Survey of Non-Volatile Memories

Lisandro Silva[1], Lizandro Oliveira[1] and Mauricio Pilla[1]
Universidade Federal de Pelotas - UFPel
Pelotas, Brasil
{lldsilva, lsoliveira, pilla}@inf.ufpel.edu.br

***Abstract*— O consumo de energia é tão importante quanto o desempenho em sistemas embarcados alimentados por bateria, pois cada vez mais estes sistemas precisam processar computação intensiva com um baixo consumo energético. Devido à alta contribuição do acesso à memória no consumo total de energia de sistemas embarcados, a arquitetura de memória influencia fortemente os objetivos dos projetos dos dispositivos embarcados. Novas técnicas são propostas devido aos problemas enfrentados com o avanço da tecnologia, como por exemplo, a memória tradicional baseada em SRAM (*Static Random Access Memory*) *on-chip* tornou-se um gargalo em consumo energético para o projeto de sistemas embarcados, devido principalmente ao seu alto *leakage*. As tecnologias emergentes de memórias não voláteis (*NVM, Non-Volatile Memories*) são soluções candidatas para os futuros sistemas de memória, pois elas possuem algumas vantagens sobre as memórias SRAMs (*Static Random-Access Memory*) e DRAMs (*Dynamic Random-Access Memory*) tradicionais, como por exemplo, um menor *leakage*, uma maior densidade e não volatilidade.**

I. INTRODUCÃO

Segundo Hennessy [1], em um sistema de computador moderno o gargalo dominante na obtenção de alto desempenho e eficiência energética é a distância tecnológica entre o desempenho do processador e a memória tradicional. Esta distância torna-se mais significativa em sistemas embarcados, pois o sistema de memória é um dos principais fatores de desempenho e consumo energético, especialmente nos sistemas embarcados que utilizam bateria [2]. O projeto de sistemas embarcados apresenta muitas restrições e requisitos rígidos. De um modo geral a descrição dos requisitos funcionais não é suficiente para o projeto de um sistema embarcado, devendo ser considerados também requisitos não-funcionais, tais como desempenho, custo, consumo de energia, tamanho físico e peso [3]. Estes requisitos não-funcionais extras ocasionam limitações nas decisões do projeto, provocando preocupações no desempenho e no consumo energético.

Os requisitos necessários para os sistemas embarcados têm motivado a investigação de técnicas de otimização. Na literatura são propostos diversos tipos de otimizações em memória, desde técnicas de otimização em *software*, em *hardware* ou técnicas mistas as quais utilizam tanto otimizações de *hardware* quanto de *software* [2]. Muitas dessas técnicas utilizam NVM's como uma forma de otimização, visto que essas memórias emergentes possuem algumas vantagens em relação as memórias tradicionais DRAM e SRAM. Embora o mercado de memórias emergentes ainda é menor que das memórias tradicionais existe a previsão que este mercado crescerá até 2021, atingindo taxas em torno de 110% ao ano, quando estas novas tecnologias serão utilizadas em vários produtos [4]. Alguns exemplos de memórias emergentes são a PCM/PCRAM (*Phase Change Memory Random-Access Memory*), a MRAM (*Magnetoresistive Random-Access Memory*), a STT-MRAM/STT-RAM (*Spin-Transfer Torque Magnetic Random-Access Memory*), a RRAM/ReRAM (*Resistive Random-Access Memory*), a FRAM/FeRAM (*Ferro-magnetic Random-Access Memory*) e a DWM (*Domain Wall Memory*)

Este artigo está organizado da seguinte maneira. Na Seção II alguns fundamentos sobre memórias NVM's serão apresentados. Na Seção III serão revisados e discutidos trabalhos empregando NVM's em diferentes níveis da hierarquia de memória, como caches, SPM's (*Scratchpad Memory*) e memórias principais, enquanto que as conclusões são discutidas na Seção IV.

II. VISÃO GERAL - MEMÓRIAS

O desempenho total do sistema de memória e o consumo de energia são severamente relacionados com o tempo de acesso à memória e o consumo de energia média, o que faz com que a arquitetura de uma memória seja uma grande preocupação em projetos de sistemas embarcados. Nesta Seção são apresentadas as características das NVM's.

A. STT-RAM

Segundo Smullen [5], a STT-RAM utiliza uma junção de túnel magnético (MTJ: *Magnetic Tunnel Junction*) como armazenamento de memória. Uma célula STT-RAM é formada por um transistor de acesso que é ligado a um elemento de memória implementado usando um MTJ, que contém duas camadas ferromagnéticas separadas por uma camada isoladora de óxido. A direção da magnetização de uma camada ferromagnética é fixa enquanto que a outra camada ferromagnética pode ser alterada pela passagem de uma corrente. A resistência do MTJ é determinada pela direção de magnetização relativa dessas duas camadas. Se as duas camadas têm direções diferentes, a resistência do MTJ é alta e vice-versa. Usando essa propriedade, um valor binário é armazenado em uma célula STT-RAM. Para ler o valor armazenado, uma pequena tensão é aplicada entre os terminais MTJ. A corrente que flui através do dispositivo é detectada, e o estado de magnetização é determinado como um resultado.

Os dispositivos não voláteis MTJs são fabricados diretamente no circuito CMOS, devido à conexão estreita entre o

circuito lógico e a memória em comparação com o circuito lógico convencional com SRAM, a lógica não volátil baseada em MTJ alcança não só a redução na área e energia de E/S, mas também a melhoria da velocidade de transferência de dados [6]. No trabalho de Meena [7] os autores afirmam que a STT-RAM é um tipo de RAM magnética com os seguintes recursos: tempos de leitura e gravação rápidos, tamanhos de células pequenas, potencialmente menores e compatibilidade com DRAM e SRAM existentes. A STT-RAM é uma tecnologia mais adequada para o futuro da MRAM produzida usando processos ultrafinos e pode ser incorporado eficientemente em gerações subsequentes de tais dispositivos semicondutores como FPGAs, microprocessadores, microcontroladores e SoC's.

Embora a STT-RAM possui densidade menor do que a PCM e a RRAM, e maior latência e energia para a operação de escrita do que a SRAM, essa memória foi amplamente utilizada para a concepção de caches devido a seu alto *endurance*. O *endurance* é o número de ciclos de gravações que podem ser aplicados a um bloco de memória flash, antes que a mídia de armazenamento torne-se inconfiável. No entanto, apesar de um valor de *endurance* de 10^{15} foi estimado, o melhor resultado do teste de *endurance* até agora é inferior a $4x10^{12}$ [8]. Outra vantagem da STT-RAM é que a sua não-volatilidade pode ser negociada para melhorar sua energia de gravação e latência.

Em Smullen [5], os autores modificam o tempo de retenção encolhendo a área planar do MTJ, enquanto que o trabalho de Jog [9] consegue isto diminuindo a espessura da camada livre e baixando a saturação da magnetização que reduz a barreira térmica do MTJ. Como exemplo, em Jog [9] mostra que para a frequência de 2 GHz, os valores de latência de gravação de uma STT-RAM de 4 MB para períodos de retenção de 10 anos, 1 segundo e 10 milissegundos são respectivamente 22, 12 e 6 ciclos. Assim, com base na característica da aplicação e no nível da hierarquia da cache, um designer pode escolher um valor apropriado de período de retenção.

Segundo Meena [7] afirma que para a utilização da STT-RAM como uma memória semicondutora universal, os desafios principais de baixa corrente de comutação e alta estabilidade térmica precisam ser resolvidos simultaneamente. Esta memória precisa ser densa (aproximadamente $10F^2$, rápida (abaixo de 10 ns de leitura e gravação) e operar em baixa potência [10]. De acordo com Endoh [6] a STT-RAM é a candidata mais promissora de memória de trabalho não volátil devido às virtudes nativas do MTJ, baixa voltagem, alta velocidade e *endurance* praticamente infinita. Entretanto esta memória possui como desafios remanescentes para um maior dimensionamento, a redução da corrente de comutação, a retenção estável de dados, a obtenção de baixa resistência e a margem operacional mais ampla (aumento o MR *ratio*).

B. FeRAM

Esta memória não volátil emergente inicialmente usava capacitores ferroelétricos como um dispositivo não volátil combinado com um circuito CMOS [6]. No entanto, o FeRAM tem dificuldade em dimensionar o tamanho da célula da memória, pois não pode ocorrer uma reação entre os materiais ferroelétricos e os materiais dos eletrodos. Como resultado, as aplicações mais adequadas para FeRAM são um pouco limitadas, para cartões IC inteligentes e microcontroladores de pequena escala com uma pequena capacidade de armazenamento.

Segundo Meena [7] a FeRAM mantém os dados sem qualquer fonte de alimentação externa, pois usa um material ferroelétrico no lugar de um material dielétrico convencional entre as placas do capacitor. A desvantagem da FeRAM é que tem um ciclo de leitura destrutiva. Espera-se a utilização da FeRAM em pequenos dispositivos de consumo, como assistentes digitais pessoais (PDAs), telefones portáteis, cartões inteligentes e em sistemas de segurança. A FeRAM possui como expectativa substituir a EEPROM (*Electrically-Erasable Programmable Read-Only Memory*) e a SRAM em algumas aplicações, tornando-se um componente-chave nos futuros produtos sem fio. Os atuais chips FeRAM oferecem desempenho que é comparável ou superior às memórias flash atuais, mas ainda mais lento do que a DRAM.

C. RRAM

A RRAM é um dispositivo simples de dois terminais de *metal-insulador-metal* (MIM), existe dois estados de condutividade distintos, sendo cada estado induzido pela aplicação de diferentes tensões nos terminais do dispositivo [7]. Uma RRAM com comutação unipolar usa um dielétrico isolador [11]. Quando é aplicada uma tensão suficientemente elevada, é formado um filamento ou um percurso condutor no dielétrico isolador. Depois disso, aplicando tensões adequadas, o filamento pode ser ajustado para *set* (o que leva a uma baixa resistência) ou *reset* (o que leva a uma alta resistência).

Entre todas as tecnologias de memória atuais, a RRAM está atraindo muita atenção, pois é compatível com os processos convencionais de semicondutores [7]. A RRAM oferece o potencial de ser uma memória simples e barata podendo competir em todo o espectro de memórias digitais, desde aplicativos de baixo custo e de baixo desempenho até memórias universais capazes de substituir todas as tecnologias atuais líderes do mercado, como unidades de disco rígido, memórias de acesso aleatório e memórias flash [12]. Já em Xu [13] a RRAM é considerada a memória mais promissora, pois opera mais rápido que a PCM e possui uma estrutura de célula mais simples e menor do que as memórias magnéticas (MRAM ou STT-RAM). Comparado a SRAM, uma cache RRAM tem alta densidade, latência de leitura comparável e possui um valor muito inferior de *leakage*.

No entanto, as limitações da RRAM são a retenção de dados, baixo *endurance* [14], cerca de 10^{11} [15], alta latência e consumo energético para as operações de escrita. Por exemplo, uma cache típica de 4 MB de RRAM tem uma latência de leitura entre 6 e 8 nanosegundos e uma latência de escrita entre 20 e 30 nanosegundos [16]. Segundo Goux [17] o uso de uma estrutura de RRAM empilhada mostrou ser um dos métodos mais promissores para melhorar as

características da memória. Um tempo de retenção de dados de mais de 10 anos podem ser extrapoladas a partir de características de retenção medidas em altas temperaturas e um *endurance* de memória de mais de 10^6 ciclos [18].

Segundo Endoh [6] existem vários tipos de princípio de armazenamento para ReRAM podendo ser categorizados em dois tipos. Uma é a RAM da ponte condutora (CBRAM: *Conductive Bridge RAM*) e a outra é a RAM de oxigênio (OxRAM: *oxide RAM*). Ambas utilizam a reação *redox* em seu princípio operacional [19], [20]. A RRAM poderá cobrir a área de aplicação de armazenamento onde o NAND flash atualmente ocupa, se a formação do dispositivo no orifício de via e/ou na camada múltipla for conseguida [21], [22].

D. PCM

A PCM usa um material de mudança de fase chamado GST, que é uma liga de germânio, antimônio e telúrio. Quando a liga é aquecida a uma temperatura muito alta e rapidamente resfriada, transita em uma substância amorfa com alta resistência elétrica (*reset*). Por outro lado, quando a liga é aquecida a uma temperatura entre a cristalização e o ponto de fusão e resfriada lentamente, cristaliza até um estado físico com menor resistência (*set*). Para a operação *set*, quando o GST é aquecido a uma temperatura entre a temperatura de cristalização ($\sim 300\,^{\circ}\mathrm{C}$) e a temperatura de fusão ($\sim 600\,^{\circ}\mathrm{C}$) durante um período de tempo, GST transforma-se no estado cristalino que corresponde ao '1' lógico. Para a operação *reset*, quando o GST é aquecido acima do ponto de fusão e extinguido rapidamente, GST transforma-se no estado amorfo que corresponde ao '0' lógico. Esta propriedade física é usada para armazenar um valor binário em uma célula PCM. A estrutura básica de uma célula da PCM consiste em um transistor NMOS e um dispositivo de mudança de fase.

Os dois desafios mais graves no uso de PCM para projetar caches *on-chip* são sua resistência limitada de escrita e alta latência de gravação. Uma vez que o tráfego de escrita para uma cache é muito mais pesado do que para uma memória principal e a *endurance* da PCM é apenas perto de 10^8 escritas [23] e [24], para diversas aplicações, uma cache que utiliza PCM pode falhar em menos de uma hora. Uma típica cache de 4 MB utilizando PCM possui uma latência de leitura entre 15 e 20 nanosegundos e uma latência de escrita entre 150 e 170 nanosegundos [16] e [24].

Segundo Meena [7] a PCM é mais rápida e possui um menor consumo energético do que a memória flash. Além disso, a tecnologia PCM tem o potencial de fornecer armazenamento não volátil de alto volume, alta velocidade e alta densidade. A estrutura física é tridimensional, maximizando o número de transistores que podem existir em um chip de tamanho fixo. A desvantagem da PCM é a alta corrente necessária para apagar a memória, entretanto, conforme o tamanho das células diminuem a corrente necessária também diminui.

De acordo com Endoh [6] a PCRAM é adequada para a área de aplicação do armazenamento do programa, onde a memória flash NOR foi tradicionalmente usada. No entanto, possui um dificuldade significativa na estabilidade ou confiabilidade da operação por causa da temperatura entre as células de memória adjacentes à medida que a tecnologia se reduz, pois usa fenômeno de transição de fase aplicando o calor de Joule como princípio de operação não-volátil. Se a resistência da ordem 10^{15} for alcançada, poderá substituir a DRAM.

E. DWM

A DWM funciona controlando a parede do domínio (DW: *Domain Wall*) em nanofios ferromagnéticos [25]. O fio ferromagnético pode ter múltiplos domínios que são separados por paredes de domínio. Esses domínios podem ser individualmente programados para armazenar um único bit (na forma de uma direção de magnetização) e assim, o DWM pode armazenar múltiplos bits por célula de memória.

Logicamente, uma macrocélula DWM aparece como uma fita, que armazena múltiplos bits e pode ser deslocada em qualquer direção. O desafio na utilização de DWM é que o tempo consumido no acesso a um bit depende da sua localização em relação à porta de acesso, o que leva a uma latência de acesso não uniforme e torna o desempenho dependente do número de operações de deslocamento necessárias por acesso. Comparado com outras NVM's, DWM é mais recente, e ainda está em fase de pesquisa e protótipos.

F. Comparativo entre as Memórias

Em um sistema embarcado genérico, o sistema de memória principal pode estar contido parcialmente dentro do chip *on-chip* e fora do chip *off-chip* [2]. As memórias *on-chip* e *off-chip* influenciam o desempenho e o consumo de energia em sistemas embarcados. No modelo on-chip as SPM's podem ser utilizadas juntamente com memórias caches, como memórias de alta velocidade. O tipo tradicional de memória empregada nas SPMs e caches são as memórias SRAMs.

Com os avanços da tecnologia CMOS, com a fabricação de transistores cada vez menores, o *leakage* da SRAM cresceu consideravelmente, resultando em uma parte significativa do consumo energético total em diversos chips de semicondutores [26]. Comparando com a memória tradicional SRAM, as memórias emergentes STT-RAM e PCRAM proporcionam um *leakage* menor e uma maior densidade. Comparando-as entre si, STT-RAM possui uma menor latência de acesso e energia dinâmica, enquanto PCRAM possui maior densidade.

Segundo Banakar [27] uma cache utilizando memória SRAM normalmente possui um consumo energético entre 25% a 45% do consumo total do chip. Os autores utilizaram uma SPM com o objetivo de reduzir o consumo de energia e de área, os resultados apresentaram uma redução de 40% no consumo energético e uma redução em média de 46% na área. Assim, a memória SPM, tem sido amplamente adotada em muitos sistemas embarcados devido a sua menor área e menor consumo energético. Além disso, a SPM pode proporcionar muitas vezes uma melhor previsibilidade e sincronização em dispositivos de tempo real, por ser uma memória gerenciada por software [28].

Ao contrário de memórias baseadas em carga, como DRAM, NVMs armazenam seus dados através da mudança do estado físico. Uma vez que uma operação de escrita para NVM envolve a mudança de seu estado físico, uma operação de gravação para NVM possui maior latência e consumo energético do que uma operação de leitura, levando a assimetria leitura-escrita [29]. Da mesma forma, a latência de escrita e o consumo energético da transição da lógica 1 para 0 é maior do que a de 0 para 1, levando a assimetria de escrita de 0/1 [30]. As NVMs também permitem o armazenamento de múltiplos bits de dados em uma única célula de memória. Isto é referido como armazenamento de células de múltiplos níveis (MLC: *Multi-Level Cell*) e conduz a um aumento significativo na densidade de armazenamento suportado por estas memórias.

Segundo Wu [31] e Sun [32] a memória STT-RAM é mais adequada para memórias de último nível, enquanto que a PCRAM é promissora como uma alternativa à DRAM na memória principal [33]. Embora a STT-RAM possui densidade menor do que a PCM e a RRAM, e maior latência e energia para a operação de escrita do que a SRAM, essa memória foi amplamente utilizada para a concepção de caches devido a seu alto *endurance*. Enquanto que a PCM é adequada para a memória principal ou hierarquias inferiores de cache, por exemplo, cache L3 ou mesmo cache L4 [31], onde a sua latência elevada pode ser tolerada e a alta densidade pode ser utilizada para evitar acessos fora do chip.

Embora STT-RAM apresente muitas características atraentes, como baixo *leakage* e alta densidade, esse tipo de memória possui alguns problemas. Ao contrário da SRAM, na qual operações de leitura e escrita consomem o mesmo tempo e energia, uma operação de gravação na STT-RAM requer muito mais tempo de latência e maior energia do que uma operação de leitura. Além disso, a latência e a energia de operações de escrita convencionais em uma STT-RAM são várias vezes maiores do que os da SRAM para um mesmo tamanho de memória.

Novos modelos de STT-RAM foram propostos para diminuir os problemas envolvidos com as operações de escrita. Segundo Amiri [34] e Tadisina [35], as PMTJ (*Perpendicular Magnetic Tunnel Junction*) foram desenvolvidas para alcançar uma baixa corrente de comutação, mantendo uma alta estabilidade térmica para as STT-RAM. Segundo Fujita [26] o modelo de memória p-STT-RAM (*Perpendicular Spin-Transfer Torque Magnetic Random-Access Memory*) possui uma maior probabilidade para substituir a SRAM do que a STT-MRAM, pois a p-STT-RAM apresenta uma maior velocidade de acesso e *endurance*. Já em Xu [36], os autores conseguiram diminuir significativamente os problemas de gravação em STT-RAM SPM, graças a um cuidadoso processo de cootimização entre os dispositivos da arquitetura.

A propriedade comum a todas as NVM's é que sua latência/energia de escrita é significativamente maior do que a latência/energia de leitura. Além disso, em condições normais, as NVM's podem reter dados durante vários anos sem a necessidade de qualquer energia de *standby* [37].

Na Tabela I apresenta uma proposta de classificação para memórias não-voláteis. São elencados os trabalhos que utilizam memórias não-voláteis para as diferentes abordagens, de acordo com as classificações propostas.

TABLE I
CLASSIFICAÇÃO DE MEMÓRIAS NÃO-VOLÁTEIS

Classificação	**Referências**
SPM	[38], [39], [40], [41], [42]
Cache	[43], [44], [45], [46], [47], [48], [49], [50]
Memória principal	[51],[52],[53],[54], [55], [56], [57], [58], [59], [60], [61]
Economia de energia	[38], [39], [40], [41], [62], [63], [64], [65]
Melhoria de desempenho	[38], [39], [41]
STT-RAM	[38], [47], [41],[42]
PCM	[39]
MRAM	[40]
Z-RAM	[40]
Algoritmos Propostos	[38], [39], [40], [41], [42]

III. TRABALHOS RELACIONADOS COM NVM'S

O emprego de NVM's tem sido investigado em diferentes níveis da hierarquia de mémoria por diversos trabalhos. Nos trabalhos de Mittal [66] e [37], os autores revisam diversas propostas de otimizações em caches e memórias principais, entretanto não abordam nenhuma utilização em SPMs. As SPM's empregando NVM's são investigadas em [38] e [39], comparando-as com tecnologias tradicionais de memória.

A. SPM's

Em Wang [38], os autores investigaram primeiramente a substituição de uma memória SPM empregando SRAM por uma SPM baseada em STT-RAM. Posteriormente, os autores também avaliaram uma SPM híbrida (SRAM+STT-RAM), onde os dados mais escritos são alocados na SRAM e os dados mais lidos são alocados na STT-RAM. Na abordagem híbrida, foram utilizadas diferentes áreas (proporções) de SRAM e STT-RAM. Para a realização do trabalho, utilizaram uma plataforma de simulação construída sobre a ferramenta SIMICS [67] juntamente com o GEMS. Os autores utilizaram o algoritmo guloso de Udayakumaran [68] para gerenciar a alocação de dados para a SPM com SRAM, já para a alocação da abordagem híbrida os dados mais escritos são alocados na SRAM e os dados mais lidos na STT-RAM. Para a abordagem híbrida os autores fixaram a área de uma SPM de 64KB SRAM como linha base para realizarem a comparação de diversas proporções. Por exemplo, os autores utilizaram a proporção de 1:1 de área para a SPM híbrida, o que significa que a SPM híbrida possui 32KB de SRAM e 128KB de STT-RAM, enquanto o SPM linha de base tem 64 KB de SRAM. Segundo Wang [38] a memória STT-RAM pode ser projetada cerca de quatro vezes mais densa que a memória SRAM, estas duas SPMs de densidades diferentes possuem uma área total semelhante de silício, devido a variação na área requerida por cada tecnologia. Na comparação do produto do desempenho e consumo energético

para as diversas proporções da SPM híbrida, a proporção 2:1 (SRAM:STT-RAM) apresentou os melhores resultados, cerca de 45% em média. Pelo trabalho de Wang [38], percebe-se que, através das explorações realizadas juntamente com as otimizações demonstradas, que a arquitetura SPM híbrida pode superar SPM's constituídas somente por SRAM ou STT-RAM. Devido as características atraentes da STT-RAM, como o baixo leakage e a alta densidade, esta é apontada como uma memória promissora para modelos híbridos de memória ou para a substituição da SRAM.

No trabalho de Hu [39], uma SPM tradicional baseada em SRAM também foi comparada com uma SPM híbrida, porém empregando como memória emergente uma PCM. Ainda nesse trabalho, os autores exploraram diferentes algoritmos para a alocação dos endereços nos tipos de memórias utilizadas, propondo novos algoritmos para alocação em memórias híbridas. De acordo com os resultados experimentais, na comparação dos algoritmos propostos MURDA e DAHS com o algoritmo Udayakumaran [68] para todos os benchmarks, o algoritmo do Udayakumaran pode terminar em menos de 1 min enquanto que os algoritmos de MURDA e DAHS juntos podem terminar em menos de 10 min. Entretanto os algoritmos propostos podem reduzir em média de 17,19% o tempo de acesso à memória, 20,84% a energia dinâmica, 76.66% no número de gravações na NVM. Segundo Hu [39] as NVM normalmente possuem um número limitado de gravações, diminuindo o número de gravações sobre a NVM, estende-se a vida útil da memória. Com a redução de 76,66% no número de gravações da NVM, o método proposto pelos autores pode estender a vida útil da NVM por mais de quatro vezes. Com a ajuda dos algoritmos propostos pelos autores, o modelo híbrido de arquitetura SPM reduziu o tempo de acesso à memória em 18,17%, a energia dinâmica em 24,29%, e o leakage em 37,34% quando comparada com uma SPM contendo SRAM com a mesma área.

Os trabalhos citados, Hu [39] e Wang [38], utilizaram o NVsim [16] com tecnologia 45nm, para a obtenção dos parâmetros das memórias PCM, STT-RAM e SRAM, como as latências de acesso e o consumo energético para as operações de escrita e leitura. Os benchmarks selecionados foram retirados do MiBench [69]. Os trabalhos conseguiram reduzir significativamente o consumo energético e melhorar o desempenho para diversos benchmarks executados em um processador ARM, através do emprego de tecnologias emergentes de memória em SPMs.

No trabalho de Qiu [40] os atores exploraram uma SPM configurada com SRAM, MRAM e Z-RAM (*Zero-capacitor RAM*) para sistemas embarcados multicores. A arquitetura utilizada nos sistema multicores com SPMs híbridos consiste em que cada núcleo é fortemente acoplado a uma SPM no chip empregando uma SRAM, uma MRAM e uma Z-RAM. Todos os núcleos acessam a memória principal *off-chip* (geralmente um dispositivo DRAM) através de um barramento compartilhado. Existe uma estrutura de anel multicanal para permitir a comunicação entre dois núcleos sem intervenção de outros núcleos. Geralmente, o acesso para uma SPM local é mais rápido e o consumo energético é mais baixo, do que buscar dados em outras SPM's, enquanto o acesso à memória principal *off-chip* proporciona uma latência mais alta e maior consumo energético. Para alocar eficientemente dados em cada memória das SPM's são propostos dois algoritmos, um é o algoritmo dinâmico multidimensional denominado MDPDA e o outro é um algoritmo genético adaptativo chamado AGADA. Os benchmarks utilizados foram retirados do PARSEC [70] e executados através do simulado M5 [71]. Também é utilizado uma versão modificada do CACTI [72] para obter os parâmetros das memórias, incluindo a latência de leitura/gravação, consumo de energia e *leakage* para a tecnologia de 65nm. Os resultados demonstraram que comparando o algoritmo MDPDA com o Udayakumaran [68], o algoritmo MDPDA pode reduzir em média 24,18% o consumo de energia dinâmica para um sistema quad-core. A partir de outros resultados, de acordo com os autores é possivel observar que a redução no consumo de energia é proporcional à redução na latência de acesso à memória. Esta redução é principalmente devido à alocação ótima do MDPDA para cada bloco de dados. Já os resultados do algoritmo AGADA para consumo dinâmico de energia mostraram que são aproximados ao MDPDA. O algoritmo AGADA consome em média 2,21% a mais de energia dinâmica do que o MDPA, entretanto, segundo Qiu [40] devido a alta complexidade do MDPDA o algoritmo AGADA é mais competitivo no desempenho geral.

No trabalho de Rodriguez [41] os autores exploraram uma SPM com STT-RAM. O trabalho baseia-se nas células voláteis STT-RAM propostas por Smullen [5] para implementar uma nova organização da memória de SPM contendo diferentes regiões com características distintas de retenção, energia e desempenho. Ao relaxar a não volatilidade da STT-RAM, as características da memória são melhoradas, como a latência, energia dinâmica, *leakage* e densidade. Possibilitando que a STT-RAM seja uma candidata promissora para a implementação de memórias *on-chip*. No trabalho é utilizado um algoritmo de alocação baseado em compilador para essa organização de SPM. Os dados de curta duração são trazidos para regiões rápidas e de baixa energia em *on-chip*. Os dados com duração mais longa são acomodados em regiões com maior retenção, capazes de explorar plenamente a sua localidade. É utilizado o sistema de modelagem e simulação STeTSiMS [73] para o design das memórias, fornecendo os parâmetros de desempenho, energia e área. Este simulador é configurado para selecionar automaticamente parâmetros arquitetônicos ótimos, otimizando a célula para diferentes objetivos de design. A célula de linha de base utilizada é um design normalizado por Smullen [73], com um aumento do tamanho do MTJ para $36F^2$ com 32nm, assegurando a retenção de 10 anos em 350K, desempenho típico de um microprocessador. A arquitetura de memória consite em uma memória cache, uma memória SPM e uma memória principal. A cache e SPM estão localizadas *on-chip* e a memória principal é assumida como uma DRAM *off-chip* possuindo uma maior latência de acesso. As experiências foram realizadas utilizando o simulador GEM5 [74], no modo de *syscall emulation*. O

sistema de memória foi modificado para incluir uma SPM. O modelo de processador usado, simula um pipeline, CPU *out-of-order* e com uma frequência de 2 GHz. A latência de acesso L1 é 1 ciclo e L2 é de 3 ciclos. As latências para SPM's simuladas e caches STT-RAM variam entre 1-3 ciclos para leituras e 3-9 ciclos para escritas, dependendo do tamanho da memória e da volatilidade. Todas as memórias simuladas possuem uma única porta de leitura/gravação. Um protocolo de coerência MESI é usado para configurações com L2 compartilhada. Os benchmarks foram retirados do SPEC CPU2006 [75] (*bwaves , cactusADM , leslie3d , e hmmer*), Mantevo [76] (*HPCCG e miniMD*), Mediabench [77] (*gsm e adpcm*) e PARSEC [70] (*canneal e streamcluster*). Cada *kernel* foi configurado para simular pelo menos 10^{10} ciclos. A avaliação experimental mostra que o projeto oferece economias superiores a 60% no consumo energético, quando comparado com diferentes projetos de memória de *on-chip* de área semelhante. O projeto fornece economias de 63% no consumo da energia dinâmica em comparação com uma SPM com STT-RAM não volátil. Quando comparado com cache SRAM, o projeto executa até 28,5% mais rápido e economiza 53% do consumo energético. Ainda os autores afirmam que os resultados podem ser melhorados através de otimizações, como desligar os bancos da SPM não utilizados durante os períodos *idle*. Além disso, o uso de SPM pode melhorar o consumo da energia dinâmica das cargas de trabalho *multithreaded*, reduzindo o compartilhamento falso.

Em Wang [42] os autores utilizaram uma STT-RAM em uma SPM para sistemas em tempo real com abordagem WCET (*Worst-case execution time*) em um ambiente multitarefa, com *tradeoffs* entre a utilização da CPU e a vida útil do sistema devido a restrições do *endurance* da NVM. Também usaram algoritmos para alocação de dados para SPM capazes de distribuir o número de gravações uniformemente, de modo a alcançar o nivelamento do desgaste e prolongar a vida útil de NVM. São utilizados dois algoritmos de otimização para minimizar a utilização da CPU do sistema, sujeitos às restrições de tempo de vida da NVM. O primeiro é um algoritmo ótimo baseado em ILP e o outro é um algoritmo heurístico eficiente que pode obter soluções próximas a ótimas. Os parâmetros do sistema de memória são obtidos a partir de Monazzah [78]. São considerados sistemas embarcados com recursos limitados baseados em microcontroladores de 8 ou 16 bits de baixo custo, com armazenamento limitado *on-chip* de memória que pode ser usado como cache ou SPM. A plataforma alvo consiste em uma CPU de 1GHz, as memórias possuem tamanhos de 2KB para NVM e 1KB para a SRAM. A latência de leitura para a NVM e SRAM possuem 1 ciclo, a latência de escrita para SRAM é de 1 ciclo e para a NVM são 10 ciclos. A latência de leitura/escrita para a memória principal são 50 ciclos. Os *benchmarks* foram retirados do Malardalen WCET [79] (*minver, adpcm,ludcmp, ndes, fir* e *edn*). Os *benchmarks* são compilados com a ferramenta *SimpleScalar* com a opção de otimização "-O2". Usa-se a ferramenta Chronos [80] para obter WCET de cada *benchmark* e obter o número de acessos de leitura/escrita para memória. São considerados seis restrições de vida diferentes (1, 2, 4, 8, 16 e 32 anos). Pelos experimentos é verificado que a utilização da CPU para a SPM com NVM de 2 KB é inferior do que para SPM com SRAM de 1 KB, principalmente quando a vida útil do sistema é curta. Segundo Wang [42] geralmente o WCET de um programa aumenta quando a vida se torna mais longa, pois mais variáveis precisam ser deslocadas da SPM para a memória principal para reduzir a pressão de gravação em SPM com STT-RAM. A utilização da CPU aumenta de 0,22 (1 ano de vida) para 0,38 (32 anos de vida). A utilização da CPU aumenta em 14% quando o tempo de vida aumenta de 1 ano para 2 anos e em 38% quando o tempo de vida aumenta de 1 ano a 4 anos. Apesar do algoritmo ILP e heurístico apresentarem resultados semelhantes para a utilização total da CPU, eles podem produzir valores diferentes de WCET. Para os *benchmarks fir* e *edn* a maioria das variáveis possuem um grande número de leituras, portanto, muitas variáveis podem ser alocadas na SPM STT-RAM com 2KB, ocasionando valores inferiores de WCET para a SPM com NVM do que a SPM com SRAM de 1 KB.

B. Caches

No trabalho de [43]...
No trabalho de [44]...
No trabalho de [45]...
No trabalho de [46]...

Em Komalan [47], os autores exploraram o impacto da substituição de uma STT-MRAM por uma memória SRAM em uma cache de dados L1. Inicialmente observou-se uma sobrecarga de 54% nas penalidades de desempenho nos benchmarks analisados. Após foram realizadas algumas modificações na arquitetura da cache de dados empregada, através da utilização de um VWB (*Very Wide Buffer*) entre a STT-MRAM e o processador. Também realizaram otimizações de software, como transformações e otimizações de código e exploraram o paralelismo através de vetorizações, alinhamentos de loop, e reduções do número de decisões de desvios. Para avaliar a eficácia das modificações propostas, os autores implementaram o projeto via o simulador GEM5 [74] e utilizaram um conjunto de benchmark do PolyBench. Através de experimentos, verificaram a influência do tamanho do VWB, na redução das penalidades. Os experimentos apontaram que ao aumentar o tamanho do VWB, reduz-se as penalidades. Entretanto, aspectos de custo e energia associados ao aumento do VWB também devem ser considerados, assim como o roteamento do layout, pois o layout do projeto torna-se mais complexo a medida em que o VWB é aumentado. Assim, os autores indicaram que o tamanho ideal para o VWB é cerca de 2KBits. As mesmas transformações são realizadas em uma SRAM, esta solução apresentou cerca de 8% de melhoramento no desempenho em comparação com a STT-MRAM. Entretanto, vale destacar que essa comparação não leva em conta as vantagens óbvias oferecidas pela cache com STT-MRAM, como uma área e gasto energético menores e não volatilidade. Após as explorações realizadas juntamente com as transformações e otimizações de código verifica-se a redução da penalidade

de desempenho introduzido pela utilização da NVM que inicialmente era de 54% para níveis toleráveis, cerca de 8%, mesmo para os piores casos testados.

No trabalho de [48]...

No trabalho de [49]...

No trabalho de [50]...

C. Memória Principal

Esta seção tem por objetivo revisar trabalhos que utilizam a tecnologia das memórias não-voláteis como memória principal.

Estas memórias são boas candidatas para serem utilizadas no lugar de memórias DRAM como memória principal pelas suas características de consumo de energia e ausência de volatilidade. No entanto, além das vantagens é preciso considerar suas desvantagens, tais como tempo de vida limitado pelo número de reescritas e também o desempenho assimétrico entre as operações de leitura e escrita.

No trabalho de Kim [51], para adotar esta nova tecnologia de memória os autores discutem um novo sistema de arquivos baseado em *tmpfs* e avaliam seu desempenho em sistemas de arquivos populares como o *ext4*. Os autores acreditam que sistemas de arquivos semelhantes ao *tmpfs* são perfeitos para sistemas de arquivos em memória não-volátil. Neste trabalho, o sistema de arquivo em memória pode armazenar suas informações constantemente na memória principal utilizando um dispositivo de memória não volátil como memória principal, ou seja, a memória principal e o sistema de arquivos podem compartilhar espaço em um mesmo dispositivo. Para avaliar o desempenho da abordagem proposta os autores executaram experimentos com o *sysbench* em três diferentes ambientes de sistemas de arquivo a fim de comparar latência de leitura e escrita entre os diferentes sistemas de arquivo. Os autores também executaram o *filebench* para avaliar como o mecanismo de *swap* trabalha com *tmpfs*.

No trabalho de [52]...

No trabalho de [53]...

No trabalho de [55]...

No trabalho de [56]...

No trabalho de [57]...

No trabalho de [58]...

No trabalho de [59]...

No trabalho de [60]...

No trabalho de [61]...

IV. CONCLUSÕES

Redigir conclusões.....

REFERENCES

[1] John L. Hennessy and David A. Patterson. *Computer architecture: a quantitative approach*. Elsevier, 2011.

[2] W. Wolf and M. Kandemir. Memory system optimization of embedded software. *Proceedings of the IEEE*, 91(1):165–182, 2003.

[3] Marilyn Wolf. *Computers as components: principles of embedded computing system design*. Elsevier, 2012.

[4] Yole Développement. Storage-class memory will be the clear go-to market for emerging non-volatile memory in 2021. [Online]. Available: http://www.yole.fr/Emerging_NVM_Market.aspx#.WYEepojyvDf, 28 July 2016.

[5] C. W. Smullen, V. Mohan, A. Nigam, S. Gurumurthi, and M. R. Stan. Relaxing non-volatility for fast and energy-efficient stt-ram caches. In *High Performance Computer Architecture (HPCA), 2011 IEEE 17th International Symposium on*, pages 50–61. IEEE, 2011.

[6] T. Endoh, H. Koike, S. Ikeda, T. Hanyu, and H. Ohno. An overview of nonvolatile emerging memories—spintronics for working memories. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 6(2):109–119, 2016.

[7] J. S. Meena, S. M. Sze, U. Chand, and T.-Y. Tseng. Overview of emerging nonvolatile memory technologies. *Nanoscale research letters*, 9(1):526, 2014.

[8] Y. Huai. Spin-transfer torque mram (stt-mram): Challenges and prospects. *AAPPS bulletin*, 18(6):33–40, 2008.

[9] A. Jog, A. K. Mishra, C. Xu, Y. Xie, V. Narayanan, R. Iyer, and C. R. Das. Cache revive: architecting volatile stt-ram caches for enhanced performance in cmps. In *Proceedings of the 49th Annual Design Automation Conference*, pages 243–252. ACM, 2012.

[10] D. Apalkov, A. Khvalkovskiy, S. Watts, V. Nikitin, X. Tang, D. Lottis, K. Moon, X. Luo, E. Chen, A. Ong, et al. Spin-transfer torque magnetic random access memory (stt-mram). *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 9(2):13, 2013.

[11] H. Li and Y. Chen. An overview of non-volatile memory technology and the implication for tools and architectures. In *Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE'09.*, pages 731–736. IEEE, 2009.

[12] S. Kim. Resistive ram (reram) technology for high density memory applications. In *4th Workshop Innovative Memory Technol MINATEC 2012; June 21-24 2012*, 2012.

[13] C. Xu, X. Dong, N. P. Jouppi, and Y. Xie. Design implications of memristor-based rram cross-point structures. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011*, pages 1–6. IEEE, 2011.

[14] A. Sawa. Resistive switching in transition metal oxides. *Materials today*, 11(6):28–36, 2008.

[15] Y.-B. Kim, S. R. Lee, D. Lee, C. B. Lee, M. Chang, J. H. Hur, M.-J. Lee, G.-S. Park, C. J. Kim, U.-I. Chung, et al. Bi-layered rram with unlimited endurance and extremely uniform switching. In *VLSI Technology (VLSIT), 2011 Symposium on*, pages 52–53. IEEE, 2011.

[16] X. Dong, C. Xu, N. Jouppi, and Y. Xie. Nvsim: A circuit-level performance, energy, and area model for emerging non-volatile memory. In *Emerging Memory Technologies*, pages 15–50. Springer, 2014.

[17] L. Goux, A. Fantini, G. Kar, Y.-Y. Chen, N. Jossart, R. Degraeve, S. Clima, B. Govoreanu, G. Lorenzo, G. Pourtois, et al. Ultralow sub-500na operating current high-performance tin\al 2 o 3\hfo 2\hf\tin bipolar rram achieved through understanding-based stack-engineering. In *VLSI technology (VLSIT), 2012 symposium on*, pages 159–160. IEEE, 2012.

[18] Y. J. Seo, H. M. An, H. D. Kim, and T. G. Kim. Improved performance in charge-traptype flash memories with an al2o3 dielectric by using bandgap engineering of charge-trapping layers. *J Korean Phys Soc*, 55(6):2679–2692, 2009.

[19] H.-S. P. Wong, H.-Y. Lee, S. Yu, Y.-S. Chen, Y. Wu, P.-S. Chen, B. Lee, F. T. Chen, and M.-J. Tsai. Metal–oxide rram. *Proceedings of the IEEE*, 100(6):1951–1970, 2012.

[20] R. Waser, R. Dittmann, G. Staikov, and K. Szot. Redox-based resistive switching memories–nanoionic mechanisms, prospects, and challenges. *Advanced materials*, 21(25-26):2632–2663, 2009.

[21] A. Kawahara, R. Azuma, Y. Ikeda, K. Kawai, Y. Katoh, Y. Hayakawa, K. Tsuji, S. Yoneda, A. Himeno, K. Shimakawa, et al. An 8 mb multi-layered cross-point reram macro with 443 mb/s write throughput. *IEEE Journal of Solid-State Circuits*, 48(1):178–185, 2013.

[22] T. Liu, T. H. Yan, R. Scheuerlein, Y. Chen, J. K. Lee, G. Balakrishnan, G. Yee, H. Zhang, A. Yap, J. Ouyang, et al. A 130.7-mm^2 2-layer 32-gb reram memory device in 24-nm technology. *IEEE Journal of Solid-State Circuits*, 49(1):140–153, 2014.

[23] J. Wang, X. Dong, Y. Xie, and N. P. Jouppi. i2wap: Improving non-volatile cache lifetime by reducing inter-and intra-set write variations. In *High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on*, pages 234–245. IEEE, 2013.

[24] Y. Joo, D. Niu, X. Dong, G. Sun, N. Chang, and Y. Xie. Energy-and endurance-aware design of phase change memory caches. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010*, pages 136–141. IEEE, 2010.

[25] R. Venkatesan, V. Kozhikkottu, C. Augustine, A. Raychowdhury, K. Roy, and A. Raghunathan. Tapecache: a high density, energy

efficient cache based on domain wall memory. In *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*, pages 185–190. ACM, 2012.

[26] S. Fujita, H. Noguchi, K. Ikegami, S. Takeda, K. Nomura, and K. Abe. Technology trends and near-future applications of embedded stt-mram. In *Memory Workshop (IMW), 2015 IEEE International*, pages 1–5. IEEE, 2015.

[27] R. Banakar, S. Steinke, B.-S. Lee, M. Balakrishnan, and P. Marwedel. Scratchpad memory: design alternative for cache on-chip memory in embedded systems. In *Proceedings of the tenth international symposium on Hardware/software codesign*, pages 73–78. ACM, 2002.

[28] L. Li, L. Gao, and J. Xue. Memory coloring: A compiler approach for scratchpad memory management. In *Parallel Architectures and Compilation Techniques, 2005. PACT 2005. 14th International Conference on*, pages 329–338. IEEE, 2005.

[29] R. Bishnoi, F. Oboril, M. Ebrahimi, and M. B. Tahoori. Avoiding unnecessary write operations in stt-mram for low power implementation. In *Quality Electronic Design (ISQED), 2014 15th International Symposium on*, pages 548–553. IEEE, 2014.

[30] R. Bishnoi, M. Ebrahimi, F. Oboril, and M. B. Tahoori. Asynchronous asymmetrical write termination (aawt) for a low power stt-mram. In *Proceedings of the conference on Design, Automation & Test in Europe*, page 180. European Design and Automation Association, 2014.

[31] X. Wu, J. Li, L. Zhang, E. Speight, R. Rajamony, and Y. Xie. Hybrid cache architecture with disparate memory technologies. In *ACM SIGARCH computer architecture news*, volume 37, pages 34–45. ACM, 2009.

[32] G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen. A novel architecture of the 3d stacked mram l2 cache for cmps. In *High Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th International Symposium on*, pages 239–249. IEEE, 2009.

[33] B. C Lee, E. Ipek, O. Mutlu, and D. Burger. Architecting phase change memory as a scalable dram alternative. In *ACM SIGARCH Computer Architecture News*, volume 37, pages 2–13. ACM, 2009.

[34] P. K. Amiri, Z. M. Zeng, J. Langer, H. Zhao, G. Rowlands, Y.-J. Chen, I. N. Krivorotov, J.-P. Wang, H. W. Jiang, J. A. Katine, et al. Switching current reduction using perpendicular anisotropy in cofeb–mgo magnetic tunnel junctions. *Applied Physics Letters*, 98 (11):112507, 2011.

[35] Z. R. Tadisina, A. Natarajarathinam, B. D. Clark, A. L. Highsmith, T. Mewes, S. Gupta, E. Chen, and S. Wang. Perpendicular magnetic tunnel junctions using co-based multilayers. *Journal of Applied Physics*, 107(9):09C703, 2010.

[36] C. Xu, D. Niu, X. Zhu, S. H Kang, M. Nowak, and Y. Xie. Device-architecture co-optimization of stt-ram based memory for low power embedded systems. In *Proceedings of the International Conference on Computer-Aided Design*, pages 463–470. IEEE Press, 2011.

[37] S. Mittal and J. S. Vetter. A survey of software techniques for using non-volatile memories for storage and main memory systems. *IEEE Transactions on Parallel and Distributed Systems*, 27(5):1537–1550, 2016.

[38] P. Wang, G. Sun, T. Wang, Y. Xie, and J. Cong. Designing scratchpad memory architecture with emerging stt-ram memory technologies. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*, pages 1244–1247. IEEE, 2013.

[39] J. Hu, C. J. Xue, Q. Zhuge, W.-C. Tseng, and E. H.-M. Sha. Data allocation optimization for hybrid scratch pad memory with sram and nonvolatile memory. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 21(6):1094–1102, 2013.

[40] M. Qiu, Z. Chen, and M. Liu. Low-power low-latency data allocation for hybrid scratch-pad memory. *IEEE Embedded Systems Letters*, 6 (4):69–72, 2014.

[41] G. Rodríguez, J. Tourino, and M. T. Kandemir. Volatile stt-ram scratchpad design and data allocation for low energy. *ACM Transactions on Architecture and Code Optimization (TACO)*, 11(4):38, 2015.

[42] Z. Wang, Z. Gu, M. Yao, and Z. Shao. Endurance-aware allocation of data variables on nvm-based scratchpad memory in real-time embedded systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(10):1600–1612, 2015.

[43] J. Li, L. Shi, Q. Li, C. J. Xue, Y. Chen, Y. Xu, and W. Wang. Low-energy volatile stt-ram cache design using cache-coherence-enabled adaptive refresh. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 19(1):5, 2013.

[44] Q. Li, J. Li, L. Shi, M. Zhao, C. J. Xue, and Y. He. Compiler-assisted stt-ram-based hybrid cache for energy efficient embedded systems. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(8):1829–1840, 2014.

[45] K. Qiu, M. Zhao, Q. Li, C. Fu, and C. J. Xue. Migration-aware loop retiming for stt-ram-based hybrid cache in embedded systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(3):329–342, 2014.

[46] M. Komalan, J. I. G. Pérez, C. Tenllado, P. Raghavan, M. Hartmann, and F. Catthoor. Feasibility exploration of nvm based i-cache through mshr enhancements. In *Proceedings of the conference on Design, Automation & Test in Europe*, page 21. European Design and Automation Association, 2014.

[47] M. P. Komalan, C. Tenllado, J. I. G. Pérez, F. T. Fernández, and F. Catthoor. System level exploration of a stt-mram based level 1 data-cache. In *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, pages 1311–1316. EDA Consortium, 2015.

[48] Q. Li, Y. He, J. Li, L. Shi, Y. Chen, and C. J. Xue. Compiler-assisted refresh minimization for volatile stt-ram cache. *IEEE Transactions on Computers*, 64(8):2169–2181, 2015.

[49] C. Lin and J.-N. Chiou. High-endurance hybrid cache design in cmp architecture with cache partitioning and access-aware policies. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(10): 2149–2161, 2015.

[50] J. Ahn, S. Yoo, and K. Choi. Prediction hybrid cache: An energy-efficient stt-ram cache architecture. *IEEE Transactions on Computers*, 65(3):940–951, 2016.

[51] H. Kim, J. Ahn, S. Ryu, J. Choi, and H. Han. In-memory file system for non-volatile memory. In *Proceedings of the 2013 Research in Adaptive and Convergent Systems*, pages 479–484. ACM, 2013.

[52] J. Zhao, S. Li, D. H. Yoon, Y. Xie, and N. P. Jouppi. Kiln: Closing the performance gap between systems with and without persistence support. In *Microarchitecture (MICRO), 2013 46th Annual IEEE/ACM International Symposium on*, pages 421–432. IEEE, 2013.

[53] I. Moraru, D. G. Andersen, M. Kaminsky, N. Tolia, P. Ranganathan, and N. Binkert. Consistent, durable, and safe memory management for byte-addressable non volatile main memory. In *Proceedings of the First ACM SIGOPS Conference on Timely Results in Operating Systems*, page 1. ACM, 2013.

[54] T. Gao, K. Strauss, S. M. Blackburn, K. S. McKinley, D. Burger, and J. Larus. Using managed runtime systems to tolerate holes in wearable memories. In *ACM SIGPLAN Notices*, volume 48, pages 297–308. ACM, 2013.

[55] S. Kannan, A. Gavrilovska, K. Schwan, and D. Milojicic. Optimizing checkpoints using nvm as virtual memory. In *Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on*, pages 29–40. IEEE, 2013.

[56] J.-Y. Jung and S. Cho. Memorage: Emerging persistent ram based malleable main memory and storage architecture. In *Proceedings of the 27th international ACM conference on International conference on supercomputing*, pages 115–126. ACM, 2013.

[57] A. Sampson, J. Nelson, K. Strauss, and L. Ceze. Approximate storage in solid-state memories. *ACM Transactions on Computer Systems (TOCS)*, 32(3):9, 2014.

[58] B. Li, S.C. Shan, Y. Hu, and X. Li. Partial-set: write speedup of pcm main memory. In *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*, pages 1–4. IEEE, 2014.

[59] Z. Zhang, Z. Jia, P. Liu, and L. Ju. Energy efficient real-time task scheduling for embedded systems with hybrid main memory. *Journal of Signal Processing Systems*, 84(1):69–89, 2016.

[60] Q. Hu, G. Sun, J. Shu, and C. Zhang. Exploring main memory design based on racetrack memory technology. In *2016 International Great Lakes Symposium on VLSI (GLSVLSI)*, pages 397–402, May 2016.

[61] G. Wang, Y. Guan, Y. Wang, and Z. Shao. Energy-aware assignment and scheduling for hybrid main memory in embedded systems. *Computing*, 98(3):279–301, 2016.

[62] Sparsh Mittal, Jeffrey S. Vetter, and Dong Li. Improving energy efficiency of embedded dram caches for high-end computing systems. In *Proceedings of the 23rd International Symposium on High-performance Parallel and Distributed Computing*, HPDC '14, pages 99–110. ACM, New York, NY, USA, 2014. ISBN 978-1-4503-2749-7. URL `http://doi.acm.org/10.1145/2600212.2600216`.

[63] A. Agrawal, A. Ansari, and J. Torrellas. Mosaic: Exploiting the spatial locality of process variation to reduce refresh energy in on-

chip edram modules. In *2014 IEEE 20th International Symposium on High Performance Computer Architecture (HPCA)*, pages 84–95, Feb 2014. ISSN 1530-0897.

[64] J. Ahn and K. Choi. Lasic: Loop-aware sleepy instruction caches based on stt-ram technology. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 22(5):1197–1201, May 2014. ISSN 1063-8210.

[65] Y. Wang, L. Ni, C. H. Chang, and H. Yu. Dw-aes: A domain-wall nanowire-based aes for high throughput and energy-efficient data encryption in non-volatile memory. *IEEE Transactions on Information Forensics and Security*, 11(11):2426–2440, Nov 2016. ISSN 1556-6013.

[66] S. Mittal, J. S. Vetter, and D. Li. A survey of architectural approaches for managing embedded dram and non-volatile on-chip caches. *IEEE Transactions on Parallel and Distributed Systems*, 26(6):1524–1537, 2015.

[67] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hallberg, J. Hogberg, F. Larsson, A. Moestedt, and B. Werner. Simics: A full system simulation platform. *Computer*, 35(2):50–58, 2002.

[68] S. Udayakumaran and R. Barua. Compiler-decided dynamic memory allocation for scratch-pad based embedded systems. In *Proceedings of the 2003 international conference on Compilers, architecture and synthesis for embedded systems*, pages 276–286. ACM, 2003.

[69] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown. Mibench: A free, commercially representative embedded benchmark suite. In *Workload Characterization, 2001. WWC-4. 2001 IEEE International Workshop on*, pages 3–14. IEEE, 2001.

[70] Christian Bienia. *Benchmarking modern multiprocessors*. Princeton University, 2011.

[71] N. L. Binkert, R. G. Dreslinski, L. R. Hsu, K. T. Lim, A. G. Saidi, and S. K. Reinhardt. The m5 simulator: Modeling networked systems. *IEEE Micro*, 26(4):52–60, 2006.

[72] N. Muralimanohar, R. Balasubramonian, and N. Jouppi. Optimizing nuca organizations and wiring alternatives for large caches with cacti 6.0. In *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 3–14. IEEE Computer Society, 2007.

[73] C. W. Smullen, A. Nigam, S. Gurumurthi, and M. R. Stan. The stetsims stt-ram simulation and modeling system. In *Proceedings of the International Conference on Computer-Aided Design*, pages 318–325. IEEE Press, 2011.

[74] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, et al. The gem5 simulator. *ACM SIGARCH Computer Architecture News*, 39(2):1–7, 2011.

[75] J. L. Henning. Spec cpu2006 benchmark descriptions. *ACM SIGARCH Computer Architecture News*, 34(4):1–17, 2006.

[76] M. A. Heroux, D. W. Doerfler, P. S. Crozier, J. M. Willenbring, H. C. Edwards, A. Williams, M. Rajan, E. R. Keiter, H. K. Thornquist, and R. W. Numrich. Improving performance via mini-applications. *Sandia National Laboratories, Tech. Rep. SAND2009-5574*, 3, 2009.

[77] Chunho Lee, Miodrag Potkonjak, and William H Mangione-Smith. Mediabench: a tool for evaluating and synthesizing multimedia and communicatons systems. In *Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture*, pages 330–335. IEEE Computer Society, 1997.

[78] A. M. H. Monazzah, H. Farbeh, S. G. Miremadi, M. Fazeli, and H. Asadi. Ftspm: A fault-tolerant scratchpad memory. In *Dependable Systems and Networks (DSN), 2013 43rd Annual IEEE/IFIP International Conference on*, pages 1–10. IEEE, 2013.

[79] J. Gustafsson, A. Betts, A. Ermedahl, and B. Lisper. The mälardalen wcet benchmarks: Past, present and future. In *OASIcs-OpenAccess Series in Informatics*, volume 15. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2010.

[80] X. Li, Y. Liang, T. Mitra, and A. Roychoudhury. Chronos: A timing analyzer for embedded software. *Science of Computer Programming*, 69(1):56–67, 2007.