

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DO PARANÁ
Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial

DISSERTAÇÃO
apresentada ao CEFET-PR
para obtenção do título de

MESTRE EM CIÊNCIAS

por

HENRY PONTI MEDEIROS

SISTEMA INTEGRADO DE AQUISIÇÃO E PROCESSAMENTO DE IMAGENS

Banca Examinadora:

Presidente e Orientador:

Prof. Dr. HUMBERTO REMIGIO GAMBA

CEFET-PR

Examinadores:

Prof. Dr. RAIMES MORAES

UFSC

Prof. Dr. JOAQUIM MIGUEL MAIA

CEFET-PR

Prof. Dr. PEDRO MIGUEL GEWEHR

CEFET-PR

Curitiba, 28 de abril de 2005

HENRY PONTI MEDEIROS

SISTEMA INTEGRADO DE AQUISIÇÃO E PROCESSAMENTO DE IMAGENS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial do Centro Federal de Educação Tecnológica do Paraná, como requisito parcial para a obtenção do título de “Mestre em Ciências” – Área de Concentração: Engenharia Biomédica.

Orientador: Prof. Dr. Humberto Remigio Gamba.

Curitiba

2005

Este trabalho é dedicado a Michelle e
Bernardo.

AGRADECIMENTOS

Agradecimentos aos professores e pesquisadores do Laboratório de Imagem e Multimídia (LABIEM) do CEFET-PR. Em especial a Jefferson Osowsky e Gustavo Benvenuti Borba pela constante disposição para discutir problemas e propor soluções.

Agradecimentos ao professor Miguel Antonio Sovierzoski por todo o apoio, técnico ou não, desde os projetos de iniciação científica até a conclusão deste trabalho.

Ao professor Pedro Miguel Gewehr por sua constante disponibilidade para revisar e corrigir voluntariamente inúmeros artigos, propostas e projetos.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelas bolsas de iniciação científica que serviram como base para este trabalho e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) que, por meio de uma bolsa de mestrado, viabilizou este projeto.

Finalmente, agradecimentos ao professor Humberto Remigio Gamba, que, mesmo diante das adversidades sofridas pela ciência em nosso país, não se deixa abater e emprega todos os seus esforços para garantir a continuidade da pesquisa científica.

SUMÁRIO

LISTA DE FIGURAS	XI
LISTA DE TABELAS.....	XIX
LISTA DE SIGLAS.....	XXI
RESUMO	XXIII
ABSTRACT	XXV
1 INTRODUÇÃO	1
1.1 OBJETIVOS.....	9
1.2 ESTRUTURA DA DISSERTAÇÃO	10
2 FUNDAMENTAÇÃO TEÓRICA.....	11
2.1 INTRODUÇÃO	11
2.2 SENSORES DE IMAGEM.....	11
2.2.1 Fotodetectores	12
2.2.1.1 Capacitor MOS	13
2.2.2 Mecanismo de Transferência de Cargas.....	16
2.2.2.1 Transferência de Cargas com Três Fases	16
2.2.2.2 Transferência de Cargas com Duas Fases	18
2.2.2.3 Transferência de Cargas com Fase Virtual.....	19
2.2.3 Dispositivo de Saída.....	20
2.2.4 Arquiteturas de Sensores CCD.....	22
2.2.4.1 Sensores Lineares	22
2.2.4.2 Sensores de Imagem <i>Full Frame</i>	23
2.2.4.3 Sensores de Imagem <i>Frame Transfer</i>	24
2.2.4.4 Sensores de Imagem <i>Interline</i>	25
2.2.5 Fontes de Ruído em Sensores CCD	27
2.3 <i>UNIVERSAL SERIAL BUS</i>	29

2.3.1	O Processo de Enumeração	33
2.3.2	A Interface entre <i>Hardware</i> e <i>Software</i> no <i>Host</i>	34
2.4	PROCESSADORES DIGITAIS DE SINAIS	36
2.4.1	Unidade de Multiplicação e Acumulação	37
2.4.2	Barramentos de Acesso à Memória.....	38
2.4.3	Endereçamento de Memória.....	39
2.4.4	Paralelismo em Nível de Instrução.....	40
2.4.5	<i>Pipelining</i>	41
2.4.6	<i>Very Long Instruction Word</i>	43
2.4.7	Processadores Multimídia	44
3	MATERIAIS E MÉTODOS.....	47
3.1	ARQUITETURA DO SISTEMA.....	47
3.2	SISTEMA DE AQUISIÇÃO DE IMAGENS	48
3.2.1	Arranjo Mecânico.....	49
3.2.2	Sensor de Imagem	50
3.2.3	Dispositivo Termo-elétrico de Resfriamento	52
3.3	CONTROLE DO SENSOR DE IMAGEM	53
3.4	SISTEMA DE CONVERSÃO ANALÓGICO-DIGITAL.....	56
3.5	SISTEMA DE PROCESSAMENTO DAS IMAGENS.....	60
3.5.1	Interface com os <i>Analog Front Ends</i>	61
3.5.2	Interface com o DSP TMS320LF2402.....	62
3.5.3	Interface USB	63
3.5.4	Estrutura do <i>Firmware</i> do DSP TMS320C6416	64
3.6	SOFTWARE PARA COMUNICAÇÃO COM O <i>HOST</i>	68
3.6.1	Comunicação com a Câmera.....	69
3.6.2	Apresentação das Imagens	71
4	RESULTADOS	73
4.1	INTRODUÇÃO	73
4.2	DISTÂNCIA FOCAL	74
4.3	RESFRIAMENTO DO SENSOR DE IMAGEM	75
4.4	TEMPO DE INTEGRAÇÃO DO SENSOR DE IMAGEM.....	75
4.5	DESLOCAMENTO PARA O NÍVEL DE PRETO ÓPTICO	78

4.6	NÍVEL DE RUÍDO DAS IMAGENS ADQUIRIDAS.....	79
4.7	TEMPO DE AQUISIÇÃO.....	80
4.8	TRANSMISSÃO DAS IMAGENS PARA O MICROCOMPUTADOR....	80
4.9	PROCESSAMENTO DAS IMAGENS.....	81
5	DISCUSSÃO E CONCLUSÕES.....	83
5.1	DISCUSSÃO.....	83
5.1.1	Aquisição das Imagens.....	83
5.1.2	Transmissão das Imagens para o Microcomputador.....	83
5.2	CONCLUSÕES.....	84
5.3	TRABALHOS FUTUROS.....	85
5.3.1	Sistema de Gerenciamento de Tarefas.....	85
5.3.2	Sistema de Armazenamento e Controle de Algoritmos.....	85
5.3.3	Transmissão das Imagens para o Microcomputador.....	86
5.3.4	Versão Definitiva do <i>Hardware</i>	86
	REFERÊNCIAS BIBLIOGRÁFICAS.....	87

LISTA DE FIGURAS

- Figura 1 – Diagrama em blocos de uma câmera inteligente. 4
- Figura 2 – Diagrama de bandas de energia de um material semiconductor ilustrando geração de portadores livres devido a absorção de fótons. Os fótons com energia hf são absorvidos pelo material deslocando portadores da banda de valência, cuja energia máxima é ϵ_v , para a banda de condução, cuja energia mínima é ϵ_c (modificado de Jenkins, 1987). 12
- Figura 3 – Estrutura do capacitor MOS. Um substrato de silício dopado com impurezas (tipo P no exemplo) é conectado a um eletrodo por meio de uma interface de isolante (geralmente dióxido de silício). Regiões fortemente dopadas, conhecidas como *channel stop* limitam a extensão lateral da região de depleção formada no substrato do capacitor MOS quando o eletrodo é polarizado. 14
- Figura 4 – Diagrama de níveis de energia do capacitor MOS polarizado. Na interface entre o substrato do capacitor e a camada de dióxido de silício é formada uma região com níveis de energia inferiores aos do substrato, conhecida como poço de potencial, na qual portadores livres podem ser armazenados. 14
- Figura 5 – Coeficiente de absorção do silício policristalino fortemente dopado com fósforo (modificado de Ristic, 1994). 15
- Figura 6 – Capacitor MOS de canal enterrado. Entre o substrato e a camada de dióxido de silício é inserida uma região levemente dopada com portadores do tipo oposto ao do substrato, o que faz com que o poço de potencial se forme a alguns micrometros da interface entre o substrato e o dióxido de silício. 16
- Figura 7 – (a) Secção transversal de duas células de um sensor CCD de três fases. (b) Ilustração do mecanismo de transferência de cargas. No instante de tempo inicial, o primeiro eletrodo de cada célula é polarizado, formando um poço de potencial sob ele. Para realizar a transferência de cargas, os eletrodos adjacentes são polarizados de forma a fazer com que os poços de potencial se expandam sob pares de eletrodos e, em seguida, comprimam-se sob os eletrodos seguintes, até que as cargas alcancem as células adjacentes. 17

- Figura 8 – Ilustração do mecanismo de transferência de cargas em um sensor CCD de duas fases. Devido à dopagem diferenciada sob os eletrodos, são formados poços de potencial de diferentes profundidades, de forma que, com a polarização dos eletrodos adjacentes, as cargas são forçadas a migrar direção ao eletrodo com maior dopagem. 18
- Figura 9 – Estrutura de um sensor CCD de duas fases utilizando dois eletrodos. Sob cada eletrodo existem duas regiões com dopagens diferenciadas. 19
- Figura 10 – Ilustração do mecanismo de transferência de cargas em um sensor CCD de fase virtual. Quando o eletrodo é polarizado, o potencial sob ele é inferior ao potencial da região não coberta pelo eletrodo (fase virtual). Quando a polarização é retirada do eletrodo, o potencial da região sob ele torna-se maior que o da fase virtual, obrigando os portadores a migrarem para ela. 20
- Figura 11 – Circuito *Floating Diffusion*, usado para converter as cargas elétricas em níveis de tensão na saída de sensores CCD. As cargas são transferidas para o *floating diffusion* devido à polarização do eletrodo de saída. O *floating diffusion* é conectado a um amplificador, o qual converte as cargas armazenadas em níveis de tensão. Após a conversão, as cargas são removidas do sensor através do dreno de *reset* devido à polarização do eletrodo de *reset*. 21
- Figura 12 – Arquitetura do sensor CCD linear. Os elementos fotodetectores, dispostos em uma única linha são conectados, através de eletrodos de transferência, a um registrador serial responsável pelo deslocamento das cargas para o amplificador de saída do dispositivo. 22
- Figura 13 – Arquitetura e funcionamento do sensor CCD *Full Frame*. Após a exposição à luz dos elementos fotodetectores, dispostos em forma de matriz, as cargas armazenadas em cada linha do sensor são deslocadas para a linha subjacente. As cargas armazenadas na última linha são transferidas para o registrador serial, através do qual são transferidas seqüencialmente para o amplificador de saída do dispositivo. Esse processo é repetido para cada linha de elementos fotodetectores do sensor. 24
- Figura 14 – Arquitetura e funcionamento do Sensor CCD *Frame Transfer*. O sensor possui uma região de elementos fotodetectores exposta à luz e outra região idêntica, porém protegida da incidência de luz. Após o período de exposição do sensor, as cargas presentes em cada linha exposta à luz são transferidas

	para a linha subjacente, até que todas as linhas sejam transferidas para a região de armazenamento. Em seguida, cada linha da região de armazenamento é transferida para a linha subjacente, de forma que a última linha é transferida para o registrador serial, através do qual os elementos são deslocados seqüencialmente para o amplificador de saída do sensor. Esse processo é então repetido para cada linha da região de armazenamento.	25
Figura 15 –	Arquitetura e funcionamento do sensor CCD <i>Interline</i> . A cada linha de elementos fotodetectores exposta à luz é intercalada uma linha de elementos fotodetectores protegida da incidência de luz. Após a exposição do sensor à imagem, todos os elementos presentes nas colunas expostas são transferidos simultaneamente para as colunas protegidas de incidência de luz. Em seguida, cada linha de elementos protegidos é transferida para a linha subjacente, de forma que a última linha é transferida para o registrador serial, através do qual as cargas são deslocadas seqüencialmente para o amplificador de saída do dispositivo. Esse processo é repetido para todas as linhas do sensor.	26
Figura 16 –	Topologia do barramento USB apresentando a conexão de dispositivos ao <i>host</i> por meio de <i>hubs</i> (fonte: USB-IF, 2000).....	29
Figura 17 –	Pilha de <i>drivers</i> USB no sistema operacional Windows. O <i>driver</i> do controlador <i>host</i> faz a interface com o <i>hardware</i> . Os <i>drivers</i> do barramento USB gerenciam a transferências. Os <i>drivers</i> de classes de dispositivos fazem a interface entre o aplicativo do usuário e dispositivos padrão como mouses ou impressoras, enquanto os <i>drivers</i> de dispositivos específicos fazem a interface com dispositivos que não pertencem a classes padrão.....	35
Figura 18 –	Arquitetura da unidade de multiplicação e acumulação de um processador digital de sinais. Dois registradores são conectados à entrada de um multiplicador, cuja saída é conectada a uma unidade lógico-aritmética. À outra entrada da unidade lógico-aritmética é conectada sua própria saída através de um registrador acumulador.	37
Figura 19 –	(a) Arquitetura <i>von Neuman</i> , na qual um único barramento é utilizado para acessar as memórias de dados e de programa. (b) Arquitetura <i>Harvard</i> , na qual existem barramentos independentes para acesso às memórias de dados e de programa.....	38

Figura 20 – Estrutura de um <i>buffer</i> circular. Um ponteiro indica a posição da amostra mais atual do sinal obtido e registradores definem as posições inicial e final do <i>buffer</i>	39
Figura 21 – Estágios de um <i>pipeline</i> . Cada uma das unidades funcionais é responsável pela execução de uma etapa do ciclo de instrução do processador.....	41
Figura 22 – Sobreposição de instruções em um <i>pipeline</i> de cinco estágios. A cada ciclo de <i>clock</i> do processador as unidades funcionais executam diferentes estágios de diversas instruções simultaneamente.	42
Figura 23 – Diagrama em blocos do sistema desenvolvido.	47
Figura 24 – Plano focal do sensor de imagem (fonte: Texas Instruments, 2001).	49
Figura 25 – (a) Foto da peça para acoplamento da lente. (b) Peça com lente <i>C- Mount</i> acoplada.	49
Figura 26 – Vista longitudinal ilustrando aspectos construtivos da montagem do sensor de imagem, lente e módulo de resfriamento <i>Peltier</i>	50
Figura 27 – Unidades funcionais do sensor de imagem CCD TC237. O sensor é do tipo <i>frame transfer</i> , contando com uma área sensível à imagem e uma área de armazenamento, ambas separadas por 4 linhas escuras. Na área sensível existem 22 elementos protegidos de incidência de luz para servirem de referência em etapas de processamento do sinal. As cargas podem ser lidas do sensor através de dois registradores seriais.....	51
Figura 28 – Resposta espectral do sensor CCD TC237 (modificado de Texas Instruments, 2001).	52
Figura 29 – Curva de temperatura versus potência térmica do módulo <i>Peltier</i> (modificado de Melcor Corporation, 2002).	52
Figura 30 – Pulsos de leitura do sensor CCD TC237 em modo progressivo com um único registrador. A leitura do sensor se divide em quatro etapas: apagamento, integração, transferência para a memória e leitura serial (fonte: Texas Instruments, 2001).	54
Figura 31 – Diagrama de fluxo do <i>firmware</i> do DSP de controle. O DSP pode receber comandos para gravar novos valores para o tempo de integração do sensor ou para iniciar a geração dos pulsos de leitura do sensor CCD. O DSP gera o pulso de apagamento, aguarda o período de integração do sensor, gera os pulsos de transferência para a região de armazenamento e os pulsos de sincronismo horizontal para o AFE. O número de pulsos gerados pelo AFE	

é controlado pelo intervalo entre os pulsos de sincronismo gerados pelo DSP de controle.....	56
Figura 32 – Diagrama em blocos do <i>analog front end</i> AD9949. O dispositivo conta com uma etapa de <i>correlated double sampling</i> (CDS), um amplificador de <i>pixels</i> (PxGA), um amplificador de ganho variável (VGA), uma etapa de deslocamento para o nível de preto óptico (CLAMP), um conversor analógico-digital de 12 bits, além de circuitos de geração de pulsos internos controlados pelos sinais de sincronismo horizontal (HD) e vertical (VD). A programação dos registradores do dispositivo é feita por meio de uma interface serial síncrona (fonte: Analog Devices, 2004b).....	57
Figura 33 – Divisão do <i>clock</i> do AD9949. Cada ciclo de <i>clock</i> é dividido em 48 posições que podem ser utilizadas para programar os sinais de controle do CCD (fonte: Analog Devices, 2004b).....	59
Figura 34 – Posições programáveis dos sinais de controle do CCD gerados pelo AD9949. Através destes sinais é possível programar a largura dos pulsos de <i>reset</i> do amplificador de saída do sensor CCD, dos pulsos de leitura serial e os instantes de amostragem do circuito de <i>correlated double sampling</i> (fonte: Analog Devices, 2004b).....	59
Figura 35 – Procedimento de escrita via interface serial do <i>analog front end</i> AD9949. Os primeiros oito bits compõem o endereço do registrador a ser gravado, os 24 bits seguintes compõem o dado (fonte: Analog Devices, 2004b).....	61
Figura 36 – Configuração dos <i>endpoints</i> do <i>serial interface engine</i> . Através do <i>endpoint</i> 2, configurado para transferências do tipo IN com um <i>buffer</i> de 3.072 bytes, são transferidas imagens do SI-API para o <i>host</i> . Através do <i>endpoint</i> 8, configurado para transferências do tipo OUT com um <i>buffer</i> de 1.024 bytes, são transmitidos comandos do <i>host</i> para o SI-API.....	64
Figura 37 – <i>Buffer</i> para armazenamento de imagens. Para armazenar as imagens adquiridas do sensor CCD, a memória SDRAM do sistema foi dividida em 24 posições de 342.000 <i>words</i> de 16 bits.....	65
Figura 38 – Diagrama de fluxo do <i>firmware</i> do DSP TMS320C6416. Os blocos em linhas pontilhadas representam os quatro comandos implementados no <i>firmware</i> : configuração do tempo de integração, aquisição de imagem através sensor CCD, aplicação do algoritmo de Sobel e transmissão de imagem para o <i>host</i> via interface USB.....	67

Figura 39 – Sistema operacional instalando os <i>drivers</i> do equipamento.	68
Figura 40 – Tela principal do <i>software</i> para comunicação com o <i>host</i> apresentando uma imagem adquirida pelo sistema.	69
Figura 41 – Menu de comunicação.	69
Figura 42 – Menu de configuração.	70
Figura 43 – Tela de configuração do tempo de integração.	70
Figura 44 – Tela de montagem de seqüência de comandos.	71
Figura 45 – Menu imagem.	72
Figura 46 – Fotografia do <i>hardware</i> do sistema desenvolvido, incluindo o sistema de aquisição e processamento de imagens e a fonte de alimentação.	73
Figura 47 – Fotografias (a) frontal e (b) traseira do protótipo construído.	74
Figura 48 – Imagens obtidas com lentes do tipo (a) <i>CS-Mount</i> de 3,5 mm, (b) <i>C-Mount</i> de 8,5 mm e (c) <i>C-Mount</i> de 12 mm. Resolução de 684 x 500 pixels, tempo de integração 3 ms, 256 níveis de cinza.....	75
Figura 49 – Imagens obtidas com o obturador da lente completamente aberto e tempos de integração de (a) 2,5 ms, (b) 4,5 ms e (c) 7 ms. Resolução de 684 x 500 <i>pixels</i> , 256 níveis de cinza. É possível observar a saturação nas regiões mais claras das imagens conforme se aumenta o tempo de integração.....	76
Figura 50 – Variação da intensidade média dos <i>pixels</i> da região de interesse em função do tempo de integração do sensor CCD com o obturador da lente <i>C-mount</i> de 12 mm completamente aberto. A variação obtida foi linear com inclinação de aproximadamente 22 níveis de cinza por milissegundo até 9,5 ms quando tem início o efeito de saturação.....	77
Figura 51 – Imagem obtida com o obturador da lente em 30% de sua abertura máxima e tempo de integração de 17 ms.....	77
Figura 52 – Variação da intensidade média dos <i>pixels</i> da imagem em função do tempo de integração do sensor CCD com o obturador da lente em 30% de sua abertura máxima. A variação obtida foi linear com inclinação de 6 níveis de cinza por milissegundo.....	78
Figura 53 – Imagens adquiridas (a) sem e (b) com deslocamento para o nível de preto óptico.....	78
Figura 54 – Histogramas das imagens adquiridas sem deslocamento para o nível de preto óptico (linha pontilhada), figura 53 (a) e com deslocamento (linha contínua), figura 53 (b).	79

Figura 55 – (a) Imagem obtida sem iluminação incidindo sobre o sensor, (b) histograma da imagem.....	80
Figura 56 – (a) Imagem adquirida pelo sistema, (b) imagem após a aplicação do algoritmo de Sobel. Resolução de 684 x 500 <i>pixels</i> , tempo de integração 3 ms, 256 níveis de cinza.	81

LISTA DE TABELAS

Tabela 1 – Tipos de pacotes especificados pela interface USB.	32
Tabela 2 – Estados lógicos definidos pela interface USB.....	32
Tabela 3 – Descritores definidos pela especificação USB.....	33
Tabela 4 – Ordem dos resultados de uma transformada rápida de Fourier de 8 pontos.	40
Tabela 5 – Mapa de memória do sistema.....	61
Tabela 6 – Rotinas de comunicação com os <i>analog front ends</i>	62
Tabela 7 – Comandos do <i>host</i> para o sistema.	65
Tabela 8 – Variação da temperatura sobre a superfície do sensor de imagem em função da corrente aplicada ao elemento <i>Peltier</i>	75
Tabela 9 – Tempos de transmissão de imagens via interface USB.....	80

LISTA DE SIGLAS

AFE	<i>Analog Front End</i>
ALU	<i>Arithmetic Logic Unit</i>
API	<i>Application Programming Interface</i>
ASIC	<i>Application Specific Integrated Circuit</i>
CCD	<i>Charge Coupled Device</i>
CDS	<i>Correlated Double Sampling</i>
CLP	Controlador Lógico Programável
CMOS	<i>Complementary Metal Oxide Semiconductor</i>
CPU	<i>Central Processing Unit</i>
CRC	<i>Cyclic Redundancy Check</i>
DMA	<i>Direct Memory Access</i>
DSK	<i>DSP Starter Kit</i>
DSP	<i>Digital Signal Processor</i>
EDMA	<i>Enhanced Direct Memory Access</i>
EMIF	<i>External Memory Interface</i>
EPIC	<i>Explicitly Parallel Instruction Computer</i>
FFT	<i>Fast Fourier Transform</i>
FIFO	<i>First In First Out</i>
FIR	<i>Finite Impulse Response</i>
FPGA	<i>Field Programmable Gate Array</i>
FPN	<i>Fixed Pattern Noise</i>
HD	<i>Horizontal Driver</i>
HID	<i>Human Interface Device</i>
IIR	<i>Infinite Impulse Response</i>
IAG	<i>Image Area Gate</i>
ILP	<i>Instruction Level Parallelism</i>
ITO	<i>Indium-Tin Oxide</i>
LED	<i>Light Emitting Diode</i>

MAC	<i>Multiply and Accumulate</i>
McBSP	<i>Multi-Channel Buffered Serial Port</i>
MIPS	<i>Millions of Instructions per Second</i>
MOPS	Milhões de Operações por Segundo
MOS	<i>Metal Oxide Semiconductor</i>
MPEG	<i>Moving Picture Expert Group</i>
NRZI	<i>Non Return to Zero Inverted</i>
OBC	<i>Optical Black Clamping</i>
ODB	<i>Output Drain Bias</i>
PDA	<i>Personal Digital Assistant</i>
PID	<i>Package Identifier</i>
Pixel	<i>Picture Element</i>
PxGA	<i>Pixel Gain Amplifier</i>
QCIF	<i>Quarter Common Intermediate Format</i>
RDI	Região de Interesse
RST	<i>Reset Gate</i>
SAG	<i>Storage Area Gate</i>
SDRAM	<i>Synchronous Dynamic Random Access Memory</i>
SI-API	Sistema Integrado de Aquisição e Processamento de Imagens
SIE	<i>Serial Interface Engine</i>
SRG	<i>Serial Register Gate</i>
UART	<i>Universal Asynchronous Receiver Transmitter</i>
USB	<i>Universal Serial Bus</i>
VD	<i>Vertical Driver</i>
VGA	<i>Variable Gain Amplifier</i>
VLIW	<i>Very Long Instruction Word</i>
WLAN	<i>Wireless Local Area Network</i>
JTAG	<i>Joint Test Action Group</i>

RESUMO

Este trabalho apresenta o projeto, a implementação e os testes de um sistema integrado de aquisição e processamento de imagens (SIAPI). O sistema desenvolvido é constituído por um sensor de imagem CCD (*Charge Coupled Device*), um processador digital de sinais (*Digital Signal Processor* – DSP) de baixo custo responsável pelo controle do sensor, um DSP de alto desempenho para realizar o processamento das imagens adquiridas, uma interface USB (*Universal Serial Bus*) para realizar a comunicação do dispositivo com um microcomputador e um módulo de resfriamento termo-elétrico baseado no efeito *Peltier* para minimizar a quantidade de ruído térmico gerado no sensor de imagem. O sistema permite que se controle todos os parâmetros do sensor de imagem, bem como o processamento das imagens adquiridas em tempo real. Para avaliar as características e o desempenho do sistema, construiu-se um protótipo baseado em um DSK (*DSP Starter Kit*) e desenvolveu-se um *software* em linguagem C para comunicação com o sistema. Foram levantadas as características de resposta do sensor de imagem em diferentes configurações. O sistema demonstrou ser funcional e seu desempenho na execução de algoritmos de processamento de imagens foi bastante superior ao de um microcomputador de alto desempenho.

ABSTRACT

This work presents the design and implementation of a new integrated system for image acquisition and processing (SIAPI). The system is composed of a charge coupled device (CCD) image sensor, a low end DSP (Digital Signal Processor) to control the image sensor, a high end DSP to process the acquired images, a universal serial bus (USB) interface to connect the system to a microcomputer and a thermo-electric cooler device based on the Peltier effect which is meant to reduce the amount of thermal noise generated by the image sensor. All the functional parameters of the image sensor can be controlled through the system, and the images acquired can be processed in real time. To evaluate the functions of the system and to measure its performance, a prototype was built based on a DSK (DSP Starter Kit). Also, a computer program in C language was developed to communicate with the system. The response characteristics of the image sensor were evaluated under different conditions. It has been concluded that the system is completely functional and that it outperforms a high performance microcomputer when executing image processing algorithms.

CAPÍTULO 1

INTRODUÇÃO

A capacidade de processamento dos computadores vem crescendo a taxas maiores a cada ano. Durante a década de setenta, o desempenho dos computadores aumentava a aproximadamente 25% ao ano, atualmente esta taxa chega a 50% (HENNESSY e PATTERSON, 1996). Dois fatores levam a este aumento crescente de desempenho: o surgimento de novas tecnologias e o desenvolvimento de arquiteturas mais eficientes.

O grande marco das inovações tecnológicas na área de computação foi a criação do microprocessador, a partir do qual computadores tornaram-se economicamente viáveis em praticamente qualquer aplicação. A partir de então, surgiram os computadores pessoais e, não muito mais tarde, computadores passaram a ser integrados aos mais diversos tipos de equipamentos, desde controladores industriais até eletrodomésticos.

O surgimento de sistemas com propósitos específicos integrados a computadores, os chamados computadores embarcados, levou ao desenvolvimento de processadores com arquiteturas projetadas especialmente para cumprir determinadas tarefas. O principal resultado desta nova tendência foi o desenvolvimento do processador digital de sinais (*Digital Signal Processor* – DSP), que foi projetado para processar digitalmente informações freqüentemente obtidas em forma analógica em tempo real. Para alcançar esse objetivo, os DSPs apresentam várias modificações em relação a processadores de propósito geral, como por exemplo unidades de multiplicação implementadas em *hardware*.

Uma consequência natural do aumento da capacidade de processamento dos computadores foi a possibilidade de se encontrar soluções para sistemas cada vez mais complexos. Os primeiros métodos de processamento digital de sinais tinham como finalidade simular sistemas analógicos (DUDGEON e MERSEREAU, 1984). Contudo, verificou-se que sistemas digitais poderiam fazer muito mais do que simplesmente simular sistemas analógicos de forma eficiente e a área de processamento digital de sinais começou a se ampliar rapidamente. Sistemas digitais são mais flexíveis e eficientes que sistemas analógicos e, atualmente, muitos dos métodos de processamento digital de sinais não possuem equivalentes analógicos.

Durante as décadas de sessenta e setenta, devido ao alto custo dos computadores digitais, o processamento digital de sinais se restringia a algumas aplicações críticas. O processamento de sinais de radares e sonares, assim como o processamento de sinais biomédicos foram algumas das áreas pioneiras a utilizar métodos digitais (SMITH, 1999). Após a revolução dos computadores pessoais nas décadas de oitenta e noventa, a área se expandiu na mesma proporção e, hoje, o processamento digital de sinais está presente em inúmeros equipamentos cotidianos, como modems, telefones celulares ou CD *players*.

Devido à grande quantidade de informações a serem processadas, e a conseqüente necessidade de grandes quantidades de memória e processadores de elevado desempenho, o processamento de sinais multidimensionais é uma área que surgiu mais recentemente. Entretanto, os avanços obtidos na área neste período relativamente curto demonstram seu grande potencial para solucionar diversos dos problemas atuais da civilização.

Um dos principais tipos de sinais multidimensionais que se pretende processar e que vem sendo objeto de muitas pesquisas são imagens. Embora as primeiras aplicações de processamento de imagens datem da década de vinte (GONZALEZ e WOODS, 1992), a área somente passou a ter avanços significativos a partir dos computadores digitais. Desde então, o processamento digital de imagens se expandiu de tal forma que são inúmeras suas aplicações práticas.

Em geral, sistemas de aquisição e processamento de imagens são compostos por cinco elementos fundamentais (GONZALEZ e WOODS, 1992). O primeiro elemento do sistema é um dispositivo sensível a alguma região do espectro eletromagnético. Este elemento pode ser um sensor de raios-X ou arranjos de fotodetectores. Acoplado ao sensor há um elemento para conversão dos sinais obtidos, em geral na forma analógica, para a forma digital. O próximo elemento necessário em sistemas de processamento de imagens é um dispositivo de armazenamento dos sinais digitalizados, como discos rígidos ou memórias semicondutoras. O terceiro elemento do sistema é o dispositivo de processamento das imagens. Atualmente, a maioria dos sistemas de processamento de imagens emprega processadores de propósito geral para esta função, mas a busca por maior desempenho em sistemas de alta complexidade tem levado ao crescente desenvolvimento de *hardwares* específicos para esta finalidade. Dependendo da aplicação do sistema, podem ainda ser necessários elementos de transmissão e apresentação das imagens, como *links* de satélite e monitores de computadores.

Um dos ramos do processamento digital de imagens que já se demonstrou muito promissor é a denominada visão computacional. Em sistemas de visão computacional as imagens são adquiridas seqüencialmente de modo a formar um sinal de vídeo, do qual são

extraídas informações utilizadas pelo sistema para diversos propósitos. A área de visão computacional já apresenta diversas aplicações bem sucedidas em navegação de robôs (GOTO et al., 1999), sistemas de manufatura (CONCI e PROENÇA, 2002), controle de tráfego de veículos (TECHMER, 2001), vigilância (WU et al., 2003), entre inúmeras outras.

Tradicionalmente, sistemas de visão computacional são compostos por uma câmera de aquisição de vídeo, a qual produz um sinal analógico no formato RS-170. Para digitalizar o sinal gerado pela câmera, normalmente são utilizadas placas de aquisição de vídeo (*frame grabbers*), as quais acoplam as câmeras ao último elemento do sistema, um microcomputador de propósito geral que executa as tarefas de processamento, armazenamento e apresentação das imagens. Entretanto, embora tais sistemas apresentem bons resultados em diversas áreas, são muitas as suas limitações.

As câmeras utilizadas atualmente, em sua maioria, utilizam sensores de imagem do tipo *charge coupled devices* (CCD). Sensores do tipo CCD fornecem sinais analógicos, estes sinais devem ser digitalizados para que possam ser transformados em sinais de vídeo. Dessa forma, uma câmera baseada em um sensor CCD que forneça um sinal de vídeo analógico, como o formato RS-170, por exemplo, deve realizar duas conversões analógico-digital e digital-analógico antes de fornecer o sinal para o próximo elemento do sistema, o qual deverá novamente converter o sinal para a forma digital para poder processá-lo. Cada etapa de conversão introduz erros e ruídos ao sinal original, consome tempo, aumenta os custos e a potência despendida pelo sistema, reduzindo seu desempenho como um todo.

Outra característica das câmeras tradicionais é que a aquisição das imagens e a formação dos quadros do sinal de vídeo são realizadas de forma pré-programada e fixa, sem a possibilidade de qualquer tipo de controle externo. Esta característica não permite flexibilidade no intervalo de aquisição das imagens e faz com que sistemas que exijam qualquer tipo de sincronização entre aquisição e processamento se tornem extremamente complexos quando não completamente inviáveis.

Além disso, o formato de varredura entrelaçado utilizado pelas câmeras comuns, no qual dois campos de informação são adquiridos em um intervalo de 1/60 segundos e apresentados simultaneamente com suas linhas entrelaçadas como se fossem uma imagem obtida em um único instante, pode apresentar sérios problemas em aplicações nas quais os objetos observados se movimentam. Se o objeto observado se movimentar em relação à câmera, ou vice-versa, a uma velocidade tal que seu deslocamento em 1/60 segundos possa ser observado, a imagem formada terá deslocamentos entre linhas (*zig-zag pattern*), os quais

freqüentemente impossibilitam a aplicação da maioria dos algoritmos de processamento digital de imagens (EASTON, 1996).

Sistemas de processamento baseados em microcomputadores de propósito geral também apresentam limitações. O processamento digital de imagens, assim como o de qualquer outro tipo de sinal, exige grandes quantidades de cálculos e seu desempenho está diretamente relacionado à quantidade de adições e multiplicações que um processador pode executar a cada ciclo de *clock* (SMITH, 1999). A arquitetura de processadores de propósito geral não é projetada especificamente para este tipo de tarefa e, embora seja possível lhes integrar funcionalidades deste tipo, os custos em termos de consumo de potência e preço são proibitivos.

Em resumo, um sistema embarcado de aquisição e processamento de imagens, composto por um sensor de imagem e um processador digital de sinais pode ser construído de forma a ser mais eficiente e flexível do que a abordagem tradicional utilizando-se uma câmera de vídeo, uma placa de aquisição e um microcomputador pessoal. Sistemas deste tipo vêm sendo desenvolvidos por diversos grupos de pesquisa (BRAMBERGER et al., 2004; MATSUSHITA et al., 2003; WOLF et al., 2003a, 2003b; LEE e SOBH, 2001) e seu elevado desempenho, aliado ao seu menor custo, consumo de potência e tamanho em relação aos sistemas tradicionais, levam a crer que a tendência no futuro é de que tais sistemas dominem o mercado de processamento digital de imagem.

Embora a expressão câmera inteligente (*smart camera*) seja ainda confusa, diversos pesquisadores (BRAMBERGER et al., 2004; WOLF et al., 2003a; LEE e SOBH, 2001) concordam que o termo se aplica a um dispositivo portátil, capaz de adquirir imagens, realizar processamento e fornecer informações sobre as imagens processadas. Atualmente existem diversas arquiteturas de câmeras inteligentes, mas, de forma geral, todas se enquadram no diagrama em blocos ilustrado na figura 1.

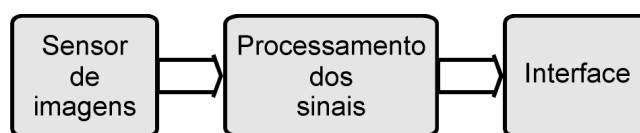


Figura 1 – Diagrama em blocos de uma câmera inteligente.

O sensor de imagem consiste normalmente em um único sensor de estado sólido do tipo CCD ou CMOS (*Complementary Metal Oxide Semiconductor*). O bloco de processamento dos sinais é, sem dúvida, o mais amplamente discutido atualmente.

Implementações variam desde dispositivos programáveis como FPGAs (*Field Programmable Gate Arrays*) e ASICs (*Application Specific Integrated Circuits*), processadores digitais de sinais, até *hardwares* específicos integrados ao próprio sensor de imagem. O terceiro bloco do sistema, responsável pela interface com outros dispositivos, permite que a câmera receba comandos e/ou entregue informações específicas obtidas a partir do processamento das imagens realizado em tempo real.

Assim como existem diversas arquiteturas de câmeras inteligentes, suas aplicações são ainda mais numerosas. Câmeras inteligentes podem substituir os sistemas tradicionais de processamento de vídeo e possibilitar muitas outras aplicações até recentemente inviáveis pelas limitações impostas pelos sistemas tradicionais.

No controle de linhas de produção industrial, as câmeras inteligentes já são amplamente empregadas e existem vários sistemas comerciais disponíveis no mercado (NATIONAL INSTRUMENTS, 2004; ARVOO 2003; CORECO 1998). O principal modelo sobre o qual se estruturam os sistemas comerciais empregados em aplicações industriais é o de câmeras inteligentes programáveis pelo usuário, as quais são integradas a *softwares* dotados de bibliotecas de processamento digital de imagens compostas de funções úteis para o controle de produção e inspeção de qualidade de produtos. Por meio do *software*, o usuário define o processamento que a câmera deve executar e as ações que deve tomar. Vários sistemas possuem inclusive interfaces para controladores lógicos programáveis (CLPs), permitindo que as câmeras possam atuar diretamente na linha de produção. Um exemplo desse tipo de sistema é o CVS-1450 da National Instruments (NATIONAL INSTRUMENTS, 2004), o qual conta com uma interface para até três câmeras digitais, rede Ethernet e entradas e saídas digitais. Embora apresente bom desempenho e permita, em muitos casos, processamento em tempo real, a utilização de um processador de propósito geral como unidade de processamento faz com que seu consumo de potência seja bastante elevado, podendo alcançar até 22 watts, inviabilizando sua aplicação em sistemas móveis. Diversas aplicações de câmeras inteligentes em ambientes industriais são descritas por Lee et al. (1993, 1994, 2000) e Conci e Proença (2002).

Pesquisas sobre sistemas de monitoração e/ou controle de tráfego de veículos, devido à dificuldade em se enviar dados a um computador central e a necessidade de processamento em tempo real, podem se beneficiar com o uso de câmeras inteligentes. Câmeras inteligentes podem realizar, em tempo real, tarefas de determinação da velocidade média dos veículos, detecção de acidentes ou veículos parados na pista, reconhecimento de placas de identificação (IMBIRIBA, 2002) e compactação e criptografia de imagens (FERRAZ, 2004) a serem

transmitidas para uma estação central. Bramberger et al. (2004) descrevem um sistema composto por um sensor de imagem CMOS, dois processadores digitais de sinais de alto desempenho e uma rede sem fio WLAN (*Wireless Local Area Network*). A avaliação demonstrou que o sistema é capaz de detectar veículos estacionários em uma rodovia e informar a um computador central a uma taxa de 2,4 quadros por segundo. A esta velocidade é possível informar à unidade central de controle de tráfego a ocorrência de situações perigosas em uma rodovia em aproximadamente 417ms.

Na área de engenharia biomédica, protótipos de sistemas de visão empregando câmeras inteligentes estão sendo desenvolvidos por diversos grupos de pesquisa. Schwiebert et al. (2002) propõem um protótipo de retina artificial para ser implantada em pacientes portadores de deficiência visual. O sistema é composto de uma câmera inteligente acoplada a óculos que transmitem imagens a um microsensor implantado no olho do paciente. O microsensor é capaz de estimular a retina do paciente por meio de impulsos elétricos. Embora o trabalho de Schwiebert e colaboradores esteja apenas em um estágio inicial, e muitas pesquisas em diversas áreas ainda sejam necessárias até que próteses de retinas possam ser utilizadas por portadores de deficiências visuais, a abordagem de se utilizar uma câmera inteligente vêm apresentando resultados e parece promissora.

Outra área que vem apresentando bons resultados nos últimos anos e que conta inclusive com diversos produtos no mercado (GADELLA, 1997) é a microscopia baseada em fluorescência resolvida no tempo. Essa técnica utiliza membranas, as quais, uma vez excitadas por uma fonte luminosa, apresentam fluorescência cujo tempo de decaimento é proporcional à concentração de um determinado tipo de reagente. Por meio de sensores de imagem do tipo CCD e técnicas de processamento de imagens é possível detectar esse decaimento e gerar imagens representando a concentração espacial do reagente. Microscópios baseados em fluorescência resolvida no tempo são utilizados atualmente para gerar imagens de diversos tipos de reagentes químicos e mesmo organismos biológicos. A técnica da fluorescência resolvida no tempo não se restringe, entretanto, à microscopia. Trabalhos desenvolvidos por Jakubiak et al. (2000) e Branco et al. (1996) demonstraram a viabilidade de se produzir membranas sensíveis à concentração de oxigênio em substratos de polímeros excitadas por LEDs (*Light Emitting Diodes*), cujas respostas encontram-se dentro do espectro visível. Acoplando-se tal sistema a uma câmera CCD de alto desempenho, é possível gerar imagens da concentração de oxigênio em áreas em contato com a membrana, como, por exemplo, tecidos biológicos.

Outra área da engenharia biomédica em que câmeras inteligentes são amplamente utilizadas é na construção de ambientes de realidade virtual para microcirurgias ou cirurgias à distância. Ohba et al. (2000) propõem a utilização do sistema MAPP2200 (IVP, 1997), o qual integra um sensor de imagem CMOS e um DSP em um único circuito integrado, acoplado a uma microlente com ajuste eletrônico de foco para desenvolver um sistema de realidade virtual para microcirurgias. O sistema proposto por Ohba e colaboradores é capaz de gerar modelos tridimensionais do ambiente da cirurgia, no qual tanto o objeto da cirurgia como os instrumentos cirúrgicos permanecem constantemente focalizados e a noção de profundidade, isto é, a terceira dimensão do ambiente é mantida. Embora os primeiros resultados obtidos pelo projeto exijam um tempo de processamento de dois segundos entre os quadros, os pesquisadores pretendem obter um protótipo capaz de realizar o processamento a uma taxa de 30 quadros por segundo.

Bastante relacionadas à realidade virtual, estão as pesquisas sobre ambientes inteligentes. Ambientes inteligentes são capazes de detectar a presença de indivíduos, suas ações e auxiliá-los em suas tarefas cotidianas (ABOWD et al., 2002). Esses ambientes visam melhorar a qualidade de vida e tentar aumentar a independência de idosos ou qualquer pessoa com limitações físicas. Da mesma forma que os ambientes de realidade virtual, ambientes inteligentes precisam construir modelos digitais do mundo real em alta velocidade, de modo a obter informações que lhes permitam interagir com o usuário. O principal desafio para se obter ambientes inteligentes eficientes é o reconhecimento de atividades humanas em tempo real. Wolf e Ozer (2001) acreditam que para obter êxito na tarefa de reconhecer eficientemente atividades humanas em tempo real são necessárias pesquisas simultâneas sobre os algoritmos de reconhecimento das imagens e as arquiteturas nas quais os algoritmos são implementados. Baseados nessa filosofia, eles desenvolveram uma câmera inteligente baseada em sensores de imagem CCD, processadores digitais de sinais de alto desempenho e um computador pessoal. Os resultados obtidos pelo protótipo desenvolvido por Lv et al. (2002) alcançaram desempenho em tempo real e demonstraram que uma câmera inteligente composta por dois processadores digitais de sinais de alto desempenho e um processador de propósito geral operando com um espaço de memória compartilhado é adequada para a tarefa. Os pesquisadores estão desenvolvendo trabalhos atualmente sobre a estrutura mais adequada para se integrar câmeras inteligentes em rede de forma a possibilitar a distribuição das tarefas entre as câmeras (WOLF et al., 2003b).

Em ambientes de realidade ampliada (FEINER et al., 1993), isto é, ambientes em que elementos gráficos gerados por computador são inseridos em imagens do mundo real, as

câmeras inteligentes também são amplamente empregadas. Matsushita et al. (2003), apresentam um sistema composto por uma câmera inteligente acoplada a um *Personal Digital Assistant* (PDA), capaz de localizar dispositivos de identificação no ambiente e apresentar ao usuário imagens de realidade ampliada, por exemplo, condições de tráfego em trechos de rodovias. A câmera inteligente consiste em um sensor de imagem CMOS para aquisição de imagens em alta velocidade, uma FPGA para processamento dos sinais, e uma interface USB (*Universal Serial Bus*) para conexão com o PDA. Os dispositivos de identificação são fontes luminosas que piscam em alta frequência, emitindo, de forma serial, seu código de identificação. Ao localizar o identificador, a câmera decodifica seu código e acrescenta os elementos de realidade ampliada nas coordenadas da imagem definidas por ele.

O interesse em se desenvolver câmeras inteligentes é tanto que diversos grupos de pesquisa compostos por pesquisadores de várias instituições trabalham de forma conjunta para obter arquiteturas eficientes e viáveis. Na União Européia, o projeto IST-2001-34410 CAMELLIA (*Core for Ambient and Mobile Intelligent Imaging Applications*), que teve início em abril de 2002, está sendo desenvolvido por uma parceria entre universidades e centros de pesquisa privados. O consórcio para o desenvolvimento do projeto conta com a participação da *Philips Research* da Holanda, *Philips Semiconductors* de Hamburgo, da Renault, da *École des Mines* de Paris e das universidades de Hannover na Alemanha e Las Palmas na Espanha. O objetivo do projeto é desenvolver uma câmera inteligente eficiente, flexível e de baixo custo, capaz de executar em tempo real diversos algoritmos de processamento de imagem, desde compactação de vídeo a localização e rastreamento de objetos. Inicialmente, as principais aplicações da câmera seriam em veículos automotivos, ambientes inteligentes e em comunicações móveis. A câmera desenvolvida no projeto CAMELLIA consiste em um processador de propósito geral conectado a processadores de imagem desenvolvidos especificamente para o projeto, uma interface de entrada para os dados do sensor de imagem e uma interface de saída (JACHALSKY et al., 2003). O processador de propósito geral tem a finalidade de coordenar a comunicação entre os processadores de imagem e realizar todas as tarefas de controle das interfaces e interpretação de comandos. Cada um dos processadores de imagem tem sua arquitetura otimizada para determinadas categorias de algoritmos de processamento de imagens. Contudo, o objetivo final do projeto é encontrar uma única arquitetura capaz de executar qualquer algoritmo de processamento de imagens de forma otimizada.

As aplicações de câmeras inteligentes são inúmeras e a tendência, conforme se demonstre sua eficiência, é que elas aumentem cada vez mais. Entretanto, não existe ainda um

consenso sobre a arquitetura mais adequada. Certamente, qualquer experimento de avaliação de desempenho de estruturas específicas representa uma contribuição científica importante para esta área ainda jovem, mas que tende a representar uma parte significativa do processamento digital de imagens no futuro próximo.

1.1 OBJETIVOS

O objetivo deste trabalho é projetar, implementar e avaliar o desempenho de um protótipo de uma câmera digital inteligente, capaz de realizar aquisição e processamento de imagens de forma flexível e em tempo real.

O protótipo proposto utiliza um sensor de imagem CCD acoplado a um processador digital de sinais de alto desempenho. Todo o controle do sensor de imagem, bem como dos circuitos auxiliares do sistema, é realizado por um segundo processador digital de sinais de baixo custo, permitindo que o processador de alto desempenho ocupe-se somente com as tarefas de processamento das imagens. A comunicação entre os dois processadores ocorre por meio de uma interface serial assíncrona. O sensor de imagem é resfriado por um dispositivo termo-elétrico de resfriamento baseado no efeito *Peltier*, com a finalidade de minimizar a geração de ruído térmico no interior do sensor, permitindo a aquisição de imagens com baixa intensidade luminosa. A câmera conta ainda com uma interface serial USB para troca de informações com outros dispositivos. Inicialmente, a interface permite que a câmera se comunique com um microcomputador, entretanto, devido à elevada taxa de transmissão permitida pela interface USB, as funcionalidades do protótipo podem ser ampliadas futuramente de forma que a câmera possa se comunicar com outras câmeras semelhantes de modo a realizar processamento distribuído de imagens. Por meio da interface USB, é possível que o usuário configure parâmetros da câmera como o tempo de exposição do sensor, a taxa de aquisição das imagens, entre diversos outros. Através da interface é possível ainda definir algoritmos a serem aplicados sobre as imagens adquiridas, como técnicas de filtragem, limiarização e equalização de histograma, bem como visualizar os resultados obtidos em tempo real.

O protótipo desenvolvido, devido à sua grande flexibilidade e desempenho, é uma importante ferramenta de pesquisa na área de processamento digital de imagens. O fato de se utilizar um processador digital de sinais como elemento central para o processamento das imagens, permite que qualquer algoritmo desenvolvido em um microcomputador, em

linguagens de programação de alto nível, ou mesmo em ferramentas de modelagem de algoritmos, possa ser facilmente utilizado na câmera.

1.2 ESTRUTURA DA DISSERTAÇÃO

Esta dissertação está dividida em cinco capítulos. O capítulo 1 apresenta uma introdução ao trabalho, mostrando as razões que motivaram o desenvolvimento do projeto assim como os objetivos a serem alcançados. No capítulo 2 é realizada uma breve revisão da fundamentação teórica necessária para a compreensão do projeto. São descritos o funcionamento dos sensores de imagem CCD, características da interface USB, além de alguns aspectos relevantes da arquitetura de processadores digitais de sinais e de seus sucessores na área de processamento de sinais multimídia, os *mediaprocessors*. O capítulo 3 apresenta as etapas envolvidas no desenvolvimento da câmera. São apresentados aspectos relevantes sobre o controle do sensor de imagem, a conversão analógico-digital dos sinais obtidos, o sistema de processamento das imagens e o *software* para comunicação com o sistema. No capítulo 4 são apresentados resultados de alguns experimentos práticos realizados com o sistema, mostrando a resposta do sensor de imagem, a capacidade de transmissão e processamento do sistema e outras características relevantes. O capítulo 5 apresenta as conclusões obtidas com o trabalho, a discussão de seus aspectos mais importantes e propostas para trabalhos futuros.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

2.1 INTRODUÇÃO

Este capítulo apresenta um estudo e levantamento bibliográfico dos elementos envolvidos no desenvolvimento da dissertação. Na primeira seção é realizada uma breve introdução aos fundamentos dos elementos fotodetectores que compõem um sensor de imagem. Em seguida, são abordadas as técnicas de transferência de cargas e as arquiteturas de disposição dos fotodetectores em sensores de imagem do tipo CCD. A seção sobre sensores de imagem é encerrada com uma discussão sobre as principais fontes de ruídos em sensores CCD e algumas das técnicas para minimizá-los. Na seção seguinte, é descrito o funcionamento da interface USB. A principal finalidade desta seção é permitir a familiarização com os diversos termos da nomenclatura empregada pela especificação da interface, bem como conhecer todos os elementos envolvidos no processo de comunicação. Finalmente, na última seção deste capítulo, são apresentados os processadores digitais de sinais, com destaque às características comuns desta categoria de processadores e aos mais recentes processadores digitais de sinais de alto desempenho, conhecidos como processadores multimídia.

2.2 SENSORES DE IMAGEM

Sensores de imagem são compostos, de forma geral, por arranjos de fotodetectores em forma de matrizes, dispostos em uma ou mais linhas, acoplados a mecanismos para transferência das cargas coletadas pelos fotodetectores para circuitos de saída. As próximas seções apresentam conceitos básicos sobre os elementos que compõem os sensores de imagem.

2.2.1 Fotodetectores

Dispositivos semicondutores detectores de luz baseiam-se no princípio da geração de pares elétrons-lacunas em um material semicondutor, conforme ilustrado na figura 2.

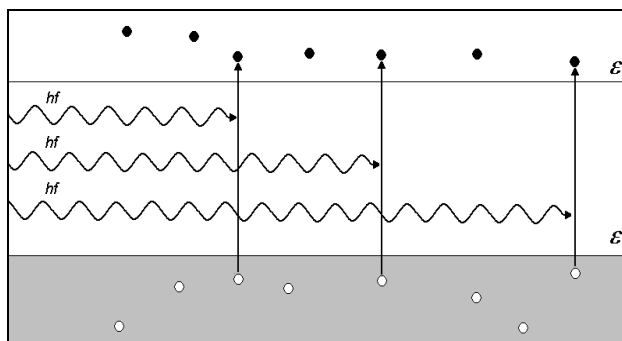


Figura 2 – Diagrama de bandas de energia de um material semicondutor ilustrando geração de portadores livres devido a absorção de fótons. Os fótons com energia hf são absorvidos pelo material deslocando portadores da banda de valência, cuja energia máxima é ϵ_v para a banda de condução, cuja energia mínima é ϵ_c . (modificado de Jenkins, 1987).

Os portadores existentes na banda de valência do material, cuja energia máxima é dada por ϵ_v , podem ser excitados para a banda de condução se receberem energia suficiente para ultrapassar a energia mínima da banda de condução, dada por ϵ_c . Isso ocorre quando fótons com energia superior à energia de *gap* ϵ_g do material, dada por $\epsilon_c - \epsilon_v$, incidem sobre o material. Os fótons são então absorvidos gerando pares elétron-lacuna. A energia de um fóton é dada pela equação de Planck (JENKINS, 1987):

$$\begin{aligned}\epsilon_{ph} &= hf \\ \epsilon_{ph} &= \frac{hc}{\lambda}\end{aligned}\tag{1}$$

onde ϵ_{ph} é a energia do fóton, c é a velocidade da luz, h é a constante de Planck, f e λ são a frequência e o comprimento de onda do fóton, respectivamente.

Assim, existe um limiar de absorção λ_{th} acima do qual não há absorção de fótons pelo material. Esse limiar, em condições ideais é obtido substituindo-se a energia de *gap* do material na equação 1:

$$\lambda_{th} = \frac{hc}{\varepsilon_g} \quad (2)$$

Fótons com comprimento de onda acima de λ_{th} não possuem energia suficiente para excitar os portadores através do *gap* de energia do material. Esse comprimento de onda limiar não é, entretanto, completamente estável e varia com a temperatura e a concentração de dopantes no material.

Os pares elétrons-lacunas, produzidos pela absorção dos fótons pelo material semiconductor, são coletados nos terminais do dispositivo, gerando sinais elétricos. Esse processo de coleta deve ser realizado antes que ocorra recombinação com outros portadores. Assim, esses pares devem ser gerados o mais próximo possível da região de depleção de uma junção P-N (reversamente polarizada), pois ao alcançarem a região de depleção são imediatamente afastados um do outro pelo campo elétrico presente na junção.

2.2.1.1 Capacitor MOS

Existem diversas estruturas de detectores ópticos semicondutores. A mais utilizada para a fabricação de sensores CCD é o capacitor MOS (*Metal Oxide Semiconductor*) (WILSON e HAWKES, 1983; RIEKE, 1994; BASS, 1995), o qual consiste em um substrato de silício recoberto por uma fina camada de dióxido de silício e um eletrodo, conforme ilustra a figura 3. Esta estrutura é utilizada em lugar da simples junção P-N, pois a capacitância que se forma entre o substrato e a camada de dióxido de silício é bastante superior à da junção, o que permite que o capacitor MOS armazene quantidades de cargas consideravelmente superiores (RIEKE, 1994).

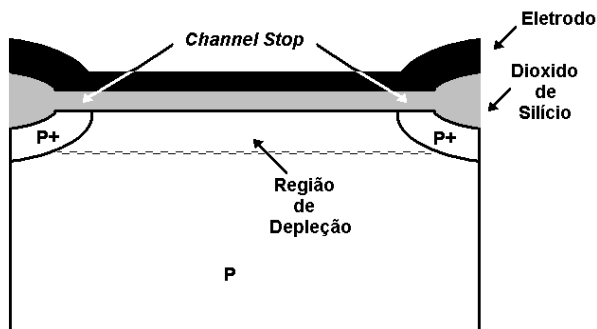


Figura 3 – Estrutura do capacitor MOS. Um substrato de silício dopado com impurezas (tipo P no exemplo) é conectado a um eletrodo por meio de uma interface de isolante (geralmente dióxido de silício). Regiões fortemente dopadas, conhecidas como *channel stop* limitam a extensão lateral da região de depleção formada no substrato do capacitor MOS quando o eletrodo é polarizado.

Na figura 3, considera-se que o substrato é composto de silício dopado tipo P, dessa forma, quando se aplica uma tensão positiva ao eletrodo, forma-se uma região de depleção entre a camada de dióxido de silício e o substrato tipo P. O diagrama de níveis de energia resultante dessa polarização é apresentado na figura 4.

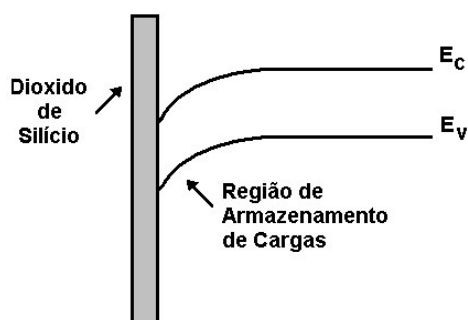


Figura 4 – Diagrama de níveis de energia do capacitor MOS polarizado. Na interface entre o substrato do capacitor e a camada de dióxido de silício é formada uma região com níveis de energia inferiores aos do substrato, conhecida como poço de potencial, na qual portadores livres podem ser armazenados.

Em ambas as extremidades do capacitor MOS existem regiões fortemente dopadas com impurezas do mesmo tipo do substrato. Essas regiões, conhecidas como *channel stop*, não sofrem depleção durante a polarização de forma que confinam os portadores gerados à região de depleção do capacitor, impedindo sua fuga lateral (RIEKE, 1994; BASS, 1995).

Quando um feixe de luz incide sobre um capacitor MOS, parte dele é refletida, parte é absorvida pelo eletrodo e o restante penetra no substrato de silício. Os portadores gerados no substrato movem-se por difusão até que sejam coletados na região de depleção ou que se recombinem. Os portadores coletados na região de depleção são armazenados na barreira de

potencial formada na interface entre o silício e o dióxido de silício, conforme indicado no diagrama da figura 4 (RIEKE, 1994).

Um dos principais problemas do capacitor MOS é a grande absorção de luz que ocorre nos eletrodos para os baixos comprimentos de onda. A figura 5 apresenta o coeficiente de absorção do silício policristalino fortemente dopado com fósforo, um dos materiais mais utilizados nos eletrodos de capacitores MOS. O elevado coeficiente de absorção na faixa de 400 nm faz com que a eficiência de conversão de fótons em elétrons, isto é, a eficiência quântica do dispositivo seja baixa para a luz azul (RISTIC, 1994).

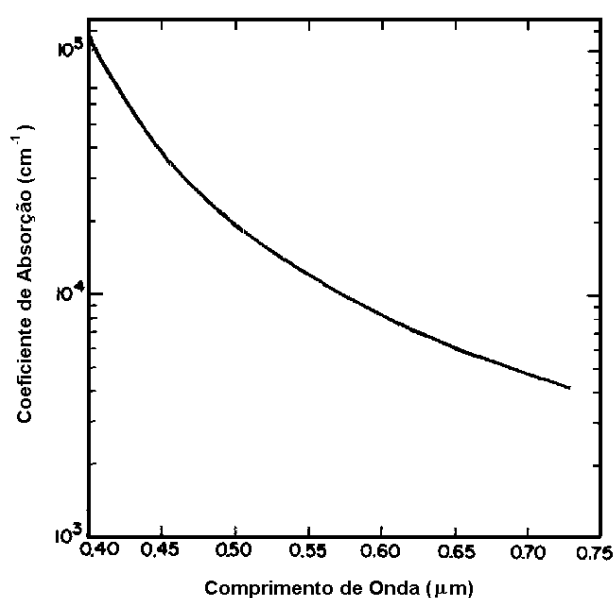


Figura 5 – Coeficiente de absorção do silício policristalino fortemente dopado com fósforo (modificado de Ristic, 1994).

Para superar o problema da baixa eficiência dos capacitores MOS nos pequenos comprimentos de onda, utiliza-se um eletrodo transparente como o óxido de índio-estanho (ITO), o qual possui baixíssima absorção na faixa de comprimentos de onda visíveis (KOSMAN et al., 1990).

Devido às imperfeições da rede cristalina na região de interface entre o substrato e o dióxido de silício, surgem nessa região níveis de energia conhecidos como armadilhas que podem aprisionar portadores e impedir que eles sejam coletados durante a leitura do dispositivo. Para superar esse problema foi desenvolvida a estrutura conhecida como capacitor MOS de canal enterrado (HARTGRING e KLEEFSTRA, 1978). Nessa estrutura, uma região tipo N levemente dopada é inserida entre o substrato e a camada de dióxido de silício, conforme representado na figura 6. Quando a estrutura é polarizada, a região de

depleção se estende por toda a região levemente dopada, formando um poço de potencial a poucos micrometros de distância da interface, de forma que os portadores gerados são armazenados nesse poço de potencial e não correm riscos de serem capturados pelas armadilhas (RIEKE, 1994; RISTIC, 1994; BASS, 1995).

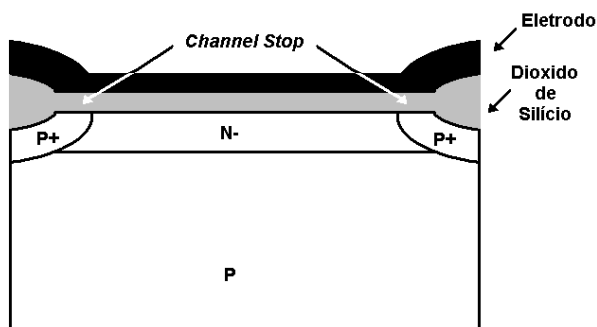


Figura 6 – Capacitor MOS de canal enterrado. Entre o substrato e a camada de dióxido de silício é inserida uma região levemente dopada com portadores do tipo oposto ao do substrato, o que faz com que o poço de potencial se forme a alguns micrometros da interface entre o substrato e o dióxido de silício.

2.2.2 Mecanismo de Transferência de Cargas

Sensores de imagem são compostos por arranjos de fotodetectores adjacentes. O fotodetector mais utilizado é o capacitor MOS de canal enterrado. Em sensores de imagem do tipo *charge coupled device* a carga armazenada em cada capacitor é transferida ao capacitor adjacente utilizando-se eletrodos posicionados sobre o capacitor. A polarização sequencial dos eletrodos faz com que as regiões de depleção dos capacitores movimentem-se, deslocando as cargas armazenadas até que alcancem o elemento de saída do sensor (JENKINS, 1987; RIEKE, 1994; BASS, 1995).

2.2.2.1 Transferência de Cargas com Três Fases

As figuras 7 (a) e (b) ilustram o processo de transferência de cargas em sensores de imagem que utilizam três eletrodos sobre cada elemento de armazenamento de cargas.

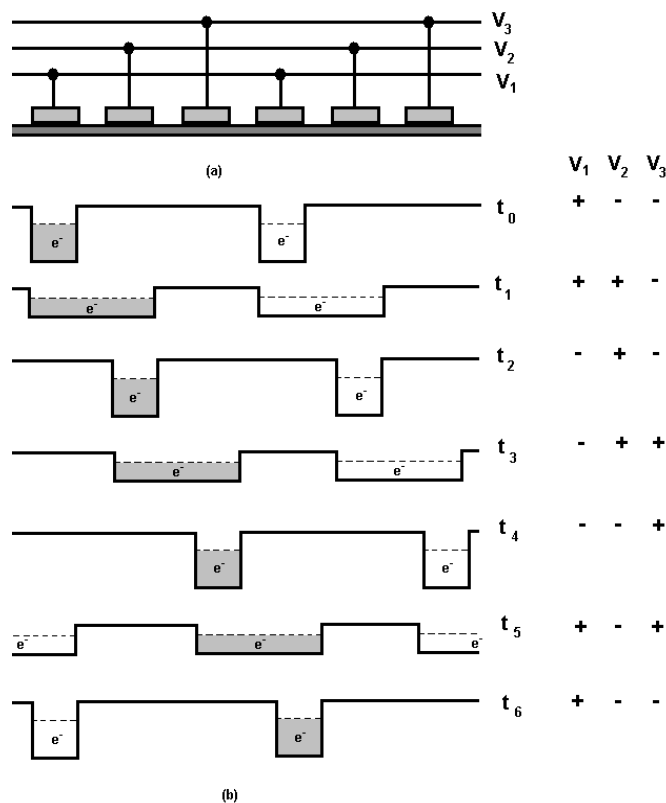


Figura 7 – (a) Secção transversal de duas células de um sensor CCD de três fases. (b) Ilustração do mecanismo de transferência de cargas. No instante de tempo inicial, o primeiro eletrodo de cada célula é polarizado, formando um poço de potencial sob ele. Para realizar a transferência de cargas, os eletrodos adjacentes são polarizados de forma a fazer com que os poços de potencial se expandam sob pares de eletrodos e, em seguida, comprimam-se sob os eletrodos seguintes, até que as cargas alcancem as células adjacentes.

Assumindo-se, com finalidade de ilustração, que o substrato do dispositivo apresentado na figura 7 seja do tipo P, quando um potencial positivo é aplicado ao eletrodo, forma-se uma região de depleção, ou poço de potencial, sob ele. Para facilitar a visualização do processo de transferência, a carga armazenada no capacitor MOS esquerdo foi destacada. Para que as cargas armazenadas nesse poço de potencial sejam transferidas, aplica-se inicialmente um potencial positivo sobre o eletrodo V_1 . Essa primeira etapa do processo está representada na figura 7(b) pelo instante de tempo t_0 , durante o qual as cargas permanecem concentradas sob o eletrodo V_1 . Em seguida, no instante de tempo t_1 , aplica-se um potencial positivo no eletrodo V_2 , de forma que o poço de potencial expande-se sob ambos os eletrodos. Em t_2 , o potencial positivo é retirado de V_1 , de modo que o poço de potencial concentra-se sob o eletrodo V_2 . No instante seguinte, o eletrodo V_3 é polarizado de forma que o poço de potencial expande-se novamente, mas dessa vez sob os eletrodos V_2 e V_3 . O processo se repete, retirando-se, em t_4 , o potencial do eletrodo V_2 , o que faz com que o poço se concentre sob V_3 . Em t_5 polariza-se novamente V_1 e as cargas se distribuem sob os eletrodos V_3 do

capacitor onde as cargas foram originadas e o eletrodo V_1 do capacitor adjacente. Finalmente, o processo é reiniciado, polarizando-se apenas V_1 de forma que as cargas são transferidas para o poço de potencial subjacente ao primeiro eletrodo do próximo capacitor (WILSON e HAWKES, 1983; RIEKE, 1994; BASS, 1995; KODAK 2001a).

CCDs que utilizam três eletrodos para realizar a transferência de cargas são conhecidos como CCDs de três fases. O mesmo processo de transferência de cargas pode ser utilizado em CCDs com quatro fases (RIEKE, 1994; BASS, 1995).

2.2.2.2 Transferência de Cargas com Duas Fases

Dopando-se diferentemente as regiões sob os eletrodos é possível obter poços de potencial com diferentes profundidades, o que possibilita a construção de CCDs de duas ou mesmo uma única fase (RIEKE, 1994; BASS, 1995; KODAK, 2001a). A figura 8 ilustra o processo de transferência de cargas em um CCD de duas fases, novamente o substrato do dispositivo foi definido como tipo P. No instante de tempo t_0 , quando o eletrodo V_1 é polarizado, a região sob o eletrodo com menor quantidade de dopante age como uma barreira de potencial, obrigando os portadores a se deslocarem para o poço sob o eletrodo mais fortemente dopado. Na próxima etapa da transferência, representada por t_1 , o potencial positivo é retirado de V_1 e aplicado em V_2 , formando duas barreiras de potencial que fazem com que os portadores se desloquem para a região mais fortemente dopada sob o eletrodo V_2 (RIEKE, 1994; BASS, 1995; KODAK, 2001a).

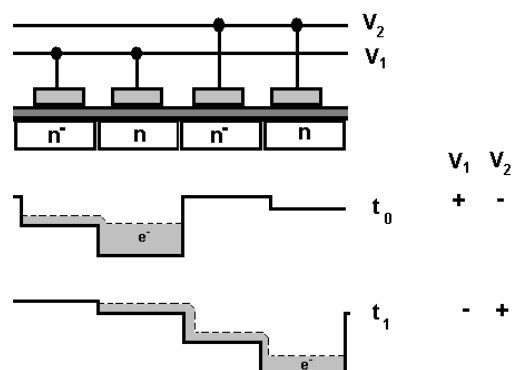


Figura 8 – Ilustração do mecanismo de transferência de cargas em um sensor CCD de duas fases. Devido à dopagem diferenciada sob os eletrodos, são formados poços de potencial de diferentes profundidades, de forma que, com a polarização dos eletrodos adjacentes, as cargas são forçadas a migrar direção ao eletrodo com maior dopagem.

Existem duas técnicas para a construção de CCDs de duas fases, na técnica apresentada na figura 8, cada fase é formada por dois eletrodos conectados. A dopagem sob um dos eletrodos é menor que a dopagem sob o outro, de forma que se formam os poços e barreiras de potencial. Uma outra abordagem é a de se construir cada fase utilizando-se somente um eletrodo e dopando-se diferentemente a região sob ele, conforme representa a figura 9 (BASS, 1995).

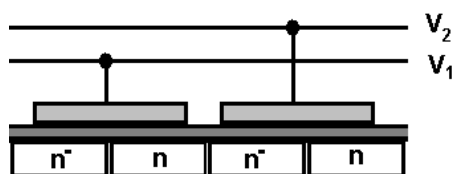


Figura 9 – Estrutura de um sensor CCD de duas fases utilizando dois eletrodos. Sob cada eletrodo existem duas regiões com dopagens diferenciadas.

O princípio de funcionamento do CCD de duas fases utilizando apenas dois eletrodos é idêntico ao do CCD de duas fases utilizando quatro eletrodos (o qual é conhecido como *pseudo-two-phase CCD*), com a vantagem que a área de cada *pixel* (*Picture Element*) pode ser reduzida devido à necessidade de apenas dois eletrodos sobre cada *pixel* (KODAK, 2001a).

CCDs de duas fases são amplamente utilizados devido à possibilidade de se utilizar circuitos de controle consideravelmente mais simples que os necessários em CCDs de quatro ou três fases (BASS, 1995).

2.2.2.3 Transferência de Cargas com Fase Virtual

Uma técnica semelhante à utilizada na construção de CCDs de duas fases pode ser utilizada para a construção de CCDs de uma única fase, ou fase virtual. Em CCDs de fase virtual, a região sob o eletrodo é construída utilizando-se dopagens diferentes como em CCDs de duas fases, de modo a se obter uma barreira de potencial e um poço de potencial sob o eletrodo. A região adjacente ao eletrodo é fortemente dopada, de forma a se obter uma nova barreira e um novo poço de potencial, como em um eletrodo mantido sob uma polarização constante, conforme ilustrado pela figura 10 (BASS, 1995).

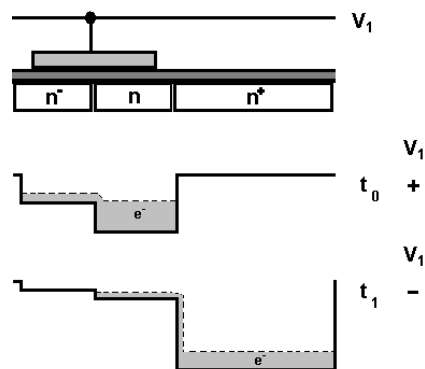


Figura 10 – Ilustração do mecanismo de transferência de cargas em um sensor CCD de fase virtual. Quando o eletrodo é polarizado, o potencial sob ele é inferior ao potencial da região não coberta pelo eletrodo (fase virtual). Quando a polarização é retirada do eletrodo, o potencial da região sob ele torna-se maior que o da fase virtual, obrigando os portadores a migrarem para ela.

Durante o período em que o eletrodo permanece polarizado, representado por t_0 na figura 10, a região da fase virtual age como uma barreira de potencial para os portadores armazenados sob o eletrodo. Em t_1 , quando a polarização é retirada do eletrodo, a situação se inverte e a carga armazenada flui para a fase virtual (BASS, 1995).

CCDs de fase virtual necessitam de circuitos de controle mais simples que CCDs com duas ou mais fases e apresentam maior eficiência quântica, devido ao fato que uma área consideravelmente menor do *pixel* é encoberta pelo eletrodo (KODAK, 2001a).

2.2.3 Dispositivo de Saída

Na saída do sensor CCD há um circuito para converter as cargas armazenadas em cada *pixel* em um sinal de tensão. O circuito mais utilizado é conhecido como *floating diffusion* acoplado a um amplificador seguidor de fonte (RIEKE, 1994; RISTIC, 1994; BASS, 1995; KODAK, 2001a), representado esquematicamente na figura 11.

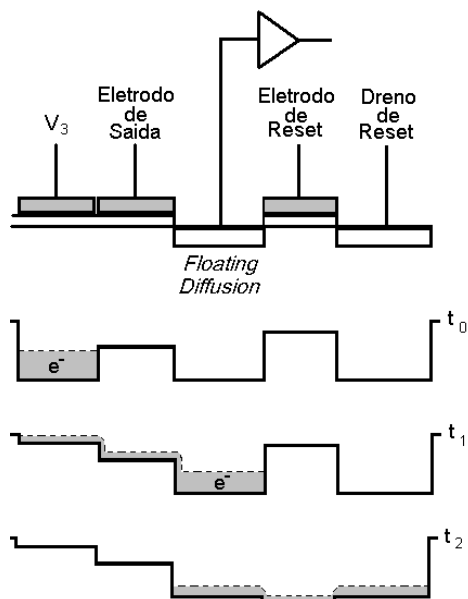


Figura 11 – Circuito *Floating Diffusion*, usado para converter as cargas elétricas em níveis de tensão na saída de sensores CCD. As cargas são transferidas para o *floating diffusion* devido à polarização do eletrodo de saída. O *floating diffusion* é conectado a um amplificador, o qual converte as cargas armazenadas em níveis de tensão. Após a conversão, as cargas são removidas do sensor através do dreno de *reset* devido à polarização do eletrodo de *reset*.

Conforme ilustra a figura 11, no instante t_0 as cargas são transferidas para o último eletrodo do CCD, no caso V_3 . O eletrodo de saída é mantido em um potencial constante de forma a criar uma barreira de potencial para as cargas armazenadas sob V_3 . Quando a polarização é retirada do eletrodo V_3 , as cargas são transferidas para o *floating diffusion* através do eletrodo de saída, conforme representado pelo instante de tempo t_1 . As cargas armazenadas no *floating diffusion* fazem com que a tensão na entrada do amplificador varie de acordo com a equação 3 (RIEKE, 1994):

$$V = \frac{Nq}{C} \quad (3)$$

onde N é o número de elétrons e C é a associação da capacitância do *floating diffusion*, da interconexão com o amplificador seguidor de fonte e da capacitância de entrada do amplificador.

Em geral a capacitância equivalente é da ordem de fentofarads, o que leva a grandes variações de tensão para pequenas quantidades de carga. A relação de conversão de cargas em tensão é um dos fatores mais importantes no projeto de sensores CCD. Atualmente, são

obtidas relações da ordem de dezenas de microvolts por elétron, lineares por várias décadas (RIEKE, 1994).

Após observar a variação de tensão no amplificador, as cargas armazenadas no *floating diffusion* precisam ser removidas antes que as cargas da próxima célula possam chegar. Para remover estas cargas, o eletrodo de *reset* é polarizado, conforme ilustrado pelo instante de tempo t_2 na figura 11. Neste instante, as cargas são transferidas para o dreno de *reset*, o qual é mantido em um potencial positivo, de modo que as cargas são removidas e o processo de leitura pode ser reiniciado (RIEKE, 1994).

2.2.4 Arquiteturas de Sensores CCD

Existem duas categorias de sensores de imagem de estado sólido: os lineares e os de área. Sensores lineares são utilizados em aplicações de geração de imagens por varredura, como, por exemplo, *scanners* ópticos. Sensores de imagem de área, por sua vez, são utilizados para aquisição de imagens sem a necessidade do processo de varredura. Um sensor de imagem de área é composto por um arranjo bidimensional de fotodetectores e registradores de deslocamento em duas direções. Existem três arquiteturas de sensores CCD de área: *full frame*, *frame transfer* e *interline* (RIEKE, 1994; RISTIC, 1994; BASS, 1995; INOUÉ e SPRING, 1997; KODAK 2001a).

2.2.4.1 Sensores Lineares

A figura 12 apresenta a estrutura de um sensor CCD linear (BASS, 1995; KODAK, 2001a).

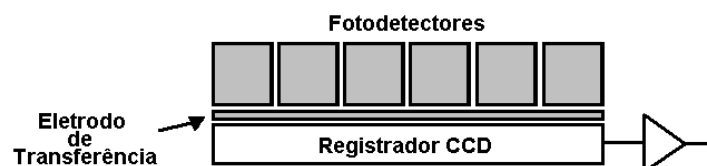


Figura 12 – Arquitetura do sensor CCD linear. Os elementos fotodetectores, dispostos em uma única linha são conectados, através de eletrodos de transferência, a um registrador serial responsável pelo deslocamento das cargas para o amplificador de saída do dispositivo.

O sensor consiste em uma linha de detectores ópticos (geralmente capacitores MOS de canal enterrado) acoplada a um registrador de deslocamento através de um eletrodo de transferência. Durante a operação, o sensor é deslocado até o início do objeto cuja imagem se deseja capturar (em geral um documento). Após o tempo necessário para que os fotodetectores se carreguem, as cargas armazenadas em cada um deles são transferidas simultaneamente para o registrador de deslocamento CCD. Em seguida, as cargas são transferidas seqüencialmente para o amplificador de saída, o sensor é deslocado para a próxima linha da imagem a ser adquirida e o processo é reiniciado (BASS, 1995; KODAK, 2001a).

Embora a figura 12 apresente a arquitetura mais comum em sensores CCD lineares, existem diversas variantes, como, por exemplo, estruturas que utilizam dois ou mais registradores de deslocamento e amplificadores de saída para acelerar o processo de leitura do sensor (BASS, 1995).

Para aquisição de imagens coloridas, são utilizados três sensores paralelos, construídos no mesmo chip de silício, espaçados normalmente pela distância equivalente a uma linha da imagem. Cada um dos sensores é recoberto por um filtro de cor de modo que cada sensor seja responsável pela aquisição de uma das componentes de cor da imagem (BASS, 1995).

2.2.4.2 Sensores de Imagem *Full Frame*

Um sensor de imagem *full frame* consiste em uma matriz de fotodetectores completamente exposta à luz, conforme ilustra a figura 13.

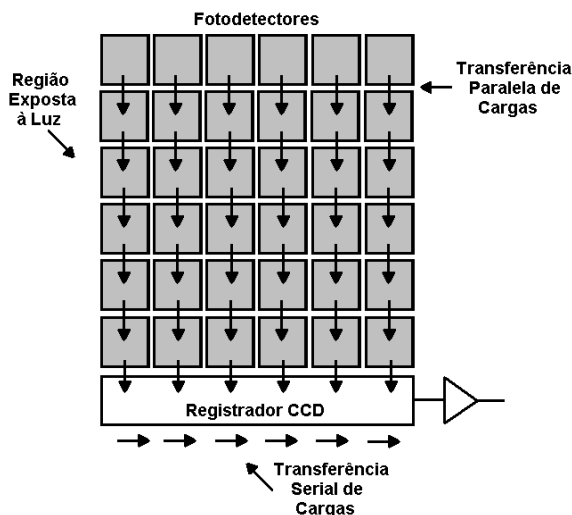


Figura 13 – Arquitetura e funcionamento do sensor CCD *Full Frame*. Após a exposição à luz dos elementos fotodetectores, dispostos em forma de matriz, as cargas armazenadas em cada linha do sensor são deslocadas para a linha subjacente. As cargas armazenadas na última linha são transferidas para o registrador serial, através do qual são transferidas seqüencialmente para o amplificador de saída do dispositivo. Esse processo é repetido para cada linha de elementos fotodetectores do sensor.

As cargas armazenadas nos fotodetectores de cada linha do sensor são transferidas para a linha subjacente pelo mecanismo CCD até que alcancem o registrador serial horizontal abaixo da última linha do sensor. As cargas armazenadas no registrador são então transferidas seqüencialmente para o amplificador de saída (RIEKE, 1994; RISTIC, 1994; BASS, 1995; INOUÉ e SPRING, 1997; KODAK, 2001a).

Embora os sensores CCD do tipo *full frame* apresentem excelente fator de preenchimento, isto é, espaçamento entre *pixels* bastante pequeno, eles apresentam a grande desvantagem de manter a região sensível à luz exposta durante a leitura dos *pixels*, o que pode gerar imagens borradas caso não seja utilizado um obturador mecânico. A necessidade de um obturador mecânico faz com que sensores *full frame* sejam utilizados em aplicações estáticas como fotografia digital (RIEKE, 1994).

2.2.4.3 Sensores de Imagem *Frame Transfer*

Para aplicações que exigem aquisição de imagens em alta velocidade, um obturador mecânico não é conveniente. Para superar essa dificuldade foram desenvolvidos os sensores CCD do tipo *frame transfer*. Um sensor *frame transfer* funciona de modo bastante semelhante ao *full frame*, exceto pelo fato que as imagens são transferidas inicialmente para uma região

não exposta à luz para somente então ser efetuada a leitura serial dos *pixels* (RIEKE, 1994; RISTIC, 1994; BASS, 1995; INOUÉ e SPRING, 1997; KODAK, 2001a). A figura 14 apresenta um diagrama de um sensor CCD do tipo *frame transfer*.

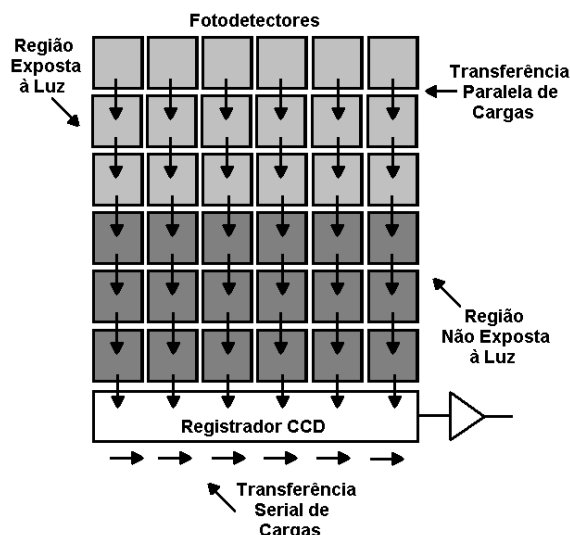


Figura 14 – Arquitetura e funcionamento do Sensor CCD *Frame Transfer*. O sensor possui uma região de elementos fotodetectores exposta à luz e outra região idêntica, porém protegida da incidência de luz. Após o período de exposição do sensor, as cargas presentes em cada linha exposta à luz são transferidas para a linha subjacente, até que todas as linhas sejam transferidas para a região de armazenamento. Em seguida, cada linha da região de armazenamento é transferida para a linha subjacente, de forma que a última linha é transferida para o registrador serial, através do qual os elementos são deslocados seqüencialmente para o amplificador de saída do sensor. Esse processo é então repetido para cada linha da região de armazenamento.

Uma vez que o tempo de transferência da imagem para a região não exposta à luz é bastante inferior ao tempo necessário para a transferência serial dos *pixels*, sensores CCD do tipo *frame transfer* apresentam consideravelmente menor borramento que sensores do tipo *full frame* operando sem obturadores (RIEKE, 1994), entretanto, em determinadas aplicações mesmo esse pequeno borramento pode não ser aceitável (INOUE e SPRING, 1997).

2.2.4.4 Sensores de Imagem *Interline*

Em sensores CCD do tipo *interline* colunas de elementos expostos e não expostos à luz são intercaladas conforme ilustra a figura 15.

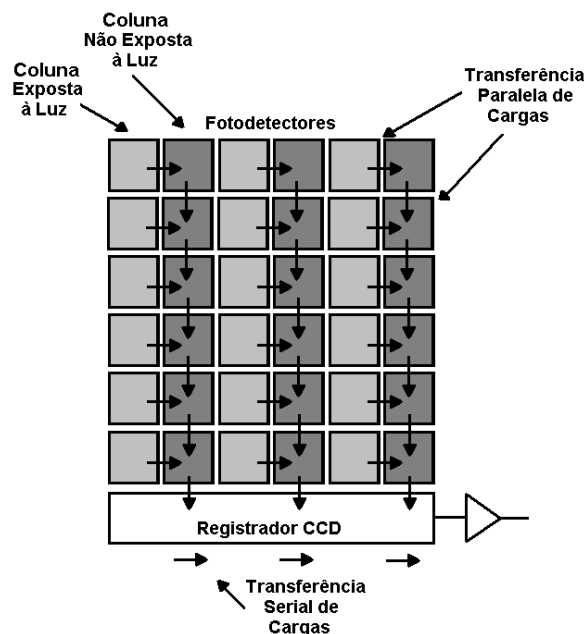


Figura 15 – Arquitetura e funcionamento do sensor CCD *Interline*. A cada linha de elementos fotodetectores exposta à luz é intercalada uma linha de elementos fotodetectores protegida da incidência de luz. Após a exposição do sensor à imagem, todos os elementos presentes nas colunas expostas são transferidos simultaneamente para as colunas protegidas de incidência de luz. Em seguida, cada linha de elementos protegidos é transferida para a linha subjacente, de forma que a última linha é transferida para o registrador serial, através do qual as cargas são deslocadas seqüencialmente para o amplificador de saída do dispositivo. Esse processo é repetido para todas as linhas do sensor.

Após o período de aquisição da imagem, as cargas armazenadas sob os elementos fotosensíveis são transferidas simultaneamente para os elementos não expostos à luz. Em seguida, as cargas são transferidas para o registrador horizontal para então serem transferidas seqüencialmente para o amplificador de saída (RIEKE, 1994; RISTIC, 1994; BASS, 1995; INOUÉ e SPRING, 1997; KODAK, 2001a).

Embora os sensores CCD do tipo *interline* apresentem consideravelmente menos borramento que os sensores do tipo *full frame* ou *frame transfer*, devido ao espalhamento da luz incidente sobre os elementos fotosensíveis, uma pequena quantidade de cargas é absorvida pelos elementos não expostos à luz, o que ocasiona também um pequeno borramento (BASS, 1995). Outro problema inerente aos sensores CCD do tipo *interline* é que seu fator de preenchimento é inferior ao dos outros tipos de CCD (WILSON e HAWKES, 1983).

2.2.5 Fontes de Ruído em Sensores CCD

Como sensores de imagem CCD consistem em arranjos de fotodetectores acoplados a um amplificador de detecção de cargas, todos os ruídos inerentes a ambos os tipos de dispositivos podem ser encontrados nos sensores.

O primeiro tipo de ruído ao qual o sensor CCD está sujeito é a flutuação provocada pela natureza aleatória da própria luz. É impossível prever com exatidão quantos fótons incidirão sobre o sensor em um determinado período de tempo. A incidência é aleatória e obedece à distribuição de Poisson, de forma que a incerteza associada à medição é igual à raiz quadrada do valor médio do sinal (PAPOULIS, 1991). Assim, considerando-se S o número de fótons incidente sobre o sensor, o valor do ruído dado pela incerteza do número de fótons incidentes, conhecido como *photon noise* (BARBE, 1975) ou *shot noise* (KODAK, 2001c) é dado por:

$$\sigma_p = \sqrt{S} \quad (4)$$

Outra fonte de ruído ocorre devido às imperfeições da rede cristalina no interior do dispositivo. Essas imperfeições dão origem a estados de energia dentro da banda proibida que podem vir a ser ocupados por elétrons de valência, os quais podem então se deslocar para a banda de condução e serem armazenados nos poços de potencial do dispositivo. A energia necessária para deslocar os elétrons para os estados gerados pelas imperfeições tem origem térmica e o ruído associado a esse fenômeno é conhecido como corrente de escuro (*dark current*), pois ocorre mesmo sem a presença de uma fonte de luz (AMELIO et al., 1971).

Uma vez que o número de imperfeições varia ao longo da área do sensor, uma diferente quantidade de corrente de escuro é gerada em cada *pixel* da imagem. Esse fenômeno dá origem a um ruído de padrão fixo (*Fixed Pattern Noise – FPN*), isto é, um ruído que não varia entre diferentes aquisições e que pode ser visto como uma espécie de “imagem de fundo”. Devido à natureza aleatória da geração térmica de portadores ao longo do tempo, a corrente de escuro também produz ruído com variação temporal, o qual obedece à distribuição de Poisson (BARBE, 1975).

A corrente de escuro apresenta uma forte dependência com a temperatura, portanto, resfriando-se o sensor, é possível reduzi-la consideravelmente. Segundo Barbe (1975), a corrente de escuro pode ser reduzida por um fator igual a 2, a cada 10°C de redução de temperatura.

Conforme descrito na seção 2.2.3, durante o período de *reset* do estágio de saída de sensores CCD, as cargas presentes na entrada do amplificador são removidas por meio de um dreno, o qual consiste, geralmente, em um canal MOS. O nível de tensão do amplificador é então levado a um valor conhecido. Entretanto, devido à resistência do canal, ruído térmico é introduzido no sinal, levando a uma incerteza no nível de tensão presente no amplificador antes do início do acúmulo de cargas do próximo *pixel*. Esse ruído, conhecido como ruído de *reset*, pode ser calculado pela equação 5 (CARNES e KOSONOCKY, 1972):

$$\sigma_r = \frac{\sqrt{kTC}}{q} \quad (5)$$

onde k é a constante de Boltzman, T é a temperatura, C é a capacitância do elemento sensor e q é a carga do elétron. O ruído de *reset* σ_r é dado pela equação 5 em número de elétrons.

Uma técnica bastante empregada para se minimizar os efeitos do ruído de *reset* é conhecida como amostragem dupla correlacionada (*Correlated Double Sampling – CDS*). Essa técnica consiste em se amostrar o sinal de saída do sensor CCD em dois momentos, um durante o período de *reset* do amplificador e outro durante o período em que o sinal do *pixel* está disponível (WHITE e BLAHA, 1974). A diferença entre os dois valores é então calculada, eliminando-se a necessidade de um nível de *reset* constante.

O amplificador de saída do CCD também está sujeito a ruído térmico. O ruído nesse caso é gerado devido à resistência de saída do amplificador e pode ser calculado, em volts, a partir da equação 6 (KODAK, 2001c). Por não apresentar relação com a frequência, o ruído térmico gerado no amplificador de saída do CCD é conhecido como ruído branco.

$$\sigma_w = \sqrt{4kTBR_{out}} \quad (6)$$

onde B é a largura de banda equivalente de ruído e R_{out} é a resistência de saída do amplificador.

O ruído de cintilação (*flicker*) ou ruído $1/f$, encontrado em praticamente qualquer sistema natural, apresenta uma relação aproximadamente linear com o inverso da frequência. O ruído de cintilação tem origem devido à comutação de estados de elementos com constantes de tempo aleatórias. No caso dos sensores CCD, os elementos que sofrem transições são portadores que são capturados por armadilhas na interface entre o silício e o isolante, isto é,

estados de energia discretos gerados na banda proibida por imperfeições na rede cristalina. As constantes de tempo são os tempos de permanência dos portadores nas armadilhas. Devido à dependência com o inverso da frequência, em geral, em sensores de imagem com altas frequências de leitura, o ruído de cintilação pode ser desprezado (KODAK, 2001c).

A última fonte de ruídos em sensores CCD deve-se às interferências geradas no sinal de saída pelos sinais de *clock* do dispositivo. Quanto maior a frequência de operação do sensor maior a interferência gerada pelos sinais e, conseqüentemente, maior o ruído de *clock* (KODAK, 2001c).

2.3 UNIVERSAL SERIAL BUS

A *Universal Serial Bus* é uma interface serial assíncrona desenvolvida por um grupo de grandes fabricantes de microcomputadores, *softwares* e periféricos com a finalidade de desenvolver um meio eficiente, simples e padronizado de se conectar microcomputadores e periféricos.

Uma rede USB consiste em um *host*, responsável pelo controle do tráfego, conectado a até 128 dispositivos. Para permitir a conexão de vários dispositivos ao *host* são utilizados *hubs*, conforme ilustra a figura 16. Um *hub* raiz (*root hub*) faz a interface entre o *host* e as portas USB propriamente ditas, nas quais os periféricos ou outros *hubs* são conectados.

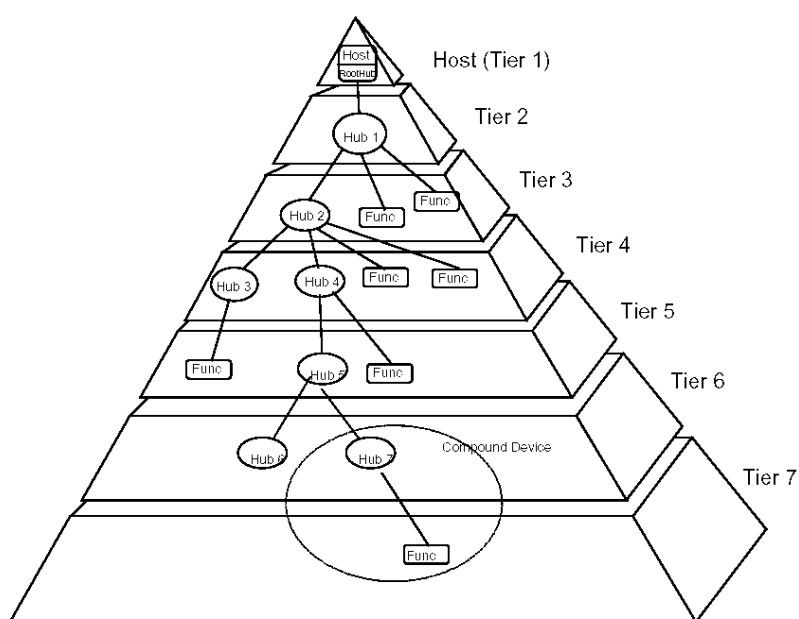


Figura 16 – Topologia do barramento USB apresentando a conexão de dispositivos ao *host* por meio de *hubs* (fonte: USB-IF, 2000).

Todos os dispositivos compartilham um único caminho físico para trocas de informações com o *host*, o computador que gerencia a interface. O *host* deve gerenciar o fluxo de dados, verificar erros nas transmissões, identificar a conexão de novos dispositivos e fornecer alimentação para os dispositivos que solicitem alimentação via barramento. O fluxo de dados do *host* para os periféricos é denominado *downstream* e o fluxo no sentido oposto *upstream* (USB-IF, 2000).

Cada dispositivo USB pode ser composto por até 16 *endpoints*. Um *endpoint* é um *buffer* capaz de armazenar dados de uma transmissão USB (AXELSON, 2001). Os dados armazenados pelo *endpoint* podem ser dados recebidos do *host* ou dados aguardando para serem enviados pelo dispositivo ao *host*. Cada *endpoint* em um dispositivo possui um endereço que pode ser um número entre 0 e 15 e uma direção de fluxo de dados (IN ou OUT). A especificação USB define a direção do fluxo de dados do ponto de vista do *host*, isto é, a direção IN define transmissões para o *host* e a direção OUT para o *endpoint* (USB-IF, 2000). Desta forma, a associação do endereço do dispositivo com o endereço do *endpoint* e sua direção de fluxo de dados define um endereço único em uma rede USB.

Todo dispositivo USB deve ter o *endpoint* 0 (zero), conhecido como *endpoint* de controle. O *endpoint* 0 consiste, na verdade, em um par de *endpoints*, um IN e outro OUT, possibilitando dessa forma a comunicação de dados em ambas as direções. Através do *endpoint* de controle, o *host* solicita e envia ao dispositivo informações de configuração.

Com a finalidade de atender a maior gama possível de dispositivos, quatro tipos de transferências são definidos pela especificação USB: controle, interrupção, isócrona e *bulk*.

Transferências de controle, que ocorrem geralmente entre o *host* e o *endpoint* 0, são transferências através das quais o *host* solicita informações de configuração do dispositivo. É o único tipo de transferência USB que apresenta comandos definidos pela especificação. Por meio destes comandos é possível identificar e configurar dispositivos sem a necessidade de qualquer tipo de interação do usuário. Esse processo é conhecido como enumeração e será abordado em maiores detalhes na próxima seção.

Transferências de interrupção são utilizadas por dispositivos que devem ser atendidos periodicamente pelo *host*. Nesse tipo de transferência, pequenos pacotes de dados são transmitidos em intervalos de tempo definidos. Mouses e teclados são exemplos de dispositivos que costumam utilizar transferências de interrupção.

Transferências isócronas, por sua vez, são utilizadas para transferir dados que exijam uma taxa de transmissão garantida. Isto é, o *host* aloca a quantidade de largura de banda

solicitada pela transferência antes de seu início, o que garante a transmissão dos dados dentro do tempo desejado. Em contrapartida, esse tipo de transferência não possui um sistema de controle de erros, ou seja, alguns bytes das informações transmitidas podem ser corrompidos. Transmissões isócronas podem ser utilizadas, por exemplo, para transmitir sinais de áudio, os quais podem aceitar erros ocasionais sem prejudicar sua qualidade, mas não toleram atrasos na transmissão.

As transferências do tipo *bulk* são utilizadas para transferir grandes quantidades de dados com correção de erro. A transferência do tipo *bulk* apresenta baixo *overhead* e é o mais eficiente tipo de transmissão em um barramento USB não compartilhado por vários dispositivos. Entretanto, quando o barramento é compartilhado, não há garantia de taxa de transmissão, isto é, o *host* atende a todos os dispositivos da melhor forma possível, dedicando uma parte da banda disponível para cada *endpoint*, o que pode levar a baixas taxas de transmissão por *endpoint* em barramentos congestionados. Transferências do tipo *bulk* são freqüentemente utilizadas por sistemas de aquisição de imagens como *scanners* e câmeras fotográficas digitais.

Transferências USB são compostas por transações. Transações são comunicações que devem ser completadas dentro de um único quadro de transmissão, isto é, transações não podem ser interrompidas. Um quadro de transmissão USB é um intervalo de um milissegundo definido pelo *host* através do envio de comandos de sincronismo específicos. A finalidade dos quadros de transmissão é permitir que o *host* administre o fluxo de informações dos diversos periféricos conectados a um único barramento USB, dedicando a cada periférico uma parcela do tempo disponível.

Uma transferência USB completa pode ocorrer dentro de uma única transação ou separada em diversas transações, caso consista em uma grande quantidade de dados. Uma transação contém informações de identificação (contendo endereço de dispositivo e *endpoint*), verificação de erros, *status* e controle. A especificação USB define três tipos de transações: *setup* para enviar solicitações de controle para um dispositivo, IN para receber dados de um dispositivo e OUT para enviar dados para um dispositivo.

Uma transação USB é composta por até três fases: *token*, dados e *handshake*. Cada fase, por sua vez, é composta por um ou dois blocos de informação com formatos definidos, os quais são denominados de pacotes. Cada pacote é composto por um campo de sincronismo (SYNC), um identificador de pacote (*Package Identifier* - PID) e outros campos que dependem do tipo de pacote, como por exemplo, endereços, dados, informações de *status* e

bits de verificação de erro (AXELSON, 2001). A tabela 1 apresenta os nove pacotes implementados na versão atual da especificação USB (USB-IF, 2000).

Tabela 1 – Tipos de pacotes especificados pela interface USB.

Tipo de Pacote (fase da transação)	Descrição	Nome do Pacote	PID	Descrição
Token	Identificam tipos de transações	OUT	0001	Endereça um <i>endpoint</i> para transações no sentido OUT (<i>host-dispositivo</i>)
		IN	1001	Endereça um <i>endpoint</i> para transações no sentido IN (<i>dispositivo-host</i>)
		SOF	0101	Marcador de início de quadro
		SETUP	1101	Endereça um dispositivo para transações de controle
Dados	Transmitem informações	DATA0	0011	Transmite dados
		DATA1	1011	Transmite dados
Handshake	Transmitem códigos de <i>status</i>	ACK	0010	Informa o recebimento correto de dados
		NACK	1010	Informa recebimento incorreto de dados
		STALL	1110	Informa solicitações que não são suportadas pelo dispositivo ou que o dispositivo não pode atender

No barramento USB, a comunicação ocorre por meio de dois sinais diferenciais D+ e D-, em modo NRZI (*Non-Return-to-Zero Inverted*). O modo NRZI é muito utilizado em sistemas assíncronos, pois permite uma boa sincronização dos dados transmitidos. A tabela 2 apresenta os três estados lógicos definidos pelo padrão USB (ZELENOVSKY e MENDONÇA, 2002).

Tabela 2 – Estados lógicos definidos pela interface USB.

estado	D+	D-
J	alto	baixo
K	baixo	alto
SE0	baixo	baixo

O estado SE0 reflete a ausência de dispositivos conectados ao barramento. As transições entre os estados lógicos J e K definem os valores dos bits no barramento. No modo NRZI, quando se deseja transmitir um bit 0, é necessário inverter os sinais presentes no barramento, ou seja, gerar uma transição entre os estados J e K durante o ciclo de *clock* seguinte. Quando se deseja transmitir um bit 1, o barramento deve permanecer no estado

anterior durante o próximo período de *clock* do sinal. Para evitar longos períodos sem transições, transmissões no modo NRZI frequentemente empregam bits redundantes, técnica conhecida como *bit stuffing* (ZELENOVSKY e MENDONÇA, 2002). No caso da especificação USB, a cada seis bits 1 transmitidos, um bit 0 deve ser inserido.

2.3.1 O Processo de Enumeração

Quando o *host* detecta uma nova conexão de um dispositivo ao barramento USB, ele dá início ao processo de identificação do dispositivo, técnica conhecida como “enumeração”. Durante o processo de enumeração, o *host* faz uma série de solicitações de informações por meio de transferências de controle para o *endpoint* 0 do dispositivo.

Durante o processo de enumeração, primeiramente o *host* define um endereço para o dispositivo, garantindo que cada dispositivo conectado à rede USB possua um endereço único. Após a definição do endereço do dispositivo, o *host* faz uma série de solicitações de informações ao dispositivo. As informações são retornadas pelo dispositivo por meio de descritores, os quais consistem em estruturas de dados com formatos definidos contendo informações que permitam a comunicação entre o dispositivo e o *host*. A tabela 3 apresenta os quatro descritores padrão definidos pela especificação USB que devem estar presentes em todo dispositivo USB (USB-IF, 2000).

Tabela 3 – Descritores definidos pela especificação USB.

Tipo de descritor	Descrição
Dispositivo	Contém informações gerais sobre o dispositivo
Configuração	Define as características de alimentação do dispositivo e o número de interfaces
Interface	Define classes de interfaces do dispositivo e o conjunto de <i>endpoints</i>
<i>Endpoint</i>	Contém informações sobre cada <i>endpoint</i> do dispositivo

O descritor de dispositivo é o primeiro descritor solicitado pelo *host* e contém informações como os números de identificação do fabricante e do produto, os quais são utilizados pelo sistema operacional para carregar o *device driver* adequado para o dispositivo em questão. No descritor de dispositivo é definido também o tamanho dos pacotes de dados do *endpoint* 0, esta informação é utilizada nas solicitações dos próximos descritores. Caso o

dispositivo pertença a uma classe padrão como *hubs* ou impressoras, a informação do tipo de classe do dispositivo também pode ser enviada pelo descritor de dispositivo.

O descritor de configuração contém informações sobre o tipo de alimentação utilizada pelo dispositivo (via barramento ou alimentação própria), a quantidade de corrente consumida e se o dispositivo pode atender a solicitações para entrar em modo de baixo consumo. O descritor de configuração também contém informações sobre o número de interfaces presentes no dispositivo.

Os descritores de interface contêm informações sobre o número de *endpoints* utilizados pelo dispositivo e, conseqüentemente, o número de descritores de *endpoints* que o *host* deve solicitar. Um único dispositivo pode contar com várias interfaces. Um dispositivo composto, isto é, um dispositivo com múltiplas funcionalidades que podem ser utilizadas simultaneamente, possui várias interfaces simultâneas. Em dispositivos que podem ser utilizados de várias formas diferentes, existem diversas interfaces que são habilitadas quando necessário. Da mesma forma que o descritor de dispositivo, os descritores de interface possuem campos para identificação da classe da interface, de modo que o dispositivo pode ser composto por várias interfaces de classes padrão. As informações contidas nos descritores de interface são utilizadas pelo sistema operacional juntamente com as informações do descritor de dispositivo para carregar *device drivers* para o dispositivo. Cada interface pode, inclusive, necessitar de um *device driver* específico.

Exceto pelo *endpoint* 0, cujas características devem seguir o padrão definido pela especificação para permitir o processo de enumeração, cada *endpoint* do dispositivo possui seu respectivo descritor de *endpoint*, o qual contém informações para permitir a comunicação entre o *host* e o *endpoint* específico. O descritor de *endpoint* define o endereço e a direção do *endpoint*, o tipo de transmissão a ser utilizada por este *endpoint* (interrupção, isócrona ou *bulk*) e o tamanho dos pacotes de dados do *endpoint*.

2.3.2 A Interface entre *Hardware* e *Software* no *Host*

Nos sistemas operacionais, a interface entre *hardware* e *software* é feita por meio de *device drivers*. Segundo Hazzah (1997), um *driver* é qualquer trecho de código que realize um acesso a um dispositivo de *hardware*. O sistema operacional Windows divide diferentes partes do acesso ao *hardware* em camadas de *drivers*, de forma que cada camada gerencia uma parte do processo de comunicação com o *hardware*. A vantagem deste sistema está na

possibilidade de que dispositivos que possuem tarefas semelhantes compartilhem *drivers*. A figura 17 apresenta a estrutura de camadas de *drivers*, conhecida como pilha de *drivers*, para a comunicação USB no Windows.

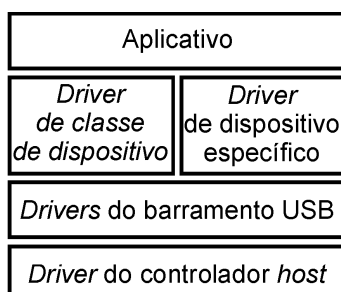


Figura 17 – Pilha de *drivers* USB no sistema operacional Windows. O *driver* do controlador *host* faz a interface com o *hardware*. Os *drivers* do barramento USB gerenciam a transferência. Os *drivers* de classes de dispositivos fazem a interface entre o aplicativo do usuário e dispositivos padrão como mouses ou impressoras, enquanto os *drivers* de dispositivos específicos fazem a interface com dispositivos que não pertencem a classes padrão.

A camada mais baixa da pilha de *drivers* da figura 17, no caso o *driver* do controlador *host*, faz a comunicação com o *hardware*. A camada superior, correspondente aos *drivers* do barramento USB, gerencia os processos de enumeração, fornecimento de alimentação aos dispositivos e as transações USB propriamente ditas. A separação entre as camadas de acesso ao *hardware* e de tratamento dos protocolos envolvidos na interface USB se justifica, pois existem diferentes tipos de controladores USB que podem ser utilizados no Windows e cada um deles necessita de um *driver* diferente. Essas duas camadas de *drivers* são incluídas no sistema operacional Windows.

Acima das camadas dos *drivers* do controlador *host* e do barramento USB, estão os *drivers* que fazem a interface entre os *drivers* do sistema operacional e os aplicativos. Esses *drivers* permitem que os aplicativos se comuniquem com dispositivos USB utilizando rotinas da API (*Application Programming Interface*) do sistema operacional. O Windows (HAZZAH, 1997), assim como a especificação USB (USB-IF, 2000) dividem dispositivos em classes, isto é, dispositivos com interfaces semelhantes pertencem a uma mesma classe. Dispositivos como impressoras, câmeras de vídeo ou dispositivos HID (*Human Interface Devices*) possuem *drivers* específicos, os quais são fornecidos pelo Windows. Caso o dispositivo não pertença a uma classe cujos *drivers* sejam fornecidos pelo sistema operacional, é necessário implementar um *driver* de dispositivo específico. Em geral, os próprios fabricantes dos dispositivos fornecem os *drivers* necessários para fazer a interface com seus produtos.

2.4 PROCESSADORES DIGITAIS DE SINAIS

Ao longo dos anos, computadores digitais comprovaram sua eficiência em duas áreas: manipulação de dados e cálculos matemáticos. Manipulação de dados consiste, em linhas gerais, em operações de ordenação e armazenamento de informações como, por exemplo, em um banco de dados. Embora cálculos matemáticos sejam necessários em sistemas de manipulação de dados, eles ocorrem com pouca frequência e têm pouca influência sobre o desempenho total do sistema. Por outro lado, algoritmos de processamento digital de sinais têm seu desempenho limitado quase que unicamente pela capacidade do processador de executar operações de multiplicação e adição (SMITH, 1999). Embora seja possível construir arquiteturas de computadores otimizadas para ambas as funções, a complexidade do sistema o tornaria inviável em diversos aspectos, em especial, custo e consumo de energia.

Para superar esses problemas, desenvolveu-se uma categoria de processador de propósito específico, cuja arquitetura foi otimizada para realizar operações matemáticas eficientemente. Os processadores desta categoria foram denominados processadores digitais de sinais (*Digital Signal Processors* – DSP).

Mais do que simplesmente executar operações matemáticas mais rapidamente que processadores de propósito geral, o objetivo dos DSPs é permitir o processamento de sinais em tempo real. Em outras palavras, um DSP deve ser capaz de processar sinais amostrados em um tempo inferior ao tempo de amostragem (KUO e LEE, 2001). Por exemplo, se um sinal deve ser amostrado a uma frequência de amostragem f_s , o DSP deve ser capaz de processar cada amostra do sinal em um tempo menor que $1/f_s$, de forma que a próxima amostra do sinal não seja perdida.

Arquiteturas de DSPs variam consideravelmente entre fabricantes. Hays (2004) acredita que essa diversidade de arquiteturas deve-se, principalmente, à relativa imaturidade da área e que, no futuro, as arquiteturas devem convergir para um núcleo comum aceito universalmente com diferenças apenas em pequenos detalhes, como ocorre atualmente com processadores de propósito geral. Entretanto, diversos pontos comuns entre DSPs disponíveis atualmente no mercado fazem com que se possa considerá-los dispositivos pertencentes a uma mesma categoria. Esses pontos comuns são determinados pelas necessidades computacionais dos algoritmos de processamento digital de sinais.

2.4.1 Unidade de Multiplicação e Acumulação

A grande maioria dos algoritmos de processamento digital de sinais baseia-se na operação de convolução discreta, representada pela equação 7:

$$y[n] = \sum_{k=0}^N x[n-k]h[k] \quad (7)$$

Conforme se pode observar na equação 7, a operação de convolução discreta consiste, essencialmente, em uma série de multiplicações e adições. Para cada valor de k , o resultado da multiplicação entre os valores de x e h deve ser somado ao valor de y . A esse processo dá-se o nome de acumulação. A operação de convolução é utilizada em algoritmos de filtros FIR (*Finite Impulse Response*) e IIR (*Infinite Impulse Response*), assim como em multiplicação de matrizes (fundamentais para processamento de sinais bidimensionais) e em inúmeras outras técnicas de processamento digital de sinais (DUDGEON e MERSEREAU, 1984; SMITH, 1999).

Para executar eficientemente operações de multiplicação e acumulação, processadores digitais de sinais contam com uma unidade MAC (*Multiply-Accumulate*). Uma unidade MAC consiste em uma unidade lógica aritmética (*Arithmetic Logic Unit – ALU*) em cujas entradas estão conectados a saída de um multiplicador e a saída da própria ALU, conforme ilustra a figura 18 (HAYES, 1986; SMITH, 1999).

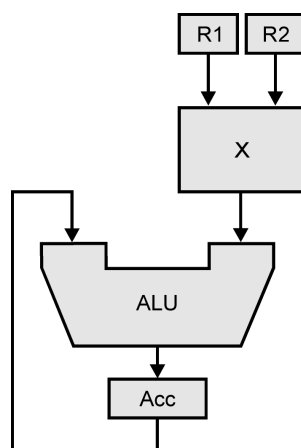


Figura 18 – Arquitetura da unidade de multiplicação e acumulação de um processador digital de sinais. Dois registradores são conectados à entrada de um multiplicador, cuja saída é conectada a uma unidade lógico-aritmética. À outra entrada da unidade lógico-aritmética é conectada sua própria saída através de um registrador acumulador.

Através dos registradores R1 e R2, os operandos são passados para o multiplicador cujo resultado, por sua vez, é enviado para ALU. Na ALU, o produto é adicionado ao valor presente no acumulador (Acc). Na maioria dos DSPs, esse processo pode ser executado em um único ciclo de *clock*.

2.4.2 Barramentos de Acesso à Memória

Um dos principais fatores limitadores do desempenho de processadores digitais de sinais é o acesso à memória (SMITH, 1999). O acesso à memória inclui a leitura das instruções que o processador deve executar, assim como os dados a serem processados. A maioria dos processadores de propósito geral é baseada na arquitetura *von Neuman*, na qual um único barramento é utilizado para acessar a memória de programa e a memória de dados.

Devido à natureza dos algoritmos de processamento digital de sinais, a maioria das instruções necessita de um ou dois operandos. Se a arquitetura *von Neuman* fosse utilizada, a cada instrução seriam necessários ao menos dois acessos seqüenciais à memória. A maioria dos DSPs utiliza a arquitetura conhecida como *Harvard*, na qual existem dois barramentos, um para acessar a memória de programa e outro para a memória de dados (BALCH, 2003). Por meio da arquitetura *Harvard*, é possível acessar uma instrução e um operando simultaneamente. As figuras 19 (a) e (b) ilustram sistemas com arquiteturas *von Neuman* e *Harvard*, respectivamente.

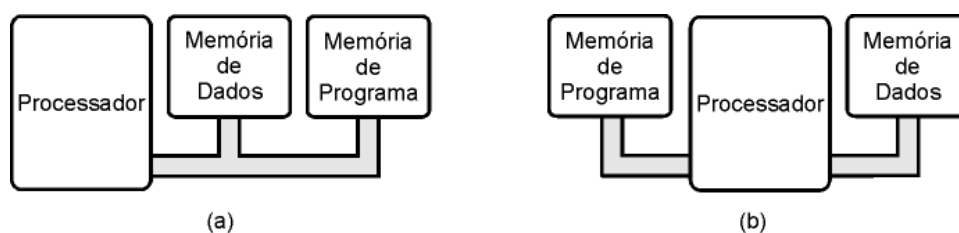


Figura 19 – (a) Arquitetura *von Neuman*, na qual um único barramento é utilizado para acessar as memórias de dados e de programa. (b) Arquitetura *Harvard*, na qual existem barramentos independentes para acesso às memórias de dados e de programa.

Com a finalidade de aumentar ainda mais o desempenho do acesso à memória, fabricantes de DSPs desenvolveram diversas arquiteturas baseadas na arquitetura *Harvard*. A Analog Devices desenvolveu a arquitetura *Super Harvard* (ANALOG DEVICES, 2004a) acrescentando ao barramento de programas uma memória *cache*, de forma que comandos

executados em um laço precisam ser lidos da memória somente uma vez. A Texas Instruments, por sua vez, desenvolveu a arquitetura *Harvard* modificada, na qual existem dois barramentos de acesso à memória de dados, de forma que dois operandos podem ser lidos simultaneamente (TEXAS INSTRUMENTS, 2000).

2.4.3 Endereçamento de Memória

Algoritmos de processamento digital de sinais necessitam de um determinado número de amostras do sinal de entrada para produzir o sinal de saída. Essas amostras são armazenadas em *buffers* finitos. Esses *buffers* precisam ser continuamente atualizados, armazenando a amostra mais atual do sinal e descartando a amostra mais antiga. Uma forma eficaz de se implementar esse tipo de armazenamento é por meio de *buffers* circulares. A figura 20 apresenta a estrutura de um *buffer* circular.

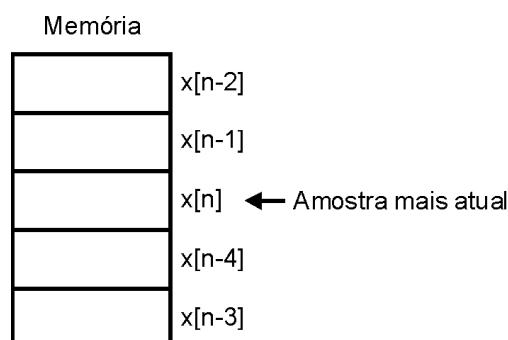


Figura 20 – Estrutura de um *buffer* circular. Um ponteiro indica a posição da amostra mais atual do sinal obtido e registradores definem as posições inicial e final do *buffer*.

Um ponteiro identifica a última amostra armazenada no *buffer*, conforme novas amostras são adquiridas, o ponteiro é incrementado e um *hardware* de controle de endereçamento verifica se o fim do *buffer* já foi alcançado. Ao alcançar o fim do *buffer*, o ponteiro é carregado novamente com a posição inicial do *buffer*, retornando ao seu início. Dessa forma, o fim do *buffer* é conectado ao seu início, permitindo o processamento contínuo das últimas N amostras do sinal. Para gerenciar esse processo, o DSP possui um *hardware* de controle do *buffer*, um registrador com o endereço da primeira posição do *buffer*, um registrador com o endereço da última posição e um ponteiro para a última amostra adquirida (HEATH, 1991; HENNESSY e PATTERSON, 1996; SMITH 1999), permitindo o controle do *buffer* sem a intervenção do processador.

Outro modo de endereçamento, conhecido como bit reverso, tem se tornado cada vez mais comum em DSPs (HENNESSY e PATTERSON, 1996; SMITH 1999). Esse modo de endereçamento tem a finalidade de ser aplicado exclusivamente em algoritmos de FFT (*Fast Fourier Transform*). Os resultados do processamento de algoritmos FFT são apresentados em uma ordem característica. A tabela 4 ilustra a ordem dos resultados para um algoritmo FFT de 8 pontos.

Tabela 4 – Ordem dos resultados de uma transformada rápida de Fourier de 8 pontos.

Entrada	Saída
0 (000) ₂	0 (000) ₂
1 (001) ₂	4 (100) ₂
2 (010) ₂	2 (010) ₂
3 (011) ₂	6 (110) ₂
4 (100) ₂	1 (001) ₂
5 (101) ₂	5 (101) ₂
6 (110) ₂	3 (011) ₂
7 (111) ₂	7 (111) ₂

Conforme se observa na tabela 4, os endereços dos dados de saída são equivalentes aos endereços dos dados de entrada com seus bits invertidos. A ordenação desses resultados necessitaria de uma quantidade considerável de processamento. Para evitar esse problema, diversos DSPs implementam unidades em *hardware* para controlar esse processo.

Em geral, em processadores de propósito geral, a tarefa de geração de endereços é executada pela própria CPU (*Central Processing Unit*). DSPs, por sua vez, contam com unidades específicas para geração de endereços, não somente nos modos de endereçamento circular ou bit reverso, mas para todos os modos de geração de endereços do DSP.

2.4.4 Paralelismo em Nível de Instrução

A maneira mais evidente de aumentar o desempenho de um processador é aumentar sua frequência de *clock*, isto é, aumentar a taxa de execução de instruções. Entretanto, essa técnica depende diretamente da tecnologia disponível para a fabricação do processador. Outra forma de obter maior desempenho é aumentar o número de instruções executadas simultaneamente pelo processador (TANENBAUM, 1999).

Para que se possa compreender as formas de implementação do paralelismo em nível de instrução, é necessário conhecer o ciclo de instrução de um processador, isto é, os passos

executados pelo processador durante uma instrução. Em linhas gerais, um ciclo de instrução de um processador é composto pelos seguintes passos (HAYES, 1986):

- busca da próxima instrução na memória, armazenamento em um registrador de execução e atualização do *program counter*;
- decodificação da instrução;
- determinação do endereço do(s) operando(s) da instrução (caso existam) e busca do(s) operando(s) da instrução;
- execução da instrução;
- armazenamento de resultados.

Essa seqüência, também conhecida como ciclo de busca-decodificação-execução (*fetch-decode-execute*), é o elemento central de qualquer microprocessador. Em um microprocessador tradicional, essas etapas são executadas seqüencialmente, uma após a outra. O paralelismo em nível de instrução (*Instruction-Level Parallelism – ILP*) é obtido inserindo-se no projeto dos microprocessadores unidades funcionais específicas, as quais são responsáveis por diferentes etapas desse processo e capazes de operar paralelamente (TANENBAUM, 1999).

2.4.5 *Pipelining*

Não apenas DSPs, mas praticamente todo microprocessador desde meados da década de 80 utiliza uma técnica de paralelismo em nível de instrução conhecida como *pipelining* (HENNESSY e PATTERSON, 1996; NULL e LOBUR, 2003). Um *pipeline* consiste em uma série de unidades funcionais, conhecidas como estágios, responsáveis pela execução de diferentes etapas do ciclo de instrução do processador, conforme ilustra a figura 21.

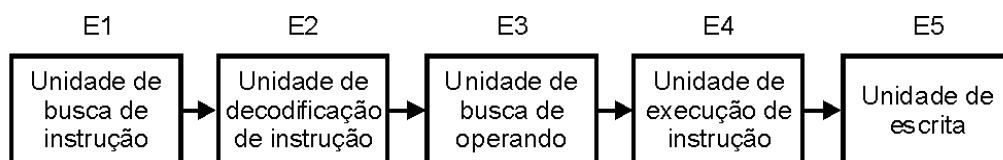


Figura 21 – Estágios de um *pipeline*. Cada uma das unidades funcionais é responsável pela execução de uma etapa do ciclo de instrução do processador.

O estágio E1 é responsável por buscar a instrução na memória e armazená-la em um *buffer*. No estágio seguinte, E2, a instrução é decodificada, isto é, o tipo de instrução e os operandos necessários à sua execução são identificados. O estágio E3 lê o(s) operando(s) da instrução e os armazena em registradores. No estágio E4, as instruções são efetivamente executadas, o que, na maioria das instruções, significa enviar os operandos para a unidade lógico-aritmética do processador. Finalmente, no estágio E5, os resultados obtidos pelas instruções são armazenados na memória.

Freqüentemente, cada estágio do *pipeline* pode ser dividido em estágios ainda mais específicos. Cada unidade de busca pode, por exemplo, ser dividida em uma unidade de geração de endereço e uma unidade de acesso à memória (HAYES, 1986). O DSP TMS320C6416 da Texas Instruments (assim como os outros DSPs de ponto fixo da família C6000) possui um *pipeline* de 11 estágios (CHASSAING, 2002).

O princípio básico através do qual o *pipeline* proporciona o paralelismo é o de sobreposição (*overlapping*) de instruções, isto é, o processo de execução de uma instrução tem início antes da conclusão da instrução anterior (HENNESSY e PATTERSON, 1996; NULL e LOBUR, 2003). Assim que um estágio do *pipeline* conclui sua tarefa sobre uma instrução e a envia para o estágio seguinte, uma nova instrução é inserida no estágio. A figura 22 ilustra esse processo.

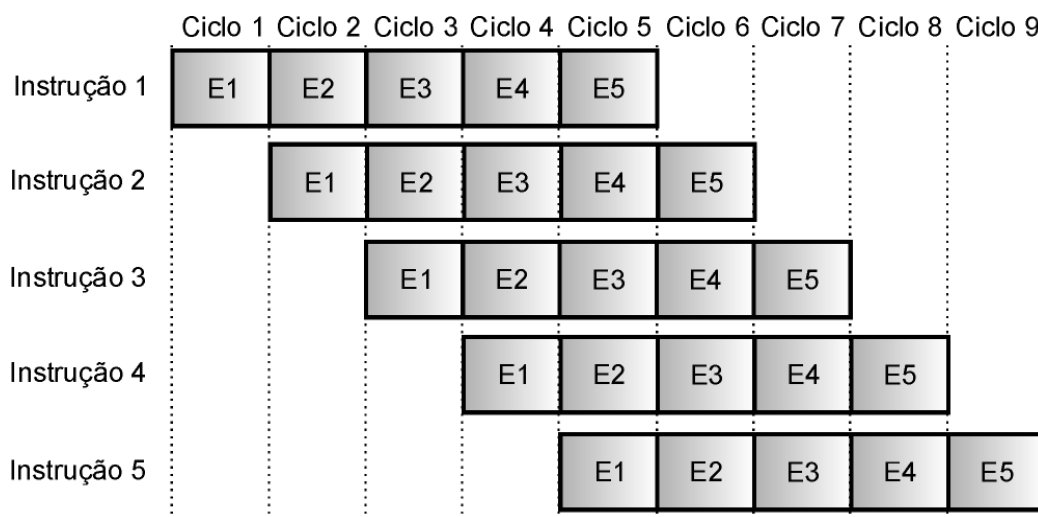


Figura 22 – Sobreposição de instruções em um *pipeline* de cinco estágios. A cada ciclo de *clock* do processador as unidades funcionais executam diferentes estágios de diversas instruções simultaneamente.

Conforme é possível observar na figura 22, durante o primeiro ciclo de *clock* a instrução 1 encontra-se no estágio E1. Assumindo-se que os estágios têm as funções

apresentadas na figura 21, o estágio E1 corresponde à unidade de busca de instruções. No ciclo de *clock* seguinte, enquanto a instrução 1 está sendo decodificada pelo estágio E2, a instrução 2 está sendo lida da memória pelo estágio E1. No terceiro ciclo de *clock*, enquanto os operandos da instrução 1 são lidos da memória pelo estágio E3, o estágio E2 decodifica a instrução 2 e o estágio E1 lê a instrução 3 da memória. No ciclo seguinte, a instrução 1 é executada pelo estágio E4, as instruções 2 e 3 passam pelos estágios de busca de operandos (E3) e decodificação (E2), respectivamente e a instrução 4 é lida da memória. Finalmente, no quinto ciclo de *clock*, os resultados da instrução 1 são armazenados na memória pelo estágio E5 enquanto a instrução 2 é executada, os operandos da instrução 3 são lidos, a instrução 4 é decodificada e uma quinta instrução é lida da memória.

A analogia mais utilizada para se compreender melhor o funcionamento de *pipelines* de processadores é a de uma linha de montagem (TANENBAUM, 1999; NULL e LOBUR, 2003). Em uma linha de montagem de veículos, por exemplo, não é necessário concluir a montagem do primeiro veículo antes de se iniciar a montagem do veículo seguinte. Ao invés disso, vários veículos são montados simultaneamente, um em cada etapa da linha de montagem.

Para que *pipelines* sejam eficientes, é importante que cada estágio necessite aproximadamente da mesma quantidade de tempo para executar sua tarefa, caso contrário, após algum tempo de execução, os estágios mais rápidos permanecerão aguardando os estágios mais lentos, o que prejudicará o desempenho total do *pipeline* (NAVAUX, 1990; NULL e LOBUR, 2003).

Teoricamente, após o período de latência do *pipeline*, durante o qual a primeira instrução trafega por todos os estágios, uma instrução seria executada a cada ciclo de *clock*. Entretanto, problemas como conflitos de recursos, instruções de desvio condicional e dependências de dados entre instruções fazem com que essa taxa de execução não possa ser alcançada (HENNESSY e PATTERSON, 1996; NULL e LOBUR, 2003). Diversos métodos e algoritmos são propostos para se tentar minimizar esses problemas (HENNESSY e PATTERSON, 1996).

2.4.6 *Very Long Instruction Word*

Outra abordagem para se obter paralelismo em nível de instrução é conhecida como VLIW (*Very Long Instruction Word*). Em processadores que utilizam a técnica VLIW, as

instruções são organizadas na memória em forma de pacotes de busca (*fetch packets*) (CHASSAING, 2002). Cada pacote de busca é composto por um determinado número de instruções (em geral de quatro a oito). A cada ciclo de instrução, o processador lê um pacote de busca completo, decodifica cada uma das instruções e as envia para as unidades de processamento correspondentes, as quais executam as instruções paralelamente. Para tanto, processadores VLIW são compostos por várias unidades funcionais, muitas delas duplicadas (HENNESSY e PATTERSON, 1996). Por exemplo, um DSP VLIW pode ter duas ALUs de ponto fixo, duas ALUs de ponto flutuante, um multiplicador de ponto fixo e um de ponto flutuante.

Para que as instruções possam ser processadas em paralelo, é necessário que cada pacote de busca seja composto exclusivamente por instruções independentes. A tarefa de determinar a dependência entre instruções é bastante complexa e, na arquitetura VLIW, é de responsabilidade do compilador (HENNESSY e PATTERSON, 1996).

Uma variante da arquitetura VLIW, conhecida com EPIC (*Explicitly Parallel Instruction Computer*), inclui nos pacotes de busca informações sobre as instruções que podem ser executadas em paralelo (NULL e LOBUR, 2003). Essa abordagem não garante a execução em paralelo de todas as instruções de um pacote de busca, mas evita a ocorrência de conflitos de recursos, isto é, mais de uma instrução sendo processada pela mesma unidade funcional. Muito freqüentemente, entretanto, as denominações VLIW e EPIC são utilizadas sem distinção.

A utilização da técnica VLIW não elimina os *pipelines*, pelo contrário, processadores desse tipo, em geral, possuem um *pipeline* global responsável pelas etapas de busca e decodificação das instruções (no caso pacotes de busca) e cada unidade funcional possui seu próprio *pipeline* de execução (HENNESSY e PATTERSON, 1996).

2.4.7 Processadores Multimídia

A crescente demanda por sistemas de processamento de alto desempenho para sinais de áudio e vídeo deu origem a uma nova categoria de processadores conhecida como processadores multimídia (*mediaprocessors*). A necessidade computacional de algoritmos de processamento e transmissão de vídeo em tempo real é extremamente alta. A decodificação do padrão de compactação de vídeo MPEG-4 (*Moving Picture Expert Group*), no formato QCIF (*Quarter Common Intermediate Format*) de 144 linhas por 176 colunas e 30 quadros

por segundo, requer aproximadamente 700 MOPS (Milhões de Operações por Segundo), consideravelmente maior do que a capacidade de processamento de DSPs convencionais (KNEIP et al., 1998).

Até recentemente, sistemas que exigiam grandes capacidades de processamento eram implementados exclusivamente por meio de ASICs (DALLY et al., 2004). O grande problema dessa abordagem está na falta de flexibilidade dos ASICs, os quais, uma vez programados, não podem ser modificados. O constante desenvolvimento de novos padrões de codificação, transmissão e compactação de sinais, bem como a grande quantidade de novos algoritmos desenvolvidos faz com que sistemas baseados em ASICs tornem-se rapidamente obsoletos. Além do que, não é possível realizar atualizações (*upgrades*) sem modificação do *hardware*. Outro aspecto negativo da utilização de ASICs está no custo de desenvolvimento de projetos, visto que, para cada novo produto, uma nova unidade de processamento deve ser desenvolvida desde o início. A solução para estes problemas é a utilização de processadores programáveis de alto desempenho, tais como os processadores multimídia (KUMURA et al., 2002).

Processadores multimídia incorporam uma estrutura VLIW, capaz de executar diversas instruções simultaneamente, e praticamente todas as funcionalidades dos DSPs, tais como unidades de geração de endereços, que permitem a execução de laços sem *overhead*, a implementação de *buffers* circulares e outros modos de endereçamento específicos, além de unidades de multiplicação implementadas em *hardware* (KURODA e NISHITANI, 1998).

Uma das características mais importantes dos processadores multimídia é a capacidade de processar dados em blocos, ou seja, vários *bytes* do sinal podem ser processados simultaneamente como um único dado (KURODA e NISHITANI, 1998). Os processadores de ponto fixo da família C6000 da Texas Instruments, como o TMS320C6416, por exemplo, são capazes de realizar operações simultaneamente em até oito pacotes de oito *bits* (CHASSAING, 2002).

A interface dos processadores multimídia com os dispositivos de entrada e saída também deve apresentar alto desempenho. Em geral, os processadores contam com controladores de acesso direto à memória (*Direct Memory Access* - DMA) capazes de acessar qualquer endereço de memória externo ou interno sem consumir tempo de processamento algum da CPU, assim como interfaces de alta velocidade com memórias de grande capacidade como SDRAMs (*Synchronous Dynamic Random Access Memory*) (KURODA e NISHITANI, 1998).

CAPÍTULO 3

MATERIAIS E MÉTODOS

3.1 ARQUITETURA DO SISTEMA

Para realizar a aquisição e o processamento de imagens em tempo real projetou-se, implementou-se e avaliou-se um Sistema Integrado de Aquisição e Processamento de Imagens (SIAPI) composto por um sensor de imagem e dois processadores digitais de sinais. A figura 23 apresenta o diagrama em blocos do sistema desenvolvido. Suas principais características são a velocidade de aquisição das imagens, a qual pode chegar a 73 quadros/segundo, utilizando a resolução máxima do sensor (500 linhas por 684 colunas), e a capacidade de processar cada quadro simultaneamente à próxima aquisição. Essa última característica somente é possível graças ao uso de processadores de sinais de alto desempenho e à natureza paralela do sistema desenvolvido, no qual dois processadores trabalham simultaneamente e independentemente, trocando mensagens de sincronização somente quando necessário.

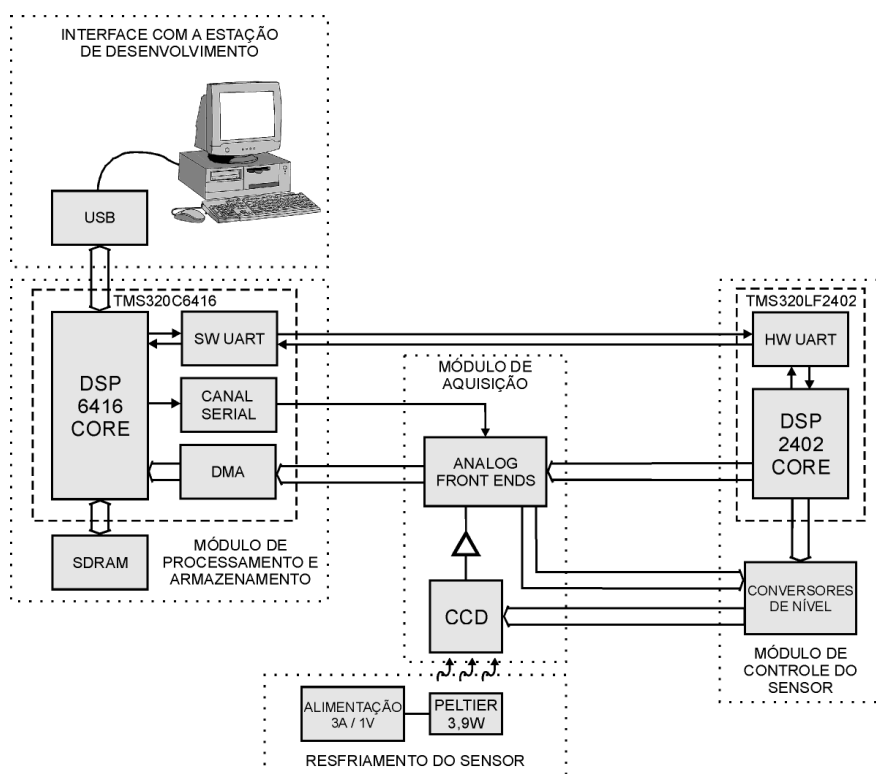


Figura 23 – Diagrama em blocos do sistema desenvolvido.

Conforme ilustra a figura 23, o sistema proposto está dividido em cinco blocos principais, a saber: 1) aquisição das imagens; 2) controle do processo de aquisição; 3) processamento e armazenamento das imagens; 4) interface USB para comunicação com a estação de desenvolvimento e 5) resfriamento do sensor de imagem.

De forma geral, as imagens são adquiridas por um sensor do tipo CCD responsável pela conversão do nível de iluminação de uma cena projetada sobre o sensor em níveis de tensão. O CCD possui uma matriz de sensores na qual cada elemento, denominado de *pixel*, produz um sinal de tensão cuja amplitude varia de acordo com o nível de iluminação sobre aquele elemento.

Os níveis de tensão de cada elemento da imagem são então convertidos para a forma digital por meio de dois dispositivos denominados de *analog front ends* (AFEs). Os processos de aquisição da imagem pelo CCD e de digitalização pelos *analog front ends* requerem vários sinais de controle perfeitamente sincronizados. A geração desses sinais de controle é realizada por um segundo processador digital de sinais, o TMS320LF2402.

Após a conversão, as informações digitalizadas de cada *pixel* são transmitidas para o DSP de alto desempenho, no caso o TMS320C6416, cuja finalidade é realizar o processamento das imagens em tempo real.

Com o intuito de propiciar um sistema capaz de adquirir imagens sob condições de baixa intensidade de iluminação, implementou-se um sistema de resfriamento do CCD, reduzindo assim o ruído térmico do sensor. O resfriamento é realizado por um módulo de resfriamento termo-elétrico de 3,9 watts, baseado no efeito *Peltier* (SCOTT, 1974).

O sistema é conectado a um microcomputador por meio de uma interface USB, para receber comandos ou enviar imagens processadas. A seguir, detalha-se cada uma das etapas envolvidas no processo de aquisição e processamento das imagens.

3.2 SISTEMA DE AQUISIÇÃO DE IMAGENS

O sistema de aquisição de imagens é composto por um sensor de imagem CCD TC237 (Texas Instruments, 2001), dois *analog front ends* AD9949 (Analog Devices, 2004b) e um módulo de resfriamento do tipo *Peltier*. Para possibilitar o resfriamento fez-se um arranjo mecânico para acoplar o sensor à lente e ao dispositivo de resfriamento.

3.2.1 Arranjo Mecânico

Para acoplar o sensor de imagem à lente, foi confeccionada uma peça em alumínio com arestas de 35x35 mm e 14,5 mm de altura. No centro da peça, há um furo de 25,4 mm (uma polegada) de diâmetro com rosca para conexão de lentes padrão *CS-Mount* de 32 voltas por polegada. O sistema mecânico foi projetado de forma que a distância entre a extremidade posterior da lente e o ponto focal fosse de 12,5 mm.

O sensor de imagem utilizado apresenta uma distância de 2,08 mm entre sua base e seu plano focal, conforme representa a figura 24, a qual ilustra a visão lateral do sensor. Assim, a referência a partir da qual devem ser medidos os 12,5 mm exigidos pelo padrão *CS-Mount* fica 2,08 mm distante da placa de circuito impresso, o que levou à confecção de uma peça para acoplamento da lente com uma altura de 14,5 mm.

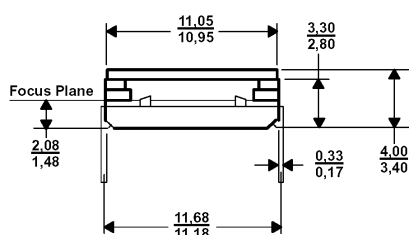


Figura 24 – Plano focal do sensor de imagem (fonte: Texas Instruments, 2001).

O padrão *CS-Mount* foi utilizado para que também se pudessem utilizar lentes do tipo *C-Mount*. Para isso, é necessário o uso de um espaçador de 5,0 mm de altura, visto que a distância entre a extremidade posterior da lente e o ponto focal é de 17,5 mm no padrão *C-Mount*. As figuras 25 (a) e (b) mostram, respectivamente, uma fotografia da peça construída e a peça com a lente acoplada.

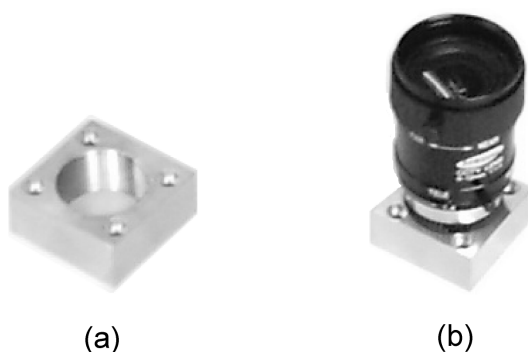


Figura 25 – (a) Foto da peça para acoplamento da lente. (b) Peça com lente *C-Mount* acoplada.

Para realizar o acoplamento térmico entre o sensor de imagem e o módulo *Peltier*, foi feito um furo retangular de 8x13 mm na região da placa de circuito impresso exatamente sob o sensor CCD, no qual foi inserido um *cold plate*. Um *cold plate* consiste em uma placa de material termicamente condutivo, em geral metal, utilizado para se conectar duas superfícies que necessitem trocar calor. O *cold plate* utilizado no projeto foi construído em alumínio com dimensões de 8x13x5 mm. Sob o *cold plate* foi posicionado o módulo *Peltier* e sob ele um dissipador de calor, o qual tem também a função de realizar a fixação mecânica do módulo *Peltier*. Para reduzir a resistência térmica entre os componentes do sistema foi introduzida uma camada de pasta térmica entre cada um deles. A figura 26 apresenta um diagrama ilustrando o acoplamento dos elementos do sistema.

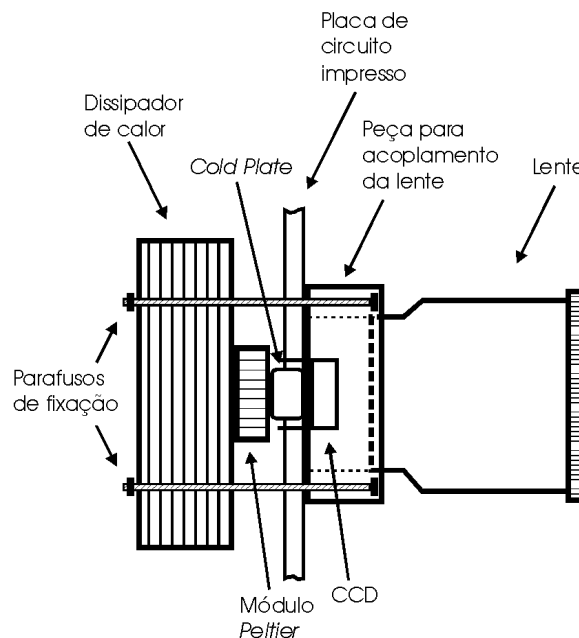


Figura 26 – Vista longitudinal ilustrando aspectos construtivos da montagem do sensor de imagem, lente e módulo de resfriamento *Peltier*.

3.2.2 Sensor de Imagem

Para adquirir as imagens foi utilizado um sensor CCD TC237 fabricado pela Texas Instruments, cujas unidades funcionais são ilustradas no diagrama da figura 27.

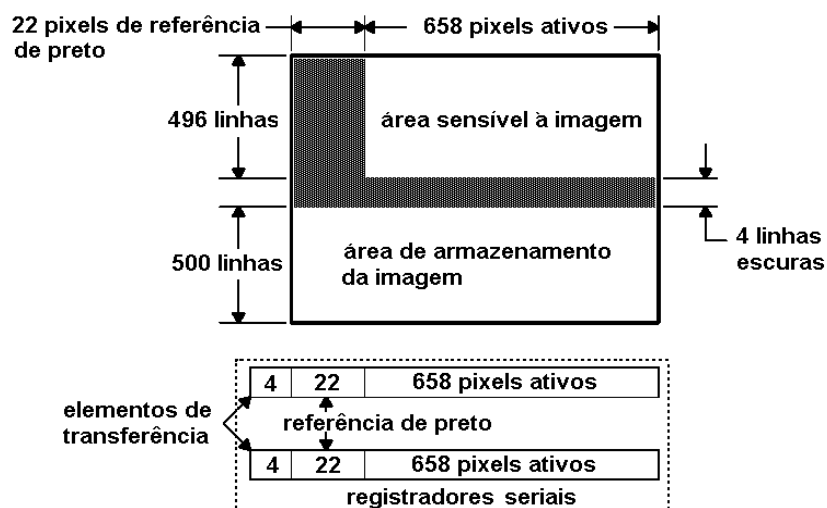


Figura 27 – Unidades funcionais do sensor de imagem CCD TC237. O sensor é do tipo *frame transfer*, contando com uma área sensível à imagem e uma área de armazenamento, ambas separadas por 4 linhas escuras. Na área sensível existem 22 elementos protegidos de incidência de luz para servirem de referência em etapas de processamento do sinal. As cargas podem ser lidas do sensor através de dois registradores seriais.

O sensor apresenta arquitetura do tipo *frame-transfer* e é composto por 500 linhas e 680 colunas, totalizando 342.000 *pixels* quadrados com arestas de 7,4 μm . Em cada linha existem 22 elementos não expostos à luz, os quais são utilizados como referência de preto em etapas de processamento posteriores à leitura dos *pixels*. As imagens podem ser lidas de modo entrelaçado (linhas pares e ímpares obtidas em aquisições separadas) ou progressivo (linhas pares e ímpares obtidas na mesma aquisição). O sensor conta com dois registradores seriais de saída, assim, no modo de leitura progressivo, é possível ler simultaneamente dois *pixels*, reduzindo pela metade o tempo de leitura da imagem. Em cada registrador serial existem quatro elementos não sensíveis à imagem, utilizados para conectar as saídas dos registradores aos amplificadores de saída do sensor. O sensor é do tipo fase-virtual, ou seja, um único sinal é necessário para controlar cada uma de suas unidades funcionais e sua frequência máxima de operação é de 12 MHz, o que possibilita a aquisição de até 73 imagens por segundo. O sensor é sensível a todo o espectro visível de luz, com seu pico de sensibilidade na região próxima a 750 nm. A figura 28 apresenta a curva de resposta em frequência do sensor TC237 para um tempo de integração de 16,6 ms. A resposta espectral de sensores de imagem é dada normalmente em função da responsividade, a qual é medida em A/W e reflete a quantidade de elétrons gerada por segundo para cada watt de potência óptica incidente sobre o sensor (KODAK, 2001b).

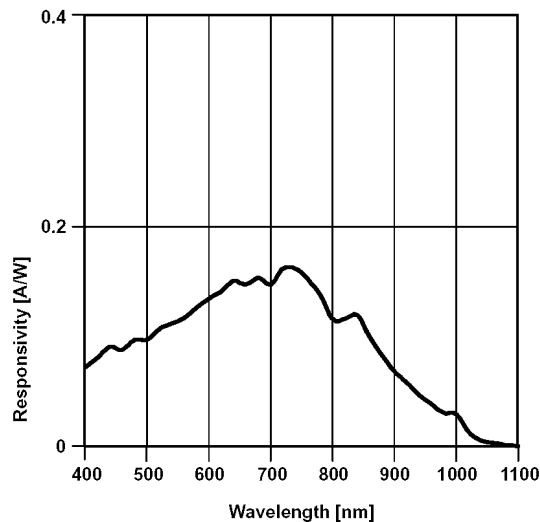


Figura 28 – Resposta espectral do sensor CCD TC237 (modificado de Texas Instruments, 2001).

3.2.3 Dispositivo Termo-elétrico de Resfriamento

O módulo *Peltier* CP 1.0-17-06L da Melcor Corporation, utilizado para reduzir o ruído térmico gerado no sensor, apresenta uma potência térmica nominal de 3,9 watts e é capaz de produzir uma diferença de temperatura de até 60° C entre suas extremidades. A figura 29 apresenta a curva da temperatura obtida no lado frio do módulo em função da carga térmica fornecida pelo componente a ser resfriado, para uma temperatura ambiente de 27° C, assumindo-se que no lado quente da célula exista uma dissipação adequada de calor.

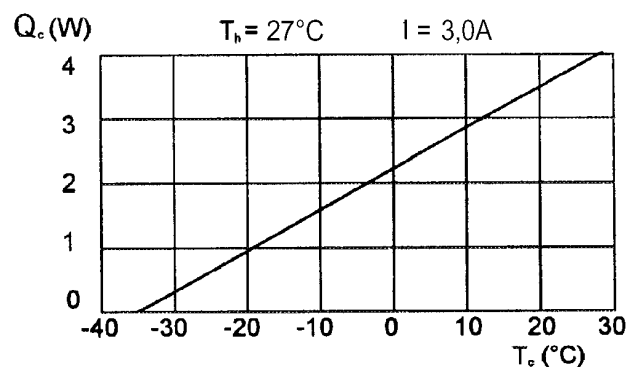


Figura 29 – Curva de temperatura versus potência térmica do módulo *Peltier* (modificado de Melcor Corporation, 2002).

O sensor de imagem gera uma carga térmica da ordem de 500 mW. Em função da baixa resistência térmica oferecida pelo *cold plate*, a queda de temperatura entre o módulo

Peltier e o sensor de imagem é da ordem de 2°C, conforme é possível comprovar pelas equações 8, 9 e 10.

$$\Delta T = \frac{Q * K * A}{L} \quad (8)$$

$$\Delta T = \frac{500.10^{-3} * 196,85 * 8.10^{-3} * 13.10^{-3}}{5.10^{-3}} \quad (9)$$

$$\Delta T = 2,04^{\circ} C \quad (10)$$

onde Q é a potência térmica gerada pelo sensor CCD, K é o coeficiente de condutibilidade térmica do alumínio, A é a área do *cold plate* e L o seu comprimento.

A potência térmica gerada pelo ambiente e absorvida pelo módulo *Peltier* é de aproximadamente 1 watt (SCOTT, 1974). Assim, a carga térmica total apresentada pelo sistema é de aproximadamente 1,5 watts. Observa-se na curva de temperatura versus potência térmica do módulo *Peltier*, figura 29, que é possível obter temperaturas da ordem de -11°C para uma carga térmica de aproximadamente 1,5 watts. Levando-se em conta as perdas térmicas no *cold plate*, é possível alcançar temperaturas da ordem de -9°C no sensor.

3.3 CONTROLE DO SENSOR DE IMAGEM

O controle do sensor de imagem é realizado pelo processador digital de sinais TMS320LF2402 fabricado pela Texas Instruments. O TMS320LF2402 é um DSP de 16 bits com velocidade de processamento de 40 MIPS (*Millions of Instructions per Second*), desenvolvido especificamente para exercer funções de controle de sistemas analógico-digitais. O processador conta com 32 kwords de memória de programa interna do tipo *flash*, o que permite atualizações de *firmware* do dispositivo via *software*.

Para reduzir o tempo de desenvolvimento do sistema, foi utilizado o kit didático desenvolvido por Matsuda et al. (2003), o qual contém o DSP TMS320LF2402 e oferece acesso a todos os seus pinos. O uso do kit didático eliminou as etapas de desenvolvimento do *layout* da placa de circuito impresso para o DSP e seus circuitos auxiliares, bem como a necessidade de depuração do *hardware* desenvolvido.

O *firmware* do sistema foi desenvolvido utilizando-se o ambiente de desenvolvimento *Code Composer Studio* 4.10.36 para DSPs da família C2000 e a depuração foi realizada por

meio de um emulador JTAG (*Joint Test Action Group*) modelo XDS510-PP fabricado pela *Spectrum Digital*.

A finalidade do DSP TMS320LF2402 é gerar os sinais para controlar o sensor CCD e realizar a movimentação das cargas armazenadas no sensor até a sua saída. Os sinais são gerados nas saídas digitais do DSP e passam por circuitos de conversão de nível para fornecer os níveis de tensão adequados ao sensor CCD. A figura 30 apresenta a seqüência de pulsos que deve ser aplicada ao sensor para efetuar a leitura dos dados no modo progressivo utilizando-se somente um dos registradores de saída.

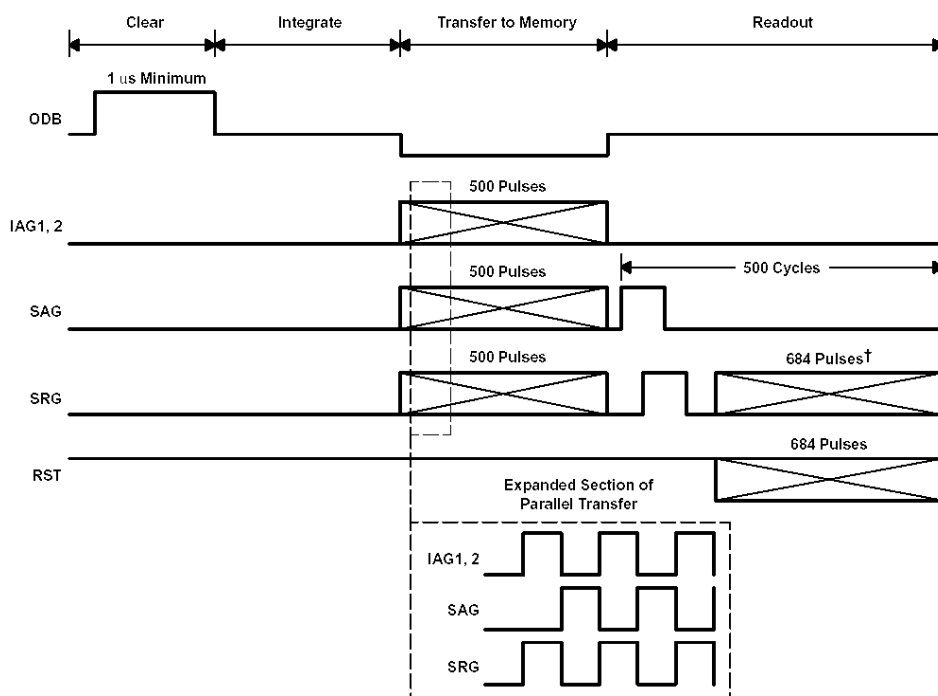


Figura 30 – Pulsos de leitura do sensor CCD TC237 em modo progressivo com um único registrador. A leitura do sensor se divide em quatro etapas: apagamento, integração, transferência para a memória e leitura serial (fonte: Texas Instruments, 2001).

A transferência de dados no sensor é dividida em quatro etapas, conforme ilustra a figura 30. Durante a etapa de apagamento (*clear*), um pulso é aplicado ao pino ODB (*Output Drain Bias*) do sensor por no mínimo 1µs. Esse pulso remove todas as cargas armazenadas nos poços de potencial do sensor, apagando qualquer imagem registrada anteriormente. Durante o período de integração (*integrate*) subsequente ao pulso de apagamento, o sensor permanece exposto à luz ambiente e cargas elétricas correspondentes à nova imagem são armazenadas nos poços de potencial. A quantidade de cargas armazenadas pode ser

controlada variando-se o período de integração. O TMS320LF2402 gera o pulso de apagamento do sensor e controla o tempo de integração por meio de seu temporizador.

O final da etapa de integração é definido pelo início dos pulsos de transferência paralela do sensor, conforme se pode observar no início do período definido como *transfer to memory* na figura 30. Durante esta etapa, as cargas elétricas armazenadas na região sensível são transferidas para a área de armazenamento não exposta à luz. Esta transferência é realizada aplicando-se uma série de 500 pulsos aos pinos IAG1, IAG2 (*Image Area Gates*) e SAG (*Storage Area Gate*) do sensor. Os pinos IAG1 e IAG2 estão conectados, respectivamente, às linhas ímpares e pares da área sensível à imagem. Cada pulso em um desses pinos faz com que todas as cargas armazenadas em uma linha da área sensível sejam transferidas simultaneamente para a linha subjacente. O sinal SAG é conectado a todas as linhas da área de armazenamento da imagem e cada pulso aplicado nele transfere uma linha da área de armazenamento para a linha subjacente. Todos os sinais da transferência paralela são gerados pelo DSP e aplicados ao sensor após terem seus níveis de tensão convertidos para valores adequados.

A última etapa da leitura do sensor consiste em se retirar serialmente cada *pixel* da imagem. Este processo é dividido em dois passos: no primeiro, uma linha da região de armazenamento é transferida para o registrador serial, por meio da aplicação de um pulso no pino SAG e outro no pino SRG (*Serial Register Gate*); em seguida, uma seqüência de 684 pulsos é aplicada aos pinos RST (*Reset Gate*) e SRG. Cada pulso aplicado ao pino RST remove os dados presentes nos amplificadores de saída no sensor; os pulsos aplicados ao pino SRG, por sua vez, deslocam as cargas presentes em um poço de potencial para o poço adjacente, de forma que as cargas presentes no último poço são transferidas para o registrador de saída.

A geração dos pulsos para os sinais SAG e SRG para a transferência das linhas para os registradores seriais é realizada pelo DSP de controle do CCD. A geração da seqüência de pulsos nos pinos RST e SRG é realizada pelos *analog front ends*, ficando sob o controle do DSP os sinais de sincronismo vertical e horizontal enviados aos AFEs para indicar o início de uma linha ou de uma nova imagem. O número de pulsos gerados pelos AFEs para leitura serial dos *pixels* do sensor CCD (684 no caso) é definido pelo intervalo entre os pulsos de sincronismo horizontal gerado pelo DSP.

A figura 31 apresenta o diagrama de fluxo do *firmware* desenvolvido para o DSP TMS320LF2402. Por meio de uma interface serial assíncrona, o TMS320LF2402 recebe comandos do TMS320C6416. Dois comandos foram implementados: início de leitura e

configuração do tempo de integração. O comando de início de leitura faz com que o DSP inicie uma seqüência de leitura de uma nova imagem, desde o pulso de apagamento até a leitura serial. O comando de configuração do tempo de integração permite que o TMS320C6416 solicite que o TMS320LF2402 ajuste o tempo de exposição à luz do sensor.

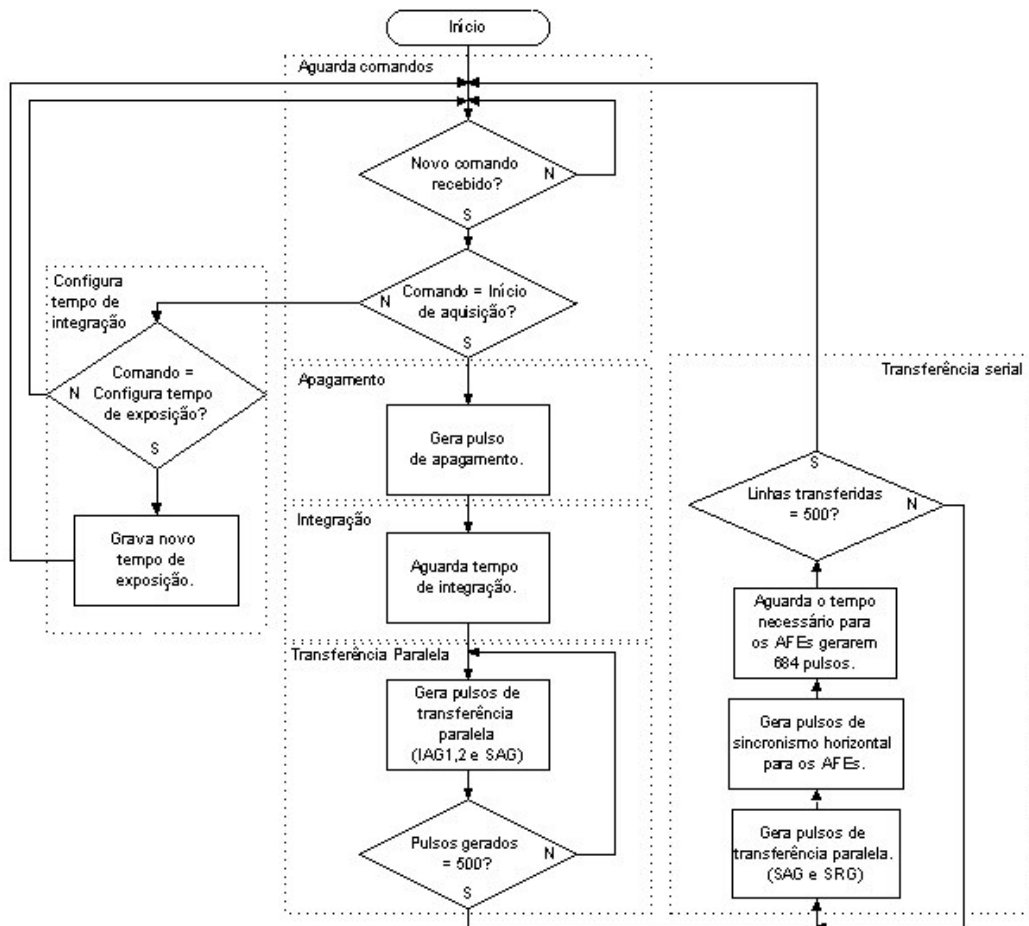


Figura 31 – Diagrama de fluxo do *firmware* do DSP de controle. O DSP pode receber comandos para gravar novos valores para o tempo de integração do sensor ou para iniciar a geração dos pulsos de leitura do sensor CCD. O DSP gera o pulso de apagamento, aguarda o período de integração do sensor, gera os pulsos de transferência para a região de armazenamento e os pulsos de sincronismo horizontal para o AFE. O número de pulsos gerados pelo AFE é controlado pelo intervalo entre os pulsos de sincronismo gerados pelo DSP de controle.

3.4 SISTEMA DE CONVERSÃO ANALÓGICO-DIGITAL

Os *pixels* que deixam os registradores de saída do sensor de imagem passam por amplificadores seguidores de emissor para fornecer níveis adequados de corrente para os *analog front ends*. AFEs são conversores analógico-digitais que realizam etapas de

processamento analógico do sinal de saída do CCD antes de realizar a conversão para o modo digital. A cada registrador de saída do CCD foi acoplado um AFE AD9949, fabricado pela Analog Devices, o qual possui um conversor analógico-digital de 12 bits e é capaz de converter sinais a uma taxa de até 36 MHz. Ressalta-se que o processo de aquisição da imagem é completamente dependente do AFE, o qual, além da conversão A/D do sinal analógico do CCD, também elimina ruídos e, de forma geral, estabiliza as variações de níveis de cinza da imagem. Para que seja possível o completo entendimento do processo de controle do CCD faz-se necessário entender o funcionamento do AFE, assim a figura 32 apresenta o diagrama em blocos do AD9949.

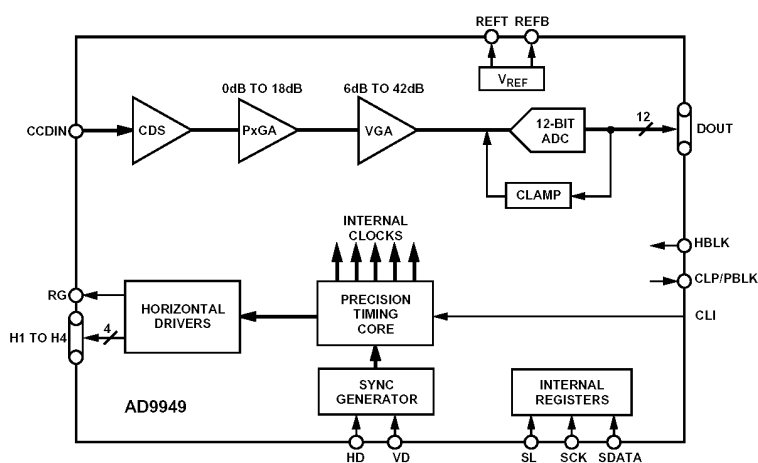


Figura 32 – Diagrama em blocos do *analog front end* AD9949. O dispositivo conta com uma etapa de *correlated double sampling* (CDS), um amplificador de *pixels* (PxGA), um amplificador de ganho variável (VGA), uma etapa de deslocamento para o nível de preto óptico (CLAMP), um conversor analógico-digital de 12 bits, além de circuitos de geração de pulsos internos controlados pelos sinais de sincronismo horizontal (HD) e vertical (VD). A programação dos registradores do dispositivo é feita por meio de uma interface serial síncrona (fonte: Analog Devices, 2004b).

Ao entrar no AD9949 através do pino CCDIN, o sinal de saída do CCD passa por um estágio de *correlated double sampling*, o qual amostra o nível de tensão correspondente a um determinado *pixel* em dois instantes, o primeiro durante o nível de referência e o segundo durante o período que contém a informação sobre a quantidade de cargas armazenadas no poço de potencial do *pixel* em questão. Essa etapa de processamento tem a finalidade de manter constante a diferença entre o nível da imagem e o nível de preto, removendo o ruído de *reset* do sensor. Os instantes em que são realizadas as amostras são programáveis por meio dos registradores de controle do AFE.

Após o CDS, o AD9949 possui um amplificador de ganho programável (*Pixel Gain Amplifier* - PxGA) utilizado em aplicações coloridas. O PxGA pode ser programado de modo a amplificar diferentemente sinais de cada componente de cor do sensor. Como o sensor utilizado no protótipo desenvolvido é monocromático, o estágio PxGA do AFE foi desabilitado.

O próximo estágio do AFE consiste em um amplificador de ganho variável (*Variable Gain Amplifier* – VGA) para todos os *pixels* do sinal. O amplificador é linear em decibéis e apresenta uma faixa de ganho entre 6 e 42 dB. A finalidade do VGA é de ajustar a amplitude do sinal de saída do sensor CCD para a faixa adequada do conversor analógico-digital. A programação do ganho é realizada de forma digital através dos registradores do AFE. A equação 11 fornece o ganho do VGA em decibéis a partir do código digital armazenado em seu registrador de controle.

$$G(dB) = (0,0351 * Cod) + 6dB \quad (11)$$

onde “*Cod*” é o valor digital armazenado no registrador que pode variar entre 0 e 1023.

Ao deixar o amplificador de ganho variável, o sinal passa pelo estágio de deslocamento para o nível de preto óptico (*Optical Black Clamping* - OBC). Durante este estágio, o AFE calcula o nível de preto obtido pelos elementos de referência não expostos à luz, presentes nos 22 primeiros *pixels* de cada linha do CCD. A diferença entre este sinal e o nível programado como preto no registrador correspondente do AFE é então introduzida ao sinal digitalizado, de forma a se evitar oscilações no nível de preto entre linhas da imagem digital.

Além de realizar as etapas de processamento analógico do sinal e a conversão para o modo digital, o AD9949 possui um módulo de geração de pulsos para controle do CCD. A geração desses pulsos é controlada pelos sinais de sincronismo horizontal (*Horizontal Driver* - HD) e vertical (*Vertical Driver* - VD) e pelos valores nos registradores do AFE.

O AFE gera três sinais de saída, RG, H1/H3 e H2/H4. Os sinais RG e H1/H3 têm suas polaridades e transições de subida e descida programáveis, enquanto H2/H4 corresponde sempre ao inverso de H1/H3. Os instantes das transições de subida e descida podem ser programados em função do sinal de *clock* (CLI) do AFE, o qual é dividido internamente em 48 posições, conforme ilustra a figura 33.

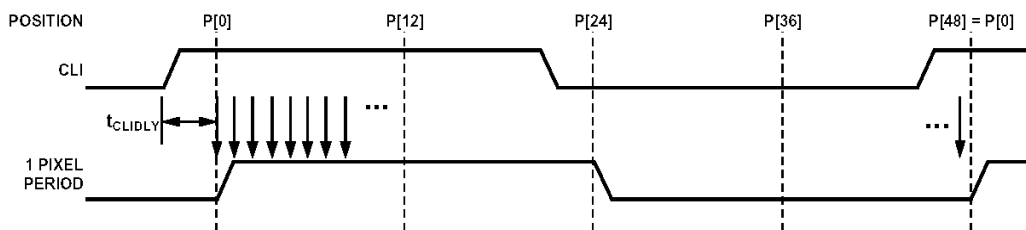


Figura 33 – Divisão do *clock* do AD9949. Cada ciclo de *clock* é dividido em 48 posições que podem ser utilizadas para programar os sinais de controle do CCD (fonte: Analog Devices, 2004b).

Cada uma das 48 posições pode ser utilizada como instante de transição dos sinais de saída do AD9949, assim como dos pulsos de amostragem do CDS, conforme ilustra a figura 34.

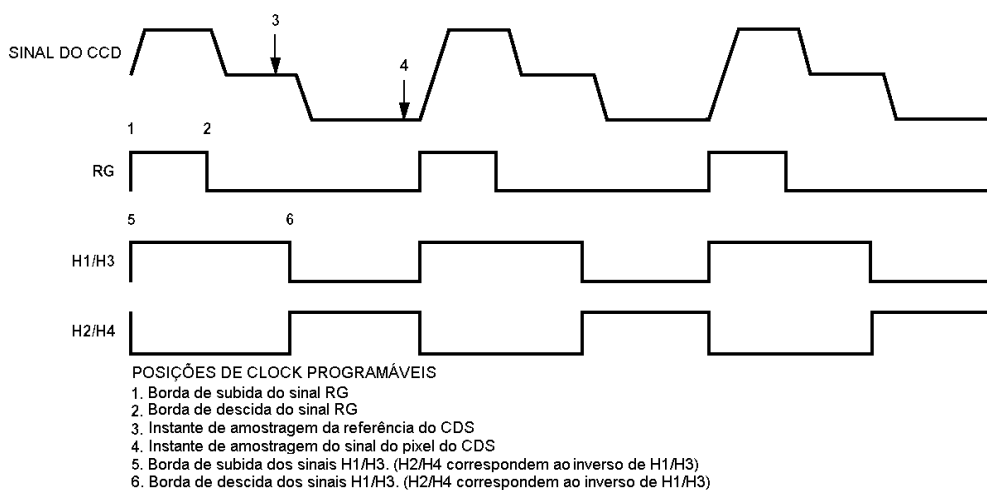


Figura 34 – Posições programáveis dos sinais de controle do CCD gerados pelo AD9949. Através destes sinais é possível programar a largura dos pulsos de *reset* do amplificador de saída do sensor CCD, dos pulsos de leitura serial e os instantes de amostragem do circuito de *correlated double sampling* (fonte: Analog Devices, 2004b).

No protótipo desenvolvido, o sinal H1 foi utilizado para controlar o sinal SRG do sensor de imagem e o sinal RG para controlar o sinal RST durante a leitura serial. O AFE foi programado para que seus sinais de saída permaneçam estáticos durante a aplicação dos pulsos de sincronismo, de forma que o DSP de controle da figura 23, possa assumir o controle do CCD durante este período.

A programação dos registradores do AD9949 é feita por meio de uma interface serial síncrona composta por três sinais: SL, SCK e SDATA. O sinal SL habilita a interface serial do AFE e os sinais SCK e SDATA correspondem ao *clock* e aos dados enviados à interface,

respectivamente. Na seção 3.5.1 o funcionamento da interface serial do AD9949 é apresentado mais detalhadamente.

3.5 SISTEMA DE PROCESSAMENTO DAS IMAGENS

Para processar as imagens adquiridas, foi utilizado o processador digital de sinais de alto desempenho TMS320C6416 de 720 MHz fabricado pela Texas Instruments Inc. O TMS320C6416 é um DSP de 32 bits de ponto fixo, com 64 registradores de propósito geral de 32 bits, dois multiplicadores de 16 bits e seis unidades lógico-aritméticas independentes, o que permite que o DSP seja capaz de desempenhar até 5760 MOPS. O processador possui 1 Mbyte de memória RAM interna, 16 kbytes de memória *cache* de dados e 16 kbytes de memória *cache* de programa.

A arquitetura do TMS320C6416 é do tipo VLIW, na qual pacotes de 256 bits são lidos pelo processador a cada ciclo de *clock*. Cada pacote de 256 bits pode conter até oito instruções de 32 bits as quais podem ser executadas paralelamente nas unidades independentes do processador. O conjunto de instruções do processador permite operações com dados de 8, 16, 32 ou 64 bits, sendo que operações com dados de tamanhos inferiores a 64 bits podem ser realizadas de forma paralela. Para ilustrar a capacidade de processamento desse DSP, é interessante mencionar que com ele é possível realizar, por exemplo, 8 multiplicações de 8 bits em um único ciclo de *clock*, ou 1,39 ns.

O processador possui uma interface de memória externa programável (*External Memory Interface* – EMIF) que permite a interface do dispositivo com diversos tipos de memória sem a necessidade de lógica de interconexão. Os acessos à memória externa podem ser controlados por meio de um controlador de acesso direto à memória (*Enhanced Direct Memory Access* – EDMA), o qual pode ser programado para realizar transferências de dados unidimensionais ou bidimensionais sem interromper a CPU. Para transmitir dados de forma serial, o TMS320C6416 possui três canais seriais (*Multi-channel Buffered Serial Port* – McBSP) programáveis.

Para simplificar o processo de desenvolvimento foi utilizado o kit de desenvolvimento TMS320C6416 DSK (*DSP Starter Kit*) fabricado pela Spectrum Digital, o qual é composto por um DSP TMS320C6416 de 720 MHz, 32 Mbytes de memória SDRAM e diversos periféricos como conversores analógico-digitais (os quais não foram utilizados nesse

trabalho). O kit possui um sistema de emulação JTAG, o qual facilita a depuração do sistema utilizando-se o ambiente integrado de desenvolvimento Code Composer Studio C6000 v 2.10.

3.5.1 Interface com os *Analog Front Ends*

A conexão das saídas digitais dos AFEs à EMIF do DSP foi realizada por meio de *latches* de 16 bits com saídas em alta impedância, de forma que os dados de cada AFEs pudessem ser lidos independentemente. Os AFEs correspondentes aos registradores de saída 1 e 2 do CCD foram mapeados diretamente na memória do DSP nos endereços 0A000-0004 e 0A000-0000, respectivamente. A tabela 5 apresenta o mapa de memória do sistema.

Tabela 5 – Mapa de memória do sistema.

0000-0000 a 0010-0000	Memória interna do TMS320C6416
8000-0000 a 9000-0000	SDRAM
A000-0000 e A000-0004	Registradores de Saída do Sensor de imagem
B000-0000 a B000-0004	Controlador USB

A transferência dos dados do CCD para a memória do DSP é realizada por meio do controlador de DMA, de modo que a CPU não dedica tempo de processamento a esta tarefa. Os dados são transferidos das saídas dos AFEs diretamente para a memória SDRAM. Os acessos à memória SDRAM, assim como a geração dos sinais de *refresh* são controlados pela EMIF do próprio TMS320C6416.

As interfaces seriais de ambos os AFEs foram conectadas ao canal serial McBSP 2 do TMS320C6416 operando a 10 MHz e um pino de I/O foi utilizado para habilitar o AFE com o qual se deseja comunicar. A figura 35 ilustra o procedimento de escrita de um registrador através da interface serial do AD9949.

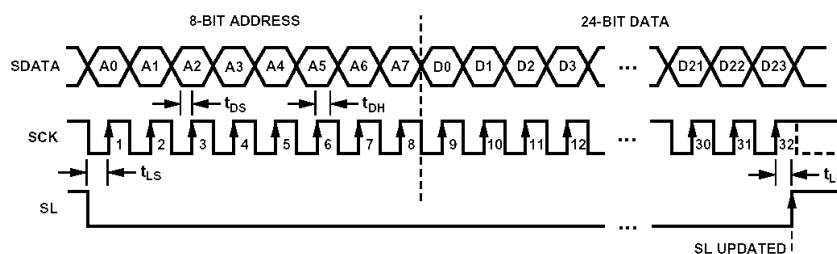


Figura 35 – Procedimento de escrita via interface serial do *analog front end* AD9949. Os primeiros oito bits compõem o endereço do registrador a ser gravado, os 24 bits seguintes compõem o dado (fonte: Analog Devices, 2004b).

Conforme pode ser observado na figura 35, o dado transmitido a cada escrita serial é composto por 8 bits do endereço do registrador seguido por mais 24 bits de dados a serem gravados no registrador. Embora diversos registradores do AD9949 não possuam 24 bits de dados, todas as transmissões seriais devem enviar 24 bits contendo informações nos bits menos significativos. Com base nas características da interface serial dos AFEs, os canais seriais do TMS320C6416 foram programados para enviar dados de 32 bits iniciando pelo bit menos significativo. A tarefa de agrupar endereço e dados na mesma transmissão foi realizada por *software*. A tabela 6 apresenta um resumo do conjunto de funções e macros desenvolvidas em linguagem C para realizar a comunicação com os AFEs, bem como uma pequena descrição das suas funcionalidades.

Tabela 6 – Rotinas de comunicação com os *analog front ends*.

AFE_initCom(MCBSP_Handle hMcbasp);	Inicializa o McBSP conectado à interface serial do AFE.
AFE_writeReg(Uint8 address, Uint32 data, MCBSP_Handle hMcbasp);	Escreve o valor em <i>data</i> no registrador cujo endereço é dado por <i>address</i> .
Uint16 AFE_ccdRead(Uint32 address);	Lê os dados da saída digital do CCD quando não é utilizada leitura via EDMA.
AFE1_enableSerial();	Habilita o canal serial do AFE 1.
AFE2_enableSerial();	Habilita o canal serial do AFE 2.
AFE_config(MCBSP_Handle hMcbasp, AFE_Config *AFE_cfg);	Inicializa o AFE a partir dos dados da estrutura AFE_cfg.

3.5.2 Interface com o DSP TMS320LF2402

Uma vez que os canais seriais do TMS320C6416 não permitem comunicação assíncrona, para que se pudesse realizar a comunicação com o TMS320LF2402 foi necessário implementar uma UART (*Universal Asynchronous Receiver Transmitter*) por meio de rotinas de *software*. Dois pinos de I/O foram utilizados para transmissão e recepção de dados, os tempos de transmissão e recepção de cada bit foram definidos por meio de laços de *software*.

Para garantir a recepção correta dos dados pelo TMS320LF2402 foi implementado um protocolo de *handshake* no qual o dado recebido pelo TMS320LF2402 é retransmitido para o TMS320C6416, o qual compara o dado recebido com o dado transmitido previamente e retransmite o dado caso a comparação não seja bem sucedida.

3.5.3 Interface USB

Para realizar a comunicação do SI-API com um microcomputador (*host*) utilizou-se o SIE (*Serial Interface Engine*) CY7C68001, fabricado pela Cypress. Um SIE é um dispositivo que realiza a função de “serialização” dos dados, isto é, a conversão dos dados recebidos paralelamente em pacotes para serem enviados em modo serial e vice-versa. Entre as tarefas do SIE estão a separação dos sinais de *clock* e dados do barramento USB, montagem e reconhecimento dos pacotes de informações das transações USB, assim como dos bytes de CRC (*Cyclic Redundancy Check*), utilizados para verificar possíveis erros de comunicação. O CY7C68001 inclui ainda um *transceiver* para conversão dos níveis elétricos do barramento USB.

O CY7C68001 é compatível com a interface USB 2.0, podendo operar nos modos *full-speed* a 12 Mbits/s ou *high-speed* a 480 Mbits/s. Uma memória FIFO (*First In First Out*) de 4 kbytes é compartilhada entre 4 *endpoints*, os quais podem ser configurados independentemente. O dispositivo é compatível com os quatro tipos de transações definidas pelo padrão USB e cada *endpoint* pode operar em qualquer tipo de transação independentemente dos demais. A interface paralela do CY7C68001 possui um barramento de 16 bits que pode ser conectada a qualquer microprocessador capaz de se comunicar com uma memória do tipo FIFO, síncrona ou assíncrona.

Durante o processo de inicialização do CY7C68001 os descritores do equipamento devem ser gravados na RAM do dispositivo para permitir que o *host*, no caso o microcomputador, inicie a enumeração do equipamento. Existem dois métodos para se gravar os descritores no CY7C68001. Se uma EEPROM for conectada à interface serial do dispositivo, ele procura pelos descritores na EEPROM e o processo de enumeração ocorre automaticamente após a leitura dos descritores. Caso não haja uma EEPROM conectada à interface serial do CY7C68001, o processador de controle, no caso o DSP TMS320C6416, é responsável por escrever os descritores na memória para que então a enumeração tenha início. Após a gravação dos descritores na memória RAM, o CY7C68001 responde automaticamente a todas as transações de controle solicitadas pelo processador *host* durante o processo de enumeração.

O CY7C68001 possui um sistema de geração de interrupções, o qual é capaz de informar ao processador de controle qualquer solicitação que não possa ser atendida automaticamente. Existem também três pinos de *flags* programáveis para informar ao processador de controle sobre o estado das FIFOs do dispositivo.

Apenas dois dos quatro *endpoints* do CY7C68001 foram habilitados, ambos para transações do tipo *bulk*, de forma a se obter a maior taxa de transmissão possível com correção de erros. O *endpoint 8* foi configurado como OUT, para que o *host* possa enviar comandos para o SI-API, e o *endpoint 2* foi configurado como IN, para enviar as imagens processadas para o *host*. Uma vez que o fluxo de informações no sentido IN, isto é, do SI-API para o *host*, é composto por imagens de 342.000 *words*, e o fluxo no sentido oposto é composto exclusivamente por comandos de, no máximo, 3 bytes, a tendência é que o tráfego de informações no sentido IN seja muito maior que no sentido oposto. Com base neste critério, foram mapeados 3.072 kbytes do *buffer* do SIE para o *endpoint 2* e 1024 kbytes para o *endpoint 2*. A figura 36 ilustra a configuração dos *endpoints* do SIE.

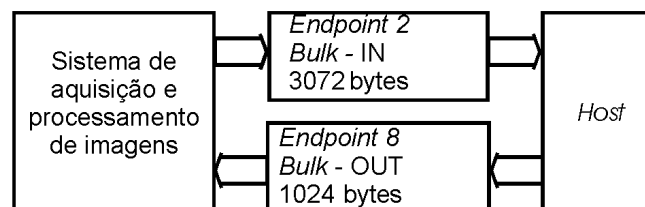


Figura 36 – Configuração dos *endpoints* do *serial interface engine*. Através do *endpoint 2*, configurado para transferências do tipo IN com um *buffer* de 3.072 bytes, são transferidas imagens do SI-API para o *host*. Através do *endpoint 8*, configurado para transferências do tipo OUT com um *buffer* de 1.024 bytes, são transmitidos comandos do *host* para o SI-API.

Quando o *buffer* do *endpoint 8* passa do estado vazio para não-vazio, é gerada uma interrupção para o TMS320C6416, que então dá início à leitura dos dados recebidos. Quando os dados recebidos correspondem a um comando de solicitação de transmissão de imagens para o *host*, o TMS320C6416 começa a enviar dados para o *endpoint 2*. Quando a FIFO do *endpoint 2* alcança o seu limite, uma nova interrupção é gerada e o TMS320C6416 interrompe o processo até que o *host* leia os dados. Essa operação é repetida até que todos os dados solicitados pelo *host* tenham sido transmitidos.

3.5.4 Estrutura do *Firmware* do DSP TMS320C6416

O *firmware* do DSP TMS320C6416 foi desenvolvido de forma que, após a inicialização do sistema, o processador permaneça em um laço aguardando comandos do *host*. Uma vez que não há sincronismo entre o envio de comandos pelo *host* e a execução dos mesmos pelo DSP, criou-se uma FIFO de comandos. Através da FIFO de comandos, o *host*

pode enviar continuamente solicitações ao sistema, o qual as executa conforme a disponibilidade de recursos. Para verificar as funcionalidades do sistema, foram implementados, inicialmente, quatro comandos. A tabela 7 apresenta os comandos implementados e uma breve descrição de cada um.

Tabela 7 – Comandos do *host* para o sistema.

Comando	Valor	Descrição	Parâmetro	Descrição
CMD_USB_IMG_READ	0x41	Solicita leitura de imagem via USB.	n_img	Número da imagem a ser transmitida
CMD_CCD_IMG_ACQ	0x51	Solicita aquisição de imagem do CCD.	n_img	Número da imagem a ser adquirida
CMD_CCD_TINT	0x61	Configura o tempo de integração.	t_int_msb t_int_lsb	Byte mais significativo do tempo de integração Byte menos significativo do tempo de integração
CMD_IMG_SOBEL	0x11	Aplica o algoritmo de Sobel sobre a imagem.	n_img	Número da imagem a ser processada

Para permitir a aquisição e processamento contínuo de imagens pelo sistema, implementou-se um *buffer* na memória SDRAM do DSK. O *buffer* implementado está dividido em 24 posições permitindo o armazenamento de até 24 imagens. Em cada posição foram reservadas 342.000 *words* de 16 bits. A figura 37 ilustra o *buffer* de armazenamento de imagens.

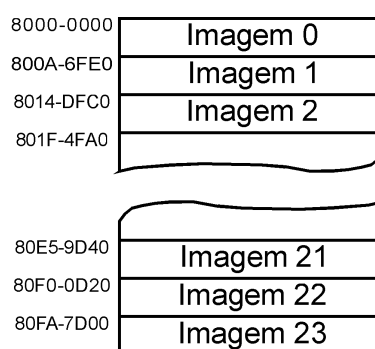


Figura 37 – Buffer para armazenamento de imagens. Para armazenar as imagens adquiridas do sensor CCD, a memória SDRAM do sistema foi dividida em 24 posições de 342.000 *words* de 16 bits.

Com o *buffer* de armazenamento de imagens é possível realizar a aquisição de uma imagem para uma determinada posição e o processamento simultâneo de uma imagem

armazenada em outra posição do *buffer*. Para sincronizar estes processos foram criados *flags* de controle, os quais são habilitados sempre que uma imagem estiver sendo adquirida.

O processo de aquisição de uma imagem tem início quando o *host* envia o comando de aquisição para o sistema (CMD_CCD_IMG_ACQ, tabela 7). O *flag* correspondente à imagem sendo adquirida é habilitado e o controlador de DMA inicia a aquisição. Após a transferência de todos os dados da imagem do CCD para a memória SDRAM, uma interrupção de conclusão de transferência é gerada pelo controlador de DMA e o *flag* correspondente à imagem é então desabilitado.

O processamento das imagens ocorre quando o *host* envia um comando de solicitação de processamento para o sistema (por exemplo, o comando CMD_IMG_SOBEL, tabela 7). O TMS320C6416 verifica então se a imagem que deve ser processada não está em aquisição para então transferi-la para a memória interna do processador. Embora a transferência da imagem consuma tempo de processamento, a aplicação dos algoritmos na memória interna do processador é muito mais eficiente. Uma vez que o TMS320C6416 é capaz de armazenar ou ler de sua memória interna até 4 dados de 16 bits simultaneamente em um único ciclo de *clock*, o tempo de acesso a dados de 16 bits é igual a $\frac{1}{4}$ do período de *clock* do DSP, ou aproximadamente 0,35 ns, comparando-se esse tempo com os 7,5 ns de tempo de acesso da memória SDRAM externa, percebe-se que o acesso a dados na memória interna do DSP é aproximadamente 21 vezes mais rápido. Após o processamento da imagem, os dados são transferidos novamente para a posição de memória correspondente à imagem no *buffer* de armazenamento de imagens da SDRAM.

Através do comando de configuração de tempo de integração, o *host* envia o novo tempo de integração para o TMS320C6416, o qual retransmite os dados por meio de sua interface serial para o TMS320LF2402, responsável pelo controle da exposição do sensor. O tempo de integração é dividido em 2 bytes (16 bits), de forma que 65.536 valores, em intervalos de 1 μ s, são possíveis.

A transmissão de imagens para o *host* é realizada por meio de um laço de *software*, durante o qual o TMS320C6416 preenche os *buffers* do *endpoint* 2 do SIE repetidamente até que a imagem seja completamente transmitida. A figura 38 apresenta um diagrama de fluxo do *firmware* do TMS320C6416 responsável pela interpretação e tratamento dos comandos recebidos do *host*.

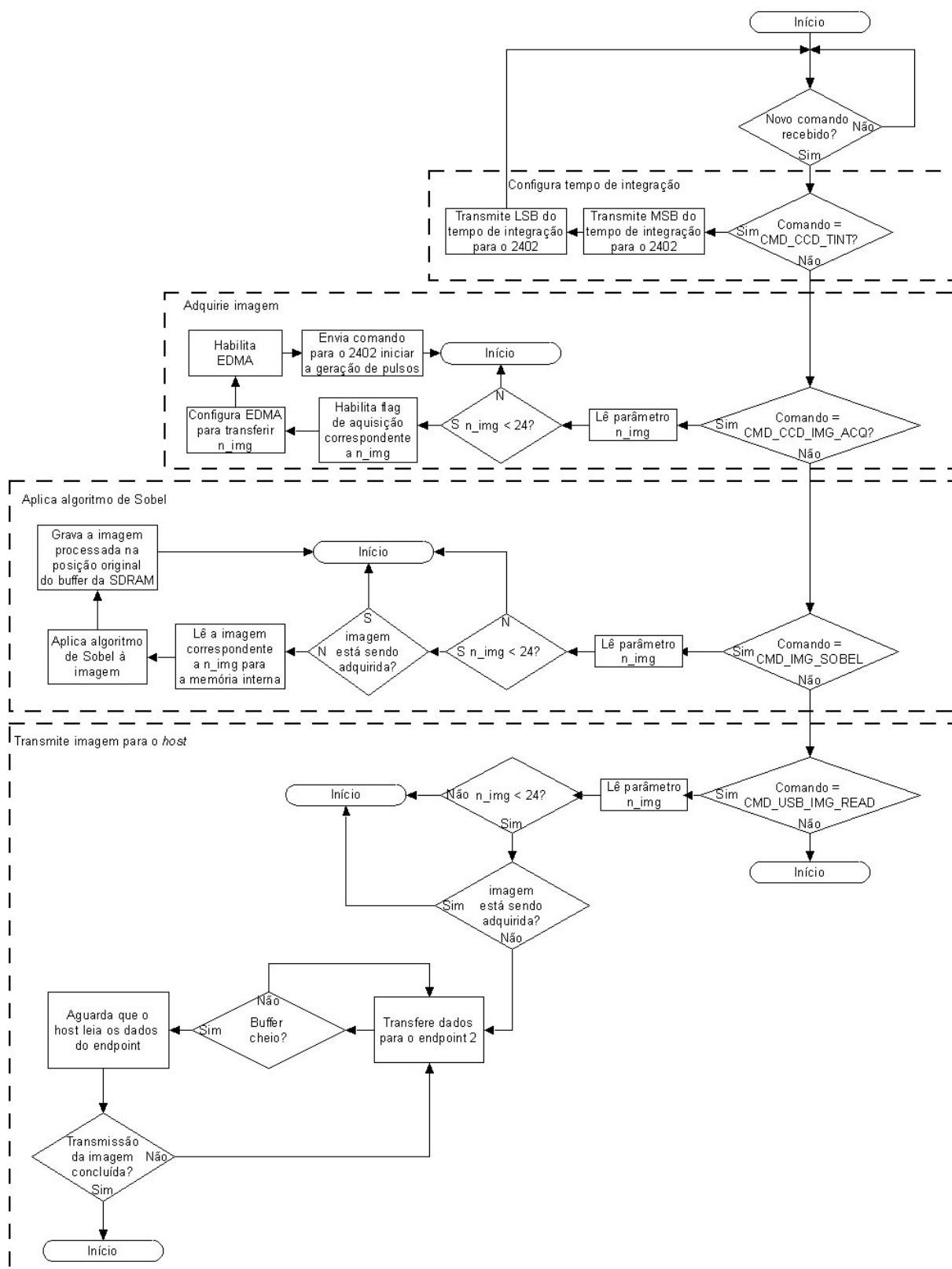


Figura 38 – Diagrama de fluxo do *firmware* do DSP TMS320C6416. Os blocos em linhas pontilhadas representam os quatro comandos implementados no *firmware*: configuração do tempo de integração, aquisição de imagem através sensor CCD, aplicação do algoritmo de Sobel e transmissão de imagem para o *host* via interface USB.

3.6 SOFTWARE PARA COMUNICAÇÃO COM O *HOST*

Para efetuar a comunicação entre o microcomputador *host* e o sistema desenvolvido implementou-se um programa em linguagem C utilizando-se o compilador Borland C++ 1.0 e o sistema operacional Windows 2000. O programa foi elaborado com o intuito de testar as funcionalidades do sistema de imagem e tem duas funções principais: enviar comandos para a câmera e ler e apresentar as imagens adquiridas.

A comunicação USB foi realizada por meio do *driver* de propósito geral EZUSB, fornecido pela Cypress. O *driver* EZUSB consiste em uma interface entre *softwares* em modo de usuário e os *drivers* do sistema operacional que acessam os controladores USB. Diversos comandos padrão da interface USB, assim como os quatro tipos de transações podem ser utilizados através do *driver* EZUSB. Para que o sistema operacional pudesse carregar o *driver* para realizar a comunicação com a câmera, foi criado um arquivo de informações do Windows (arquivo INF), o qual associa os descritores do equipamento aos arquivos do *driver* EZUSB. Após a implementação do arquivo INF, o sistema operacional foi capaz de identificar e instalar automaticamente os *drivers* do dispositivo. A figura 39 ilustra esse processo.

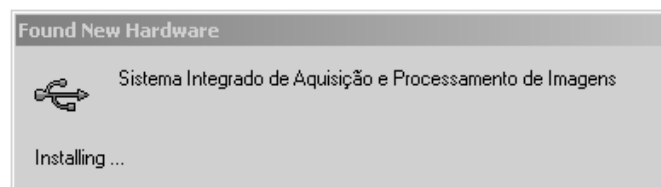


Figura 39 – Sistema operacional instalando os *drivers* do equipamento.

O programa desenvolvido possui quatro menus principais: arquivo, comunicação, imagem e configuração. A figura 40 apresenta a tela principal do *software* exibindo uma imagem adquirida pela câmera. No canto inferior esquerdo da tela principal existe o campo “Número da Imagem”, no qual pode ser inserido um valor entre 0 e 23 correspondente ao número da imagem a ser enviada ao *host* pelo SI-API, utilizando os comandos especificados na tabela 7.

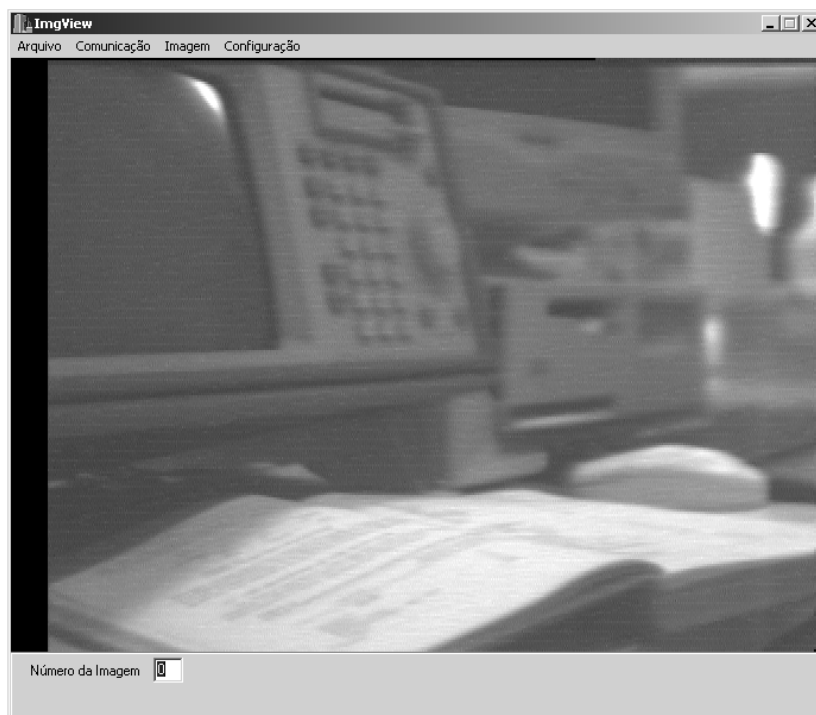


Figura 40 – Tela principal do *software* para comunicação com o *host* apresentando uma imagem adquirida pelo sistema.

3.6.1 Comunicação com a Câmera

Através da opção do menu de comunicação são solicitadas informações da câmera. O menu de comunicação apresenta três opções, conforme ilustra a figura 41.



Figura 41 – Menu de comunicação.

A opção “Adquire Imagem” envia um comando para a câmera solicitando que o TMS320C6416 inicie a aquisição de uma imagem e a armazene na posição do *buffer* de armazenamento de imagens correspondente ao valor presente no campo “Número da Imagem” do *software* de comunicação.

Através da opção “Lê Dados”, o *software* envia um comando solicitando que a câmera inicie a transmissão de uma imagem adquirida. O número da imagem a ser lida do *buffer* de armazenamento é definido pelo campo “Número da Imagem”. Após enviar o comando

solicitando a imagem, o *software* cria uma nova *thread* responsável pela leitura dos dados da USB. Uma vez que o *driver* EZUSB permite a transmissão de blocos de no máximo 64 kbytes, para transmitir uma imagem completa foi necessário implementar um laço de *software*, o qual verifica a cada iteração se o número de bytes recebidos foi inferior ao tamanho do *buffer*, o que define o fim da transmissão. Os dados recebidos são então gravados em uma variável global para armazenamento temporário das imagens adquiridas e a *thread* auxiliar é finalizada.

A opção “Solicita Aquisição Completa” envia à câmera uma seqüência de comandos solicitando a aquisição das 24 imagens que podem ser gravadas no *buffer* de armazenamento de imagens do SI-API. Essa opção faz com que a câmera grave no *buffer* uma seqüência adquirida em tempo real para processamento posterior.

No menu “Configuração” existem opções para ajustar o tempo de integração do sensor CCD e para montar seqüências de comandos específicos para serem enviadas ao equipamento, conforme ilustra a figura 42.

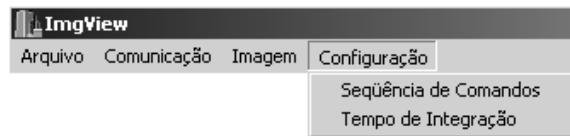


Figura 42 – Menu de configuração.

A opção “Tempo de Integração” permite que o usuário configure o tempo de exposição do sensor CCD. A figura 43 mostra a tela de configuração do tempo de integração, na qual o usuário pode inserir um novo tempo de integração em micro segundos.

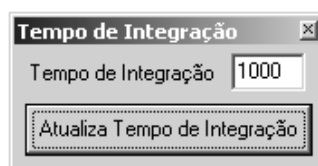


Figura 43 – Tela de configuração do tempo de integração.

A opção “Seqüência de Comandos” foi implementada para dar mais flexibilidade ao sistema. Por meio dela, o usuário pode montar uma estrutura de dados contendo uma seqüência de comandos específicos para serem executados pelo sistema. A figura 44 apresenta a tela de montagem das seqüências de comandos.

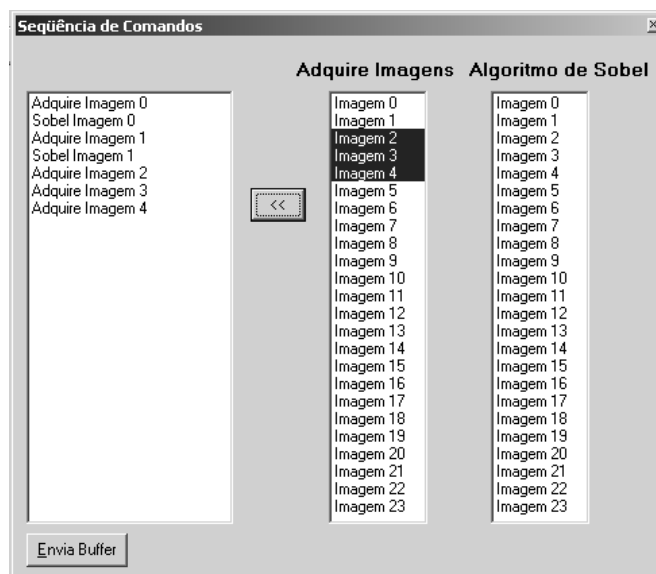


Figura 44 – Tela de montagem de seqüência de comandos.

Conforme pode ser observado na figura 44, o usuário pode selecionar em duas listas o tipo de comando que deseja aplicar e o número da imagem à qual o comando deve ser aplicado. Na implementação atual, podem ser escolhidos os comandos de aquisição de imagens e de aplicação do algoritmo de Sobel, entretanto, conforme novos algoritmos forem implementados no equipamento, novas listas de seleção podem ser acrescentadas ao *software*. A finalidade desta opção é permitir que o usuário defina a ordem em que os comandos serão executados. Por exemplo, é possível solicitar a aquisição de uma imagem e, em seguida, aplicar um algoritmo de detecção de bordas, no caso o algoritmo de Sobel, antes da aquisição da próxima imagem. Entretanto, devido à natureza paralela do sistema, a aquisição da próxima imagem terá início durante a aplicação dos algoritmos à primeira. Ressalta-se que se o tempo de execução dos algoritmos de processamento de imagem, os quais são aplicados de forma intercalada às aquisições, for menor que o tempo da aquisição, o SI-API proporciona um sistema de aquisição e processamento em tempo real.

3.6.2 Apresentação das Imagens

Para transformar os dados recebidos da câmera em padrões que possam ser interpretados pelo sistema operacional foi utilizada a biblioteca DILABIEM (ROVANI et al., 2000), a qual consiste em um conjunto de classes desenvolvidas em linguagem C++ para processamento de imagens.

A classe DIMatrixByte, utilizada para armazenar os dados recebidos da câmera, consiste em uma matriz bidimensional de dados de 8 bits, cuja finalidade é armazenar dados correspondentes a uma imagem de forma que estes possam então ser passados como parâmetro para as diversas classes de processamento de imagens.

A classe DIImageManager, por sua vez, tem a finalidade de gravar, abrir e converter imagens de diversos padrões como Windows *bitmap*, PGM, JPG, entre outras. Utilizando-se essa classe, os dados armazenados na matriz da imagem foram gravados e apresentados na tela.

A visualização das imagens obtidas é feita através das opções do menu “Imagem”, as quais podem ser observadas na figura 45.

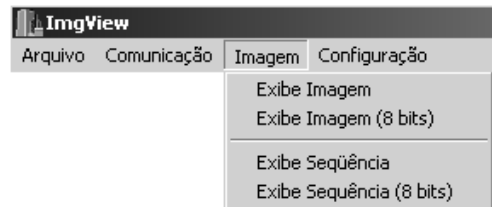


Figura 45 – Menu imagem.

As imagens adquiridas podem ser visualizadas em 16 ou 8 bits. Essa diferenciação foi feita, pois, embora as imagens sejam adquiridas com 12 bits, elas são armazenadas com 16. Os algoritmos de processamento de imagens, por outro lado, apresentam resultados de 8 bits. Para que as imagens possam ser representadas adequadamente, é necessário que o *software* tenha conhecimento da quantidade de bits utilizados.

Os comandos “Exibe Sequência”, tanto de 16 como de 8 bits, iniciam um processo de solicitação e apresentação sequencial de todas as imagens armazenadas no *buffer* de imagens do sistema de aquisição. Através deste comando é possível se ter uma idéia dos resultados obtidos pela aquisição e processamento.

CAPÍTULO 4

RESULTADOS

4.1 INTRODUÇÃO

A figura 46 apresenta uma fotografia do protótipo desenvolvido na qual se tem visão geral do SI-API acoplado à fonte de alimentação.

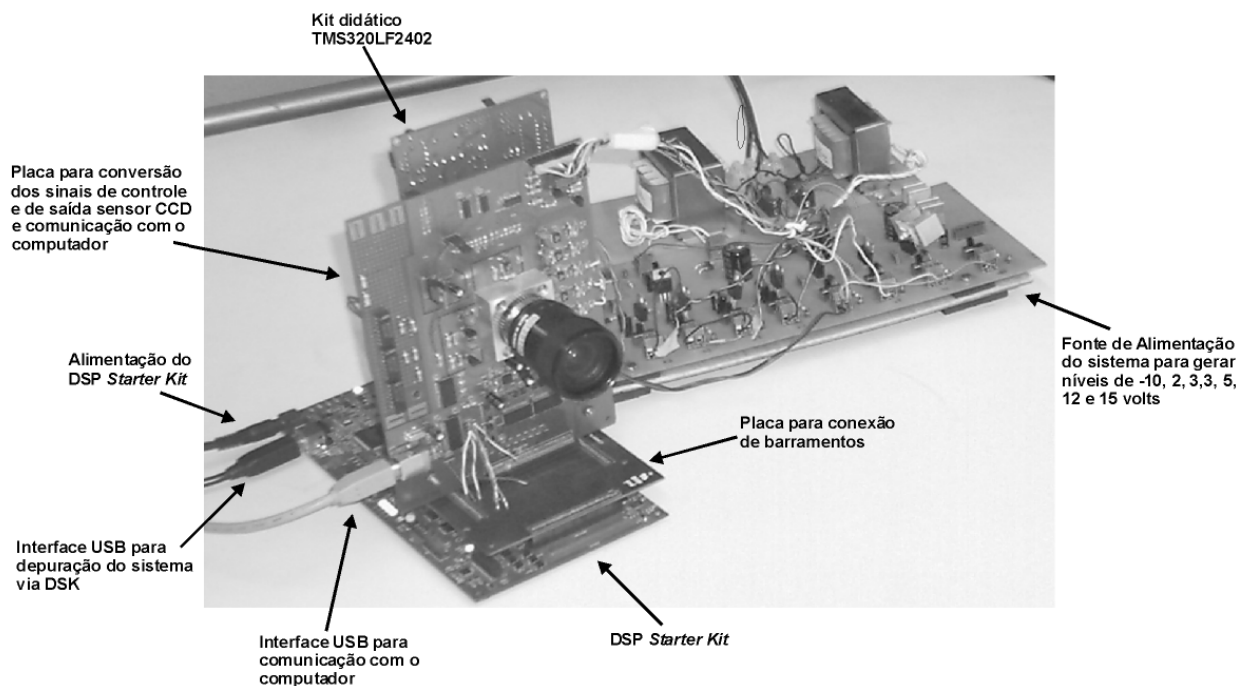


Figura 46 – Fotografia do *hardware* do sistema desenvolvido, incluindo o sistema de aquisição e processamento de imagens e a fonte de alimentação.

Na figura 46 é possível observar o kit didático do DSP TMS320LF2402 acoplado à placa responsável pela conversão dos níveis de tensão dos sinais de controle do sensor CCD e pela conversão analógica-digital dos *pixels* obtidos do sensor. Nessa placa estão também os circuitos de comunicação do SI-API com o microcomputador *host* via interface USB. Também é possível observar na figura o DSK, o qual é conectado ao microcomputador por outra interface USB através da qual é possível depurar o *firmware* do DSP TMS320C6416 por meio

do ambiente de desenvolvimento Code Composer Studio. O DSK é conectado à placa de conversão de níveis por meio de uma placa auxiliar a qual conecta o barramento do DSP aos AFEs e à interface USB. Ao lado direito do protótipo está a fonte de alimentação desenvolvida para fornecer os diversos níveis de tensão exigidos pelo sensor CCD e também para alimentar os circuitos digitais.

A figura 47 (a) apresenta uma fotografia frontal do protótipo desenvolvido. Observa-se na figura o conector para a interface USB e a lente acoplada à câmera. Na figura 47 (b), a qual apresenta uma fotografia do protótipo visto por trás, podem ser visualizados o TMS320C6416 DSK e o kit didático do DSP TMS320LF2402 acoplados ao protótipo. A figura 47 (b) destaca ainda o dissipador de calor conectado ao elemento *Peltier*.

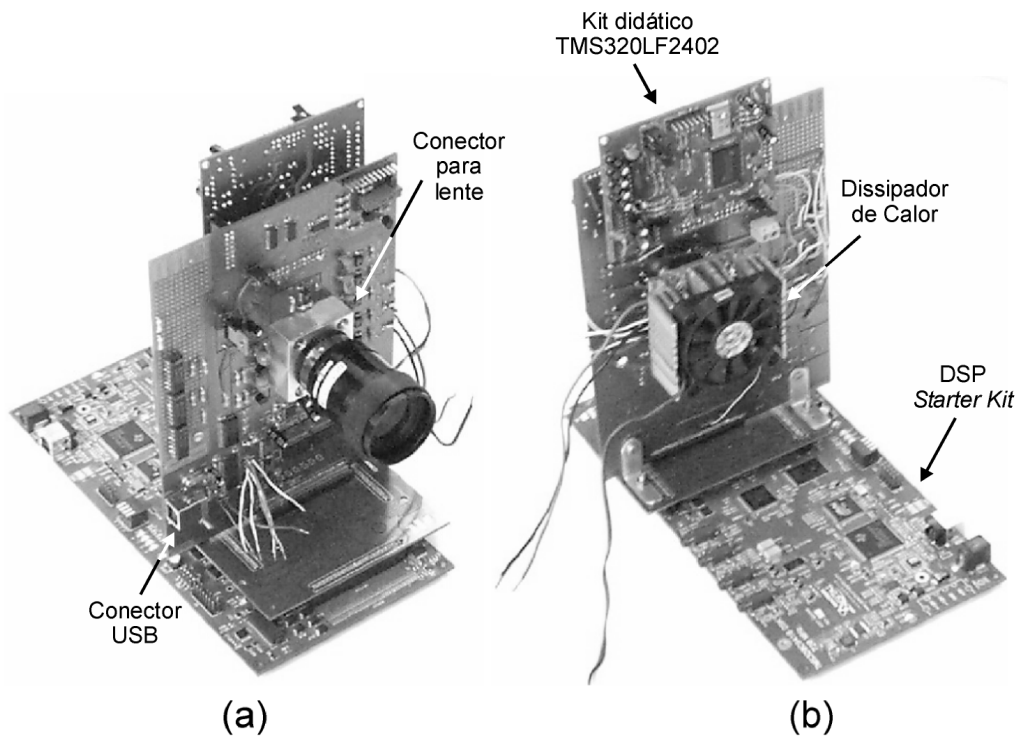


Figura 47 – Fotografias (a) frontal e (b) traseira do protótipo construído.

4.2 DISTÂNCIA FOCAL

Para testar o ajuste da distância focal do sistema foram utilizadas duas lentes do tipo *C-Mount*, com distâncias focais de 12 mm e 8,5 mm e uma lente do tipo *CS-Mount* com distância focal de 3,5 mm. As lentes *C-Mount* foram conectadas ao sistema por meio de um espaçador de 5 mm. A figura 48 (a) apresenta a imagem adquirida pela lente *CS-Mount*. As figuras 48 (b) e (c) apresentam as imagens adquiridas com as lentes *C-Mount* de 8,5 mm e 12 mm, respectivamente.

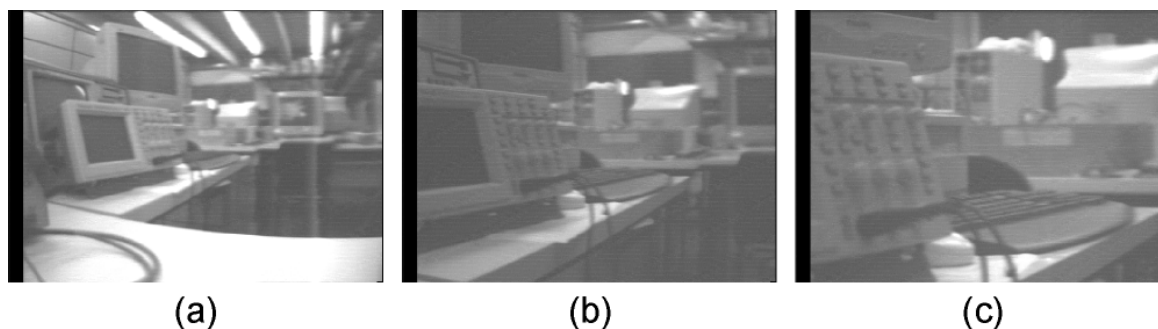


Figura 48 – Imagens obtidas com lentes do tipo (a) *CS-Mount* de 3,5 mm, (b) *C-Mount* de 8,5 mm e (c) *C-Mount* de 12 mm. Resolução de 684 x 500 pixels, tempo de integração 3 ms, 256 níveis de cinza.

4.3 RESFRIAMENTO DO SENSOR DE IMAGEM

Utilizando-se um termopar acoplado a um multímetro digital Minipa ET-2600, foram medidas as temperaturas do sensor de imagem para diversas correntes aplicadas ao elemento *Peltier*. A tabela 8 apresenta as temperaturas obtidas na superfície do sensor de imagem para diversas correntes aplicadas ao elemento *Peltier* com temperatura ambiente de aproximadamente 24°C.

Tabela 8 – Variação da temperatura sobre a superfície do sensor de imagem em função da corrente aplicada ao elemento *Peltier*.

Corrente (A)	Temperatura (°C)
0,5	10
1,0	6
1,5	2
2,0	-2
2,5	-5
3,0	-8

4.4 TEMPO DE INTEGRAÇÃO DO SENSOR DE IMAGEM

Para avaliar a resposta do sensor de imagem submetido a diferentes tempos de exposição foram adquiridas imagens variando-se o tempo de integração do sensor entre 1 ms e 17 ms, em intervalos de 500 μ s. Para garantir a menor variação possível entre as imagens, as aquisições foram realizadas em uma única seqüência, sem variação da iluminação ambiente ou dos elementos que compunham a imagem.

Com o obturador da lente completamente aberto, a partir de 4 ms de exposição, as imagens começaram a apresentar saturação, conforme pode ser observado nas figuras 49 (a), (b) e (c), as quais apresentam imagem obtidas com o sensor exposto durante 2,5, 4,5 e 7 ms, respectivamente. Conforme pode ser observado na figura 49 (a), para um tempo de integração de 2,5 ms, não ocorre saturação e a imagem adquirida é nítida. Para um tempo de 4,5 ms, apresentado na figura 49 (b), as regiões mais iluminadas da imagem começam a apresentar saturação. Finalmente, na figura 49 (c), pode-se observar que, para um tempo de integração de 7 ms, a imagem apresenta-se quase completamente saturada.

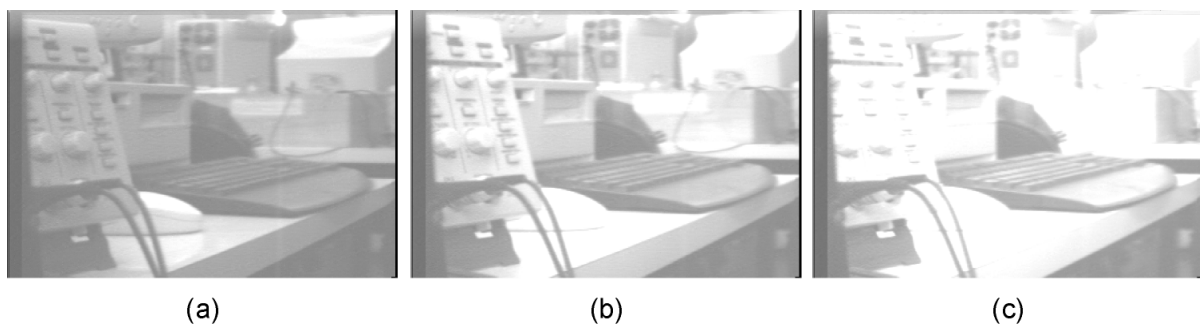


Figura 49 – Imagens obtidas com o obturador da lente completamente aberto e tempos de integração de (a) 2,5 ms, (b) 4,5 ms e (c) 7 ms. Resolução de 684 x 500 *pixels*, 256 níveis de cinza. É possível observar a saturação nas regiões mais claras das imagens conforme se aumenta o tempo de integração.

A partir dos histogramas foram calculadas as variações da intensidade média dos *pixels* das imagens em função dos tempos de integração do sensor. Para o cálculo da variação média selecionou-se uma região de interesse (RDI) de 30 por 30 *pixels* onde a intensidade média do *pixels* fosse a mais uniforme possível e cuja saturação ocorresse com o maior tempo de integração possível. A seleção da RDI foi necessária para evitar que regiões mais claras na imagem, cuja saturação ocorre antes, interferissem no cálculo da média global da imagem. Para cada tempo de integração fizeram-se três medidas da intensidade média da RDI.

Pode-se observar na figura 50 que a intensidade média dos *pixels* da RDI varia de forma linear, com uma inclinação de aproximadamente 22 níveis de cinza por milissegundo, até que tenha início o efeito de saturação, o que ocorre por volta de 9,5 ms para a região avaliada. Quando o sensor começa a sofrer os efeitos da saturação, a declividade da curva tende a zero independentemente do tempo de integração do sensor. A reta contínua sobreposta aos pontos de amostragem foi obtida utilizando-se o método dos mínimos quadrados e o coeficiente de correlação entre a reta e o valor médio das amostras é igual a 0,9962.

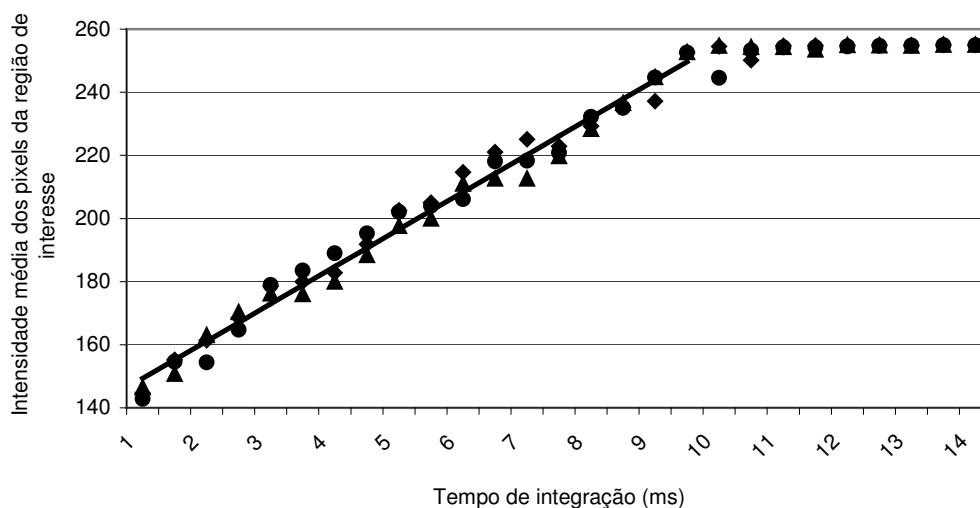


Figura 50 – Variação da intensidade média dos *pixels* da região de interesse em função do tempo de integração do sensor CCD com o obturador da lente *C-mount* de 12 mm completamente aberto. A variação obtida foi linear com inclinação de aproximadamente 22 níveis de cinza por milissegundo até 9,5 ms quando tem início o efeito de saturação.

Para que fosse possível testar a aquisição do sensor ao longo de uma faixa de integração mais ampla, a abertura do obturador da lente foi reduzida a 30% de seu valor máximo. Sob essas condições, não foi detectada saturação em qualquer imagem, conforme ilustra a figura 51, a qual apresenta uma imagem adquirida com um tempo de integração de 17 ms.



Figura 51 – Imagem obtida com o obturador da lente em 30% de sua abertura máxima e tempo de integração de 17 ms.

Conforme se pode observar na figura 52, a variação da intensidade média dos *pixels* da imagem varia linearmente com uma inclinação de 6 níveis de cinza por milissegundo ao longo de toda a faixa de tempos de exposição, a qual se estende desde 1 ms até 17 ms, em intervalos de 500 μ s. A reta sobreposta aos pontos de amostragem apresenta correlação de 0,9979 com o valor médio das amostras.

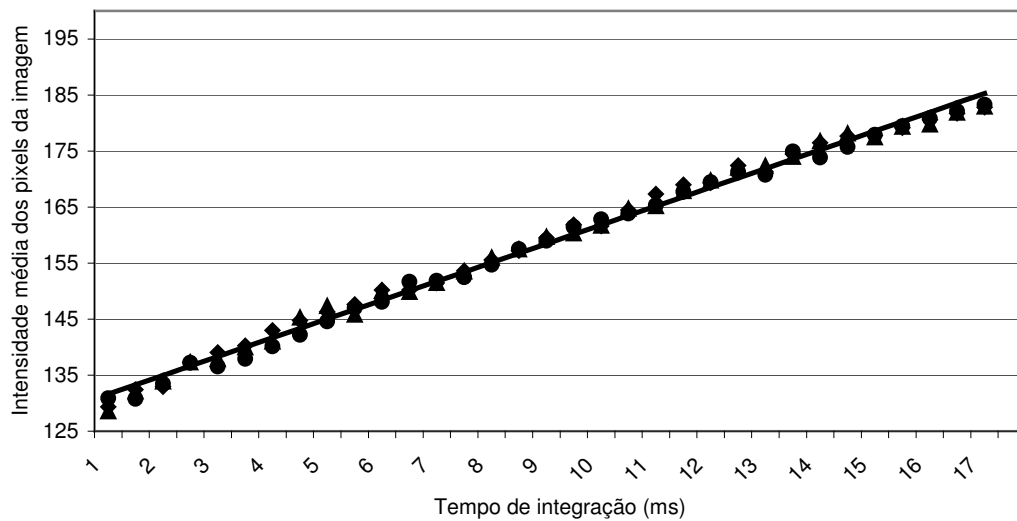


Figura 52 – Variação da intensidade média dos *pixels* da imagem em função do tempo de integração do sensor CCD com o obturador da lente em 30% de sua abertura máxima. A variação obtida foi linear com inclinação de 6 níveis de cinza por milissegundo.

4.5 DESLOCAMENTO PARA O NÍVEL DE PRETO ÓPTICO

Para se avaliar os efeitos da função de deslocamento para o nível de preto óptico das imagens (*Optical Black Clamplng* - OBC) foram adquiridas duas imagens sob condições idênticas exceto pela utilização ou não da técnica. As figuras 53 (a) e (b) apresentam, respectivamente, imagens adquiridas sem a aplicação do OBC e com a aplicação do OBC.



Figura 53 – Imagens adquiridas (a) sem e (b) com deslocamento para o nível de preto óptico

Conforme pode ser observado nas imagens das figuras 53 (a) e (b), a principal consequência de não se empregar a técnica de OBC na aquisição das imagens do sensor CCD

é uma redução considerável no contraste da imagem. Esse fenômeno fica mais claro quando os histogramas das imagens são observados simultaneamente. A figura 54 apresenta os histogramas das imagens da figura 53. A linha pontilhada representa o histograma da figura adquirida sem o deslocamento para o nível de preto óptico. O histograma da figura adquirida com a aplicação da técnica é representado pela linha contínua. Conforme é possível observar, o histograma da figura adquirida empregando-se a técnica estende-se por uma faixa consideravelmente maior de níveis de cinza.

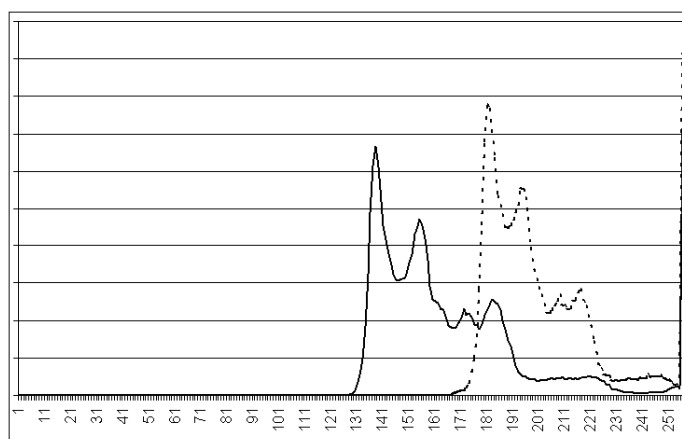


Figura 54 – Histogramas das imagens adquiridas sem deslocamento para o nível de preto óptico (linha pontilhada), figura 53 (a) e com deslocamento (linha contínua), figura 53 (b).

4.6 NÍVEL DE RUÍDO DAS IMAGENS ADQUIRIDAS

Com o obturador da lente completamente fechado foram adquiridas imagens para avaliar a intensidade de ruído no sensor de imagem. A figura 55 (a) apresenta uma imagem obtida com o obturador completamente fechado e a figura 55 (b) apresenta o histograma da imagem. Conforme pode ser observado na figura 55 (a), a imagem adquirida apresenta baixo nível de ruído. O nível médio da imagem obtida sem iluminação é aproximadamente igual a 85 e o desvio padrão é de aproximadamente 3 níveis de cinza conforme ilustra o histograma da figura 55 (b).

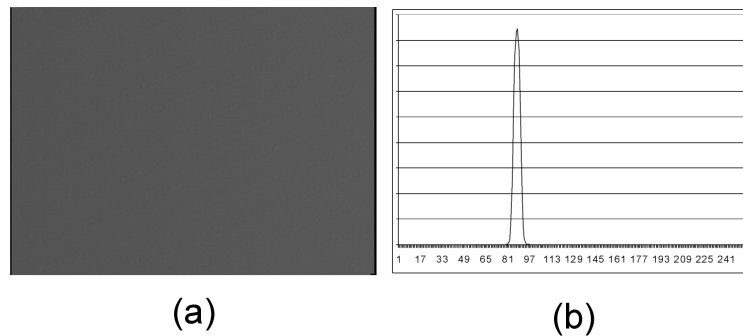


Figura 55 – (a) Imagem obtida sem iluminação incidindo sobre o sensor, (b) histograma da imagem.

4.7 TEMPO DE AQUISIÇÃO

O DSP de controle gera sinais de *clock* para os AFEs a uma frequência de 10 MHz. A essa frequência, o tempo de leitura de uma imagem é de aproximadamente 30 ms, possibilitando a aquisição de até 30 imagens por segundo. Na seção 5.1.1 são discutidas as modificações necessárias no protótipo para permitir a aquisição de imagens à velocidade de 73 quadros por segundo, limite imposto pelo sensor CCD.

4.8 TRANSMISSÃO DAS IMAGENS PARA O MICROCOMPUTADOR

Para avaliar o tempo de transmissão das imagens para o microcomputador foi utilizada a rotina da *timeGetTime()* da API do Windows, a qual retorna o número de intervalos de 1 ms transcorridos desde a inicialização do sistema. O tempo de transmissão foi calculado a partir da diferença entre os valores retornados pela rotina antes e após a transmissão da imagem. A tabela 9 apresenta os tempos transcorridos em 15 transmissões e o valor médio do tempo de transmissão.

Tabela 9 – Tempos de transmissão de imagens via interface USB.

Transmissão	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Valor Médio
Tempo (ms)	340	339	341	340	340	339	340	341	340	339	340	341	340	339	340	340

Conforme pode ser observado na tabela 9, foram necessários, em média, 340 ms para transmitir uma imagem para o computador. Uma vez que cada imagem é composta por

342.000 *pixels*, a taxa de transmissão obtida foi de aproximadamente 1 Mbyte/segundo, ou 8 Mbits/segundo.

4.9 PROCESSAMENTO DAS IMAGENS

Para avaliar a capacidade de processamento do sistema foi aplicado às imagens adquiridas o algoritmo de Sobel para detecção de bordas (GONZALEZ e WOODS, 1992). O algoritmo de Sobel aplica duas máscaras de 3×3 *pixels* sobre cada *pixel* da imagem, calcula os módulos dos resultados, soma-os e armazena o valor final em uma nova imagem. As figuras 56 (a) e (b) apresentam, respectivamente, uma imagem adquirida pelo sistema e a mesma imagem após a aplicação do algoritmo de Sobel.

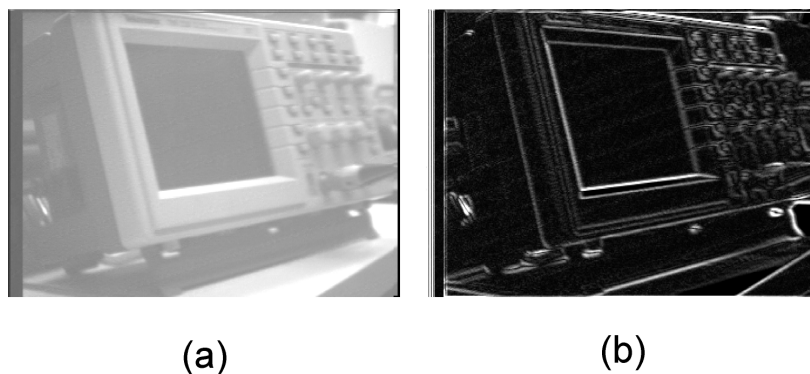


Figura 56 – (a) Imagem adquirida pelo sistema, (b) imagem após a aplicação do algoritmo de Sobel. Resolução de 684×500 *pixels*, tempo de integração 3 ms, 256 níveis de cinza.

Para avaliar o desempenho do sistema, o mesmo algoritmo foi implementado em linguagem C em um microcomputador Athlon XP 2.4 GHz. Para garantir o máximo desempenho da rotina implementada no microcomputador, as opções de otimização do compilador C foram configuradas para gerar o código *assembly* mais veloz possível. Utilizando-se a rotina *SetPriorityClass()* da API do sistema operacional Windows, a prioridade do processo de execução do algoritmo de Sobel foi definida como `REALTIME_PRIORITY_CLASS`. Dessa forma, nenhum outro processo pôde interromper a execução do algoritmo, nem mesmo processos do sistema operacional como rotinas de tratamento do mouse ou teclado.

Utilizando-se a rotina *timeGetTime()* do sistema operacional Windows, a qual apresenta resolução de 1 ms, foi avaliado o tempo de execução do algoritmo pelo microcomputador. O microcomputador executou o algoritmo em aproximadamente 10 ms.

Para indicar os instantes de início e fim da aplicação do algoritmo de Sobel a uma imagem pelo DSP TMS320C6416 foi utilizado um pino de I/O e através de um osciloscópio digital foi medido o tempo transcorrido entre esses dois eventos. Utilizando-se esse método, foi possível determinar que o SI-API aplicou o algoritmo de Sobel às imagens em 600 μ s. Ou seja, o DSP executou o algoritmo de Sobel aproximadamente 16 vezes mais rapidamente que o microcomputador.

CAPÍTULO 5

DISCUSSÃO E CONCLUSÕES

5.1 DISCUSSÃO

5.1.1 Aquisição das Imagens

Na implementação atual, o DSP de controle gera sinais de *clock* para os AFEs a uma frequência de 10 MHz, ou seja, o período de cada *pixel* do sensor CCD é de 100 ns. A esta velocidade, o tempo de aquisição de uma imagem é de aproximadamente 30 ms. O sensor TC237 é capaz de operar a até 12 MHz, permitindo a aquisição de até 36 imagens por segundo utilizando-se um único registrador de saída. Embora o *hardware* implementado permita o aumento de 20% necessário à frequência de *clock* dos AFEs, o alto tráfego no barramento de dados do sistema exige que essa modificação seja feita de forma cautelosa, de modo a evitar perda de dados no processo de leitura do sensor.

Em relação ao modo de leitura do sensor, na implementação atual do *firmware* do SI-API, apenas um dos registradores de saída do sensor de imagem está sendo utilizado. Efetuando-se as modificações necessárias no *firmware*, de modo a utilizar ambos os registradores de saída, o tempo de leitura das imagens pode ser reduzido à metade. Na implementação atual, ou seja, utilizando-se o *clock* de 10 MHz, seria possível a aquisição de até 60 imagens por segundo. Com o aumento de 20% da frequência de *clock* dos AFEs e a implementação do modo de leitura utilizando dois registradores, o sistema será capaz de adquirir imagens na velocidade máxima permitida pelo sensor de 73 quadros por segundo.

5.1.2 Transmissão das Imagens para o Microcomputador

A transmissão de imagens para o microcomputador via interface USB alcançou uma taxa de aproximadamente 8 Mbits/s. Embora a especificação USB defina a velocidade

máxima de transmissão em modo *full speed* como 12 Mbits/s, essa taxa não leva em conta o *overhead* dos pacotes de comunicação. Como os pacotes no modo *full speed* são compostos por apenas 64 *bytes* de dados, a quantidade de *overhead* é relativamente alta. Além disso, a técnica de *bit stuffing* também implica algum *overhead* na comunicação. Dessa forma, taxas de transmissão obtidas na prática utilizando-se o modo *full speed*, em geral, não costumam ultrapassar 8 ou 9 Mbits/s.

5.2 CONCLUSÕES

Nesse trabalho apresentou-se o projeto, implementação e testes de um sistema integrado de aquisição e processamento de imagens. O sistema é caracterizado pela possibilidade de se controlar os parâmetros do sensor de imagem, como o tempo de exposição à luz, velocidade de transferência das imagens e temperatura de operação do sensor, assim como pela possibilidade de processar imagens em tempo real e, simultaneamente, transferi-las a um microcomputador.

Verificou-se que todos os elementos do sistema estavam operacionais e, em sua maioria, apresentaram o melhor desempenho possível. O sistema de acoplamento das lentes apresentou bons resultados, permitindo a conexão e o ajuste focal bem sucedidos de diversos tipos de lentes. O sistema de acoplamento térmico entre o elemento de resfriamento e o sensor de imagem permitiu a transmissão adequada da potência térmica, através do *coldplate*, do sensor de imagem para o *Peltier*. A dissipação do calor gerado pelo módulo *Peltier* também foi plenamente satisfatória. O controle do sensor de imagem, embora operando a uma frequência um pouco inferior à desejada, foi capaz de proporcionar a aquisição de imagens sem qualquer tipo de distorção. As etapas de processamento analógico do sinal de saída do sensor CCD e a conversão para o modo digital também foram bem sucedidas. O DSP TMS320C6416 demonstrou ser capaz de processar imagens com um desempenho consideravelmente superior ao de um microcomputador de alto desempenho e a transmissão das imagens via USB em modo *full speed* obteve excelente taxa de transmissão.

O sistema desenvolvido será, certamente, uma eficiente ferramenta de pesquisas na área de processamento digital de imagens, permitindo o desenvolvimento de sistemas capazes de aplicar algoritmos com elevadas demandas computacionais sem a necessidade de transmitir grandes fluxos de informações para computadores centrais de alto desempenho.

5.3 TRABALHOS FUTUROS

5.3.1 Sistema de Gerenciamento de Tarefas

Na implementação atual do *firmware* do TMS320C6416, o DSP é responsável pela transmissão das imagens. Uma vez que o controlador de DMA do DSP permite a implementação de até 64 canais independentes, é possível deixar as tarefas de transmissão e recepção de informações do microcomputador sob responsabilidade do DMA, permitindo que o DSP fique livre para realizar exclusivamente tarefas de processamento das imagens.

O controle do *buffer* de armazenamento das imagens adquiridas também precisa ser aperfeiçoado. Na implementação atual, o sistema adquire imagens até preencher completamente o *buffer*, quando então passa a aguardar comandos para iniciar uma nova seqüência de aquisição. É necessário desenvolver um sistema dinâmico de controle do *buffer*, o qual permita que o usuário defina se o *buffer* deve ser tratado como uma fila circular, sobrepondo as imagens já adquiridas ou se as imagens devem permanecer armazenadas até que sejam processadas.

Implementada a transmissão de imagens para o microcomputador via DMA e o controle dinâmico do *buffer* de alocação será possível desenvolver um sistema capaz de processar imagens e transmitir dados ao microcomputador continuamente, sem a necessidade de qualquer interação por parte do usuário.

5.3.2 Sistema de Armazenamento e Controle de Algoritmos

Atualmente, a implementação de novos algoritmos de processamento de imagens depende completamente do usuário, o qual deve implementar o algoritmo e inseri-lo diretamente no *firmware* do TMS320C6416. Por meio da implementação de um sistema capaz de receber do microcomputador algoritmos devidamente depurados e compilados pelo compilador do TMS320C6416, juntamente com informações sobre os parâmetros do algoritmo, é possível fazer com que algoritmos desenvolvidos para o TMS320C6416 possam ser armazenados na memória do DSP e executados pelo usuário por meio de comandos transmitidos via microcomputador, sem a necessidade de modificar o *firmware* do DSP.

5.3.3 Transmissão das Imagens para o Microcomputador

O *hardware* desenvolvido permite a comunicação com a interface USB 2.0 em modo *high speed*, o qual possibilita a comunicação a taxas de até 480 Mbits/s. Para implementar esse modo de comunicação é necessária a configuração dos descritores do sistema para operar no modo *high speed* e pequenos ajustes no *firmware* do TMS320C6416. Ressalta-se que nessa velocidade é possível a transferência de até 170 quadros/segundo com uma resolução de 684 x 500 *pixels* e 256 níveis de cinza, o que possibilitaria a transmissão das 73 imagens adquiridas por segundo pelo sensor em resolução máxima, deixando ainda largura de banda disponível suficiente para transmitir outras 97 imagens por segundo. Essas imagens poderiam incluir, por exemplo, imagens resultantes de etapas de processamento realizadas pelo sistema.

5.3.4 Versão Definitiva do *Hardware*

O protótipo desenvolvido está baseado no TMS320C6416 DSK e no kit didático do TMS320LF2402. Para possibilitar a fabricação de várias unidades do sistema de processamento sem a necessidade da aquisição de vários *starter kits*, seria necessário desenvolver uma placa de circuito impresso contendo o DSP TMS320C6416, bancos de memória SDRAM para armazenamento de imagens, memória FLASH para armazenamento de programa e interface USB. A placa desenvolvida seria então acoplada à outra placa contendo o sensor de imagem e seus circuitos de controle e conversão analógico-digital.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABOWD, G.D.; BOBICK, A.; ESSA, I.; MYNATT, E.; ROGERS, W. *The Aware Home: Developing Technologies for Successful Aging*. Proceedings of AAAI Workshop on Automation as a Care Giver, Alberta, Canadá, 2002.
- AMELIO, G.F.; BERTRAM, W.J.; TOMPSETT, M.F. *Charge-Coupled Imaging Devices: Design Considerations*. IEEE Transactions on Electron Devices, vol. ED-18, p.986-992, 1971.
- ANALOG DEVICES. *ADSP-2106x SHARC User's Manual, Revision 2.1*. Literature Number: 82-000795-03, 2004a.
- ANALOG DEVICES. *12-Bit CCD Signal Processor with Precision Timing Core AD9949*. Literature Number: DO3751-0-11/04(B), 2004b.
- ARVOO IMAGING PRODUCTS, *Leonardo CL Product Information*, 2002-2.03, Woerden, Holanda, 2003.
- AXELSON, J. *USB Complete: Everything You Need to Develop Custom USB Peripherals*. Madison: Lakeview Research, 2001.
- BALCH, M. *Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture*. McGraw-Hill, 2003.
- BARBE, D.F. *Imaging Devices Using the Charge-Coupled Concept*. Proceedings of the IEEE, vol. 63, p.38-67, 1975.
- BASS, M. (Ed). *Handbook of Optics*. 2nd. ed. New York: McGraw-Hill, 1995.
- BRAMBERGER, M.; BRUNNER, J.; RINNER, B.; SCHWABACH, H. *Real-Time Video Analysis on an Embedded Smart Camera for Traffic Surveillance*. Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium, p.174-181, 2004.

- BRANCO, G.; JAKUBIAK, D. R.; GEWEHR, P. M. *Leds as Optical Source for a Phosphorescent Oxygen Probe*. Proceedings of the 3rd. European Conference on Optical Sensors and Biosensors (Europt(r)ode III), p. 79, Zurich, Switzerland, 1996.
- CARNES, J.E.; KOSONOCKY, W.F. *Noise Sources in Charge-Coupled Devices*. RCA Review, vol. 33, p.327-343, 1972.
- CHASSAING, R. *DSP Applications Using C and the TMS320C6x DSK*. John Willey & Sons, 2002.
- CONCI, A.; PROENÇA, C.B. *A System For Real-Time Fabric Inspection and Industrial Decision*, Proceedings of The 14th ACM International Conference on Software Engineering and Knowledge Engineering, p.707-714, 2002.
- CORECO INC. *Python/C6 Product Specification*, CODS-1093, St. Laurent, Canadá, 1998.
- DALLY, W.; KAPASI, U.; KHAILANY, B.; AHN, J.H.; DAS, A. *Stream Processors: Programmability with Efficiency*. ACM Queue Focus: DSP, v. 2, ed. 1, p.52-62, 2004.
- DUDGEON, D.E.; MERSEREAU, R.M. *Multidimensional Digital Signal Processing*, Englewood Cliffs: Prentice-Hall, 1984.
- EASTON, R. A. *Smart Digital Cameras for Product Quality Inspection*, WESCON/96, p. 402-404, 1996.
- FEINER, S.; MACINTYRE, B.; SELIGMANN, D. *Knowledge-Based Augmented Reality*, Communications of the ACM Special Issue on Computer Augmented Environments: Back to the Real World, v.36, ed.7, p.53-62, 1993.
- FERRAZ JUNIOR, A. G. *Contribuição ao Estudo de Imagens em Fiscalização Eletrônica de Velocidade*. (Dissertação de Mestrado) Centro Federal de Educação Tecnológica do Paraná (CEFET), Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Curitiba, 2004.
- GADELLA, T.W.J. *Fluorescence Lifetime Imaging Microscopy (FLIM)*, Microscopy and analysis, pp. 13-15, 1997.
- GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*, USA: Adisson-Wesley, 1992.

- GOTO, Y.; MATSUZAKI, K.; KWEON, I.; OBATAKE, T. *CMU Sidewalk Navigation System: A Blackboard-Based Outdoor Navigation System Using Sensor Fusion with Colored-Range Images*, Proceedings of ACM Fall Joint Computer Conference, p.105-113, 1999.
- HAYS, W.P. *DSPs: Back to the Future*. ACM Queue Focus: DSP, vol.1, n. 1, p. 42-15, 2004.
- HAYES, J.P. *Computer Architecture and Organization*. 7a. Ed. McGraw-Hill, 1986.
- HEATH, S. *Microprocessor Architecture and Systems: RISC, CISC & DSP*. Oxford: Newnes, 1991.
- HARTGRING, C.D.; KLEEFSTRA, M. *Quantum Efficiency and Blooming Suppression in Junction Charged-Coupled Devices*. IEEE Journal of Solid-State Circuits, vol. 13, n.5, p.728-730, 1978.
- HAZZAH, K. *Writing Windows VxDs and Device Drivers*. Lawrence: R&D Books, 1997.
- HENNESSY, J. L.; PATTERSON, D. A. *Computer Architecture: A Quantitative Approach*, USA: Morgan Kaufman, 1996.
- IMBIRIBA, R. S. *Sistema Automático de Identificação de Placas de Veículos*. (Dissertação de Mestrado) Centro Federal de Educação Tecnológica do Paraná (CEFET), Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Curitiba, 2002.
- INOUE, S.; SPRING, K.R. *Video Microscopy: the Fundamentals*. London: Plenum Press, 1997.
- IVP, *User Documentation MAPP2200 PCI System*, 1997
- JACHALSKY, J.; WAHLE, M.; PIRSCH, P.; CAPPERON, S.; GEHRKE, W.; KRUIJTZER, W.M.; NUÑEZ, A. *A Core for Ambient and Mobile Intelligent Imaging Applications*, Proceedings of the International Conference on Multimedia and Expo. ICME '03, Vol.2, p. 1-4, 2003.
- JAKUBIAK, D. R; MARTELLI, C.; BRANCO, G.; GEWEHR, P. M. *PD TBP As a New Probe for a Gaseous Oxygen Sensor*. Proceedings of the 5th. European Conference on Optical Chemical Sensors and Biosensors. Lyon, França. p. 153-154, 2000.

- JENKINS, T. E. *Optical Sensing Techniques and Signal Processing*. Englewood Cliffs: Prentice-Hall, 1987
- KNEIP, J.; BAUER, S.; VOLLMER, J.; SCHMALE, B.; KUHN, P.; REISSMANN, M. *The MPEG-4 Video Coding Standard-a VLSI Point of View*. IEEE Workshop on Signal Processing Systems, p.43-52, 1998.
- KODAK, *Charge-Coupled Device (CCD) Image Sensors*, Eastman Kodak Company, Image Sensor Solutions, CCD Primer, MTD/OS-0218, Rochester, New York, Revision N° 1, Mai, 29, 2001a.
- KODAK, *Conversion of Light (Photons) to Electronic Charge*, Eastman Kodak Company, Image Sensor Solutions, CCD Primer, MTD/OS-0217, Rochester, New York, Revision N° 1, Mai, 29, 2001b.
- KODAK, *CCD Image Sensor Noise Sources*, *Image Sensor Solutions*, Eastman Kodak Company, Application Note MTD/OS-0233, Rochester; New York, Revision 1, Ago, 8, 2001c.
- KOSMAN, S.L.; STEVENS, E.G.; CASSIDY, J.C.; CHANG, W.C.; ROSELLE, P.; MILLER, W.A.; MEHRA, M.; BURKEY, B.C.; LEE, T.H.; HAWKINS, G.A.; KHOSLA, R.P. *A Large Area 1.3-Megapixel Full-Frame CCD Image Sensor with a Lateral-Overflow Drain and a Transparent Gate Electrode*. Technical Digest of the International Electron Devices Meeting, p.287-290, 1990.
- KUMURA, T.; IKEKAWA, M.; YOSHIDA, M.; KURODA, I. *VLIW DSP for Mobile Applications*. IEEE Signal Processing Magazine, v.19, n.4, p.10-21, 2002.
- KUO, S.M.; LEE, B.H. *Real-Time Digital Signal Processing: Implementations, Applications and Experiments with the TMS320C55x*. West Sussex: John Wiley & Sons, 2001.
- KURODA, I.; NISHITANI, T. *Multimedia Processors*. Proceedings of the IEEE, v. 86, n. 6, p. 1203-1221, 1998.
- LEE, K.M.; MEYOUHAS, G.; BLENIS, R. *A Machine-Vision-Based Wrist Sensor for Direct Measurement of Three Degrees-of-Freedom Orientation*, *Mechatronics*, vol. 3, no. 5, p. 571-587, 1993.

- LEE, K.M. *Design Concept of an Integrated Vision System for Cost-Effective Flexible Part-Feeding Applications*, ASME Transactions on Engineering for Industry, vol. 116, pp. 421-428, 1994.
- LEE, K.M.; VAN KUREN, M.M.B. *Modeling and Supervisory Control of a Disassembly Automation Workcell Based on Blocking Topology*, IEEE Transactions on Robotics and Automation, vol. 6, pp. 67-77, 2000.
- LEE, K.M.; SOBH, T.M. *Visionary Prototyping*. IEEE Robotics and Automation Magazine, vol. 8, n. 3, p. 15-24, 2001.
- LV, T.; OZER, I.B.; WOLF, W. *Parallel Architectures for Video Processing in a Smart Camera System*, IEEE Workshop on Signal Processing Systems, SIPS '02, p. 9-14, 2002.
- MATSUDA, L.T.; RIELA, R.J.; SCANDELARI, L. *Material de Aula da Disciplina de Microprocessadores II*, 2003.
- MATSUSHITA, N.; HIHARA, D.; USHIRO, T.; YOSHIMURA, S.; REKIMOTO, J.; YAMAMOTO, Y. *ID CAM: A Smart Camera for Scene Capturing and ID Recognition*. Proceedings of the 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality, p.227-236, 2003.
- NATIONAL INSTRUMENTS. *IMAQ NI CVS-1450 Series User Manual*, 373610D-01, 2004.
- NAVAUX, P. O. A. *Processadores Pipeline e Processamento Vetorial*. São Paulo: VII Escola de Computação, 1990.
- NULL, L.; LOBUR, J. *The Essentials of Computer Organization and Architecture*. Sudbury: Jones and Bartlett Publishers, 2003.
- OHBA, K.; ORTEGA, C.; TANIE, K.; RIN, G.; DANGI, R.; TAKEI, Y.; KANEKO, T.; KAWAHARA, N. *Real-Time Micro Observation Technique for Tele-Micro-Operation*. Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems, p. 647-652, 2000.
- PAPOULIS, A. *Probability, Random Variables, and Stochastic Processes*. 3rd Ed. McGraw-Hill, 1991.

- RIEKE, G.H. *Detection of Light: From the Ultraviolet to the Submillimeter*. Cambridge: Cambridge University Press, 1994.
- RISTIC, L. *Sensor Technology and Devices*. Boston: Artech House, 1994.
- ROVANI, A.Z.; GAMBA, H.R.; OSOWSKY, J. *Biblioteca para Implementação de Algoritmos de Processamento Digital de Imagens*, Anais do V Seminário de Iniciação Científica e Tecnológica do CEFET-PR, p.9-12.
- SCHWIEBERT, L.; GUPTA, S.K.S.; AUNER, P.S.G.; ABRAMS, G.; IEZZI, R.; MCALLISTER, P. *A Biomedical Smart Sensor for the Visually Impaired*, *Proceedings of IEEE Sensors*, v.1, p. 693-698, 2002.
- SCOTT, A.W. *Cooling of Electronic Equipment*. New York: John Wiley & Sons, 1974.
- SMITH, S.W. *The Scientists and Engineer's Guide to Digital Signal Processing*. San Diego: California Technical Publishing, 1999.
- TANENBAUM, A.S. *Organização Estruturada de Computadores*. 4^a. Ed. Rio de Janeiro: Livros Técnicos Editora, 1999.
- TECHMER, A. *Real-Time Motion Based Vehicle Segmentation in Traffic Lanes*, *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition*, Springer-Verlag, London, p.202-207, 2001.
- TEXAS INSTRUMENTS. *TMS320C6000 CPU and Instruction Set Reference Guide*. Literature Number: SPRU189F, 2000.
- TEXAS INSTRUMENTS, *TC237B 684x500 Pixel CCD Image Sensor*, Literature Number: SOCS063, 2001.
- USB-IF, Universal Serial Bus Implementers Forum, *Universal Serial Bus Specification, Revision 2.0*, 2000.
- WHITE, M.H.; BLAHA, F.C. *Characterization of Surface Channel CCD Image Arrays at Low Light Levels*. *IEEE Journal of Solid-State Circuits*, vol. SC-9, p.1-12, 1974.
- WILSON, J.; HAWKES, J. *Optoelectronics: an Introduction*. 2nd Ed. New York: Prentice Hall, 1983.

- WOLF, W.; OZER, I.B. *A Smart Camera for Real-time Human Activity Recognition*, IEEE Workshop on Signal Processing Systems, p. 217-224, 2001.
- WOLF, W.; OZER, I.B.; LV, T. *Architectures for Distributed Smart Cameras* Proceedings of the International Conference on Multimedia and Expo, ICME '03, v. II, p. 5-8, 2003a.
- WOLF, W.; OZER, I.B.; LV, T. *Smart Cameras as Embedded Systems*, IEEE Computer, v.35, ed. 9, p. 48-53, 2003b.
- WU, G.; WU, Y.; JIAO, L.; WANG, Y-F.; CHANG, E.Y. *Multi-Camera Spatio-Temporal Fusion and Biased Sequence-Data Learning for Security Surveillance*, Proceedings of the 11th ACM international conference on Multimedia, New-York, p.528-538, 2003.
- ZELENOVSKY, R.; MENDONÇA, A. *PC: Um Guia Prático de Hardware e Interfaceamento*. 3. ed. Rio de Janeiro: MZ, 2002.

RESUMO

Este trabalho apresenta o projeto, a implementação e os testes de um sistema integrado de aquisição e processamento de imagens (SIAPI). O sistema desenvolvido é constituído por um sensor de imagem CCD (*Charge Coupled Device*), um processador digital de sinais (*Digital Signal Processor* – DSP) de baixo custo responsável pelo controle do sensor, um DSP de alto desempenho para realizar o processamento das imagens adquiridas, uma interface USB (*Universal Serial Bus*) para realizar a comunicação do dispositivo com um microcomputador e um módulo de resfriamento termo-elétrico baseado no efeito *Peltier* para minimizar a quantidade de ruído térmico gerado no sensor de imagem. O sistema permite que se controle todos os parâmetros do sensor de imagem, bem como o processamento das imagens adquiridas em tempo real. Para avaliar as características e o desempenho do sistema, construiu-se um protótipo baseado em um DSK (*DSP Starter Kit*) e desenvolveu-se um *software* em linguagem C para comunicação com o sistema. Foram levantadas as características de resposta do sensor de imagem em diferentes configurações. O sistema demonstrou-se funcional e seu desempenho na execução de algoritmos de processamento de imagens foi bastante superior ao de um microcomputador de alto desempenho.

PALAVRAS-CHAVE

Processadores digitais de sinais/Sensores de imagem/*Charge coupled devices*

ÁREA/SUB-ÁREA DE CONHECIMENTO

3.04.00.00-7 Engenharia Elétrica

3.04.03.03-0 Circuitos Eletrônicos

2005 Nº 369
