

Proposta de Interface AMBA AXI-Stream para a Integração de Aceleradores em uma Rede-em-Chip

Larissa de Sousa Gouvea
LEDS, Universidade do Vale do Itajaí, Brasil
larissagouvea@edu.univali.br

Wesley Grignani
IES, Université de Montpellier, France
wesley.grignani@umontpellier.fr

Thiago Haas Rausch
LEDS, Universidade do Vale do Itajaí, Brasil
thiagorausch@edu.univali.br

Douglas Rossi de Melo
LEDS, Universidade do Vale do Itajaí, Brasil
drm@univali.br

Abstract

The increasing complexity of modern System-on-Chip (SoC) designs has strengthened the need for scalable and efficient communication mechanisms. Network-on-Chip (NoC) architectures, such as XINA, address this demand by providing structured and reliable interconnections between processing modules. However, the current XINA Network Interface (NI) supports only a subset of the AMBA AXI protocol and lacks compatibility with AXI4-Stream, which is essential for accelerators and modules that require continuous data transfer. This work proposes extending the existing interface to include full AXI4-Stream support, enabling the integration of streaming-based modules into the NoC. The design will be derived from the current architecture and evaluated through simulations to verify feasibility and communication behavior.

Keywords

Computer Architecture, Networks-on-Chip, Network Interfaces, AXI-Stream

1 Introdução

A utilização crescente de arquiteturas System-on-Chip (SoC) em aplicações embarcadas tem intensificado a demanda por mecanismos de comunicação mais eficientes e capazes de acompanhar o aumento de complexidade desses sistemas. Nesse contexto, as Redes-em-Chip (Networks-on-Chip, NoCs) surgem como alternativa aos barramentos tradicionais, oferecendo maior escalabilidade, maior previsibilidade do tráfego e suporte a comunicações paralelas entre diversos núcleos de processamento [1].

Entre as arquiteturas de NoC descritas na literatura, a XINA (eXtensible Interconnect Network Architecture) apresenta mecanismos de parametrização dos controladores e de proteção dos elementos de memória, possibilitando sua operação em ambientes sujeitos a radiação [2]. A arquitetura é composta por roteadores interconectados e módulos responsáveis por funções como roteamento, controle de fluxo e arbitragem, além de uma interface de rede (Network Interface, NI) que conecta módulos compatíveis com o padrão AMBA AXI, especificado pela ARM, à NoC, viabilizando a integração de diferentes IPs (Intellectual Property) ao sistema [3]. A NI atualmente oferece suporte a um subconjunto do protocolo AXI-Full, mas não contempla o padrão AXI4-Stream, utilizado em aplicações que demandam transmissão contínua de dados, o que limita a integração de aceleradores e de outros módulos especializados que dependem desse padrão.

A relevância do suporte ao AXI4-Stream também pode ser observada em aplicações que demandam processamento contínuo e alta taxa de dados, como aceleradores de hardware empregados em visão computacional, processamento de sinais, compressão de dados e comunicações. Em trabalhos recentes, esse tipo de abordagem tem sido explorado, por exemplo, em aceleradores voltados à compressão de imagens hiperespectrais para missões espaciais [4].

Este trabalho propõe o aprimoramento da interface de rede existente, estendendo suas funcionalidades para oferecer suporte ao AXI4-Stream, visando atender às demandas de sistemas embarcados contemporâneos e ampliar as possibilidades de uso da XINA.

2 Fundamentação Teórica

O padrão AXI4-Stream é destinado à transmissão contínua de dados entre dois componentes, adotando um modelo de transferência direta sem endereçamento e sem operações de leitura. O protocolo assegura que o fluxo seja entregue de forma ordenada ao receptor e permite a movimentação de bytes, palavras, pacotes ou quadros completos, possibilitando que camadas superiores construam estruturas mais complexas a partir desses dados. A coordenação da comunicação ocorre por meio de um mecanismo bidirecional de *handshake*, no qual transmissor e receptor indicam sua prontidão, garantindo que cada transferência seja realizada apenas quando ambos podem enviar e receber adequadamente [5].

Segundo [6], uma Rede-em-Chip (NoC) é uma infraestrutura de interconexão utilizada em SoCs modernos para ligar componentes por meio de elementos de comutação e de links organizados em uma rede interna. Em vez de múltiplos barramentos dedicados, a NoC transporta dados de forma multiplexada, melhorando a utilização dos canais físicos e permitindo isolamento entre fluxos. Em sistemas complexos, partes do chip podem manter interconexões locais tradicionais, exigindo adaptadores para integrar protocolos como AXI à NoC, o que contribui para maior escalabilidade, flexibilidade e redução da latência no projeto de SoCs.

Nesse contexto de NoCs, a XINA se destaca como uma arquitetura de interconexão voltada a garantir comunicação eficiente e confiável entre os núcleos do sistema. Seu roteador conta com múltiplas portas e módulos especializados que empregam *buffers* FIFO e a técnica de comutação *wormhole*, além de utilizar o algoritmo XY para o roteamento e o esquema *round-robin* para a arbitragem. Com esses mecanismos e estruturas internas projetadas para aumentar a robustez, a XINA mantém desempenho e continuidade no tráfego de dados mesmo na presença de falhas internas [2].

Uma interface de rede (NI), conforme descrito por [1], é o módulo que liga um núcleo de processamento à rede de interconexão, separando as funções de computação das tarefas de comunicação e realizando a conversão de protocolo entre o IP core e o roteador. Ela organiza o fluxo de dados por meio de blocos que adaptam os sinais do núcleo, empacotam as informações para transmissão e, depois, desmontam os pacotes recebidos, permitindo que diferentes núcleos se comuniquem com a rede de forma padronizada, eficiente e independente da arquitetura interna de cada core.

3 Trabalhos Relacionados

Foram analisados estudos sobre interfaces de rede, com o objetivo de compreender suas funções, aplicações e os impactos observados em sistemas integrados. Alguns trabalhos investigam projetos de interfaces de rede para melhorar a escalabilidade, a modularidade e a eficiência da comunicação. Em especial, o trabalho de [7] apresenta o desenvolvimento de uma interface de rede compatível com a arquitetura XINA, baseada no protocolo AXI-Full, na qual a interface é organizada em *front-end* e *back-end*. Nesse trabalho, o *front-end* adapta os sinais AXI aos sinais genéricos da rede, enquanto o *back-end* é responsável pelo empacotamento de dados, pelo controle de fluxo e pela aplicação de técnicas de tolerância a falhas, como TMR e códigos de detecção e correção de erros.

Outros estudos analisam a organização interna das interfaces de rede e os mecanismos empregados para adaptar diferentes protocolos de comunicação ao modelo orientado a *flits* das NoCs. Em [8], a interface é estruturada em camadas responsáveis pela adaptação de sinais, pela montagem e desmontagem de pacotes e pela interação com a rede, o que evidencia a importância da separação de responsabilidades no projeto de interfaces escaláveis. Embora essas abordagens explorem a integração entre núcleos e Redes-em-Chip, elas não abordam explicitamente o suporte a interfaces baseadas em fluxo contínuo de dados, como o AXI4-Stream.

No contexto específico do protocolo AXI4-Stream, trabalhos como [9, 10] demonstram seu uso em sistemas de interconexão e na comunicação entre módulos de hardware que exigem altas taxas de transferência de dados. Essas propostas evidenciam a eficiência do AXI4-Stream em aplicações de transferência contínua, como a comunicação entre blocos de processamento e entre a lógica programável e os sistemas de processamento. No entanto, tais estudos não consideram a integração direta do AXI4-Stream à arquitetura XINA, o que motiva este trabalho.

4 Protocolos

A Figura 1 ilustra o protocolo de *handshaking* do AXI4-Stream entre um *AXI4-Stream Data Producer* e um *AXI4-Stream Data Consumer*, conforme descrito na documentação da AMD [11].

Primeiro, o gerente (*AXI4-Stream Data Producer*) coloca os sinais TDATA e TLAST (e, quando usado, TUSER) no barramento e sinaliza TVALID para indicar que o dado é válido. Em seguida, o subordinado (*AXI4-Stream Data Consumer*) ativa o sinal TREADY para indicar que está pronto para receber. Quando TVALID e TREADY estão ambos em nível alto, o *flit* é transferido para o subordinado, e o gerente pode prosseguir com a transmissão dos próximos valores de TDATA, TLAST (e opcionalmente TUSER) em ciclos subsequentes.

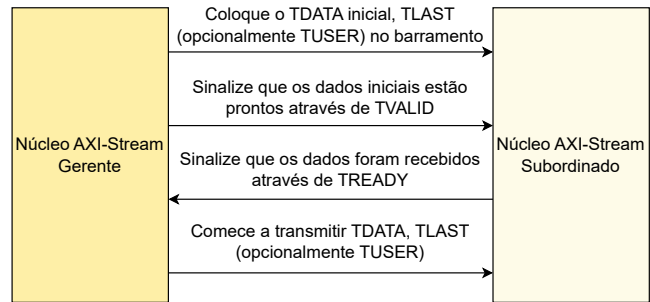


Figura 1: Comunicação AXI4-Stream gerente-subordinado

4.1 Sinais do protocolo AXI4-Stream

Conforme a especificação da ARM [5], o protocolo AXI4-Stream define os sinais básicos usados para o transporte unidirecional de dados com fluxo de controle por *handshake*. A interface AXI4-Stream é descrita com base no conjunto de sinais que compõe a configuração mínima adotada pelos IPs AXI4-Stream da própria AMD [11]. Os principais sinais dessa interface mínima são:

- TVALID – Sinal emitido pelo gerente. Indica que os dados em TDATA são válidos e prontos para transferência.
- TREADY – Sinal emitido pelo subordinado. Indica que o destino está pronto para receber a transferência de dados.
- TDATA – Barramento de dados enviado pelo gerente; carrega o payload da transação AXI4-Stream.
- TLAST – Sinal opcional gerado pelo gerente para marcar a última transferência elementar de dados realizada em um ciclo de clock válido, de um frame ou pacote.
- TKEEP – Sinal opcional gerado pelo gerente; máscara de bytes válidos em cada beat de dados.
- TSTRB – Sinal opcional gerado pelo gerente; strobes de bytes que indicam quais bytes de TDATA são significativos.
- TID – Sinal opcional emitido pelo gerente. Identifica o fluxo de dados.
- TDEST – Sinal opcional gerado pelo gerente. Fornece informações de roteamento.

4.2 Sinais da Rede-em-Chip

Na Rede-em-Chip considerada neste trabalho, baseada no roteador de Melo et al. [2], a porta local conecta o roteador à interface de rede por meio de dois canais unidirecionais (injeção e recepção). Cada canal é composto por um barramento de dados e por dois sinais de controle que implementam o protocolo de *handshake* em quatro fases (VAL/ACK) usado pelo controlador de regulação de fluxo.

- DATA – Barramento de dados que transporta os *flits* entre a interface de rede e o roteador. No canal de injeção, é encaminhado pela interface de rede para o roteador; no canal de recepção, do roteador para a interface de rede.
- VAL – Sinal de validação associado a DATA. É gerado pelo transmissor do canal para indicar que o *flit* presente em DATA é válido e pronto para ser aceito.
- ACK – Sinal de reconhecimento do *handshake*. É gerado pelo receptor do canal (roteador na injeção, interface de rede na recepção) para confirmar que o *flit* foi recebido com sucesso, autorizando o transmissor a avançar para o próximo *flit*.

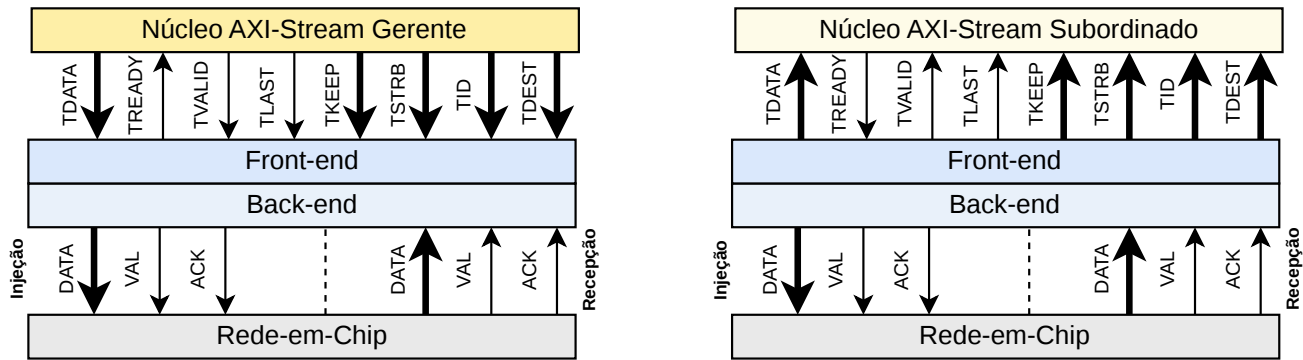


Figura 2: Estrutura da interface de rede gerente e subordinada e seus sinais

5 Projeto

A arquitetura proposta foi concebida para preservar a modularidade da XINA, mantendo a separação entre a adaptação de protocolo e o transporte na rede, ao mesmo tempo em que viabiliza a integração de módulos baseados em AXI4-Stream.

A Figura 2 apresenta as duas variantes de interface de rede utilizadas neste trabalho: subordinada e gerente. Ambas seguem a mesma organização geral, composta por um *front-end* vinculado ao núcleo AXI4-Stream e um *back-end* responsável pela interação com a porta local do roteador da NoC.

A interface gerente mantém a mesma estrutura funcional, porém atua como a entidade responsável por iniciar operações de tráfego e por coordenar o fluxo de mensagens no sistema. Seu *front-end* interage com o núcleo AXI4-Stream gerente, enquanto o *back-end* utiliza os mesmos canais da NoC para a injeção e recepção de *flits*, preservando a compatibilidade com o roteador e com a interface subordinada.

Na interface subordinada, o *front-end* recebe o fluxo de dados proveniente do núcleo AXI4-Stream e realiza a adaptação necessária ao modelo de comunicação baseado em *flits* da NoC. O *back-end* executa a injeção e recepção por meio dos canais dedicados, garantindo que a transferência ocorra de forma sincronizada com o roteador.

Em conjunto, essas duas interfaces permitem integrar blocos AXI4-Stream à NoC de forma modular, mantendo a separação clara entre a adaptação de protocolo no *front-end* e o transporte orientado a *flits* no *back-end*.

6 Considerações Finais

Este trabalho estabelece as bases conceituais para o desenvolvimento de uma interface AXI4-Stream integrada à arquitetura XINA. Ao analisar as limitações da interface atualmente disponível, restrita ao suporte AXI-Full, identificou-se a necessidade de incorporar um mecanismo capaz de atender a aplicações que dependem de transmissão contínua de dados, especialmente em aceleradores e módulos especializados presentes em sistemas embarcados.

O estudo antecipatório da XINA, do funcionamento do protocolo AXI4-Stream e dos requisitos de comunicação da NoC permitiu delinear uma arquitetura inicial composta por *front-end* e *back-end*, bem como definir as responsabilidades de cada bloco e o fluxo geral de operação.

As próximas etapas deverão envolver o detalhamento da arquitetura proposta, o refinamento dos mecanismos de adaptação e a preparação do ambiente para análises futuras. Em um momento posterior, espera-se realizar simulações para observar o comportamento da interface ao interagir com módulos aceleradores que dependem do fluxo contínuo de dados, além de avaliar métricas de desempenho.

Agradecimentos

Este trabalho foi financiado, em parte, pela Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina – FAPESC, Chamada 51/2024 (Contratos 2024TR001897 e 2025TR001565), pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – CAPES/PROSUC, e pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq.

Referências

- [1] Santanu Kundu and Santanu Chattopadhyay. *Network-on-chip: the next generation of system-on-chip integration*. Taylor & Francis, 2018.
- [2] Douglas R Melo, Cesar A Zeferino, Luigi Dilillo, and Eduardo A Bezerra. Maximizing the inner resilience of a network-on-chip through router controllers design. *Sensors*, 19(24):5416, 2019.
- [3] Thiago H Rausch, Wesley Grignani, Gustavo HS Müller, Douglas A Santos, Luigi Dilillo, and Douglas R Melo. Hardening an amba-axi network interface for a reliable network-on-chip. In *2025 IEEE 16th Latin America Symposium on Circuits and Systems (LASCAS)*, volume 1, pages 1–5. IEEE, 2025.
- [4] Wesley Grignani, Douglas A Santos, Maria Kastriotou, Carlo Cazzaniga, Luigi Dilillo, and Douglas R Melo. Implementation and characterization of a fault-tolerant CCSDS 123 hardware accelerator under neutron radiation. *Microprocessors and Microsystems*, page 105184, 2025.
- [5] ARM. *AMBA AXI-Stream Protocol Specification*, 2025. Disponível em: <https://developer.arm.com/documentation/ih0051/latest>.
- [6] David J Greaves. *Modern system-on-chip design on ARM*. Arm Education media, 2021.
- [7] Gustavo Henrique Stahl Müller, Thiago Haas Rausch, and Douglas Rossi Melo. Interface de rede AMBA-AXI para uma rede-em-chip confiável. *Anais do Computer on the Beach*, 15:172–178, 2024.
- [8] Douglas Melo, Michelle Wingham, and Cesar Zeferino. Xiru: Interface de rede extensível para integração de núcleos a uma rede-em-chip. *Revista de Informática Teórica e Aplicada*, 21(2):10–31, 2014.
- [9] Prajwal Angadi, Sanket Venkatesh Shenvi, and Sathvik Lokesh. Design and simulation of AXI4 stream interconnect using verilog. In *2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)*, pages 1–4. IEEE, 2024.
- [10] Ankit Sharma. Evaluation of axi-interfaces for hardware software communication. 2018.
- [11] Vitis High-Level Synthesis User Guide: AXI4-Stream Interfaces. Advanced Micro Devices, Inc., 2025. UG1399 – AXI4-Stream Interfaces (online). Available at: <https://docs.amd.com/r/en-US/ug1399-vitis-hls/AXI4-Stream-Interfaces>.