Extensão da Ferramenta MultiExplorer para Exploração de Projetos de GPUs e Máquinas Virtuais

Samuel Rodrigues, Ricardo Santos

[1]Faculdade de Computação – Universidade Federal de Mato Grosso do Sul (UFMS)
79070-970 – Campo Grande – MS – Brazil

{samuel_rodrigues, ricardo.santos}@ufms.br

Abstract. *This work presents the design and development of the MultiExplorer tool, which supports the exploration of multiple design flows for computer architectures. The design flows developed in this work (GPUs and virtual machines) cover everything from functional simulation, physical exploration of computational resources, to design space exploration.*

Resumo. *Este trabalho apresenta o projeto e desenvolvimento da ferramenta MultiExplorer, que oferece suporte à exploração de múltiplos fluxos de projetos de arquiteturas de computadores. Os fluxos de projeto desenvolvidos neste trabalho (GPUs e máquinas virtuais) abrangem desde a simulação funcional, a exploração física dos recursos computacionais e a exploração do espaço de projeto.*

1. Introdução

A exploração do espaço de projeto de sistemas computacionais, ou *Design Space Exploration* (DSE), refere-se ao processo de encontrar diferentes alternativas de projeto para um sistema específico antes da sua implementação [Kang et al. 2010]. Esta técnica visa maximizar parâmetros de desempenho, como tempo, densidade de potência, performance e minimizar parâmetros como custo, área, entre outros, enquanto respeita restrições de arquitetura previamente estabelecidas. Dada a crescente complexidade dos sistemas computacionais, torna-se necessário utilizar ferramentas que automatizem e auxiliem no processo de projeto arquitetural, de acordo com o orçamento disponível. Tais ferramentas possibilitam encontrar soluções viáveis em espaços de exploração complexos e diante de várias restrições e objetivos impostos pelo usuário.

O MultiExplorer [Devigo et al. 2015] [Amorim and Duenha 2023] é um *framework* desenvolvido com o intuito de automatizar a exploração de espaço de projeto de arquiteturas de computadores. Atualmente, o MultiExplorer possui uma infraestrutura que possibilita a exploração do espaço de projeto de processadores *multicore* heterogêneos.

Diante do contexto do MultiExplorer e considerando a demanda atual por projetos de sistemas computacionais que utilizam GPUs e Máquinas Virtuais (*Virtual Machines* - VMs), este trabalho estendeu as funcionalidades e alcance dessa ferramenta. Especificamente, este artigo apresenta o fluxo de desenvolvimento integrado ao MultiExplorer para a exploração de sistemas computacionais que utilizam GPUs e máquinas virtuais.

2. Trabalhos Relacionados

O Heracles [Kinsy et al. 2013] é um conjunto de ferramentas *open source* desenvolvido para ensino e pesquisa na exploração arquitetural de sistemas *multicore*, baseada em simulação RTL. Inclui módulos de hardware em HDL, compilador de aplicações, e uma interface gráfica, tornando-a uma ferramenta com exploração rápida de alternativas *multicore*.

O ZigZag [Mei et al. 2021] é um *framework* de exploração do espaço de projeto que busca encontrar soluções ótimas, a partir da definição da curva de Pareto, para arquiteturas de aceleradores de redes neurais profundas para sistemas embarcados. A ferramenta inclui algoritmos de mapeamento, desempenho, entre outros. Através de benchmarks, ZigZag provou encontrar soluções até 64% mais eficientes que outros trabalhos do mesmo tema.

O ArtA (Artificial Architect) [Paraskevopoulos et al. 2024] é uma ferramenta de exploração do espaço de projeto para arquiteturas de spin-qubit baseadas em quantum dots. Através da ferramenta de otimização ArtA, é possível explorar 29.312 arquiteturas. A ferramenta demonstrou reduzir o tempo de exploração em até 99,1% em comparação com abordagens tradicionais, como força bruta.

Os trabalhos mencionados utilizam da exploração de projetos de sistemas específicos. Diferentemente das ferramentas Heracles, Zigzag e ArtA, MultiExplorer possibilita a integração de diversos fluxos de exploração de projetos em um único ambiente de forma flexível, oferecendo ao usuário a possibilidade de explorar projetos de sistemas computacionais de diferentes arquiteturas. Adicionalmente, MultiExplorer também permite que o usuário possa atuar de maneira simplificada no fluxo de exploração, seja simulando o desempenho ou as características físicas (área, consumo energético) do sistema, podendo chegar até a exploração de novas alternativas arquiteturais.

3. MultiExplorer

MultiExplorer [Devigo et al. 2015] foi inicialmente concebido visando a exploração do espaço de projetos de arquiteturas *Multicore* and *Manycore*. Para isso, utilizou a ferramenta **MPSoCBench** [Duenha et al. 2014] para simulação de arquiteturas de processadores e **McPAT** (Multicore Power, Area, and Timing) [Li et al. 2009] para estimativas físicas do processador. Através dos resultados de desempenho do MPSoCBench e das estimativas físicas do McPAT, o projetista do sistema computacional *multicore* possui condições para avaliar se a configuração atende as restrições e objetivos previamente definidos.

Em etapas posteriores, a ferramenta McPAT, integrada ao MultiExplorer, foi estendida a fim de suportar estimativas de *dark silicon* [Silva et al. 2015]. Com essa nova funcionalidade, tornou-se necessário adicionar ao fluxo de execução uma abordagem de exploração do espaço de projeto ciente de *dark silicon*. Técnicas metaheurísticas (utilizando o algoritmo NSGA-II [Deb et al. 2002]) foram inseridas ao fluxo de exploração e modelos de performance, baseados em técnicas de aprendizado de máquina, foram integrados ao MultiExplorer [Santos et al. 2017]. O fluxo de exploração de sistemas multicore, disponíveis no MultiExplorer, é apresentado na Figura 1.

O fluxo de *multicores* foi estendido em MultiExplorer para receber como entrada informações sobre o sistema computacional de referência e as restrições de área e potência impostas. A simulação é executada através da ferramenta Sniper [Carlson et al. 2011].

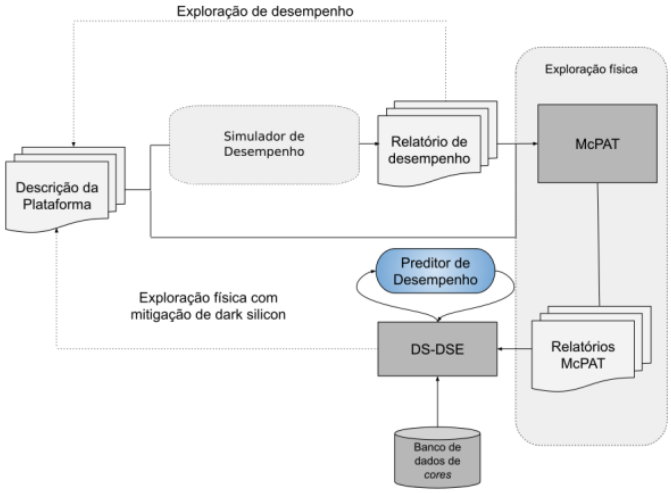


Figura 1. Fluxo do MultiExplorer para projetos de sistemas *multicores*.

A partir do desenvolvimento inicial do MultiExplorer, novos trabalhos científicos independentes foram propostos visando o desenvolvimento de fluxos abrangendo novas arquiteturas de sistemas. O trabalho de Sonohata e Duenha [Sonohata et al. 2023] desenvolveu um fluxo de exploração do espaço de projeto de sistemas heterogêneos baseados em GPUs. Os autores utilizaram ferramentas de simulação e estimativa física como o **GPGPU-Sim** [Khairy et al. 2020] e **GPUWattch** [Leng et al. 2013]. O fluxo busca maximar o desempenho ao mesmo tempo em que minimiza a densidade de potência de sistemas computacionais que utilizam GPUs.

Arigoni e Santos [Arigoni et al. 2022] desenvolveram um fluxo de exploração do espaço de projeto de sistemas de máquinas virtuais. O desempenho das máquinas vírtuais foi mensurado através do simulador **CloudSim** [Goyal et al. 2012] e a exploração através do algoritmo genético NSGA-II. O objetivo deste fluxo de exploração é maximizar o desempenho da aplicação e minimizar custo de utilização de máquinas virtuais em ambiente de nuvem computacional.

Amorim e Duenha [Amorim and Duenha 2023] projetaram uma interface gráfica e arquitetura escalável para o MultiExplorer. A interface facilita a utilização e a interação com a ferramenta. A arquitetura possibilita que novos fluxos e funcionalidades sejam mais facilmente integrados e possam ser utilizados pelos usuários.

Ressalta-se que os trabalhos supramencionados foram desenvolvidos de maneira independente, não sendo integrados em um fluxo de execução único em MultiExplorer. A ausência da integração desses vários fluxos de exploração de projetos e demais funcionalidades motivaram o desenvolvimento deste trabalho.

4. Desenvolvimento e Integração da Ferramenta MultiExplorer

O *framework* MultiExplorer foi desenvolvido na linguagem de programação Python v2.7. Uma representação do projeto da integração dos fluxos de GPUs e VMs, no MultiExplorer pode ser observado na Figura 2.

Para realizar a integração dos fluxos de GPUs e VMs na ferramenta MultiExplorer foram projetadas e desenvolvidas cinco classes Python para cada projeto: *Adapters*, *AllowedValues*, *MultiexplorerVM/GPGPU*, *Presenters* e *Steps*. A estrutura da integração

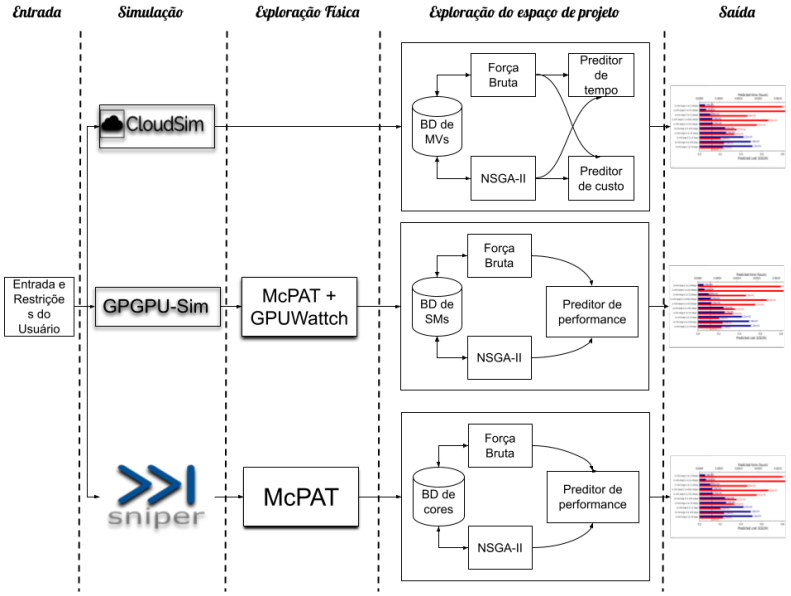


Figura 2. Fluxos de exploração de projetos integrados ao MultiExplorer.

dos fluxos é representada na Figura 3 por um diagrama de classes. Devido ao espaço limitado, o diagrama de classes foi significativamente reduzido. No entanto, as principais modificações foram mantidas.

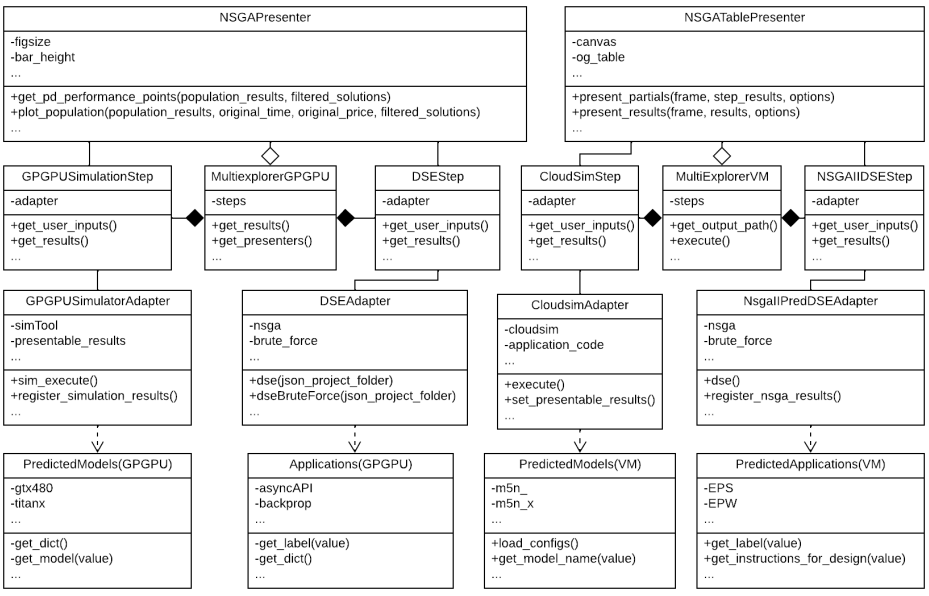


Figura 3. Diagrama de classes para os fluxos de GPU e VMs no MultiExplorer.

MultiExplorerVM/GPGPU é a classe principal que engloba todas as demais classes de um fluxo de projeto. O detalhamento de cada tipo de classe projetada para a integração dos fluxos é apresentada a seguir:

- A classe *Steps* define os passos que cada fluxo de execução possui. Cada passo se comunica com seu *Adapter* e possui informações relacionadas aos *Presenters*. Para o fluxo de GPUs, existem duas classes de *Steps*: *GPGPUSimulationStep* e

DSEStep. Para o fluxo de VMs, existem as classes *CloudSimStep* e *NsgaIIDSEStep*.

- A classe *Adapters* promove a interface entre o *front-end* e o *back-end*, definindo as entradas do usuário e como esse dados devem se comportar. As classes *Adapters* são *GPGPUSimulatorAdapter* e *DSEAdapter*, para o fluxo de GPUs; *CloudSimAdapter* e *NsgaIIPredDSEAdapter* para o fluxo de VMs.
- A classe *AllowedValues* delimita os valores disponíveis na interface gráfica. Como exemplo, para o fluxo de exploração de GPUs, define a lista de placas de vídeos disponíveis (*PredictModelsGPGPU*) e aplicações CUDA (*Applications(GPGPU)*). Para o fluxo de máquinas virtuais, dita os tipos de máquinas virtuais (*PredictModels(VM)*) e aplicações de *benchmarks* (*Applications(VM)*).
- *Presenters* possui classes responsáveis pela apresentação dos dados de saída. Essa classe define ajustes de alinhamento, cor, quantidades de dados, entre outros. Para a integração dos fluxo de VMs e GPUs, foram necessárias oito classes *Presenters*.

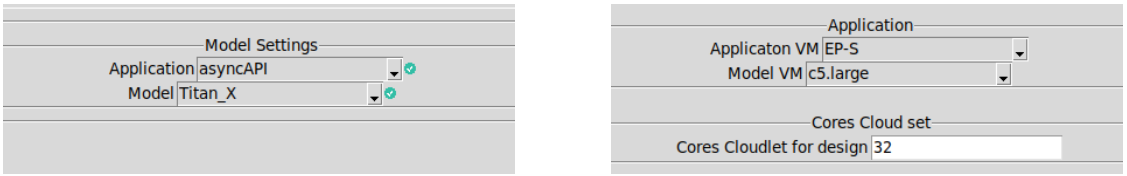
5. Cenários de Utilização e Discussão

Ao iniciar o MultiExplorer, o usuário deve escolher o fluxo de exploração do espaço de projeto e, como consequência, definir os parâmetros iniciais para a execução desse fluxo, como mostram as Figuras 4, 5 e 6.

(a) Escolha do fluxo para GPUs. (b) Escolha do fluxo para VMs.

Figura 4. Seleção do fluxo de execução.

Para as entradas do usuário na etapa de simulação do fluxo de GPUs, deve-se escolher uma aplicação (neste exemplo, AsyncAPI) de *benchmark* e a GPU inicial/original (Titan X) para iniciar a simulação e exploração. Para o fluxo de VMs (*Virtual Machine*), Figura 5(b), é escolhida a aplicação EP-S (Embaraçosamente Paralelo) e configuração de máquina virtual C5.LARGE - uma instância AWS (*Amazon Web Services*) - além da quantidade de cores demandados da *cloudlet*, uma classe que modela aplicações e serviços baseados em nuvem que são comumente empregados em *data centers* [Arigoni et al. 2022].

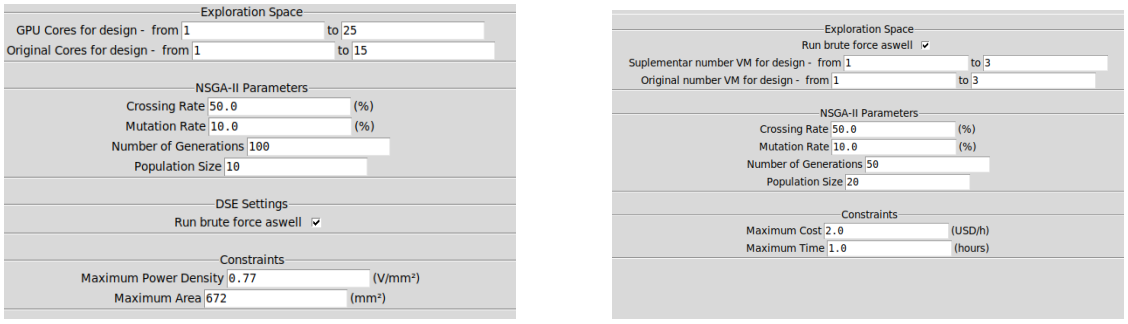


(a) Seleção da aplicação e GPU inicial (original). (b) Seleção da aplicação e configuração da VM.

Figura 5. Parâmetros para simulação e seleção da configuração inicial.

Na próxima etapa o usuário informa parâmetros de entradas relacionados com a etapa de DSE. Considerando GPUs, o usuário deve definir as faixas de alguns parâmetros como números de SMs (cores) da placa original e da placa suplementar - segunda placa

da combinação heterogênea - para o projeto, restrições de densidade de potência, área do chip, além de parâmetros sobre o algoritmo para DSE como mostra a Figura 6(a). Para VMs, deverá informar as restrições de preço e tempo, quantidade de VMs e parâmetros para o algoritmo genético, semelhante ao fluxo de GPUs, como mostra a Figura 6(b).

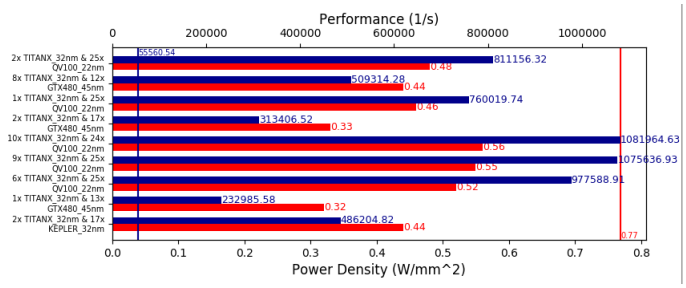


(a) Parâmetros para exploração de GPUs. (b) Parâmetros para exploração de VMs.

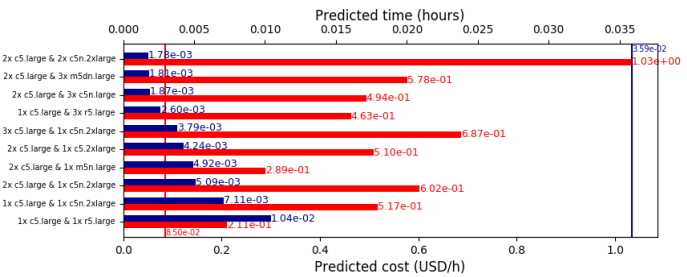
Figura 6. Parâmetros para exploração do espaço de projetos.

Observa-se que são definidos parâmetros para exploração de 1 a 25 *Streaming Multiprocessors* (SMs) para a GPU suplementar (GPU resultante da exploração de projeto) e 1 a 15 para a original (Titan X). Como parâmetros do algoritmo para exploração, determinam-se a taxa de cruzamento e mutação, quantidade de gerações e o tamanho da população. Em relação às configurações geradas pela exploração, elas devem ter área menor ou igual a 672 mm^2 e densidade de potência de até $0.77\,\mathrm{V}\,\mathrm{mm}^{-2}$. Para o fluxo de VMs, a quantidade de instâncias originais e suplementares é de 1 à 3 máquinas. Como limite de custo e tempo, o serviço deve custar no máximo 2 dólares/hora e ter uma duração de até 1 hora.

Para o fluxo de exploração de GPUs, é apresentado o gráfico da Figura 7(a).



(a) Configurações alternativas para GPUs.



(b) Configurações alternativas para VMs.

Figura 7. Gráficos com resultados da exploração do espaço de projeto.

Para o fluxo de VMs, obtém-se o gráfico da Figura 7(b). Os resultados apresentam as configurações arquiteturais viáveis encontradas (soluções da frente de Pareto) com seus respectivos valores de desempenho e densidade de potência (para GPUs) e tempo e custo (para VMs). Devido ao espaço limitado, os valores de quantificação (tabelas) foram omitidos.

Um dos resultados viáveis para as GPUs foi a de uma configuração composta por 2 Titan X e 25 QV100 com desempenho de 811156,32 e densidade de potência de 0,48. Considerando VMs, o primeiro resultado viável foi uma configuração composta por 2 VMs C5.large e 2 c5n.2xlarge com tempo de $1,78^{-3}$ horas e custo de 1,03 por hora.

6. Conclusões

Este trabalho apresentou o desenvolvimento de uma nova versão da ferramenta MultiExplorer, que engloba a integração de fluxos de exploração do projeto de diferentes arquiteturas de processadores. O trabalho projetou e implementou um conjunto de novas classes para fluxos de projetos de GPUs e máquinas virtuais. O desenvolvimento deste trabalho possibilita que usuários do MultiExplorer tenham a possibilidade de projetar e avaliar configurações de sistemas heterogêneos baseados em vários tipos de GPUs, máquinas virtuais e processadores multicore de forma integrada. Os resultados obtidos permitiram observar as opções de configurações, simulações e, especialmente, os resultados obtidos com a exploração do espaço de projetos.

A partir deste trabalho, outros desenvolvimentos futuros podem ser realizados: extensão do fluxo de execução do MultiExplorer para suportar exploração de dispositivos IoT (*Internet of Things*); adição de novas aplicações de *benchmarks*; atualização ou substituição das ferramentas de simulação para se alinharem aos novos recursos disponíveis nas mais recentes arquiteturas de hardware.

Agradecimentos

Este trabalho recebeu apoio da FUNDECT, no âmbito da Chamada Especial Fundect 07/2023 - PIBIC - Fundect

Referências

Amorim, E. and Duenha, L. (2023). Uma ferramenta para ensino e aprendizado de exploração de espaço de projeto de arquiteturas de processadores na era de dark-silicon. Master's thesis, Universidade Federal de Mato Grosso do Sul.

Arigoni, D. C. A., dos Santos, R. R., and Garanhani, L. D. D. (2022). Design space exploration of heterogeneous systems applied to the cloud resource allocation problem. In *Proceedings of the SSCAD*. SBC.

Carlson, T. E., Heirman, W., and Eeckhout, L. (2011). Sniper: exploring the level of abstraction for scalable and accurate parallel multi-core simulation. In *Proceedings of 2011 SC Conference*, New York, NY, USA. Association for Computing Machinery.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Devigo, R., Duenha, L., Azevedo, R., and Santos, R. (2015). Multiexplorer: A tool set for multicore system-on-chip design exploration. In *Proceedings of the ASAP Conference*, pages 160–161.

Duenha, L., Guedes, M., Almeida, H., Boy, M., and Azevedo, R. (2014). Mpsocbench: A toolset for mpsoc system level evaluation. In *Proceedings of the SAMOS XIV Conference*, pages 164–171.

Goyal, T., Singh, A., and Agrawal, A. (2012). Cloudsim: simulator for cloud computing infrastructure and modeling. *Procedia Engineering*, 38:3566–3572.

Kang, E., Jackson, E., and Schulte, W. (2010). An approach for effective design space exploration. In *Proceedings of the 16th FOCS*, page 33–54, Berlin, Heidelberg. Springer-Verlag.

Khairy, M., Shen, Z., Aamodt, T. M., and Rogers, T. G. (2020). Accel-sim: An extensible simulation framework for validated gpu modeling. In *Proceedings of the ISCA Conference*, pages 473–486.

Kinsy, M. A., Pellauer, M., and Devadas, S. (2013). Heracles: a tool for fast rtl-based design space exploration of multicore processors. In *Proceedings of the FPGA Conference*, page 125–134, New York, NY, USA. Association for Computing Machinery.

Leng, J., Hetherington, T., ElTantawy, A., Gilani, S., Kim, N. S., Aamodt, T. M., and Reddi, V. J. (2013). Gpuwattch: enabling energy optimizations in gpgpus. *SIGARCH Computing Architecture News*, 41(3):487–498.

Li, S., Ahn, J. H., Strong, R. D., Brockman, J. B., Tullsen, D. M., and Jouppi, N. P. (2009). Mcpat: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of the MICRO Conference*, pages 469–480.

Mei, L., Houshmand, P., Jain, V., Giraldo, S., and Verhelst, M. (2021). Zigzag: Enlarging joint architecture-mapping design space exploration for dnn accelerators. *IEEE Transactions on Computers*, 70(8):1160–1174.

Paraskevopoulos, N., Hamel, D., Sarkar, A., Almudever, C. G., and Feld, S. (2024). Arta: Automating design space exploration of spin qubit architectures.

Santos, M. T., Duenha, L., Magalhaes, F. C., and Santos, R. (2017). Avaliação de preditores de desvios por meio de simuladores como parte do processo de ensino e aprendizagem de arquitetura de computadores. In *IJCAE*.

Silva, A. C., Bignardi, T., de Palma, E., Alves, R., Hayashi, C., and Santos, R. (2015). Identificação automática de dark silicon em processadores multicore. In *Proceedings of the SSCAD*. SBC.

Sonohata, R., Arigoni, D. C. A., Fernandes, E. R., dos Santos, R. R., and Duenha, L. D. (2023). Performance predictors for graphics processing units applied to dark-silicon-aware design space exploration. *Concurrency and Computation*, 35(17).