

Projetos de Sistemas Computacionais com Suporte de Simulação, de Plataformas Abertas de Prototipação e de Hardware Reconfigurável com o CompSim

Eduardo Carlos Pereira da Silva Proto¹, Guilherme Álvaro Rodrigues Maia Esmeraldo¹, Edson Barbosa Lisboa², Edna Natividade da Silva Barros³

¹Instituto Federal do Ceará (IFCE) – Crato – CE – Brazil

²Instituto Federal de Sergipe (IFS) – Aracaju – SE – Brazil

³Centro de Informática (CIn) – Universidade Federal de Pernambuco (UFPE) – Recife – Brazil

ecpsproto@gmail.com, guilhermealvaro@ifce.edu.br,
edson.lisboa@academico.ifs.edu.br, ensb@cin.ufpe.br

Abstract. *Designing new computational systems is not a simple task, as it deals jointly with hardware and software issues. To address this complexity, the Platform Based Design approach considers the use of a predefined microprocessor-based hardware architecture (Platform), which can be programmed and modified to meet the design requirements. This paper presents CompSim, a tool for designing computational systems, which uses simulation for functional validation and supports physical implementation of the system under development on open prototyping and reconfigurable hardware platforms. The results show that it is possible to establish trade-offs among distinct metrics to support the choice of the most suitable platform.*

Resumo. *Projetar novos sistemas computacionais não é uma tarefa simples, pois lida-se conjuntamente com aspectos de hardware e de software. Para tratar essa complexidade, a abordagem de Projetos Baseados em Plataforma considera o uso de uma arquitetura de hardware predefinida baseada em microprocessador (Plataforma), que pode ser programada e modificada para atender aos requisitos de projeto. Este artigo apresenta CompSim, uma ferramenta de projeto de sistemas computacionais, que utiliza simulação para validação funcional e suporta implementação física do sistema em desenvolvimento em plataformas abertas de prototipação e de hardware reconfigurável. Os resultados mostram que é possível estabelecer trade-offs entre métricas distintas para apoiar a escolha da plataforma mais adequada.*

1. Introdução

Atualmente, o estado da arte em sistemas computacionais abrange diferentes aplicações, tais como Internet das Coisas (IoT), *Smart Cities*, Indústria 4.0, Robótica, Computação de Alto Desempenho, Inteligência Artificial, Computação em Nuvem, entre outras [Ghaffarian 2016] [Esmeraldo et al. 2020].

O projeto de novos sistemas computacionais não é uma tarefa simples, pois deve-se considerar conjuntamente aspectos de hardware e software. Além disso, no projeto de novos sistemas eletrônicos deve-se equilibrar os custos de produção com o tempo e o custo de desenvolvimento, tendo em vista as restrições da aplicação, tais como desempenho, funcionalidades e dimensões do produto [Sangiovanni-Vincentelli and Martin 2001].

Para tratar essa complexidade, na literatura, surgiram diferentes abordagens de projeto de sistemas computacionais, como é o caso de Projeto Baseado em Plataforma (*Platform Based Design* - PBD) [Davare et al. 2007]. Nesta abordagem, considera-se uma arquitetura de hardware predefinida e baseada em microprocessador, chamada de *Plataforma*, que pode ser modificada para atender aos requisitos de uma determinada aplicação ou classe de aplicações. Durante um projeto baseado em plataforma, uma aplicação sofre o particionamento de suas funções, as quais serão mapeadas para implementação em hardware ou em software. Isto pode implicar em necessidade de modificações na plataforma, tais como substituição, inclusão, configuração de componentes de hardware e ajustes na estrutura de comunicação, e/ou no software que será executado pela plataforma. Esse processo é iterativo e, a cada ciclo, novos refinamentos são realizados para estabelecer o software e a plataforma de hardware mais adequados à solução.

De uma forma geral, há duas tecnologias base que podem ser adotadas para a implementação física de plataformas, que são: ASIC (*Application Specific Integrated Circuit*) e FPGA (*Field Programmable Gate Array*) [Barkalov, Titarenko and Mazurkiewicz 2019]. Em PBD, os ASICs implementam sistemas computacionais programáveis, tais como os *System-on-Chips* (SoCs), que podem conter processadores e/ou microcontroladores, memórias *on-chip*, funções de periféricos e de aceleradores de hardware. Já um FPGA, que também possui uma estrutura fixa de fabricação, traz uma abordagem mais flexível, pois com ela pode-se modelar o comportamento do hardware, de maneira que é possível ter diferentes módulos independentes executando simultaneamente e com alta frequência de operação [Rajewski 2017].

Naturalmente, a escolha da tecnologia mais adequada para implementação da plataforma requer que o projetista caracterize os dados mais representativos da arquitetura do sistema que ele deseja implementar [Barkalov, Titarenko and Mazurkiewicz 2019]. No entanto, de forma mais pragmática, frequentemente, as escolhas são baseadas em estabelecer *trade-offs* entre quatro métricas principais: custo, tamanho físico, desempenho e consumo de energia [Kuon and Rose 2007]. Compreende-se, desta maneira, que, mesmo com a abordagem de PBD, é necessário optar por uma determinada tecnologia e realizar ajustes na plataforma e no software, visando explorar o espaço de projeto para identificar a melhor solução.

Este artigo apresenta o simulador CompSim que tem como objetivo suportar o projeto de novos sistemas computacionais, apoiado na abordagem de PBD. A ferramenta proposta inclui uma plataforma virtual, a qual é possível modificar suas configurações, programá-la, simular a execução de um sistema computacional e analisar os resultados da simulação para validação dos requisitos funcionais. Após a validação funcional, é possível exportar o sistema, de forma automatizada, para plataformas abertas de prototipação (que incluem ASICs) e/ou de hardware reconfigurável (FPGA). Com isso, pode-se estabelecer um fluxo de projeto de novos sistemas computacionais que inclui a validação funcional durante a simulação e flexibilização da sua implementação física em diferentes plataformas de hardware, visando atender aos requisitos/restrições de projeto.

Observa-se que, na literatura, há um *lack* de ferramentas de simulação que lidam com plataformas de hardware, como mostra o estudo em [Esmeraldo et al. 2019]. Pode-se destacar o simulador SimuS, que permite exportar o sistema para plataformas abertas de prototipação baseadas em Arduino [Arduino 2018], e o cMIPS, que consiste de uma implementação do processador MIPS em linguagem de descrição de hardware (HDL).

O restante do artigo está dividido da seguinte forma: A Seção 2 apresenta o simulador CompSim; As Seções 3 e 4 detalham os recursos de exportação do sistema virtual para plataformas abertas de prototipação e de hardware reconfigurável,

respectivamente; Na Seção 5, apresenta-se os resultados experimentais; e, por fim, a Seção 6 apresenta as considerações finais e demarca os trabalhos futuros.

2. O Simulador CompSim

CompSim consiste de um ambiente virtual para apoio ao ensino-aprendizado em Arquitetura e Organização de Computadores e ao projeto de novos sistemas computacionais [Esmeraldo and Lisboa 2017].

A arquitetura de software do CompSim está dividida em três camadas, que são: 1) Plataforma de Hardware Virtual (PHV): consiste de uma plataforma de hardware parametrizável e simulável, cujos componentes são desenvolvidos em software e possuem as características dos principais componentes do computador, buscando oferecer cenários de projeto semelhantes aos reais. A PHV possui os seguintes componentes: A) CPU de 16-bits com arquitetura CISC (*Complex Instruction Set Computer*), que inclui as principais características de processadores reais, tais como registradores, diferentes tipos de instruções, Unidade Lógica e Aritmética e Unidade de Controle; B) Memória Principal (RAM), capaz de armazenar dados e instruções, bem como é parametrizável no número de linhas e de colunas, e o tamanho do bloco de dados. A RAM possui espaço de endereçamento máximo de 12-bits ou 4096 posições endereçáveis; C) Memória Cache, utilizada para minimizar o número de acessos à memória RAM pela CPU, inclui os seguintes parâmetros: número de linhas e de palavras por linha, técnicas de mapeamento (*Direct*, *Associative* e *Set Associative*), políticas de atualização (*Write-Through* e *Write-Back*), de alocação (*Write-Allocate* e *Write-Around*) e de substituição (*First-In, First-Out - FIFO*, *Least Recently Used - LRU* e *Random*); D) Dois barramentos compartilhados, sendo um deles de Sistema, que conecta CPU e memórias Cache e RAM, e o outro de Periféricos; e E) Subsistema de Entrada/Saída (E/S), que é conectado ao barramento de periféricos e permite a comunicação com até 256 periféricos. A Figura 1 ilustra a PHV do CompSim, onde visualiza-se a interligação de seus componentes;

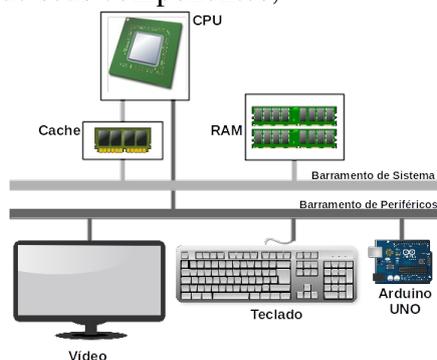


Figura 1. Plataforma de Hardware Virtual do simulador CompSim.

2) Interface Gráfica, que consiste de uma GUI (*Graphical User Interface*) que está integrada aos componentes da PHV e traz funções para criar uma plataforma, configurá-la, programá-la e simulá-la, visualizar os status dos componentes de hardware da PHV e elementos do programa, durante uma simulação, e, após uma simulação, realizar análises de desempenho do sistema computacional em desenvolvimento; e 3) Periféricos, que podem ser conectados ao Subsistema de E/S da PHV. O Subsistema de E/S permite a conexão automatizada de um novo periférico ao barramento de periféricos, tornando-o disponível para comunicação com os outros componentes da PHV. Os periféricos suportados pelo Subsistema de E/S do Simulador CompSim podem ser do tipo virtual (implementado em software e simula um periférico real) e físico (possui uma parte em software, para conexão com o barramento de periféricos, e

uma parte em hardware físico). Na Figura 1, ilustra-se os periféricos Video e Teclado, como periféricos virtuais, e o periférico Arduino UNO, como físico.

Ainda na camada de interface gráfica, há ferramentas que permitem exportar o sistema virtual em desenvolvimento para plataformas abertas de prototipação baseadas em Arduino e hardware reconfigurável (FPGA). As duas seções a seguir detalham esses recursos.

3. Exportando o Sistema para Plataformas Arduino

O simulador CompSim permite exportar o sistema simulável em desenvolvimento (PHV e o programa) sob o formato de um emulador que pode ser executado em plataformas abertas de prototipação Arduino UNO e MEGA.

O emulador consiste de um *firmware* que implementa a PHV e o respectivo programa na linguagem C/C++ nativa do Arduino. Desta maneira, ao exportar o emulador, os componentes da PHV serão copiados para a memória *flash* e sua memória RAM (com o programa para execução) será alocada na memória SRAM do microcontrolador da plataforma Arduino. Com isso, obtém-se uma implementação física do sistema computacional em desenvolvimento, com execução em modo *standalone* (que independe do simulador CompSim). A Figura 2 ilustra a ferramenta do CompSim em que é possível exportar o emulador do sistema para uma plataforma Arduino UNO.

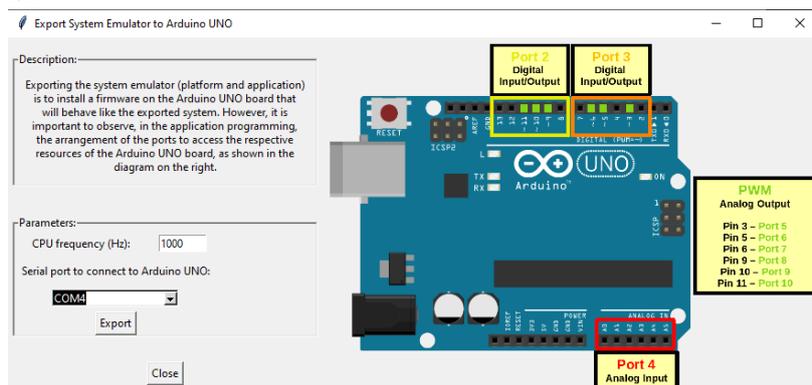


Figura 2. Ferramenta para Exportação do Sistema como Emulador para Arduino UNO.

Na Figura 2, à esquerda, deve-se configurar: 1) Frequência de operação da CPU (“*CPU frequency*”), que consiste do número de instruções executadas por segundo pela CPU da PHV, na plataforma Arduino; e 2) Porta serial (“*Serial port to connect to Arduino UNO*”), que consiste da porta serial na qual o Arduino UNO está conectado fisicamente ao computador e é utilizada como canal para gravação do *firmware* do emulador no microcontrolador do Arduino UNO, com apoio do ambiente *Arduino IDE*.

O emulador trata as operações de E/S digitais e analógicas do Arduino como periféricos da PHV. Assim, pode-se criar sistemas computacionais cujos periféricos aproveitam os recursos de plataformas Arduino para interagir com usuários, outros sistemas eletrônicos e o próprio ambiente. À direita da Figura 2, pode-se observar um diagrama que ilustra o mapeamento entre os endereços de periféricos (“portas”) da PHV e os respectivos recursos do Arduino UNO. Por exemplo, a porta 2 (“*Port 2*”) está mapeada nos pinos 8 a 13 da *board* Arduino UNO, nos quais é possível realizar operações de leitura e escrita digitais.

4. Exportando o Sistema para Plataformas com Hardware Reconfigurável

Para tornar possível a exportação dos sistemas simuláveis para plataformas com

uma operação de E/S (IORdy).

Ao contrário da abordagem que trata da exportação do sistema para Arduino, cujo mapeamento entre portas e recursos do Arduino é fixo, o CompSim traz uma ferramenta que, além de exportar o sistema para uma plataforma com hardware reconfigurável, é possível mapear os sinais entre Subsistema de E/S e periféricos nos pinos de comunicação de uma *board* com FPGA. Essa abordagem torna o desenvolvimento de sistemas em plataformas de hardware reconfigurável mais flexível, seguro e produtivo, dado que projetos que envolvem FPGA são complexos e propensos a diversos tipos de erros. A Figura 4 (a) ilustra a ferramenta do CompSim em que é possível configurar uma *board* com FPGA com o sistema descrito em VHDL.

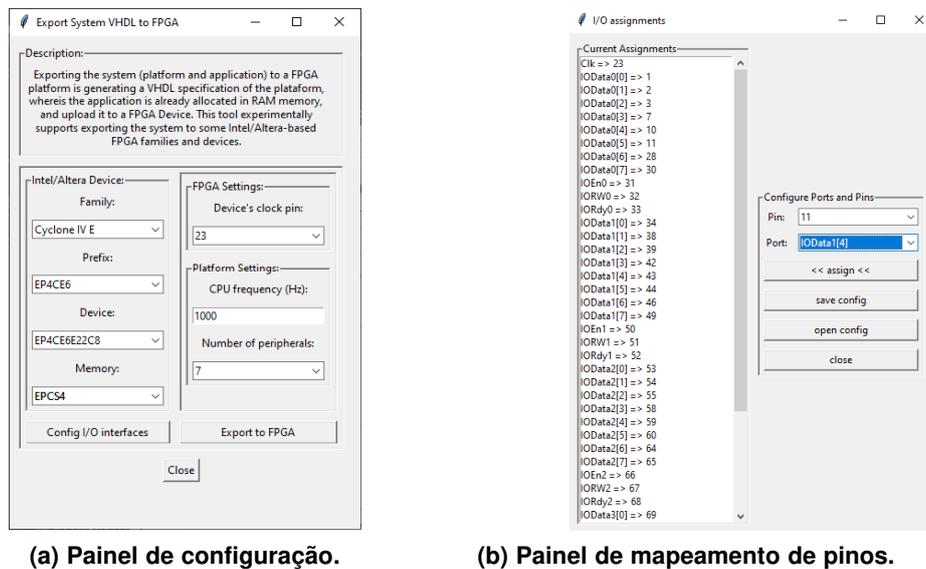


Figura 4. Ferramenta para Exportação do Sistema para FPGA.

Na Figura 4 (a), no painel à esquerda, para configuração da plataforma, o primeiro passo envolve selecionar os dados do FPGA (“*Family*”, “*Prefix*” e “*Device*”) e da memória (“*Memory*”), suportados na *board* adotada para o projeto. Ainda na Figura 4(a), no painel direito e acima, deve-se informar o pino da *board* onde estará disponível o sinal de *clock* (“*Device’s Clock Pin*”), pois ele será utilizado para conduzir as operações síncronas da PHV. No painel logo abaixo, estão as configurações da plataforma no FPGA, em que deve-se informar a frequência de operação do sistema (“*CPU frequency (Hz)*”) e o número de periféricos que a PHV suportará. Ressalta-se aqui que cada fabricante de soluções com FPGA pode incluir diferentes periféricos na *board*, tais como *leds*, chaves e *displays*, os quais são mapeados em pinos da interface do FPGA, fazendo com que o mapeamento de pinos, em alguns casos, tenha que ser personalizado. O botão “*Config I/O Interfaces*”, abre o painel ilustrado na Figura 4(b), onde é possível personalizar o mapeamento entre portas e pinos, alterando as atribuições geradas de forma automática. Em seguida, ao clicar no botão “*Export to FPGA*”, a ferramenta utilizará os *scripts* de linha de comando do ambiente de projeto de sistemas digitais *Intel Quartus Prime Design Suite (IQPDS)* para gerar as configurações do sistema descrito em VHDL (PHV com o respectivo programa alocado em sua RAM) e as devidas configurações de mapeamento de portas-pinos da *board*, para o modelo de FPGA informado. Em seguida, a ferramenta aqui apresentada utilizará o IQPDS para configurar o FPGA com o respectivo sistema digital, obtendo, dessa maneira, a implementação física do sistema em desenvolvimento.

5. Resultados Experimentais

O Simulador CompSim vem sendo utilizado por estudantes de diferentes turmas de um curso de Técnico em Eletrônica e de um Bacharelado em Sistemas de Informação. Com o suporte de exportação da sua PHV para plataformas de hardware físico, os estudantes são estimulados a criar diferentes sistemas computacionais, tais como para: piscar leds (“árvore-de-natal”, semáforos e jogos); exibição de dados em *displays* (contadores numéricos, relógio e menu interativo); monitoramento (temperatura, umidade, luminosidade, proximidade e infravermelho); sinalizadores (alarme e código morse); movimento (hélices, carrinho e braço mecânico); e que interagem com outros sistemas eletrônicos tais como o próprio Arduino e sistemas digitais em FPGA. Nesse sentido, tem-se uma abordagem de projeto mista, baseada em PBD, que envolve tanto o desenvolvimento do hardware do periférico, quanto do respectivo software para controlá-lo. A Tabela 1 apresenta um comparativo entre as plataformas adotadas para implementação física de sistemas virtuais criados no simulador CompSim.

Tabela 1. Comparativo entre plataformas para implementação física de sistemas virtuais criados no simulador CompSim.

Métrica	Arduíno UNO	Arduíno MEGA	Intel/Altera Cyclone IV EP4CE6
PHV (consumo médio)	13% de 32 KiB de memória <i>flash</i>	4% de 256 KiB de memória <i>flash</i>	22% de 6.272 elementos lógicos
Posições endereçáveis de RAM	900	3.900	4.096
Área (cm ²)	≈ 36	≈ 54	≈ 70
Frequência do oscilador (MHz)	16	16	472,5
Custo médio (R\$)*	R\$ 55,00	R\$ 100,00	R\$ 250,00

* Calculado a partir de consultas a diversos websites de vendas de produtos eletrônicos.

De acordo com a Tabela 1, ao exportar o sistema para plataformas Arduino, o emulador consome, em média, 13% e 4% das capacidades de armazenamento das memórias *flash* dos modelos UNO e MEGA, respectivamente. Já no cenário com hardware reconfigurável, a PHV consome em média 22% dos elementos lógicos do FPGA (adotou-se aqui o *device* Intel/Altera Cyclone IV EP4CE6). Observa-se ainda que, ao utilizar as plataformas Arduino UNO e MEGA, não é possível alocar todo o espaço de endereçamento da memória RAM da PHV, dadas as restrições das capacidades de armazenamento das memórias SRAM dos microcontroladores, sendo portanto possível dispor de 900 e 3.900 posições no espaço de endereçamento da memória RAM, respectivamente. Como a SRAM da plataforma com FPGA dispõe de maior capacidade de armazenamento (33.750 KiB), é possível alocar todo o armazenamento projetado para a RAM da PHV (4096 posições endereçáveis).

Ainda na Tabela 1, para a escolha da plataforma mais adequada a uma determinada aplicação, pode-se estabelecer um *trade-off* entre: capacidade de armazenamento da memória RAM da PHV, área física ocupada pela *board*, desempenho (cuja medida se baseia na frequência de operação do oscilador da *board*) e custo de aquisição da plataforma. A plataforma Arduino UNO apresenta menores custos de aquisição e área física, porém possui as menores capacidades de armazenamento de RAM e desempenho. Arduino MEGA, dentre as três plataformas, apresenta custo de aquisição e área física intermediários, bem como alta capacidade de armazenamento, porém apresenta baixo desempenho. Por fim, a plataforma com FPGA apresenta maiores custo de aquisição e área física, entretanto dispõe das maiores capacidades de armazenamento da RAM e desempenho.

6. Conclusões

Neste artigo apresentou-se o CompSim, uma ferramenta integrada para projetos de sistemas computacionais com abordagem PBD, que utiliza simulação para validação funcional, bem como plataformas abertas de prototipação baseadas em Arduino e de hardware reconfigurável para implementação física do sistema em desenvolvimento. Os resultados mostraram que é possível estabelecer compromissos entre diferentes métricas para escolha de uma plataforma física que melhor se adequa aos requisitos de uma determinada aplicação. Os trabalhos futuros compreendem a inclusão de novos componentes de simulação à PHV, para oferecer novos cenários de projeto, e o suporte de mais plataformas de hardware, visando ampliar as opções de implementações físicas.

Referências

- Arduino. (2018) "What is Arduino?", <https://www.arduino.cc/en/Guide/Introduction>, February.
- Barkalov A., Titarenko L. and Mazurkiewicz M. (2019) "Design of Embedded Systems", In: Foundations of Embedded Systems. Studies in Systems, Decision and Control, 195. Springer, Cham.
- Davare, A., Densmore, D., Meyerowitz, T., Pinto, A., Sangiovanni-Vincentelli, A., Yang, G., Zeng, H. and Zhu, Q. (2007) "A next-generation design framework for platform-based design", In: Conference on using hardware design and verification languages, 152.
- Esmeraldo, G. and Lisboa, E. B. (2017) "Uma ferramenta para exploração do ensino de organização e arquitetura de computadores", In: International Journal of Computer Architecture Education, 6(1), p. 68-75.
- Esmeraldo, G. A. R. M., Lisboa, E. B., Mendes, C. S. R., Cartaxo, L. F., Ribeiro, C. V., Santos, P. S., Morato, L. F. B. and Nascimento, M. S. (2020) "Uma Abordagem Integrada de Hardware e Software para o Aprendizado de Subsistemas de Entrada/Saída em Projetos de Sistemas Computacionais", In: International Journal of Computer Architecture Education (IJCAE), 9(1), p. 1-9.
- Esmeraldo, G. A. R. M., Mendes, C. S. R., Cartaxo, L. F. And Lisboa, E. B. (2019) "Apoio ao Aprendizado em Arquitetura e Organização de Computadores: Um Estudo Comparativo entre Simuladores Computacionais", In: Revista Tecnologias na Educação, 31, p. 1-17.
- Ghaffarian, R. (2016) "Microelectronics packaging technology roadmaps, assembly reliability, and prognostics", In: Facta universitatis-series: Electronics and Energetics, 29(4), p. 543-611.
- Kuon, I. and Rose, J. (2007) "Measuring the gap between FPGAs and ASICs". In: IEEE Transactions on computer-aided design of integrated circuits and systems, 26(2), p. 203-215.
- Lameres, B. J. (2019) "Quick start guide to VHDL". Springer.
- Rajewski, J. (2017) "Learning FPGAs: Digital Design for Beginners with Mojo and Lucid HDL". 1st Edition. O'Reilly Media.
- Sangiovanni-Vincentelli, A. and Martin, G. (2001) "Platform-based design and software design methodology for embedded systems", In: IEEE Design & Test of Computers, 18(6), p. 23-33.