Elementos de diseño**

Según afirma el artículo "Multi Buses - Theory and Practical Considerations"(2020), el diseño de buses de alto rendimiento en FPGAs se basa en dos grandes conceptos, los cuales son el Multi Frame Bus (MFB) y el Multi Value Bus (MVB). El MFB está diseñado y orientado a la transferencia de varias transacciones de datos por ciclo de reloj. Este se explica a partir de cuatro atributos principales: el número de regiones que fija el número máximo de tramas por palabra, el tamaño de la región que establece el tiempo adicional necesario para tramas cortas, el tamaño del bloque que regula el tiempo de alineación y, por último, el ancho del elemento, que describe el tamaño de la unidad de datos más pequeña. El MVB queda caracterizado por el número de ítems, el cual fijará la máxima cantidad de valores transferidos, y por el ancho del ítem, que describirá el tamaño de cada valor. De este modo, en ambos casos encontramos el objetivo principal del diseño de buses que trata de llegar a una estructura de palabra de datos que permita un posicionamiento de tramas razonablemente eficiente sin comprometer la complejidad de la lógica de procesamiento y minimiza el tiempo de alineación del bus [3].

1. **Tipos de buses**
2. **Bus de datos**

Tal y como señalan Freund, Pirker y Dür (2024), el bus de datos cuántico es considerado como una solución flexible para las redes cuánticas. Este bus de datos cuántico lleva a cabo su operación mediante un estado de acumulación bidimensional entrelazado que actúa como su recurso clave. Con la aplicación de un esquema de medida local en un camino diagonal, es capaz de establecer varias conexiones paralelas entre distintos dispositivos de la red. Cabe señalar que un rasgo fundamental de este método consiste en que las mediciones no destruyen la estructura de entrelazamiento del estado de acumulación. Esto permite que el estado que queda pueda ser utilizado de nuevo para conexiones futuras. El artículo demuestra que esta técnica puede dar lugar a muy complejas líneas de medidas que se cruzan, giran y se fusionan, lo cual es aplicable tanto a redes de área local, como a redes de larga distancia [4].

1. **Bus de direcciones**

En el artículo “Design And Verification Of Apb Bridge”, se afirma que en el protocolo APB, el bus de direcciones PADDR se usa para acceder a ciertas ubicaciones y hacer lecturas y escrituras. El protocolo APB permite que las direcciones en el bus no coincidan con la alineación con respecto al ancho del bus de datos. No obstante, un esclavo APB que recibe una solicitud con direcciones no alineadas podría no ser capaz de manejarlas, lo que conduciría a operaciones indeseadas e impredecibles. Para eliminar esta limitación, el artículo propone un puente al que denomina UTAA (Unaligned To Aligned Access), que es capaz de convertir una solicitud de dirección no alineada desde el maestro en dos solicitudes de direcciones alineadas para permitir un acceso al bus de datos. De este modo, el puente puede facilitar la reutilización de los esclavos APB que no los admiten dichas solicitudes no alineadas [5].

1. **Bus de control**

Según el artículo "A Novel Plane-Based Control Bus Design with Distributed Registers in 3D NAND Flash Memories", se ha propuesto un diseño de bus de control basado en planos que ahorra área. Está basado en el uso de registros distribuidos para memorias flash de tipo 3D NAND. Dicho diseño permite la reducción de los cables de enrutamiento de las señales de control hasta en un 99,47% respecto al diseño convencional. El bus de control conecta una lógica de control central con la agrupación de registros distribuidos que hay dentro de cada plano de memoria. Cada plano utiliza un bus de control con 27 señales de enrutamiento distintas para poder comunicarse. Las señales de control del bus son: un reloj de bus, señales de habilitación de lectura y de escritura, una dirección de grupo de registros de 7 bits, y buses de datos de lectura/escritura de 8 bits cada uno. El bus de control permite las operaciones: escritura/lectura de registros y direccionamiento de registros. Las operaciones de direccionamiento pueden ser continuas o bien podrían ser aleatorias. El bus de control funciona a una frecuencia de reloj de 50 MHz. En el artículo, también se incluye un esquema de plane gating basado en la dirección de grupo de registros con aproximación de 2,9 mW de potencia [6].

1. **Buses internos y externos**

Según el artículo “Analisis Sistem Bus USB Dan PCI Pada Organisasi Arsitektur” (2023), el bus USB es un estándar de interfaz, el cual se usa para conectar diferentes dispositivos externos a una computadora. Por otro lado, el bus PCI es un bus interno que conecta los componentes de la computadora [7].

|  |  |  |
| --- | --- | --- |
| **Característica** | **Bus USB (Universal Serial Bus)** | **Bus PCI (Peripheral Component Interconnect)** |
| **Tipos de Bus** | Externo | Interno |
| **Función Principal** | Conectar diversos dispositivos externos a la computadora, como teclados, ratones, impresoras y cámaras. | Conectar componentes internos de la computadora, como tarjetas de video, tarjetas de sonido y tarjetas de red, a la placa base. |
| **Evolución** | USB 1.1 (12 Mbps), USB 2.0 (480 Mbps), USB 3.0 (5 Gbps), USB 3.1 Gen 2 (10 Gbps), USB 3.2 (20 Gbps), USB 4.0 (40 Gbps). | PCI (33 MHz, 133 MB/s), PCI-X (133 MHz, 533 MB/s), PCI Express (PCIe) con velocidades que van desde 250 MB/s por carril hasta 1 GB/s por carril y más en revisiones posteriores. |
| **Tipo de Conexión** | Comunicación en serie. | PCI y PCI-X utilizan un bus compartido, mientras que PCIe utiliza un enfoque de comunicación punto a punto. |
| **Versiones de Puerto** | USB Type-A (el más común), USB Type-C (más nuevo, reversible y multifuncional). | Las ranuras PCI solo aceptan tarjetas PCI. PCIe tiene diferentes tamaños de ranura (x1, x4, x8, x16) |
| **Compatibilidad** | Son compatibles con versiones más nuevas de USB, pero la velocidad de transferencia se limita a la versión más baja. | Creado por Intel y se convirtió en el estándar para sistemas Pentium |

1. **Rol en la transferencia de datos entre componentes**
2. **Interacción entre CPU, memoria y periféricos**

Según el documento "Stop Taking the Scenic Route: the Shortest Distance Between the CPU and the NIC is MMIO" (2025), la relación entre la CPU, la memoria y los dispositivos periféricos como la NIC ha pasado a convertirse en un tema de creciente interés a causa de la oferta de enlaces de alta velocidad y la presión para interfaces de NIC eficientes para la CPU. Históricamente, la mejor manera de transportar datos de la CPU hacia la NIC ha sido por medio del acceso directo a memoria (DMA), donde el único trabajo de la CPU es iniciar la transferencia. La NIC recupera los datos de la memoria del host en forma asíncrona. Desafortunadamente, esta forma de acceso a memoria por DMA produce un acceso de memoria para cada palabra de datos, el cual es no contiguo y produce retrasos en las transferencias de datos. El artículo mencionado propone que MMIO, un tipo de E/S programada más general, supera a la implementación E/S por DMA en latencia y ancho de banda al copiar los datos directamente en los registros de la NIC. Los autores desafían la sabiduría tradicional que menosprecia el MMIO como el mecanismo más importante para mover datos y afirman que se puede conseguir un alto rendimiento de escritura con MMIO si se sueltan las restricciones del orden. Proponen un hardware eficiente para regenerar el orden en el NIC, lo que permite disfrutar las ganancias de rendimiento proveniente de las escrituras de MMIO sin orden. El estudio muestra cómo un único núcleo de CPU puede mover más de 100 Gbps de tráfico a un dispositivo por medio de MMIO con escrituras combinadas, sobrepasando la tasa de línea de la NIC, mientras que el punto de partida de las instrucciones de barrera de memoria [8].

1. **Gestión del flujo de datos en sistemas multiprocesador**

Tal como se menciona en el artículo "NoC-based hardware software co-design framework for dataflow thread management"(2023), la manera en que se gestionan los flujos de datos en los sistemas multiprocesador se puede entender en el marco del modelo de trabajo de un diseño de hardware y software, pero plenamente basado en la propuesta del modelo de ejecución del flujo de datos. Con esta idea se plantea implementar un sistema de gestión de hilos sobre la NoC (Red en Chip), buscando proporcionar una mejor eficiencia energética y utilización de recursos en los sistemas de muchos núcleos. El marco de trabajo que se propone opera en tres fases o pasos fundamentales: una política de distribución de hilos eficaz y rápida, la inclusión de adaptabilidad de la NoC y la exploración de una topología híbrida que combina topología de mallas 2D y anillos. Todo ello incide en que la comunicación entre los hilos en el flujo de datos es mejor y permitiendo que un hilo se ejecute solo cuando todos sus datos de entrada están disponibles, lo que es fundamental en el paradigma del flujo de datos [9].

1. **Diseño de Buses:**
2. **Realizar diagramas que representen la estructura y función de un bus.**

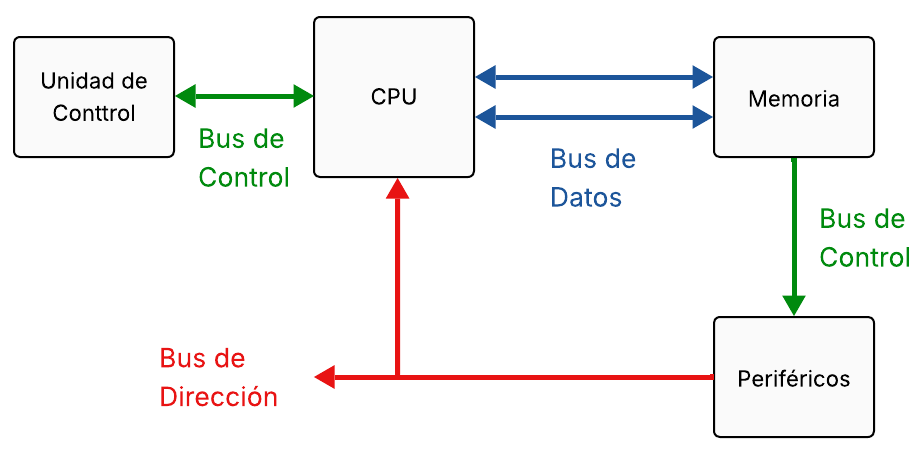
****

Figura 1. Estructura y función de un bus

**Bus de Datos:** Transfiere información entre CPU, memoria y periféricos. Es bidireccional.

**Bus de Dirección:** Indica la dirección de memoria o del periférico al que acceder. Es unidireccional desde la CPU.

**Bus de Control:** Coordina las operaciones mediante señales como lectura, escritura, interrupción y reloj.

La función de un bus es permitir la comunicación ordenada de la CPU y la memoria.

1. **Identificar y describir los diferentes tipos de buses y su impacto en la velocidad de transferencia.**

|  |  |  |
| --- | --- | --- |
| **Tipo de bus** | **Descripción** | **Impacto en velocidad** |
| **Interno** | Une partes dentro de la CPU. | Muy rápido. |
| **Externo** | Une CPU con memoria y periféricos. | Más lento que el interno. |
| **Datos** | Lleva la información. | Más líneas = más bits a la vez. |
| **Direcciones** | Indica dónde leer o escribir. | Más líneas = más lugares posibles. |
| **Control** | Envía órdenes y señales. | Afecta coordinación y eficiencia. |
| **Paralelo** | Varios bits al mismo tiempo. | Rápido en corto alcance, problemas en largas distancias. |
| **Serie** | Bits uno por uno. | Más lento, pero estable en largas distancias. |

**BIBLIOGRAFÍA**

[1] A. Das, M. Palesi, J. Kim, and P. Pratim Pande, “Chip and Package-Scale Interconnects for General-Purpose, Domain-Specific, and Quantum Computing Systems—Overview, Challenges, and Opportunities,” *IEEE J Emerg Sel Top Circuits Syst*, vol. 14, no. 3, pp. 354–370, Sep. 2024, doi: 10.1109/JETCAS.2024.3445829.

[2] H. Weerasena and P. Mishra, “Security of Electrical, Optical and Wireless On-Chip Interconnects: A Survey,” Sep. 2023.

[3] L. Kekely, J. Cabal, V. Pus, and J. Korenek, “Multi Buses: Theory and Practical Considerations of Data Bus Width Scaling in FPGAs,” in *2020 23rd Euromicro Conference on Digital System Design (DSD)*, IEEE, Aug. 2020, pp. 49–56. doi: 10.1109/DSD51259.2020.00020.

[4] J. Freund, A. Pirker, and W. Dür, “Flexible quantum data bus for quantum networks,” *Phys Rev Res*, vol. 6, no. 3, p. 033267, Sep. 2024, doi: 10.1103/PhysRevResearch.6.033267.

[5] Krishnakanth Katteri Mahadeva Murthy, “DESIGN AND VERIFICATION OF APB BRIDGE,” *International Research Journal of Modernization in Engineering Technology and Science*, Sep. 2022, doi: 10.56726/IRJMETS30019.

[6] H. CAO, Q. WANG, F. LIU, and Z. HUO, “A Novel Plane‐Based Control Bus Design with Distributed Registers in 3D NAND Flash Memories,” *Chinese Journal of Electronics*, vol. 31, no. 4, pp. 647–651, Jul. 2022, doi: 10.1049/cje.2021.00.283.

[7] Putri Andini Maulana, Muhammad Yusuf Habibi, Agus Gilang Hermawan, and Didik Aribowo, “Analisis Sistem Bus USB Dan PCI Pada Organisasi Arsitektur Komputer,” *Jurnal Kendali Teknik dan Sains*, vol. 1, no. 4, pp. 115–122, Oct. 2023, doi: 10.59581/jkts-widyakarya.v1i4.1463.

[8] W. S. Liew, M. A. Rahaman, J. McMahon, R. Stutsman, and V. Nagarajan, “Stop Taking the Scenic Route: the Shortest Distance Between the CPU and the NIC is MMIO,” in *Proceedings of the Workshop on Hot Topics in Operating Systems*, New York, NY, USA: ACM, May 2025, pp. 144–150. doi: 10.1145/3713082.3730389.

[9] S. Mazumdar, A. Scionti, S. Zuckerman, and A. Portero, “NoC-based hardware software co-design framework for dataflow thread management,” *J Supercomput*, vol. 79, no. 16, pp. 17983–18020, Nov. 2023, doi: 10.1007/s11227-023-05335-8.