

Uma Arquitetura de Alta Velocidade para Sistemas Digitais Heterogêneos

Duarte L. Oliveira¹, Diego Bompean¹, Lester A. Faria¹ e João Luis V. Oliveira²

¹Divisão de Engenharia Eletrônica do Instituto Tecnológico de Aeronáutica – ITA – IEEA
Praça Marechal Eduardo Gomes, 50, CEP 12228-900, SJC, SP, Brasil

²Instituto de Pesquisas Eldorado – Campinas, SP, Brasil

Resumo — Projeto digital VLSI (*Very Large Scale Integration*) de alta integração pode ter requisitos críticos, tais como consumo de potência, robustez, desempenho, etc. Eles normalmente são sintetizados no paradigma síncrono, na tecnologia DSM (*Deep-Sub-Micron*), usam o conceito de projeto SOC (*Systems-On-Chip*) e sinal de *clock* global. A exigência destes requisitos e a plataforma de síntese trazem sérias dificuldades no projeto por causa do sinal de *clock* global. Um estilo de projeto SOC que elimina estas dificuldades é permitir interação de módulos funcionais com temporizações diferentes, formando um SOC heterogêneo. Este artigo propõe uma arquitetura para implementar sistemas digitais heterogêneos, onde o sinal de *clock* global é eliminado. A arquitetura proposta permite a comunicação de dados ponto-a-ponto entre os módulos funcionais, seja do tipo síncrono ou/e assíncrono. A comunicação entre os módulos é realizada no protocolo *handshake* de duas fases, portanto acarretando um aumento de velocidade. Uma análise da arquitetura proposta para SOC heterogêneo é realizada mostrando o seu desempenho.

Palavras-chave— Lógica assíncrona, GALS, interface assíncrona, controlador port, FPGA

I. INTRODUÇÃO

Sistemas digitais contemporâneos devem ser necessariamente baseados no conceito de “*System-on-chip – SOC*”. O motivo é satisfazer uma demanda implacável por alto desempenho, reusabilidade e requisitos de baixa potência [1,2]. Circuitos SOC são compostos por módulos funcionais, que podem ser de propriedade intelectual (*intellectual property – IP*) das mais diferentes empresas. Os módulos *IP* são pré-projetados, verificados, testados e possuem alto desempenho. Eles permitem reduzir custo e principalmente o tempo do projeto. Circuitos SOC quando implementados com o sinal de *clock* global afetam desempenho e potência (*clock skew* e rede de distribuição), como dificultam a análise de temporização [3]. Circuitos SOC também podem ser implementados em FPGAs (*Field Programmable Gate Array*), apesar de que o problema de *clock skew* se agrava. O motivo é que o atraso entre as macro-células, devido ao meio de comunicação é significativo.

Duarte L. Oliveira, duarte@ita.br, Tel +55-12-3947-6813, Fax +55-12-3947-6930, Diego Bompean, diegobompean@yahoo.com.br, Lester A. Faria, lester@ita.br, João Luis V. Oliveira, jaovillar@gmail.com - ITA - IEEA - Praça Marechal Eduardo Gomes, 50, Vila das Acácias - São José dos Campos - SP - Brasil.

As FPGAs têm se tornado um meio popular para implementar circuitos digitais, isto é devido ao seu baixo custo e tempo curto de projeto. Tecnologia FPGA tem crescido consideravelmente nos últimos anos, gerando FPGAs que suportam milhões de portas, permitindo assim que sistemas digitais complexos possam ser programados em tais dispositivos [4,5].

O avanço da microeletrônica segue a lei de Moore, que oferece aos projetistas de sistemas digitais VLSI (*Very Large Scale Integrated*) a possibilidade de construir cada vez mais circuitos digitais mais complexos (tecnologia 35nm, 10^9 transistores — *Deep-sub-micron – DSM*), mas ao mesmo tempo cria novos problemas, quando se constroem em tais estruturas. A tecnologia DSM é sensível à interferência eletromagnética, alto consumo de potência, e o atraso nas linhas pode ser maior que o atraso nas portas [6,7]. Devido a estes problemas, o grande desafio é a distribuição do relógio global através de todo o *chip* com defasagem mínima (*clock skew*) sem implicar em um grande esforço de projeto e área ocupada [8]. A diferença de fase no sinal de *clock* não pode ultrapassar uma porcentagem do tempo de ciclo (relacionado com a tecnologia usada), se não acarreta mau funcionamento [9]. Ao aumentar a complexidade dos sistemas digitais e da frequência de relógio, metodologia totalmente sincronizada procura solucionar este grande desafio, que é o problema da fase defasada (*clock skew*) e a distribuição do *clock* global, com um numeroso número de *buffers* e um cuidadoso projeto de distribuição (árvore) do *clock*, que introduz uma considerável área e um aumento de potência. Em uma CPU de alto desempenho quase 40% do consumo da potência total do circuito é consumida pelo *clock* [10].

Uma solução que reduz significativamente os problemas relacionados com o *clock* global é a metodologia síncrona localmente e assíncrona globalmente (*Globally Asynchronous Locally Synchronous – GALS*), proposta por Chapiro em seu PhD [11]. Um sistema GALS consiste de módulos funcionais síncronos, que cada módulo tem o seu próprio relógio com frequência não relacionada, e que se comunicam uns com os outros na forma assíncrona. Este paradigma é uma combinação dos paradigmas síncrono e assíncrono, mas parte de que todos os módulos do sistema são síncronos. Para manusear a comunicação assíncrona entre os módulos, um circuito de interface é adicionado em torno de cada módulo síncrono, gerando um invólucro assíncrono (*asynchronous wrapper*) (ver Fig. 1) [12]. Esta interface, por exemplo, pode

ser composta de relógio local, FIFO, controlador assíncrono de comunicação (*Inputs Ports, Output Ports*). Techan et al. [13] mostra os diferentes estilos de interface assíncrona voltada para sistemas GALS. Interfaces assíncronas que usam *ports* de comunicação são interessantes porque elas permitem remover o esquema *handshake* assíncrono do módulo síncrono, assim o módulo síncrono pode ser projetado com técnicas usuais (*standard*) do projeto síncrono. Este estilo, viabilizada fortemente o uso de módulos *IPs* síncronos. Os sistemas GALS solucionam os problemas relacionados com o *clock* global, reduzem drasticamente o problema de análise de temporização e incentivam fortemente a reusabilidade, através dos *IPs*. Uma crítica que se faz aos sistemas GALS é o custo de comunicação de dados entre os módulos e que se agrava porque na maioria das interfaces assíncronas propostas, a comunicação entre os módulos usa protocolo *handshake* de quatro fases (ver Fig. 2), com isso há um aumento do *overhead* na comunicação [14], porque os sinais devem passar pela fase zero.

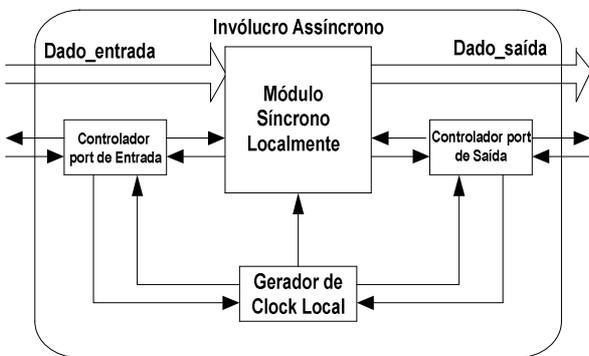


Fig. 1. Invólucro assíncrono.

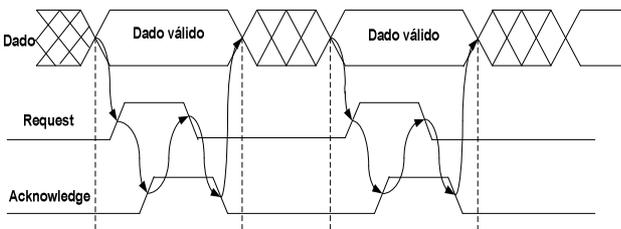


Fig. 2. Protocolo *handshake* de quatro fases.

Outra solução para o problema do *clock* global é uma variante de GALS que permite módulos assíncronos interagirem com módulos síncronos, gerando o que se chama de sistemas heterogêneos [15,16]. Para muitas aplicações podemos ter também módulos assíncronos, principalmente quando se deseja alto desempenho, onde há requisitos críticos tais como tempo de processamento de alguma tarefa, muito comum em sistemas de tempo real. Módulos funcionais assíncronos permitem uma redução no custo de comunicação e há um potencial na redução da potência dissipada como também um aumento de desempenho.

Este artigo propõe uma nova arquitetura para sistemas heterogêneos que realiza a comunicação de dados ponto-a-ponto. Esta proposta apresenta um novo invólucro assíncrono

para os módulos síncronos. Esta arquitetura também suporta módulos assíncronos e todos os módulos se comunicam no protocolo *handshake* de duas fases (ver Fig. 3). Os módulos assíncronos também são encapsulados para permitir a comunicação dos dados no protocolo de duas fases. Também apresentamos uma análise de desempenho desta nova arquitetura e comparamos com uma arquitetura similar da literatura, mostrando um aumento de desempenho na comunicação dos dados.

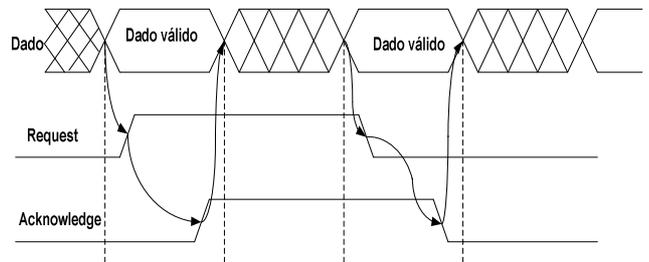


Fig. 3. Protocolo *handshake* de duas fases.

Este artigo está estruturado como segue: seção II discute as diferentes topologias para os sistemas heterogêneos; seção III apresenta o novo invólucro assíncrono de duas fases; seção IV apresenta o esquema de comunicação do módulo assíncrono; seção V discute a proposta da nova arquitetura; seção VI as nossas conclusões e trabalho futuro.

II. COMUNICAÇÃO EM SISTEMAS HETEROGÊNEOS

A comunicação de dados entre módulos sejam síncronos ou assíncronos em um sistema heterogêneo pode ocorrer no estilo ponto-a-ponto ou no estilo múltiplos pontos. Fig. 4 mostra uma topologia de comunicação de dados ponto-a-ponto de um sistema heterogêneo, onde somente há uma entrada e uma saída para cada módulo. Fig. 5 mostra uma topologia que contém uma comunicação dados bidirecional, no caso os módulos 1 e 2. No módulo 2 há duas saídas, portanto para cada saída devemos ter um protocolo *handshake* correspondente. No módulo 1 há duas entradas, que podem ser solicitadas concorrentemente, portanto precisamos de um arbitrador [17]. Fig. 6 mostra uma topologia geral de N módulos para um sistema heterogêneo com comunicação de dados de múltiplos pontos.

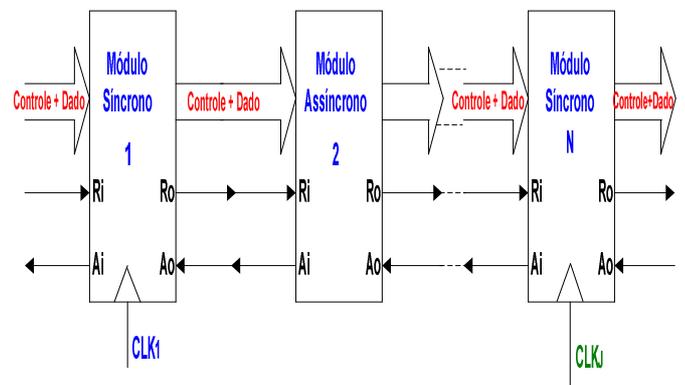


Fig. 4. Sistemas heterogêneos ponto-a-ponto.

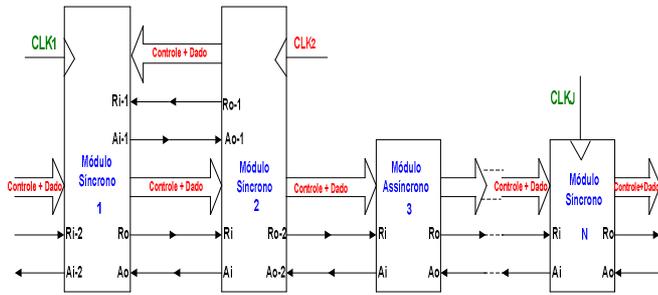


Fig. 5. Sistemas heterogêneos com comunicação bidirecional.

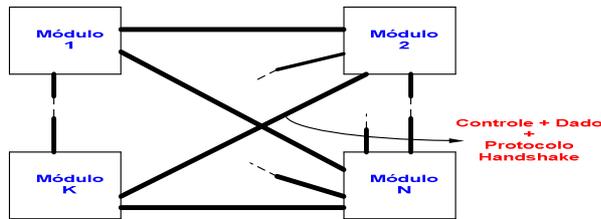


Fig. 6. Sistemas heterogêneos com comunicação múltiplos pontos.

III. INVÓLCURO ASSÍNCRONO DE DUAS FASES

O invólucro assíncrono proposto está voltado para comunicação ponto-a-ponto. Ele é baseado no invólucro assíncrono de [17,18], que realiza a comunicação de dados no protocolo *handshake* de quatro fases. Diferente do invólucro de [17,18], o invólucro proposto realiza a comunicação de dados inteiramente no protocolo de duas fases. Fig. 7 mostra a arquitetura do nosso invólucro assíncrono. Fig. 8 e 9 mostram o controle de entrada e a descrição do controlador *port* de entrada. Os *ports* propostos foram descritos na especificação modo *burst* [19] e sintetizados na ferramenta Minimalist [20]. Fig. 10 e 11 mostram o controle de saída e o controlador *port* de saída descrito no modo *burst*. Fig. 12 mostra o nosso gerador *gated-clock* com duas interrupções proposto em [21]. O gerador usa uma simples sincronização, mas é robusto a metaestabilidade [9].

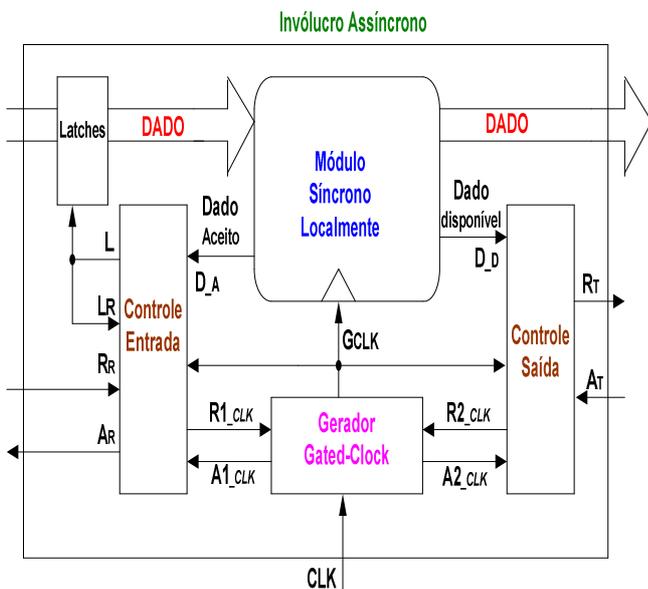


Fig. 7. Invólucro assíncrono de 2-Fases: comunicação ponto-a-ponto.

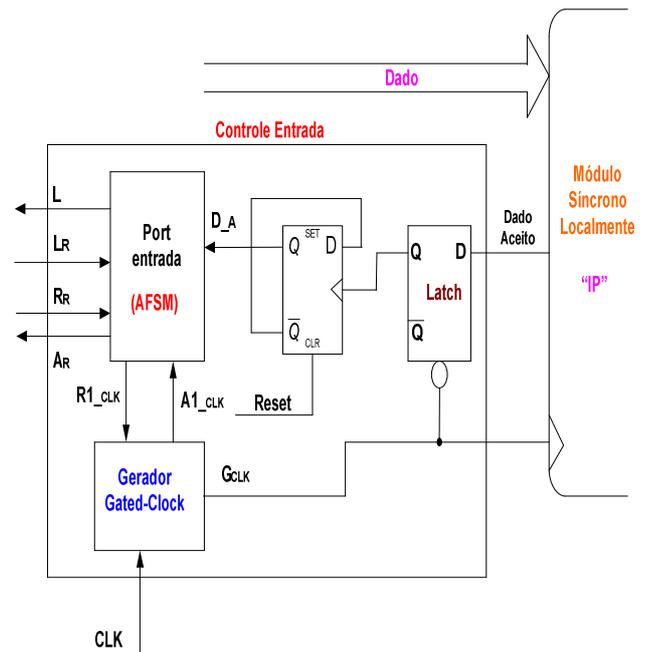


Fig. 8. Esquema geral: controle de entrada.

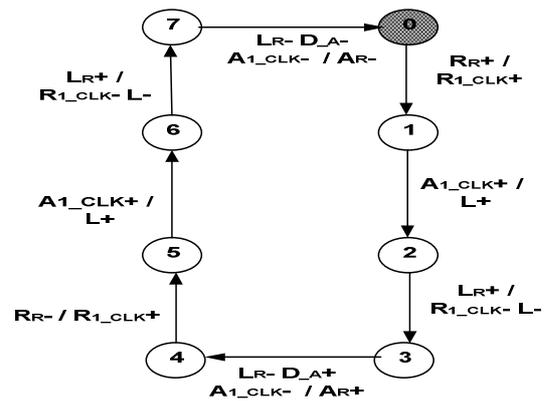


Fig. 9. Especificação modo *burst*: controlador *port* de entrada.

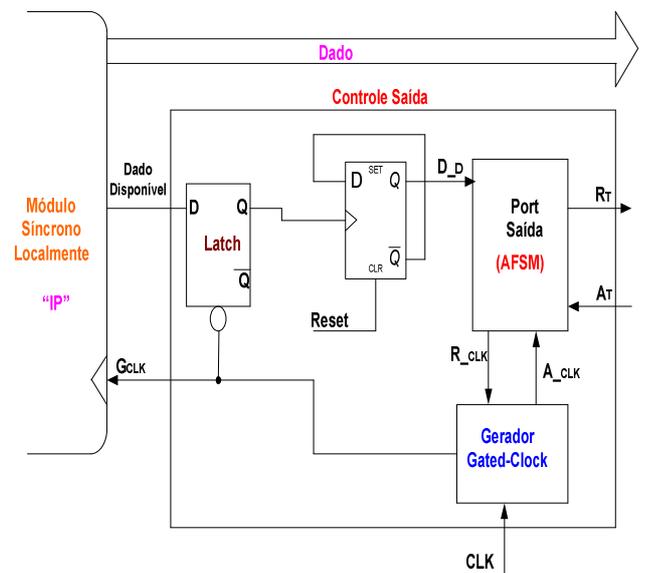


Fig. 10. Esquema geral: controle de saída.

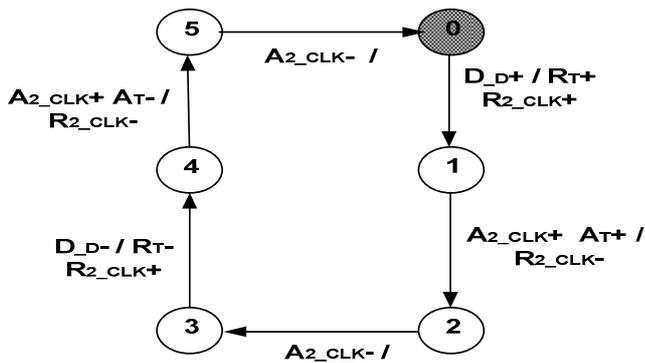


Fig. 11. Especificação modo *burst*: controlador *port* de saída.

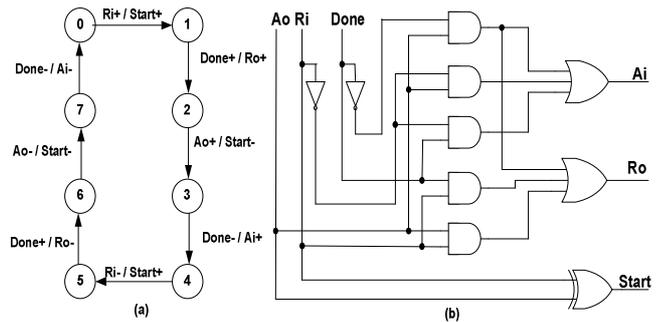


Fig. 14. Conversor de protocolo: a) especificação modo *burst*; b) circuito lógico.

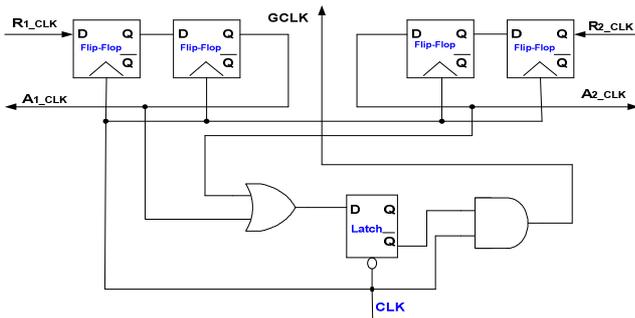


Fig. 12. Gerador *gated-clock* com duas interrupções.

IV. COMUNICAÇÃO DE DUAS FASES DOS MÓDULOS ASSÍNCRONOS

O projeto de um módulo assíncrono pode ser realizado em diferentes classes de circuitos [22]. A classe é definida pelo modelo de atraso em que o circuito opera corretamente, como o modo de interação com o ambiente [22]. O projeto mais natural de um módulo assíncrono qualquer que seja a sua classe é o uso do protocolo *handshake* de quatro fases, portanto a comunicação com o ambiente ocorre neste protocolo. Como a comunicação de quatro fases tem um alto custo, os nossos módulos assíncronos são encapsulados para permitirem comunicação dos dados no protocolo de duas fases. Fig. 13 mostra o encapsulamento usado para os módulos assíncronos, proposto em [23]. Fig. 14a mostra a especificação do conversor de protocolo, que permite a comunicação de dados seja realizada no protocolo de duas fases. Fig. 14b mostra o circuito lógico do conversor sintetizado na ferramenta Minimalist [20].

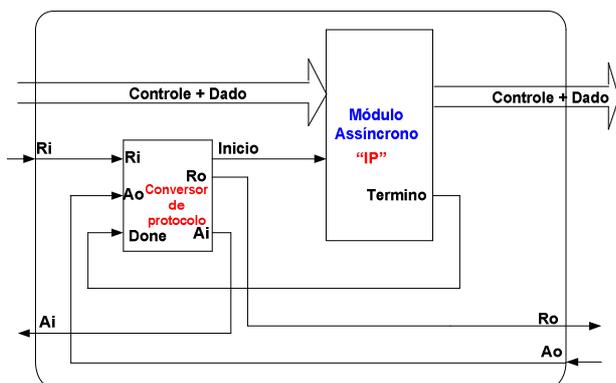


Fig. 13. Módulo assíncrono de duas fases ponto-a-ponto.

V. PROJETO E RESULTADOS

As grandes vantagens dos sistemas heterogêneos são: a) a possibilidade de reusar *IPs* existentes; b) os *IPs* síncronos operam na frequência original; c) uso de *IPs* assíncronos permitem reduzir o overhead de comunicação [24]; d) redução drástica no esforço da análise de temporização; e) redução da interferência eletromagnética; f) potencial na redução do consumo de energia; g) eliminação do problema de *clock skew* e a distribuição do *clock*. O invólucro assíncrono proposto foi descrito em VHDL estrutural do ALTERA Quartus II versão 9.0, da família Cyclone III no dispositivo EP3C16F484C6 [25]. Os controladores *ports* foram descritos no VHDL do invólucro através das suas equações de próximo estado e de saída que foram obtidas pela ferramenta de síntese Minimalist.

Projeto & simulação dos controladores *ports*

Fig. 15 e 16 mostram respectivamente os circuitos lógicos dos *ports* de entrada e de saída, que foram implementados na arquitetura de Huffman com saída realimentada. Fig. 17 e 18 mostram respectivamente as suas simulações, que são livres de hazard e satisfaz as especificações para um conjunto de estímulos. Os *ports* entrada/saída propostos necessitaram respectivamente de seis e duas LUTs. Quando comparado com os *ports* de [18] teve uma redução respectivamente de 75% e 50% no número de LUTs. O *port* de entrada de [18] necessitou de inserir um elemento de atraso para solucionar hazard essencial.

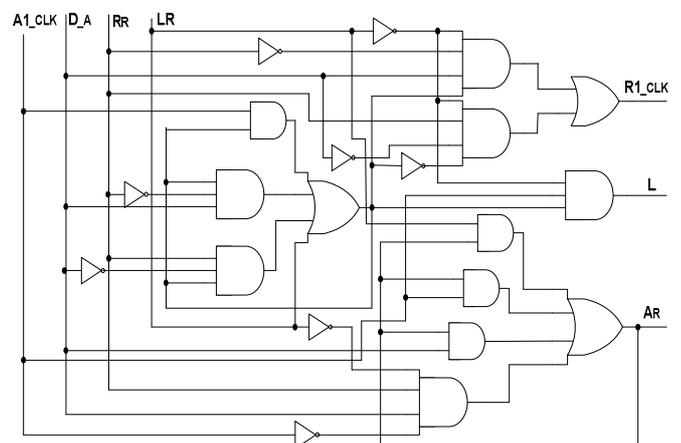


Fig. 15. Circuito lógico: controlador *port* de entrada.

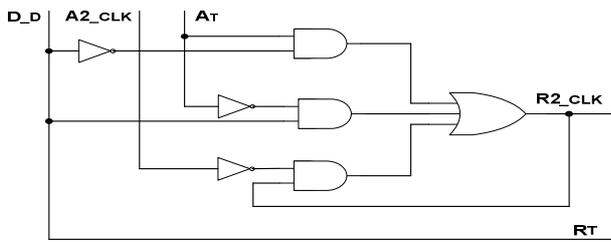


Fig. 16. Circuito lógico: controlador port de saída.

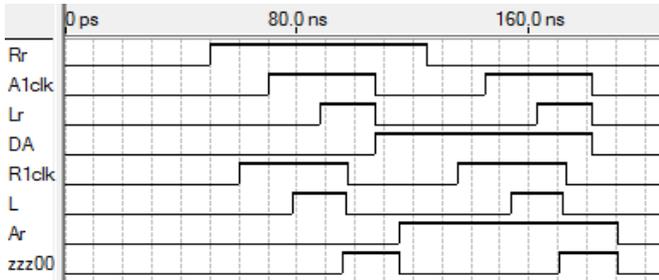


Fig. 17. Simulação: controlador port de entrada.



Fig. 18. Simulação: controlador port de saída.

Simulação do invólucro assíncrono de 2-Fases

Fig. 19 mostra a simulação do invólucro assíncrono proposto. Ela foi feita parcialmente, demonstrando os momentos onde o sinal de *clock* é “bloqueado”. Na etapa de escrita, foram percorridos os estados de 0 a 4 do *port de entrada* e na etapa de leitura, foram percorridos os estados 0 a 3 do *port de saída*. A simulação em Fig. 19 mostra o invólucro operando corretamente, seguindo a especificação estabelecida.

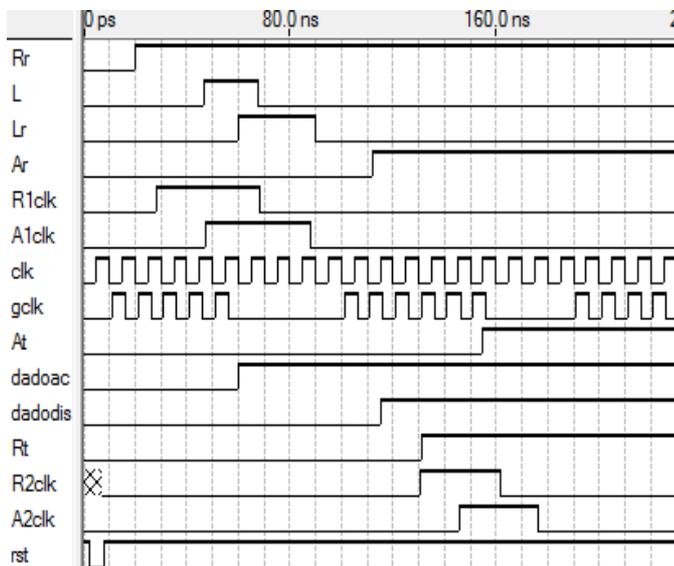


Fig. 19. Simulação: interface assíncrona.

Desempenho na comunicação dos dados

O invólucro assíncrono proposto se comunica no protocolo de 2-fases. A comunicação de dados entre dois módulos síncronos (ver Fig. 20) tem como tempo mínimo 86ns. Fazendo uma comparação com os *ports* de [18], a nossa comunicação de dados teve uma redução de 35%. Fig. 21 mostra uma comunicação entre os módulos síncrono e assíncrono, tendo como tempo mínimo 54ns.

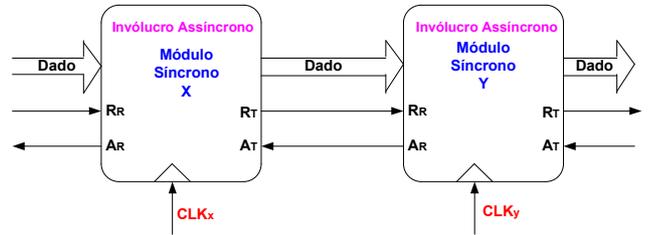


Fig. 20. Comunicação de dados: módulos síncronos.

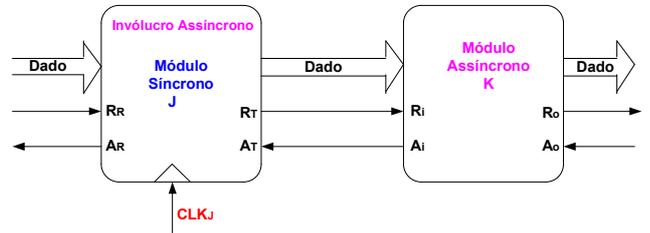


Fig. 21. Comunicação de dados: módulo síncrono → módulo assíncrono.

VI. CONCLUSÃO

Projeto de sistemas heterogêneos é um estilo interessante de projeto digital. Ele elimina os problemas relacionados com o *clock* global e reduz drasticamente os problemas relacionados com a síntese assíncrona. Este artigo propõe uma nova interface para sistemas heterogêneos, onde a comunicação de dados entre os módulos ocorre no protocolo de 2-Fases, permitindo um aumento de desempenho. Para trabalhos futuros aplicar esta interface em radio definido por software.

REFERÊNCIAS

- [1] G. DE Micheli, “An Outlook on Design Technologies for Future Integrated Systems,” *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol.28, no.6, pp. 777-789, June 2009.
- [2] K. D. Muller-Glaser, et. al. “Multiparadigm Modeling in Embedded Systems Design”, *IEEE Trans. on Control Systems Technology*, vol. 12, no. 2, March 2004.
- [3] A. J. Martin and M. Nystrom, “Asynchronous Techniques for System-on-Chip Design,” *Proc. of the IEEE*, vol.94, no. 6, pp.1089-1120, June 2006.
- [4] J. J. Rodriguez, et. Al., “Features, Design Tools, and Applications Domains of FPGAs”, *IEEE Trans. on Industrial Electronics*, vol. 54, No. 4, pp.1810-1823, August 2007.
- [5] P. P. Czapski and A. Sluzek, “A Survey on System-Level Techniques for Power Reduction in Field Programmable Gate Array (FPGA)-Based Devices”, *The Second Int. Conf. on Sensor Technologies and Applications*, pp.319-327, 2008.
- [6] R. ho, K. W. Mal and M. A. Horowitz, “The Future of Wires,” *Proc. of the IEEE*, vol. 89, no. 4, pp. 490-504, April 2001.
- [7] D. Sylvester and K. Keutzer, “Getting to the Bottom of Deep Submicron,” *Proc. ICCAD*, pp. 203-211, November 1998.
- [8] E. G. Friedman, “Clock Distribution Networks in Synchronous Digital Integrated Circuits,” *Proc. of. The IEEE*, vo. 89, pp. 665-692, 2001.

- [9] D. G. Messerschmitt, "Synchronization in Digital System Design" IEEE Journal on Selected Areas in Commun. , vol. 8, no 8, pp. 1404-1419, October 1990.
- [10] A. Jain *et al.*, "A 1.2 GHz alpha microprocessor with 44.8 GB/s chip pin bandwidth," in *IEEE Int. Solid-State Circuits Conf. Tech. Dig.*, Feb. 2001, pp. 240–241.
- [11] D. M. Chapiro, *Globally-Asynchronous Locally-Synchronous Systems*, PhD thesis, Stanford University, October 1984.
- [12] D. S. Bormann and P. Y. K., "Asynchronous Wrappers for Heterogeneous Systems," Proc. Int. Conf. Computer Design (ICCD), pp.307-314, October 1997.
- [13] P. Techan, M. Greenstreet, and G. Lemieux, "A Survey and Taxonomy of GALS Design Styles," *IEEE Design & Test of Computers*, vol. 24, pp.418-428, September-October 2007.
- [14] S. Zhuang, et al., "An Asynchronous Wrapper with Novel Handshake Circuits for GALS Systems," Proc. IEEE, Communications, Circuits and Systems and West Sino Expositions, pp.1521-1525, 2002.
- [15] L. Jozwiak, et al., "Multi-objective Optimal Controller Synthesis for Heterogeneous embedded Systems," Int. Conf. on Embedded Computer Systems: Architectures, Modeling and Simulation, pp. 177-184, 2006.
- [16] A. Kumala, et al., "Reliable GALS Implementation of MPEG-4 Encoder with Mixed Clock FIFO on Standard FPGA," Int. Conf. on Field Programmable Logic and Application, pp. 1-6, 2006.
- [17] A. Reddy Ravi, "Globally-Asynchronous, Locally-Synchronous Wrapper Configurations for Point-to-Point and Multi-Point Data Communication," Master of Science, University of Central Florida, 2001.
- [18] J. Muttersbach, "Globally-Asynchronous Locally-Synchronous Architectures for VLSI Systems," Ph.D. Thesis, ETH, Zurich, 2001.
- [19] S. M. Nowick, *Automatic Synthesis of Burst-Mode Asynchronous Controller*, PhD thesis, Stanford University, 1993.
- [20] . R. M. Fuhrer, et al., "Minimalist: An environment for the Synthesis, verification and testability of burst-mode machines," Technical Report, Columbia University, TR-CUCS-020-99, 1999.
- [21] D. L. Oliveira, et al., "An Asynchronous Interface with Robust Control for Globally-Asynchronous Locally-Synchronous Systems," Journal of Aerospace Technology and Management, vol. 5, nro. 1, pp. 91-102, 2013.
- [22] C. J., Myers, "*Asynchronous Circuit Design*", Wiley & Sons, Inc., 2004, 2a edition.
- [23] D. L. Oliveira, et al., "Synthesis of Asynchronous Digital Systems of High Performance using Simpler Approach," Proc. XVIII Iberchip Workshop, pp.74-79, 2012.
- [24] W. Hardt, et. al., "Architecture Level Optimization for Asynchronous IPs", Proc. 13th Annual IEEE Int. Conf. ASIC/SOC, pp.158-162, 2000.
- [25] Altera Corporation, 2013, www.altera.com.