

cLUPA – Ampliador Digital de Documentos Impressos sobre uma Plataforma Multicore

Edmar A. Bellorini
Departamento de Informática
UFPR
Curitiba - Brasil
edmar.bellorini@gmail.com

Roberto A. Hexsel
Departamento de Informática
UFPR
Curitiba - Brasil
roberto@inf.ufpr.br

Marcio S. Oyamada
Ciência da Computação
Unioeste
Cascavel - Brasil
marcio.oyamada@unioeste.br

Resumo— O cLUPA é um sistema embarcado portátil para ampliar e aplicar contraste em documentos impressos utilizando uma câmera de vídeo para a captura e um monitor para a exibição. O *hardware* foi modelado em VHDL, contém dois núcleos cMIPS que se comunicam através de uma fila e componentes de interface USB e VGA. O cLUPA permite ampliar a parte central do documento e/ou exibir a imagem com contraste cinza, verde e vermelho. O cLUPA foi validado e seu desempenho avaliado através de simulações. Com relógio de 50MHz, os resultados de simulações indicam que o cLUPA pode processar de 9.3 a 2.4 quadros por segundo, dependendo das combinações de ampliação e contraste.

Keywords— *Sistema embarcado, ampliador digital, multicore, cMIPS.*

I. INTRODUÇÃO

Em 2010, o IBGE estimou que 48 milhões de brasileiros sofriam com algum tipo de deficiência visual, auditiva ou mental. A deficiência visual predomina com 19% da população (cerca de 35,6 milhões de pessoas) e é classificada de acordo com a severidade da deficiência, como “legalmente cego” (600 mil), visão “subnormal alta” (6 milhões) e visão “subnormal baixa” (29 milhões) [1].

As tecnologias assistivas (TAs) permitem que portadores de deficiência aumentem suas experiências motoras e sensoriais e consigam melhor inclusão no meio social [2]. Pessoas com visão subnormal alta dependem de ampliadores analógicos, ou do auxílio de outra pessoa, para ler documentos impressos, como manuais ou folhetos informativos [2], [3]. Para prover maior independência a estas pessoas, os ampliadores de documentos impressos são, geralmente, sistemas embarcados que através de uma câmera de vídeo capturam documentos impressos e os exibem em um monitor ou televisão. O xLupa Embarcado [4], [5] é um projeto de ampliador de documentos impressos derivado do xLupa [3], que é uma “lupa digital para ambiente Linux”.

Este artigo descreve o projeto de um ampliador de documentos impressos, chamado cLUPA, modelado em VHDL e compilado e simulado com GHDL, que baseia-se nas funcionalidades de ampliação e aplicação de contraste do xLupa Embarcado. Os resultados da avaliação de desempenho indicam que o cLUPA processa de 9.3 a 2.4 quadros por segundo, dependendo das combinações de contraste e ampliação. Estes resultados se baseiam num módulo

de desenvolvimento com relógio de 50MHz. O cLUPA foi projetado para implementação no *kit* de desenvolvimento com um FPGA Altera Cyclone IV [6].

II. cLUPA

O cLUPA emprega os quatro componentes mostrados na Figura 1: (A) o documento impresso a ser exibido, posicionado abaixo da câmera; (B) braço de sustentação com câmera; (C) módulo de processamento do cLUPA que contém os botões que definem a aplicação de ampliação e contraste (representados pelas letras Z e C) e é ligado à câmera e ao monitor por cabos USB e VGA, respectivamente; e (D) monitor que exibe a imagem capturada pela câmera e ampliada ou com contraste alterado.

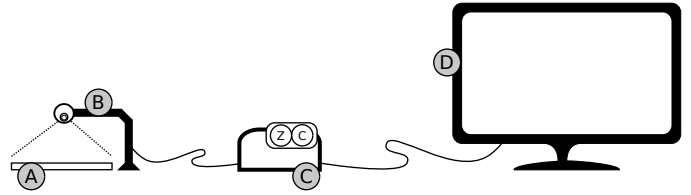


Figura 1: Modo de uso do cLUPA

Para fazer uso do cLUPA, o usuário deve posicionar o documento impresso abaixo da câmera, que captura e envia a imagem ao módulo de processamento. Este verifica a configuração de ampliação e contraste, a aplica, e envia a imagem tratada para o monitor. O usuário deve movimentar o documento impresso no sentido vertical em relação à câmera de acordo com a posição desejada de leitura (início, meio ou fim da página) e no sentido horizontal no caso de ampliação ativa. O usuário determina se a ampliação será ativada – sempre em um fator de 200% – e se contraste verde, vermelho ou cinza, será aplicado.

III. PLATAFORMA DE HARDWARE

Um diagrama de blocos da plataforma de *hardware* do cLUPA é mostrado na Figura 2 e é composta por 12 módulos. O Módulo-USB efetua a comunicação da câmera com o DMA-USB. O módulo DMA-USB contém registradores que capturam quatro bytes (um *pixel*) emitidos pelo Módulo-USB, e os envia para a memória RAM-0. O árbitro que controla o MULTIPLEXADOR-0 dá prioridade de acesso

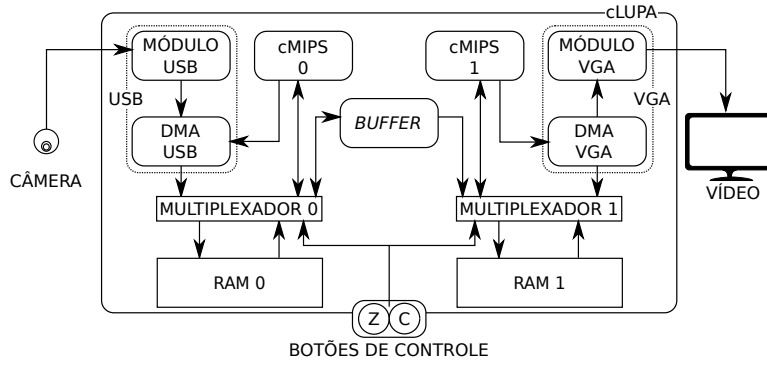


Figura 2: Plataforma de *hardware* do cLUPA

à memória para o componente DMA-USB porque este é um gargalo na comunicação entre o dispositivo de captura (DMA-USB) e armazenamento (RAM-0). O processamento das imagens é dividido entre dois núcleos cMIPS. O cMIPS é um modelo em VHDL de uma implementação completa do conjunto de instruções MIPS32r2 [7]. O modelo “implementa” um *pipeline* de cinco estágios similar ao descrito em [8].

O tratamento de contraste é aplicado *pixel-a-pixel* pelo núcleo cMIPS-0 e então os *pixels* alterados são gravados no BUFFER, que é uma fila de dados usada como único meio de comunicação entre os dois núcleos. O núcleo cMIPS-1 lê um *pixel* do BUFFER e calcula o endereço para finalizar o processamento e o armazena na memória RAM-1.

O módulo DMA-VGA lê, da memória RAM-1, todos os *pixels* da imagem processada e os envia ao Módulo-VGA. O árbitro que controla o MULTIPLEXADOR-1 dá prioridade de acesso à memória para o componente DMA-VGA porque a comunicação entre estes dois componentes é crítica para o desempenho do sistema. O Módulo-VGA gera os sinais analógicos de sincronização da imagem (*hsync* e *vsync*).

Os Botões de Controle permitem ao usuário configurar a ampliação e o contraste. O botão de ampliação, indicado por Z (*zoom*), determina a ativação da configuração de ampliação. O botão de contraste, indicado por C, altera a configuração de contraste para a próxima configuração da lista circular de sequência inativo, vermelho, verde e cinza.

Considerando o FPGA EP4CE30F23 do *kit* de desenvolvimento *Mercurio IV*, estima-se de que a implementação do cLUPA fará uso de 50% dos blocos combinacionais, 20% dos registradores, e 70% dos bits de memória [6], [7].

A comunicação do cLUPA com o meio externo (câmera, monitor) é realizada através dos módulos USB e VGA, respectivamente, e foram modelados a partir dos componentes disponíveis no *kit* de desenvolvimento de modo a simular os componentes reais.

Para efeitos de simulação, o componente Módulo-USB recupera cada byte de um arquivo binário (*img.data*), que contém apenas os *pixels* de uma imagem no formato *bitmap* de 32 bits e o Módulo-VGA escreve os *pixels* tratados no arquivo de saída (*vga.data*).

IV. SOFTWARE EMBARCADO

O código do cLUPA foi escrito com base nas funcionalidades do xLupa Embarcado e na divisão de tarefas apropriada ao *hardware* discutido na Seção III. O cLUPA processa imagens em oito configurações de ampliação e contraste: ampliação ativa ou inativa, e contraste cinza, vermelho, verde ou desativado.

O processamento das imagens é dividido entre os dois núcleos. O núcleo cMIPS-0 executa o módulo C-0 que controla o módulo USB, inicia os cálculos de endereços dos *pixels* (se a ampliação está ativa), aplicação de contraste e armazenamento no *buffer*. O núcleo cMIPS-1 executa o módulo C-1 que lê do *buffer*, finaliza os cálculos de endereços dos *pixels*, e controla o módulo VGA.

Na operação *sem* ampliação, os *pixels* são armazenados em posições contíguas da memória a partir do endereço base da imagem.

Com ampliação, a área de exibição contém 25% dos *pixels* da imagem original, e os *pixels* são armazenados em memória em endereços *e* determinados por

$$e = (b + l \cdot s + c) \quad (1)$$

onde *b* é o endereço base da imagem, *l* é a linha do *pixel*, *s* é o tamanho da linha, e *c* é a coluna do *pixel*. Os valores de *l* e *c* variam de linha/coluna iniciais até linha/coluna finais no corte da imagem.

O contraste cinza é calculado através da média aritmética dos componentes *r*, *g* e *b*. Os contrastes verde e vermelho são calculados através da expressão

$$k' = (2 \cdot r + 4 \cdot g + 1 \cdot b) / 6. \quad (2)$$

onde *k'* é a luminosidade que substitui o componente de contraste *r* ou *g*, e os demais componentes são substituídos por “0” (branco). A expressão *k'* é uma aproximação da expressão da luminosidade do xLupa Embarcado que utiliza ponto-flutuante, recurso não implementado no cMIPS.

A implementação no *kit* de desenvolvimento, com um relógio de 50MHz, implicou numa série de escolhas de projeto para que o desempenho do cLUPA fosse satisfatório. Uma delas é a execução do *software* de aplicação diretamente sobre o *hardware*, sem um sistema operacional (SO).

A aplicação é simples e os acessos aos dispositivos de entrada e saída são todos por *polling* e portanto o tratamento de E/S é trivial e dispensa os mecanismos mais sofisticados providos por um SO. Outra escolha condicionada pelas características do *kit* é a saída de vídeo em formato VGA com resolução de 640×480 *pixels*. Estas limitações foram respeitadas na avaliação de desempenho do projeto.

V. RESULTADOS DAS SIMULAÇÕES

Para avaliar o desempenho do cLUPA foram realizadas simulações de processamento de imagens com todas as configurações de contraste e ampliação, e foram usadas seqüências de três imagens para cada configuração. Como entrada foram utilizadas imagens no formato *bitmap* de 32 bits com uma resolução de 640 por 480, totalizando 307.200 *pixels*.

As configurações de contraste são representadas por *ampliação* *contraste*, sendo a ampliação *nenhuma* (N) ou *ampliada* (A), e os contrastes cinza (C), verde (G), vermelho (R) ou nenhum (N). A combinação “sem ampliação e com contraste verde” é representada por N_G.

Uma simulação processa três imagens contidas no arquivo de entrada *img.data* e as imagens resultantes são gravadas no arquivo de saída *vga.data*. O processamento de cada imagem é dividido em quatro operações: (i) transferência USB (T-USB): transferência da imagem do módulo MÓDULO-USB para RAM-0 (módulo DMA-USB); (ii) processamento inicial (C-0): leitura da imagem de RAM-0, computação dos endereços para ampliação, computação do contraste e escrita no *buffer* (efetuado por cMIPS-0); (iii) processamento final (C-1): leitura do *buffer*, gravação da imagem e replicação dos vizinhos (com ampliação) e escrita da imagem em RAM-1 (efetuado por cMIPS-1); e (iv) transferência VGA (T-VGA): transferência da imagem da RAM-1 para o MÓDULO-VGA (módulo DMA-VGA).

Para mensurar o tempo simulado do processamento dos vários componentes do cLUPA foi usado o comando condicional *assert* da linguagem VHDL.

A avaliação se dá com duas métricas: (i) *tempo simulado de processamento* (TP) para executar uma determinada operação; e (ii) *quadros por segundo* (qps).

A. Processamento Paralelo

A Figura 3 mostra o diagrama de tempo das operações do cLUPA para N_G, com o tempo medido em ciclos de relógio (CP). O cLUPA-0 requer 77 ciclos para iniciar a transferência USB (T-USB) que leva 3.072.006 ciclos (61,44 ms) para gravar a imagem na memória. mais 73 ciclos são necessários entre leitura de configurações de contraste e ampliação e instruções do código (atribuições, laços de repetição e saltos condicionais).

Após 15 ciclos da gravação do primeiro *pixel* no *buffer* por C-0, C-1 é iniciado e as duas operações processam os *pixels* da imagem concorrentemente. C-1 encerra 10 ciclos depois do encerramento de C-0. O tempo para processar uma imagem pelas operações C-0 e C-1 é 361,6 ms ou 18.1×10^6 ciclos.

O tempo de inicialização da transferência da memória para a interface VGA (T-VGA) é de 28 ciclos. A transferência consome 614.401 ciclos. Os tempos de transferência T-VGA e T-USB são constantes para todas as simulações, pois dependem apenas do número total de *pixels* da imagem.

No instante em que a operação T-VGA é iniciada, a operação T-USB para a nova imagem é também iniciada, com processamento em paralelo durante a operação T-VGA.

A sobreposição de processamento de uma imagem entre as operações C-0 e C-1, e a sobreposição das transferências T-VGA e T-USB ocorrem sempre e portanto, na Seção V-B discute-se somente os tempos das operações C-0 e C-1.

B. Ampliação e Contraste

A Tabela I mostra as medidas de tempo das operações C-0 e C-1 para as 8 combinações de ampliação e contraste.

As medidas para o processamento *sem* contraste e *com* ampliação são as melhores porque não há processamento para alterar os *pixels* da imagem, e a imagem de saída tem 1/4 dos *pixels* originais, apesar da replicação para os vizinhos próximos [9].

Tabela I: Medidas das operações C-0 e C-1.

	com ampl.		sem ampl.	
	C-0	C-1	C-0	C-1
sem contraste	A_N		N_N	
TP [ms]	41,5	41,5	172,0	172,0
contr. cinza	A_C		N_C	
TP [ms]	59,9	59,9	258,0	258,0
contr. verde	A_G		N_G	
TP [ms]	85,8	85,8	361,6	361,6
contr. vermelho	A_R		N_R	
TP [ms]	84,5	85,5	356,4	356,4

Nos *contrastes verde e vermelho*, o valor do *pixel* é computado pela Equação 2, que tem um custo computacional maior do que o contraste inativo, e isso impacta no custo por *pixel*, que aumenta até 200%.

O tempo simulado total de uma imagem sem ampliação e com contraste verde é de 435,3 ms, este valor obtido com a sobreposição de processamento entre as operações C-0 e C-1, conforme discutido na Seção V-A.

Os resultados apresentados acima foram todos obtidos com o código compilado com otimização -O2, que é a otimização padrão nos *scripts* de compilação para facilitar a legibilidade do código em *assembly*.

A Tabela II apresenta os resultados em quadros por segundo para todas as combinações de ampliação e contraste.

Possamai e Oyamada [10] avaliam os ganhos de desempenho obtido pela otimização e sintonia fina de componentes de *software* do xLupa Embarcado. Este ampliador é implementado sobre uma Beagleboard-xM, executa Ubuntu 12.04, e faz uso da biblioteca Cairo. Os ganhos obtidos

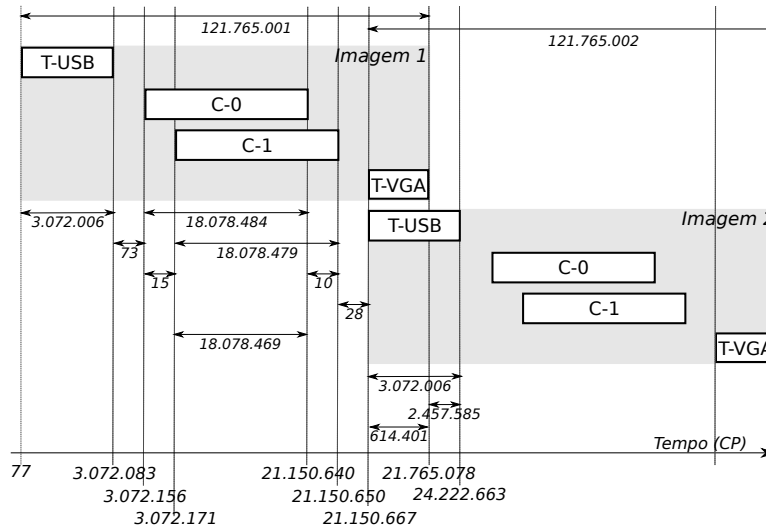


Figura 3: Diagrama de tempo das operações do cLUPA para contraste verde, sem ampliação (N_G).

Tabela II: Resultados finais [qps].

Contraste	com ampl.	sem ampl.
sem	9,3	4,2
cinza	8,0	3,1
verde	6,6	2,3
vermelho	6,7	2,4

são apreciáveis, e o desempenho *sem* contraste passa de 2,5 qps para 6,7 qps, e *com* contraste o desempenho cresce de 1,7 qps para 2,9 qps. O modelo de simulação do cLUPA indica desempenho ligeiramente superior: *sem* contraste o ganho é de $1,4 \times (9,3/6,7)$, e *com* contraste vermelho o ganho é de $2,3 \times (6,7/2,9)$.

VI. IMPLEMENTAÇÃO E TRABALHOS FUTUROS

São muitas as possibilidades por testar após a implementação do cLUPA no *kit* de desenvolvimento. Vejamos algumas delas.

As operações para cálculo do contraste empregam somas saturadas e instruções que efetuam somas saturadas não constam do conjunto de instruções implementado no cMIPS. A implementação é relativamente simples e o ganho de desempenho seguramente compensaria o esforço.

O uso de uma plataforma de hardware com interfaces de vídeo de maior capacidade do que o VGA simples abre larga avenida para experimentação e desenvolvimento.

VII. CONCLUSÃO

Este artigo descreve o projeto do ampliador de documentos impressos cLUPA. O *hardware* do cLUPA é um sistema *multicore* com dois núcleos cMIPS que se comunicam através de uma fila, modelado utilizando VHDL. O *software* foi escrito com base nas funcionalidades de ampliação e aplicação de contraste presentes no xLupa Embarcado e adaptado para a plataforma de *hardware*.

A avaliação de desempenho foi realizada através de simulações da execução do modelo em oito configurações: quatro configurações de contraste em duas configurações de ampliação. Os resultados obtidos nas simulações variam de 9,3 quadros por segundo (qps) para configuração de ampliação ativa e contraste inativo, a 2,3 qps para configuração de contraste verde e ampliação desativada.

REFERÊNCIAS

- [1] IBGE, "Censo 2010 - Pessoas de 5 anos ou mais de idade, por existência ou não de pelo menos uma das deficiências investigadas e alfabetização, segundo o sexo e os grupos de idade," disponível em <http://www.ibge.gov.br/>. Acessado em Nov/2013.
- [2] R. Bersch, "Introdução à tecnologia assistiva," 2008. CEDI - Centro Especializado em Desenvolvimento Infantil.
- [3] J. Bidarra, C. Boscarioli, and S. M. Peres, "Avaliando a ferramenta xLupa como um recurso para educação especial e inclusiva," in *Simpósio Brasileiro de Informática na Educação (XX SBIE)*, 2009.
- [4] F. F. dos Santos, J. Bidarra, and M. Oyamada, "Implementação de um ampliador de documentos impressos embarcado," to be published.
- [5] D. R. Hachmann, P. W. Santiago, J. Bidarra, and M. Oyamada, "Um ampliador de tela embarcado utilizando arquiteturas heterogêneas," in *Fórum Internacional Software Livre - Workshop de Software Livre (XXII FISL)*, 2011.
- [6] M. DHW, *Mercurio IV: Manual do Usuário*, disponível em <http://www.macnicadhw.com.br/products/mercurion-4-devkit-board>. Acessado em Nov/2014.
- [7] R. A. Hexsel, "cMIPS, a VHDL model for the classical five stage pipeline," 2013. Universidade Federal do Paraná - Departamento de Informática. Relatório Técnico.
- [8] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, 4th ed. Morgan Kaufmann Publishers Inc., 2008.
- [9] Cambridge in Colour, "Digital Image Interpolation," disponível em <http://www.cambridgeincolour.com/tutorials/image-interpolation.htm>. Acessado em Ago/2014.
- [10] D. Possamai and M. S. Oyamada, "Avaliando o impacto de otimizações em diferentes níveis em aplicações embarcadas com interface gráfica," in *IV Simp Brasileiro de Engenharia de Sistemas Computacionais (SBESC2014)*, 2014, pp. 2188–2191.