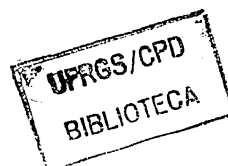


UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
PÓS GRADUAÇÃO EM CIÊNCIA DE COMPUTAÇÃO

**CV2D** - *Operador de Convolução para  
Processamento de Imagens em Tempo Real*

por

João C. Netto



Dissertação submetida como requisisto parcial para a obtenção do grau de  
Mestre em Ciência da Computação

Prof. Tiaraju Vasconcellos Wagner  
Orientador

Porto Alegre, Dezembro de 1989

3416-4

## Catalogação na Fonte

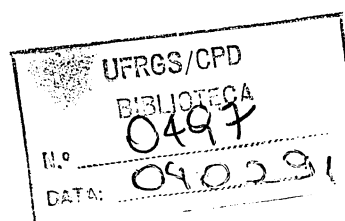
Netto, João Cesar

CV2D - Operador de convolução para processamento de imagens em tempo real. Porto Alegre, PGCC - UFRGS, 1989

1v.

Diss. (mestr. ci. comp.) UFRGS - PGCC, Porto Alegre, BR-RS, 1989.

Dissertação: Processamento de Imagens: Processamento de Sinais:  
Convolução: Arquiteturas Sistólicas.



"Senhor que queres que eu faça?"  
São Francisco de Assis

## AGRADECIMENTOS

Agradecimentos à Universidade Federal do Rio Grande do Sul que proporciona e incentiva a pesquisa mesmo com os poucos recursos destinados a este fim pelo governo federal. À Université Catholique de Louvain que, através de um convênio de cooperação, possibilitou o projeto e implementação do protótipo do circuito CV2D. Também aos órgãos financiadores públicos, CNPq e CAPES, e particulares, IBM, pelo apoio financeiro na forma de bolsa de mestrado, permitindo a dedicação ao trabalho.

Aos meus pais, que sem o esforço na orientação, formação, amizade e sacrifício financeiro não existiria este trabalho. A minha namorada, pela sua paciência, presença e amor e também pela inumerável quantidade de cartas durante minha ausência.

Fica o agradecimento a todos meus professores, pelos conhecimentos transmitidos e estímulo científico e intelectual. Em especial ao companheiro e orientador Tiaraju Wagner, que aceitou enfrentar junto este trabalho e pelas horas de bate-papo sempre produtivas e também ao prof. Philippe Navaux por sua intervenção junto a CAPES no processo de estágio no exterior. Ao prof. Raul Weber pelo auxílio na utilização dos MACs e paciência na conversão de formatos.

Ao "gurú" e amigo Altamiro Suzim, por suas sugestões e perguntas e por mostrar, através de suas atitudes, o comportamento do espírito científico e o entusiasmo de acreditar nas pessoas.

Ao pessoal que fica atrás da cortina, bibliotecárias, secretárias, telefonistas, recepcionistas e faxineiras, que acompanharam este trabalho desde o seu início e forneceram a infraestrutura para sua realização.

Ao time da engenharia, 1985, sempre dispostos a realizar os projetos mais impossíveis: Ricardo Telichevesky, Fernando Belmonte, Eduardo Todt, Régis Huf, Carlos Prolo, Luigi Carro, Cirano Silveira e aos demais que estiveram juntos.



Ao amigo e sócio Alexandre Caríssimi, pela amizade e batalhas enfrentadas juntos. Também por aquelas "broncas" deixadas como herança e pelo seu aprendizado e parceria de chimarrão.

Aos amigos belgas, Frank Vos e André Vandelmelbrok, pela acolhida no país estrangeiro, disponibilidade de tempo e principalmente pela amizade e paciência nos diálogos em francês. Também ao belgo-brasileiro Ricardo Jacobi com quem tive oportunidade de trocar muitas idéias durante os seis meses que convivemos em Louvain.

Muitas pessoas contribuíram, direta ou indiretamente, na realização deste trabalho, sem as quais este não chegaria ao fim e citar todas as pessoas que comigo participaram dos últimos três anos seria impossível. Fica no entanto o agradecimento ao longo da convivência da vida.

**De forma especial agradeço a Deus, pelo magnífico dom da vida e pelas pessoas que passaram em meu caminho.**

## SUMÁRIO

GLOSSÁRIO.....	9
LISTA DE ABREVIATURAS.....	15
LISTA DE FIGURAS.....	17
RESUMO.....	19
ABSTRACT.....	21
1 INTRODUÇÃO.....	23
1.1 <u>Conceitos Básicos</u> .....	23
1.2 <u>Escopo e Organização do Texto</u> .....	26
2 SINAL DE VÍDEO.....	29
2.1 <u>Modelo de Imagem</u> .....	29
2.2 <u>Caracterização do Sinal de Vídeo</u> .....	31
2.2.1 Características.....	31
2.2.2 Fundamentos de Cor.....	36
2.2.3 Padrões.....	37
2.2.4 Degradações.....	41
2.3 <u>Vídeo Digital</u> .....	46
2.3.1 Amostragem e Quantização.....	46
2.3.2 Efeitos da Digitalização.....	48
2.4 <u>Exibição de Imagens</u> .....	51
2.5 <u>Conclusão</u> .....	55
3 PROCESSAMENTO DE IMAGENS.....	59
3.1 <u>Componentes de um Sistema IP</u> .....	62
3.2 <u>Sistema A17</u> .....	68
3.3 <u>Convolução Bidimensional</u> .....	71
3.4 <u>Conclusão</u> .....	76
4 ARQUITETURA DE CONVOLUÇÃO.....	79
4.1 <u>Concepção e Mapeamento de Arquiteturas</u> .....	81
4.2 <u>Materialização do Algoritmo</u> .....	82
4.2.1 Método de Choen & Johnson.....	83
4.3 <u>Frentes de Onda</u> .....	87
4.4 <u>Fluxo de Dados</u> .....	87
4.5 <u>Conclusão</u> .....	89
5 CV2D.....	91
5.1 <u>Descrição Geral</u> .....	91

5.2 <u>CV2D - Descrição Funcional</u> .....	94
5.3 <u>Descrição Operacional</u> .....	99
5.4 <u>Particionamento</u> .....	102
5.4.1 <u>Unidade de Convolução</u> .....	103
5.4.1.1. <u>Elementos de Processamento</u> .....	104
5.4.1.2 <u>Sincronização</u> .....	105
5.4.1.3 <u>Unidades de Ajuste de Fase</u> .....	106
5.4.2 <u>Unidade de Deslocamento</u> .....	106
5.4.3 <u>Unidade de Programação</u> .....	106
5.4.4 <u>Unidade de Sincronização</u> .....	108
5.5 <u>Detalhes</u> .....	108
6 <u>IMPLEMENTAÇÃO</u> .....	111
6.1 <u>Síntese do Fluxo de Dados</u> .....	112
6.1.1 <u>Cadeia de Processadores</u> .....	112
6.1.2 <u>Registradores de Deslocamento</u> .....	114
6.1.3 <u>Unidade de Deslocamento</u> .....	114
6.2 <u>Síntese das Máquinas de Controle</u> .....	114
6.3 <u>Posicionamento e Interconexão</u> .....	116
6.4 <u>Testes</u> .....	117
7 <u>CONCLUSÕES</u> .....	121
7.1 <u>Condições de Desenvolvimento do Projeto</u> .....	122
7.2 <u>Melhorias</u> .....	123
7.2.1 <u>Testabilidade</u> .....	124
7.3 <u>Resultados</u> .....	126
ANEXO 1 - <u>Diagramas de Estado</u> .....	131
ANEXO 2 - <u>Distribuição de PADs</u> .....	135
ANEXO 3 - <u>Informações Preliminares</u> .....	141
REFERÊNCIAS BIBLIOGRÁFICAS.....	145

## GLOSSÁRIO

- Ad hoc - Técnica de projeto que segue, fundamentalmente, a experiência do projetista.
- Aliasing - Erro originado pela utilização de uma taxa de amostragem inferior à necessária para o sinal. Caracteriza-se pela impossibilidade da reconstrução do sinal original.
- And - Operador lógico 'E'
- Arrays - Conjunto de processadores, semelhantes ou não, que trabalham em conjunto na execução de um algoritmo.
- Barrel shifter - Elemento combinacional que realiza um deslocamento de um determinado número de bits em uma palavra.
- Bit slice - Unidade funcional de um bit que pode ser multiplicada para compor uma unidade de vários bits.
- Blurring - Degradação caracterizada pela perda de definição das bordas da imagem. É causada pela atenuação das componentes de alta frequência.
- Broadcast - Tipo de transmissão de televisão. Caracteriza-se por usar o ar como meio de propagação, permitindo que qualquer aparelho receba a informação.
- Buffer - Componente de um sistema empregado como área temporária para o armazenamento de dados.
- Checker board - Efeito de tabuleiro de xadrez. É o mesmo que pixellation.
- Clock - Relógio. Base de tempo para os sistemas digitais.
- Cross Colour - Efeito de degradação de uma imagem colorida devido a superposição da crominância e da luminância no início da banda de frequência da crominância.

Cross Luminance	- Efeito de degradação de uma imagem colorida devido a superposição da crominância e da luminância no final da banda de frequência da luminância.
Data flow	- Modelo computacional onde o controle da máquina é orientado pelo fluxo dos dados.
Delay	- Tempo de propagação de um sinal elétrico, ou atraso imposto à transferência de um dado.
Dot pitch	- O mesmo que pitch.
Filter	- Operação de filtragem.
Flicker	- Efeito de piscamento da imagem devido a baixa frequência de varredura vertical.
Floor plan	- Planta baixa. Disposição dos blocos funcionais em um circuito integrado.
Frame Grabber	- Dispositivo capaz de digitalizar e armazenar um quadro de Imagem de TV.
Frame	- Quadro de uma imagem de TV. Seus limites são dados por uma varredura vertical completa.
Handshake	- Protocolo de comunicação entre duas máquinas.
Input	- Entrada.
Kernel	- Máscara ou núcleo de uma operação de convolução. Matriz de constantes que definem os coeficientes de uma operação bidimensional de máscara.
Keying	- Capacidade de adicionar planos de texto em planos gráficos.
Layout	- Desenho da disposição física dos elementos de um circuito integrado e de suas interconexões.
Layout custom	- Layout realizado pelo próprio projetista, desenhado sem o auxílio das ferramentas automáticas do sistema de CAD utilizado.

Lentgh ( )	- Indica o tamanho do argumento parentizado.
Line flicker	- Efeito de piscamento local, que afeta apenas uma linha.
Look-up-tables	- Tabelas que fornecem o valor de saída indexado pelo seu valor de entrada.
Net list	- Lista dos componentes de um sistema e de suas interconexões.
Not	- Operador lógico de negação.
Or	- Operador lógico 'OU'
Output	- Saída.
Overflow	- Estouro de representação numérica.
Pan	- Deslocamento da janela de exibição de uma imagem na direção horizontal, dentro de seu universo.
Phosphor spacing	- Espaçamento entre os elementos de fósforo de um tubo de Imagem.
Picture Elements	- Pixels. Menor elemento de imagem. Seu conjunto forma um quadro de imagem digital.
Pitch	- Espaçamento dos pontos de fósforo de uma mesma cor em um tubo de imagem.
Pixellation	- Efeito de tabuleiro de xadrez. A imagem é vista como um conjunto de quadrados que definem seus objetos.
Pixel size	- Tamanho do pixel. Dimensão do elemento de imagem, que, quando colorida, é composto pelos três pontos de cor.
Placement	- Posicionamento dos blocos funcionais em um circuito Integrado.
Reset	- Sinal de inicialização de um equipamento ou bloco funcional.

- Retimming - Técnica que consiste em modificar a temporização do sistema a fim de melhorar seu desempenho.
- Routing - Desenho das linhas de interconexão dos elementos de um sistema.
- Row processor - Processador de linha. Nome dado ao conjunto de elementos de processamento que tratam os pixels de uma determinada linha da Imagem.
- Salt-and-peeper - Ruído caracterizado por pontos pretos e brancos na tela, causados por interferência elétrica no sinal de vídeo.
- Scanning - Processo de conversão de uma cena em um sinal elétrico, através de um feixe de elétrons que se desloca, ao mesmo tempo, nas direções horizontal e vertical.
- Scan Moiré - Efeito de batimento em áreas da imagem que são iluminadas uniformemente. Caracteriza-se por linhas horizontais nas áreas de mesma luminosidade.
- Scrool - Deslocamento da janela de exibição de uma imagem na direção vertical, dentro de seu universo.
- Sharpening - Processo de enriquecimento de imagem que reforça as altas frequências. Seu objetivo é realçar o contorno dos objetos de uma imagem.
- Skew - Atraso colocado a fim de ajustar a temporização dos dados.
- Slices - Fatias. Conjunto funcional que pode ser replicado para formar unidades maiores.
- Sliced floor plan - Planta baixa de um circuito integrado que caracteriza-se por todos os blocos possuírem forma tetragonal ( quatro lados ).

Smoothing	- Processo de enriquecimento de imagem que atenua as altas frequências. Seu objetivo é a redução de ruído.
Speckly, efeito	- Degradação em forma de manchas, que aparece em imagens geradas por fontes de luz coerentes.
Spot size	- Tamanho da malha ou grade que existe nos tubos de imagem. A função desta malha é de colimar o feixe de elétrons.
Standard cells	- Técnica de implementação semi-dedicada de circuitos integrados. Caracteriza-se por possuir blocos funcionais que possuem mesma altura, possibilitando sua justaposição.
Sync	- Pulso de sincronismo existente no sinal de vídeo fornecido por uma câmera de TV.
Throughput	- Taxa de saída de dados de um processador.
Twin tub	- Tecnologia de fabricação de circuitos integrados que utiliza poço 'P' e 'N'.
Vídeo Moiré	- Efeito que causa o aparecimento de faixas verticais estreitas, com distância múltipla do espaçamento das trilhas de fósforo.
Wavefronts	- Frentes de onda. Representam a movimentação dos dados em relação ao tempo.



## LISTA DE ABREVIATURAS

- ALU - Arithmetic Logic Unit ( Unidade Lógica e Aritmética )
- ASIC - Application Specific Integrated Circuit ( Circuito Integrado Dedicado )
- BB - Bounding Box
- CAD - Computer Aided Design ( Projeto Auxiliado por Computador)
- CAPES - Coordenadoria de Aperfeiçoamento de Pessoal e Ensino Superior
- CC - Chip Compiler
- CCD - Charged Coupled Device ( Dispositivo de Acoplamento de Carga )
- CI - Circuito Integrado
- CNPq - Conselho Nacional de Pesquisa
- CPGCC - Curso de Pós Graduação em Ciência da Computação
- CRT - Cathode Rays Tube ( Tubo de Raios Catódicos )
- DPC - Data Path Compiler
- DSP - Digital Signal Processing ( Processamento Digital de Sinais )
- $f_l$  - Frequência de linha; frequência de varredura horizontal
- $f_s$  - Frequência de amostragem
- IP - Image Processing ( Processamento de Imagens )
- LFSR - Linear Feedback Shift Register
- LS - Logic Synthesizer
- MIPS - Milhões de operações por segundo
- NTSC - National Television Standards Committee
- PE - Processing Element ( Elemento de Processamento )
- RMS - Root Mean Square ( Valor médio quadrático )
- SAW - Surface Acoustic Waves ( Ondas Acústicas de Superfície )
- SM - State Machine Compiler
- UFRGS - Universidade Federal do Rio Grande do Sul

## LISTA DE FIGURAS

Fig. 1.1	Elementos de um Sistemas de Processamento Digital de Imagens.....	24
Fig. 2.1	Modelo de Formação da Imagem.....	31
Fig. 2.2	Sinais de Sincronismo .....	35
Fig. 2.3	Formato RS170.....	39
Fig. 2.4	Níveis de quantização e Aplicações.....	48
Fig. 2.5	Razão de Visualização .....	49
Fig. 2.6	<i>Dot Pitch</i> Pv e Ph.....	52
Fig. 2.7	Índice de Modulação (m) e Razão de Contraste (c).....	53
Fig. 2.8	Relação entre <i>SpotSize- DotPitch- PixelSize</i> .....	55
Fig. 3.1	Classificação hierárquica dos algoritmos IP.....	60
Fig. 3.2	Arquitetura com Barramento Interno.....	63
Fig. 3.3	Digitalizador de Vídeo PIP.....	64
Fig. 3.4	Configurações de Processadores Especializados de Vídeo.....	65
Fig. 3.5	Arquitetura do Sistema Série 150.....	67
Fig. 3.6	Modelo de Processo no Sistema A17.....	69
Fig. 4.1	Representação Gráfica do Elemento de Delay.....	84
Fig. 4.2	Implementação das Redes de Convolução.....	86
Fig. 4.3	Implementação da Convolução Bidimensional.....	89
Fig. 5.1	Modelo do Ambiente de Funcionamento do CV2D.....	94
Fig. 5.2	Diagrama em Blocos.....	95
Fig. 5.3	Forma de Onda de Sincronização .....	97
Fig. 5.4	Formas de Onda de Programação .....	99
Fig. 5.5	Parâmetros de Programação .....	100
Fig. 5.6	Formato dos Parâmetros de Programação.....	101
Fig. 5.7	Efeito do Índice de Deslocamento.....	102
Fig. 5.8	Unidade de Convolução.....	104
Fig. 5.9	Arquitetura do Elemento de Processamento.....	105
Fig. 5.10	Unidade de Programação .....	107
Fig. 5.11	Unidade de Sincronização.....	108
Fig. 5.12	Caminho Crítico.....	110
Fig. 6.1	Layout gerado pelo DPC.....	113
Fig. 6.2	Arquitetura Alvo das Máquinas de Estado.....	115
Fig. 6.3	CV2D - Planta Baixa.....	117

Fig. 7.1	Estruturas de Teste.....	126
Fig. 7.2	Arquitetura PGCC.....	129

## RESUMO

Este trabalho descreve a concepção de um circuito integrado dedicado (ASIC), destinado a realizar a operação de convolução sobre imagens em tempo real. São analisadas as características do sinal de vídeo e algumas propostas de sistemas de processamento de imagens (IP). É caracterizado um ambiente de IP assim como o projeto de um operador de convolução chamado CV2D.

O CV2D calcula, à taxa de pixel, a convolução bidimensional sobre imagens quantizadas em 8 bits/pixel e até 512 pixels/linha. O circuito foi projetado para ser uma interface em sistemas de processamento de imagens, podendo realizar ou filtragem linear ou convolução sobre uma sequência de pixels. O processo de convolução necessita de duas linhas de varredura anteriores e o armazenamento destes dados é realizado através de um sistema de *Line Delay*, atualmente implementado fora do componente.

## ABSTRACT

This work describes the design of an Application Specific Integrated Circuit (ASIC) dedicated to perform the real time convolution over monochromatic images. The video signal characteristics and some alternatives for image processing (IP) systems are considered. An IP environment is characterized and the design of a convolution operator called CV2D is presented.

The CV2D computes, at pixel rate, the bidimensional convolution over 8 bits/pixel quantized images up to 512 pixels/line. The chip was designed to be an interface for image processing systems performing linear filtering or convolution processing of a pixel stream. The convolution process needs two delayed lines and the storage of these lines is performed by means of a line delay circuit, actually implemented out of the chip.

# 1 INTRODUÇÃO

" Pois a dúvida agrada-me não  
menos que o saber. "  
Dante (1265-1321). Inferno.

Dentro do vasto domínio de aplicações na área de tratamento de sinais, ênfase especial é dada ao ramo de processamento digital de sinais (DSP). Esta área foi fortemente impulsionada pelo aumento dos recursos computacionais disponíveis e pelo avanço e integração das técnicas digitais. A teoria matemática, ferramenta que sustenta o desenvolvimento desta matéria, encontra-se na bibliografia relacionada ao assunto, onde pode-se citar [OPP 75], [RAB 75], [OPP 78], [EDE 89] e outros.

As aplicações em DSP compreendem a realização de filtros digitais, sistemas de controle lineares e não lineares, sistemas de comunicação, radares convencionais e de abertura sintética e processamento de imagens entre outras. Uma visão deste campo é dada em [OPP 78] e [SAL 77]. As aplicações em filtragem digital vêm de longo tempo e hoje utilizam tanto técnicas puramente digitais como capacitores chaveados e SAW (Surface Acoustic Waves) para sua implementação. Referências são encontradas em [JAC 68], [HAM 77], [PEL 76] e [GRE 87].

Uma das áreas que tem merecido atenção dos pesquisadores é a classe de processamento digital de imagens, que objetiva desde a melhoria da qualidade de imagem para visualização até a implementação de máquinas de visão artificial, capazes de reconhecer e manipular objetos. Aplicações em robótica, transmissão de imagens e biomédica são frequentemente objeto de publicação.

## 1.1 Conceitos Básicos

A expressão "imagem monocromática", ou simplesmente "imagem", refere-se à função de intensidade bidimensional  $f(x,y)$ , onde  $x$  e

y são as coordenadas espaciais e o valor da função é proporcional ao brilho (ou tom de cinza) da imagem neste ponto. "Imagem digital" é uma função imagem que foi discretizada nas duas coordenadas espaciais e no brilho. Pode-se considerar uma imagem digital como uma matriz na qual os índices de linha e coluna identificam espacialmente um ponto da imagem na matriz e o valor deste corresponde ao brilho. Os componentes desta matriz são chamados de elementos de imagem ou pixels ( *picture elements* ). A dimensão da imagem está relacionada com a aplicação a que se destina e é limitada pela resolução do sensor.

Os componentes de um sistema simples de processamento de imagens pode ser visto na figura 1.1.

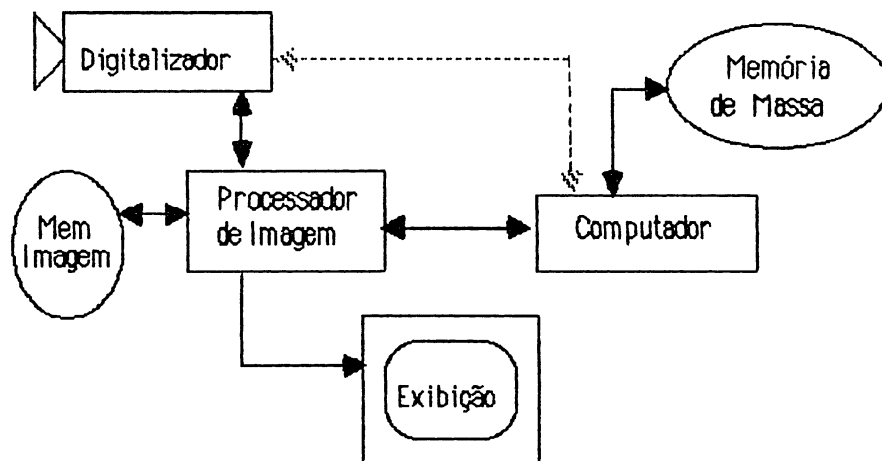


Figura 1.1 Elementos de um Sistemas de Processamento Digital de Imagens

A partir da figura identifica-se que um processador de imagens é formado por um conjunto de módulos que realizam quatro funções básicas: Aquisição, Armazenamento, Processamento básico do sinal e exibição [GON 87].

Tipicamente um módulo de aquisição possui um sensor e um digitalizador, que converte o sinal gerado em uma representação digital. Quando este módulo é capaz de digitalizar uma imagem de TV no tempo de um quadro, recebe o nome de "*Frame Grabber*". Digitalizador é o dispositivo que converte uma imagem na sua representação numérica, manipulável pelos computadores digitais. Embora a bibliografia cite vários tipos de

tipos de sensores como "*flying spot scanners*", "*image dissectors*", "*microdensitometers*" entre outros [PRA 78], [GON 87], [EKS 84], este trabalho considera apenas os sensores do tipo câmera de TV. Esta opção deve-se à facilidade de obtenção deste sensor, capacidade de exibição imediata da imagem captada em um monitor de vídeo bem como a aquisição de imagens móveis.

O módulo de armazenamento, chamado também de "*Frame Buffer*", é uma memória capaz de armazenar uma imagem completa. Uma variação na forma de endereçamento desta memória permite realizar zoom (aproximação) instantâneo na imagem, bem como "*scroll*" (deslocamento vertical) e "*pan*" (deslocamento horizontal), da mesma forma como é feito nos atuais controladores de vídeo com capacidade gráfica.

O módulo de processamento realiza operações aritméticas e lógicas. Estes processadores variam desde microcomputadores [MAT 87] até componentes especializados, ASICs, que realizam a computação dos dados [RUE 87], [RUE 89]. Este tipo de processador especializado é projetado especialmente para ser rápido e, muitas vezes, realiza o processamento dos pixels em paralelo.

A função do módulo de exibição é realizar a conversão da imagem de sua forma numérica em um sinal de vídeo analógico e exibí-lo em um monitor de TV ou outro dispositivo que traduza a informação em uma forma perceptível ao olho humano. Neste texto serão considerados apenas os vídeos, em especial os CRTs (*Cathode Rays Tubes*) que são os mais comuns e permitem a visualização de imagens móveis. Sistemas em hardware podem ser colocados a fim de realizar transformações na escala de tons, inserção de gráficos e planos alfanuméricos e tratamento de pseudo-cores.

Os demais elementos que aparecem na figura 1.1 são um computador e um dispositivo de memória de grande capacidade. O computador pode ser utilizado para controle ou processamento de mais alto nível e sua memória de massa para armazenamento das imagens ou dos resultados do processamento realizado.



## 1.2 Escopo e Organização do Texto

O presente trabalho descreve a concepção e implementação de um processador integrado e especializado, que realiza o processo de convolução bidimensional. Este componente foi projetado para ser utilizado em tarefas de processamento de imagens monocromáticas em tempo real, onde tempo real significa que o cálculo dos resultados é realizado na mesma taxa da geração das imagens do sensor. Este componente implementa uma das funções do processador de imagens da figura 1.1, e pode ser utilizado em sistemas de identificação e visualização. Com este processador de convolução é possível implementar filtros de eliminação de ruído, reforço de contorno, cálculo de gradientes e outras operações comuns em processamento de imagens.

O processo de concepção de um sistema compreende as seguintes fases:

- \* Compreensão e limitação da função a ser realizada pelo sistema; (Especificação)
- \* Detalhamento do sistema determinando os pontos críticos, em velocidade e complexidade; (Refinamento)
- \* Avaliação das diversas formas de realização e a viabilidade da aplicação a que se destina;
- \* Projeto e implementação utilizando ferramentas de CAD que viabilizem a avaliação e realização das etapas acima.
- \* Fabricação e teste do sistema.

Os capítulos a seguir abordam cada um destes tópicos, com maior ênfase na implementação do circuito.

No capítulo 2, SINAL de VÍDEO, é feita a caracterização do sinal a ser processado e introduz parâmetros básicos de vídeo digital: bits/pixel e pixels/linha. São citadas algumas degradações provenientes dos sensores e do processo de discretização. Finalmente são descritos alguns parâmetros qualitativos dos sistemas de exibição baseados em CRTs.

O tópicos de PROCESSAMENTO de IMAGENS, capítulo 3, apresenta uma classificação dos algoritmos utilizados em IP (*Image Processing*). Faz-se a apresentação do algoritmo de convolução bidimensional, extensamente utilizado em processamento de imagens, e discute-se as condições de contorno da equação. Neste mesmo capítulo são caracterizados os componentes básicos e o modelo de sistema de processamento de imagens a ser utilizado.

O capítulo 4, ARQUITETURAS DE CONVOLUÇÃO, dedica-se a discutir os conceitos arquiteturais utilizados para definição da arquitetura alvo utilizada.

No capítulo 5, CV2D, é apresentada a arquitetura implementada e a descrição dos blocos funcionais que compõem o circuito.

No título IMPLEMENTAÇÃO, capítulo 6, descreve-se brevemente as ferramentas de CAD utilizadas para o projeto do CV2D e o resultado da aplicação destas na concepção do circuito.

Finalmente no capítulo 7 são apresentados os resultados e CONCLUSÕES. São dadas também sugestões de melhorias para o circuito, avaliação da testabilidade e um sumário das características do circuito.

## 2 SINAL DE VÍDEO

Este capítulo apresenta o modelo de imagem que será utilizado ao longo do texto. É caracterizado o sinal a ser processado, sua geração, limitações e eventuais degradações. São abordados os aspectos de digitalização do sinal de vídeo, concluindo com a escolha do número de pixels/linha e do número de bits/pixel utilizados no sistema. É discutida também a representação de imagens monocromáticas a partir de sinais de cor. Tal relação pode ser utilizada no tratamento de pseudo-cores [HAL 79]. Ainda neste capítulo são tecidas considerações sobre a exibição de imagens em CRTs a fim de completar a cadeia do sinal: Formação e Geração, Aquisição, Tratamento e Exibição.

### 2.1 Modelo de Imagem

O termo imagem refere-se à função bidimensional de intensidade luminosa, descrita por  $f(x,y)$ , onde o valor de  $f$  nas coordenadas  $(x,y)$  fornece a intensidade ou brilho deste ponto de imagem. As imagens que percebemos consistem da luz refletida dos objetos de uma cena e cada um dos pontos da imagem contém a contribuição de luminosidade de todos os outros pontos vizinhos. Sendo a luz uma forma de energia, os valores de  $f$  estão nos limites:

$$0 < f_{xy} < \infty$$

A função  $f(x,y)$  pode ser caracterizada por duas componentes [PRA 78], [OPP 78], [GON 87]. A primeira é a quantidade de luz incidente sobre a cena observada e a segunda traduz a quantidade de luz refletida pelos elementos da cena. Estas componentes são chamadas de iluminação e refletância, representadas pelas funções  $i(x,y)$  e  $r(x,y)$  respectivamente. O produto destas funções define a imagem  $f(x,y)$ .

$$f_{xy} = r_{xy} * i_{xy}$$

Neste contexto a refletância é uma característica dos objetos da cena e tem seus limites entre a absorção total (zero) e a reflexão total (um) da luz incidente. A intensidade de uma imagem monocromática, chamada de tons ou níveis de cinza, normalmente é colocada no intervalo  $[0,L]$ . Estes limites representam o preto (0) e o brilho máximo possível para um elemento de imagem (L).

As imagens reais possuem propriedades de vizinhança, significando que um ponto  $(x,y)$  no plano de imagem [OPP 78] não corresponde apenas ao ponto  $(x',y')$  no plano do objeto, mas também é função da energia radiante dos infinitos vizinhos deste ponto. A imagem  $g(x,y)$  corresponde à distribuição de energia de um objeto e o somatório das contribuições infinitesimais de seus vizinhos. Assumindo linearidade nos processos pelos quais a energia radiante se acumula no plano de imagem, a formação de um quadro pode ser descrita, após alguma manipulação algébrica, pela equação abaixo. Esta representa um sistema linear de duas dimensões e invariante no deslocamento.

$$g_{xy} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_{x-x_1, y-y_1} f_{x_1 y_1} dx_1 dy_1$$

É claro que a influência de um determinado ponto diminui nos pontos mais afastados deste. Isto permite restringir a influência de um

determinado elemento em seus vizinhos sem perda significativa de informação para um posterior processamento.

Os processos de formação da imagem e seu sensoriamento podem ser idealizados no blocodiagrama da figura 2.1. Neste modelo foi incluída uma fonte de ruído aditivo e uma função de transferência que modela outras degradações possíveis no sinal de vídeo gerado.

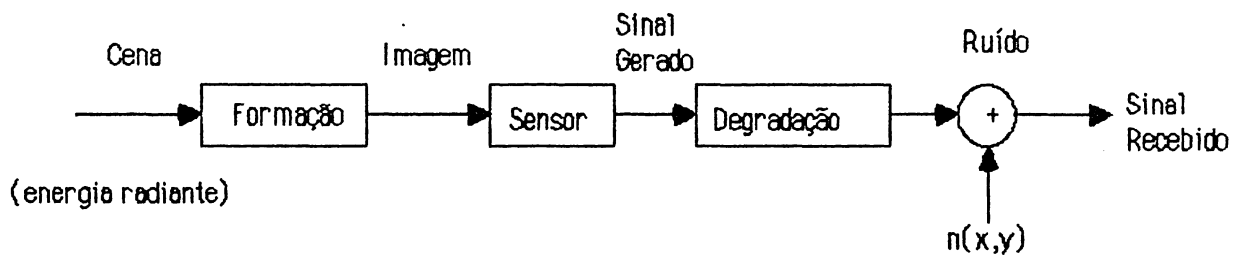


Figura 2.1 Modelo de Formação da Imagem

## 2.2 Caracterização do Sinal de Vídeo.

Para uma boa compreensão dos métodos utilizados em processamento digital de imagens e suas limitações é necessário conhecer o sinal a ser processado. A seguir são dadas as principais características do sinal fornecido pelas câmeras de vídeo.

### 2.2.1 Características

O objetivo de uma câmera de vídeo é converter uma cena que está em frente ao sensor em um sinal elétrico. Este sinal possui apenas um valor a cada instante de tempo, ou seja, uma função de uma variável ou um sinal unidimensional. No entanto a imagem é uma função de duas dimensões, como descrito anteriormente, que ao mesmo tempo possui valores diferentes para cada posição do quadro.

Um método de converter a imagem de duas dimensões em um sinal elétrico é chamado de varredura ou *scanning*. O processo de varredura consiste em um ponto sensor que desloca-se rapidamente sobre a imagem, com velocidade suficiente para realizar a captura completa da imagem antes desta modificar-se muito. Durante o movimento do ponto sensor, a saída elétrica varia em resposta ao brilho ou a cor da imagem sob o ponto de sensoriamento. O sinal elétrico proveniente deste sensor apresenta a imagem como uma série de valores separados no tempo. Este sinal será chamado de sinal de vídeo.

Normalmente a cena é explorada pela combinação de dois movimentos ortogonais do ponto sensor. Assim a varredura inicia no canto superior esquerdo e prossegue horizontalmente através da imagem, formando a chamada linha de varredura. Ao mesmo tempo o ponto sensor é movido verticalmente para baixo a uma velocidade muito menor do que o seu movimento horizontal. Quando o lado direito da imagem é atingido, o ponto de varredura retorna ao lado esquerdo. Devido a este movimento vertical o ponto estará, após retornar ao lado esquerdo, um pouco abaixo do lugar de partida da primeira linha. O processo de varredura prossegue através da próxima linha e sucessivamente até ser percorrida a última linha. Durante a linha percorrida, a saída do sensor representa a intensidade luminosa de cada ponto. No período de retorno do ponto ( conhecido como tempo de apagamento ou retraço ) o valor é colocado em zero (ou nível de apagamento). Este apagamento vertical e horizontal é originalmente dedicado a fornecer o tempo necessário para o retraço do feixe das câmeras e CRTs, porém, devido ao fato de não conterem informações de imagem, aproveita-se este tempo, no caso de transmissão de TV, para enviar outras informações, como videotexto, dados e sinais de correção. O sinal de uma varredura completa é uma sequência de linhas separadas por intervalos de apagamento. Este conjunto de linhas de varredura é chamado quadro ou *frame*. Um importante parâmetro na varredura é a relação de aspecto. Esta é a razão do comprimento de uma linha de varredura horizontal pela distância coberta verticalmente no quadro. A relação de aspecto pode ser vista como a relação largura vs altura de um frame. Em televisão a relação de aspecto padrão é 4:3 e em outros sistemas, como em cinema, pode-se chegar a razões de 1:1 e, modernamente, até 2:1.

Se o sinal obtido for utilizado para modular a intensidade de um feixe de elétrons em um CRT, com o mesmo padrão de varredura do sensor, a imagem original será reproduzida. Desta forma o sinal elétrico enviado ao monitor deve conter alguma informação adicional para garantir que a varredura do monitor mantenha sincronismo com o sensor. Esta informação é chamada de pulsos de sincronismo (SYNC), incluída dentro do próprio sinal de vídeo durante o tempo de apagamento ou enviada separadamente.

Resolução é a medida da fidelidade de reprodução dos detalhes de um quadro real através do sinal de vídeo gerado. A capacidade de variação do sinal está vinculada a dois fatores:

- Intensidade luminosa necessária para sensibilizar o dispositivo e
- área sensível do ponto explorador.

A diferença entre o valor máximo e mínimo de luminosidade capaz de estimular o sensor é chamada de faixa dinâmica. O tamanho do ponto sensor determina a resolução horizontal, onde um ponto menor dará uma maior resolução na imagem capturada, que reflete-se na capacidade de reproduzir detalhes finos. Uma das formas de testar a resolução horizontal de um sistema é colocar linhas verticais pouco espaçadas em frente a câmera. Se a área do ponto sensor for menor que o espaço entre as linhas, estas serão reproduzidas, caso contrário a saída produzida será um sinal correspondente a intensidade média entre o brilho das linhas e do intervalo entre as mesmas. Em sistemas de televisão [GRO 79] a resolução é medida pela contagem de linhas verticais pretas e brancas consecutivas capazes de serem reproduzidas. Resolução de 400 a 500 linhas são valores comuns em câmeras industriais vidicon, chegando a valores bem maiores em equipamentos especializados. Em dispositivos que utilizam um arranjo de pontos sensores, chamados matriciais, a resolução horizontal depende do tamanho do ponto e do número de sensores em cada linha da matriz.

É importante notar que nas transições abruptas de luminosidade o sensor sempre produz sinais elétricos de alta frequência, sendo importante que os circuitos utilizados para processamento e transmissão

destes sinais possuam a largura de banda adequada para suportar tais frequências. Em [LUT 89] encontramos, por exemplo, que para transmissão de televisão é necessária uma banda de 1Mhz para cada 80 linhas de resolução horizontal. Este parâmetro é importante na utilização de um sensor que possua no mínimo uma capacidade de resolução e largura de banda igual à desejada na computação digital, evitando processos de estimação de pixels em tempo de amostragem.

A resolução vertical depende do número de linhas de varredura utilizadas em um quadro, ou do número de pontos verticais no caso de um sensor matricial. Quanto maior o número de linhas, maior a resolução vertical. Por exemplo os sistemas de TV utilizam valores de 525 linhas (americano e japonês) e 625 linhas (europeu). Destas um pequeno número (em torno de 40 linhas) são utilizadas para apagamento vertical. Novos sistemas de TV de alta resolução amplia estes valores para 1000 a 1200 linhas, com relação de aspecto de 5:3 ou mesmo 16:9 [LEW 88]

A taxa de quadros exibidas no tempo deve ser suficiente para produzir um efeito de continuidade em imagens móveis. Em sistemas de transmissão padrão, taxas de 25 ou 30 quadros/s são utilizadas. Estes números, embora suficientes para representar imagens móveis, não são altos o suficiente para evitar *flicker* (efeito de piscamento da imagem). Desta forma procura-se atender taxas superiores a 50 quadros/s, que, apesar de reduzir o *flicker* apresenta resultados satisfatórios. Com esta taxa de atualização, preservando a mesma resolução horizontal é necessário um aumento nas frequências de varredura horizontal e vertical, que provoca um significativo aumento na largura de banda necessária para o sistema. A teledifusão comercial evita este problema através do uso de técnicas de entrelaçamento.

Entrelaçamento, em televisão, significa que mais do que uma varredura vertical produz um quadro. Sistemas de televisão usam entrelaçamento 2:1. Assim uma varredura vertical exhibe as linhas ímpares e outra exhibe as linhas pares, chamados campos pares e ímpares do quadro exibido. Com uma taxa de 30 campos por segundo a taxa de varredura vertical equivale a 60 quadros/s. Baseado no fato que o olho humano não percebe piscamento em pequenos objetos nesta frequência, a taxa de 30 campos/s é suficiente e a imagem total é aparentemente regenerada a uma



taxa de 60 quadros/s (*frames/s*). Desta forma, cada campo par inicia imediatamente após um retraço vertical e os campos ímpares 1/2 linha após. Durante o tempo de apagamento vertical existem os sinais de sincronização horizontal e pulsos de equalização que mantêm a relação de frequências do dispositivo de exibição e o entrelaçamento correto. A figura 2.2 mostra o formato completo do sinal, indicando o início dos campos pares e ímpares e os pulsos de equalização.

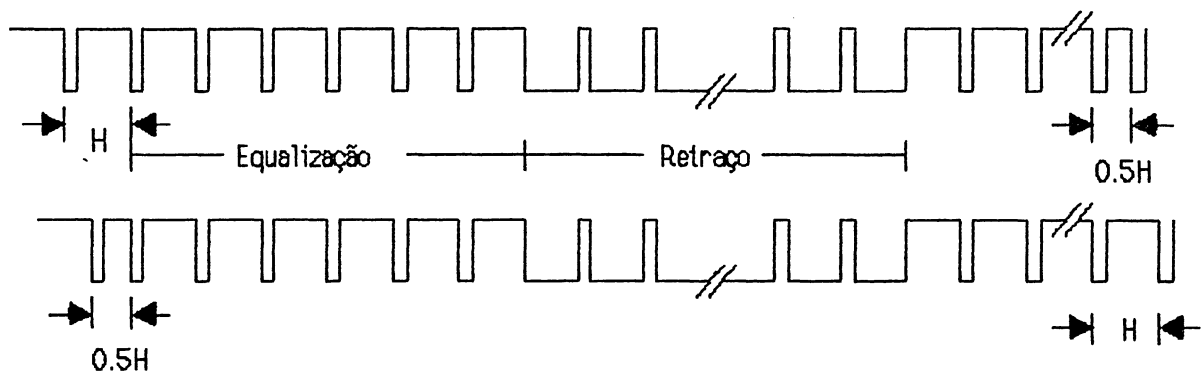


Figura 2.2 Sinais de Sincronismo

Nos sensores do tipo câmera, a imagem é opticamente focalizada em uma superfície bidimensional. Esta superfície fotosensível recolhe a luz proveniente de todos os pontos da imagem, de maneira contínua, gerando uma carga elétrica correspondente a cada ponto da imagem focalizada nesta superfície. Esta carga é convertida em um valor de tensão que compõe o sinal de vídeo.

Existem duas grandes classes destes dispositivos, que diferem na forma de armazenamento e transferência de carga. Tubos a vácuo (vidicon, staticom, plumbicom) [GRO 79] coletam a carga em uma superfície especial depositada no fundo do tubo. Um feixe de elétrons faz a varredura e gera o sinal de vídeo correspondente a carga armazenada. O segundo tipo de dispositivo é o de estado sólido, onde a carga é gerada em uma área de silício que é superposta a uma matriz de componentes que transferem ordenadamente esta, cadenciada pelos sinais de sincronismo. Estes dispositivos são conhecidos como CCD (Charged Coupled Devices).

Com o mesmo princípio alguns sensores utilizam as propriedades de armazenamento de carga utilizadas nas RAMs dinâmicas ("Optics DRAMs") [AWC 87a], [AWC 87b], [AWC 87c].

## 2.2.2 Fundamentos de Cor

Devido à estrutura do olho humano, todas as cores podem ser vistas como uma combinação variável das três chamadas cores primárias: Vermelho (R), Verde (G) e Azul (B). Estas cores primárias podem ser somadas para produzir as secundárias: Magenta (R+B), Cian (G+B) e Amarelo (R+G). Misturando as três primárias ou cada uma das secundárias com a cor primária oposta, na intensidade correta, a luz branca é produzida.

Existe diferença entre cores primárias de luz e cores primárias de pigmentos ou corantes. Nos corantes, a cor primária é definida como absorção de uma das cores primárias e reflexão das outras duas [GON 87], [GRO 79]. O sistema de cores em televisão é um exemplo de coloração de natureza aditiva e portanto composta com as cores primárias RGB. As características utilizadas para distinguir uma cor de outra são o BRILHO, MATIZ e SATURAÇÃO.

Brilho refere-se à intensidade e o Matiz é um atributo associado ao comprimento de onda dominante nas ondas luminosas ( a percepção de uma cor não se restringe a um comprimento de onda fixo e sim a uma faixa do espectro visível). Assim o matiz representa a cor dominante recebida pelo observador. Quando se chama um objeto de azul, laranja ou amarelo, está sendo estabelecido seu matiz. A saturação refere-se a pureza relativa, ou quantidade de luz branca misturada ao matiz. O espectro puro é definido como completamente saturado e o grau de saturação é inversamente proporcional à quantidade de luz branca adicionada. O conjunto de matiz e saturação é chamada de cromaticidade e, portanto, uma cor pode ser caracterizada pelo seu brilho e cromaticidade. A quantidade de R, G, B necessária para formar uma determinada cor é chamada de valores "tristimulus" e denotados por X,Y e Z respectivamente. Uma cor é então especificada por seus coeficientes tricromáticos, definidos como :

$$\begin{aligned}
 R &= \frac{R}{R + G + B} & G &= \frac{G}{R + G + B} \\
 B &= \frac{B}{R + G + B} & R + G + B &= 1
 \end{aligned}$$

Na criação da tecnologia de vídeo colorido, procurou-se manter compatibilidade com os receptores de TV monocromáticos e a mesma largura de banda para transmissão. Para atender tal requisito, o sinal de vídeo colorido é codificado em dois sinais distintos [GRO 79], [EDE 89]:

- a) O sinal Y, luminância, que contém o sincronismo e a informação de brilho, compatível com os sistemas preto e branco.
- b) O sinal de croma C, que carrega a informação de cor (matiz e saturação).

Para não haver redundância, a croma é composta apenas pelos sinais "diferença de cor" (R-Y) e (B-Y), modulados em fase e amplitude sobre uma subportadora de 3.58Mhz. Fica função do sistema de exibição ou recepção recuperar o sinal G (cor verde). De acordo com a sensação de brilho de cada cor ao olho humano, foi estabelecida a relação algébrica entre o sinal Y e os sinais de cor:

$$Y = 0.3R + 0.59G + 0.11B$$

### 2.2.3 Padrões

Vários comitês desenvolveram padrões para câmeras de vídeo. Estes, primeiramente desenvolvidos para aplicações em televisão, incluem o RS170, CCIR625 e RS343. Estes padrões satisfazem uma tecnologia

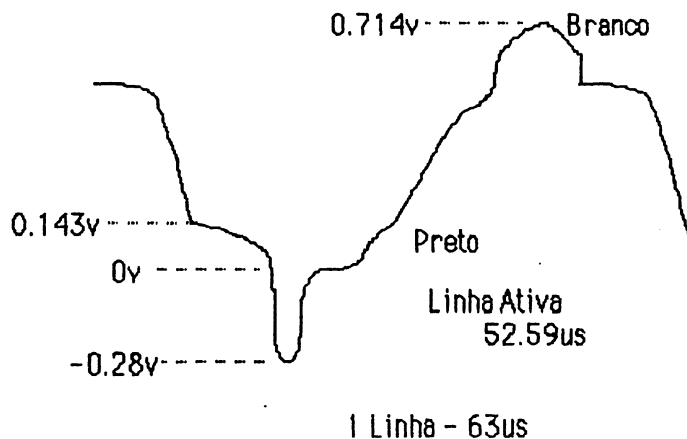
antiga, baseada em sinais monocromáticos. Assim, relacionado a esta origem, estes possuem algumas limitações, como:

- . Largura de banda apropriada para difusão, o que limita a resolução;
- . O formato entrelaçado representa um obstáculo para aquisição de imagens em alta velocidade;
- . Não existe "reset" vertical assíncrono. Desta forma uma imagem não pode ser gatilhada por eventos externos.

Para sistemas que necessitam alto desempenho não existem padrões definidos. Existem sistemas de aquisição de imagens que atendem os requisitos de alta definição e interface digital não entrelaçada.

Os padrões são conceitualmente semelhantes entre si na sua implementação, produzindo imagens entrelaçadas em 2:1 com resolução variada. Será apresentada a seguir uma visão superficial dos padrões mais utilizados. Mesmo as câmeras CCD de alta resolução fornecem sinais compatíveis aos formatos descritos, variando apenas as frequências, para permitir o não entrelaçamento.

O RS170 define a temporização e a especificação de voltagem para sinais de vídeo preto e branco. Posteriormente a codificação de sinais de diferença de cor com luminância e sinais de sincronismo conduziu a um padrão de televisão colorida, o NTSC ("National Television Standards Commitees"). O quadro RS170 é composto de 525 linhas de varredura horizontal, separadas em dois campos, par e ímpar. Durante o campo par, 262.5 linhas são fornecidas em 1/60s. As 262.5 linhas ímpares são fornecidas nos próximos 16.7ms. A cada 33.3ms, ou 1/30s, um quadro completo é atualizado. Como um tempo de 9 linhas é necessário para o intervalo de sincronismo vertical e 11 linhas são utilizadas no apagamento, a janela de vídeo fica reduzida a 242.5 linhas ativas em cada campo. Logo a resolução vertical efetiva é de apenas 485 linhas. A figura 2.3 ilustra o formato típico RS170 para um determinado campo.



262.5 linhas

9 linhas de sincronismo vertical

11 linhas de apagamento

total: 242.5 linhas ativas/campo

resolução vertical efetiva: 485 linhas/quadro

Figura 2.3 Formato RS170

A resolução horizontal efetiva é limitada pela largura de banda do amplificador de vídeo da câmera. Muitas câmeras RS170 possuem resolução de 300a 400 pixels/linha com sinais de temporização horizontal e vertical inseridos nos dados analógicos de vídeo. Conforme a figura, o envelope de vídeo possui limites de -0.28V a 0.714V, ou seja, em torno de 1Vpp. Os dados de vídeo iniciam em um pedestal de 0.143V, que corresponde ao valor de um ponto preto na imagem e um ponto completamente branco possui um valor de 0.714V. Os sinais no intervalo entre 0.143 e 0.0V são utilizados para indicar apagamento horizontal e vertical. Os sinais negativos correspondem aos pulsos de sincronização.

O CCIR625, 625 linhas, é similar ao padrão RS170, porém com diferenças na taxa de quadro e largura de banda de vídeo. Por possuir mais linhas ativas, uma resolução maior é alcançada. Em sistemas de imagens isto significa que um quadro de 512x512 pode ser capturado, uma vez que

possui 575 linhas ativas, em oposição as 480 linhas do RS170. No entanto a frequência de quadro é de 50 Hz, o que pode fazer diferença na captura de imagens rápidas. Além disso a natureza dos quadros pode causar um batimento de frequências espaciais, introduzido quando os objetos em foco estão sob iluminação fluorescente alimentadas pela rede de 60 Hz. Em conjunto com o desenvolvimento deste padrão foi criado um sistema de codificação de cor: o sistema de alternância da fase da linha (Phase Alternation Line system PAL). Neste, a cor é codificada pela fase do sinal de 3.58MHz em relação a uma referência. A fase de uma das componentes de cor é alternada linha a linha, tornando média qualquer distorção de cor. Este tipo de codificação reduz os erros de fase ao longo do processo de transmissão, dispensando o ajuste de cor existente nos receptores NTSC.

Em busca de um padrão com melhor resolução que o RS170, mantendo o fator de entrelaçamento 2:1, a EIA desenvolveu o RS343. Devido a melhor resolução e uma taxa de quadro mais alta, os sistemas que empregam RS343 utilizam principalmente câmeras VIDICON. Para este padrão existem diferentes versões, de 675 a 1023 linhas/frame e com um número de linhas ativas entre 624 e 943 respectivamente, todos com entrelaçamento 2:1. O padrão RS343 mais popular é o de 875 linhas. Não existe padronização de cor desenvolvido para este sistema.

Câmeras com sensor CCD fazem parte de um conjunto de interfaces não padronizadas. Estas podem conter até 1320x1035 pontos sensíveis a luz, cada um com uma área de  $6.8 \mu\text{m}^2$  e espaçamento de  $6.8 \mu\text{m}$  de centro a centro. Como 100% da área do sensor é sensível à luz, os efeitos de aliasing podem ser reduzidos. Uma das características deste tipo de sensor é a de possuir um conversor A/D no mesmo sistema. O conversor, geralmente de 8 bits, produz um sinal digital, neste caso em 256 tons, eliminando degradações de amplificadores, condicionadores de sinais e cabos de transmissão.

Outros sensores do tipo câmeras com interface não padronizada podem ser encontrados em [AMD 88] que sumariza alguns tipos de câmeras e suas principais características.

## 2.2.4 Degradações

Em todo sistema, alguns efeitos indesejáveis podem ocorrer por diversos motivos. Algumas destas degradações são comentadas a seguir.

A forma mais comum de ruído em sistemas de varredura é conhecido como *snow*. A presença destas manchas é devido à excessiva quantidade de ruído branco (uniformemente distribuído na banda passante) sobre o sinal. Este pode ser modelado como ruído aditivo, sendo o sinal ( $\mathbf{v}$ ) representado como  $\mathbf{v} = \mathbf{v}' + \mathbf{r}$ , onde o ruído ( $\mathbf{r}$ ) e o sinal imagem ( $\mathbf{v}'$ ) não são correlatos. Este tipo de ruído é observado quando a relação sinal/ruído cai abaixo dos 40 dB. Esta degradação pode ser introduzida nos vários caminhos de transmissão, seja em cabos ou nos sistemas de teledifusão (*broadcast*). Outros tipos de ruído manifestam-se diferentemente. Em alguns casos o nível de ruído depende da intensidade do sinal de imagem. Se o ruído for, por exemplo, proporcional ao sinal de vídeo, isto é,  $\mathbf{v} = \mathbf{v}' + r\mathbf{v}'$ , temos  $\mathbf{v} = \mathbf{v}'(1+r) \approx \mathbf{v}'r$ , que representa um ruído não correlato porém multiplicativo. Outros exemplos são os ruídos de baixa frequência que aparecem como linhas horizontais no quadro. No caso de sinal de vídeo com informação de cor, devido à estreita faixa do canal de crominância e a superposição no final da faixa de luminância, os efeitos de *cross-coloure* *cross-luminance* frequentemente aparecem.

Imagens geradas por fonte de luz coerente, como imagens de radar infravermelho [EKS 84] podem apresentar ruídos que se manifestam como manchas, chamadas de efeito *speckly*. Imagens transmitidas em um canal ruidoso ou sob interferência de ruído elétrico podem apresentar pontos pretos e brancos espalhados na imagem. Este tipo de efeito é conhecido como *salt-and-pepper*. O ruído tipo "chuvisco", também visualizado como pontos pretos e brancos ao longo da imagem, é resultado de transições de alta frequência com tempo de duração considerável e característica de distribuição aleatória.

Para os ruídos de alta frequência, um filtro passa baixas pode ser colocado, desde que não ocorra perda significativa de resolução. Em [ODG 81] encontramos que para uma imagem de 512 x 512, uma largura de

banda de 7Mhz para o sistema de exibição e suficiente para que não haja degradações pela perda de altas frequências.

As distorções devido à perda de alta frequência, afetam a percepção das transições. Este tipo de degradação pode ser introduzido devido a perda de foco nas lentes, movimento ou turbulência atmosférica, ou pela limitação da banda de passagem do sinal. Este tipo de degradação é chamada de *blurring* ou ainda de ruído de convolução. A perda na amplitude e, no caso de sinais coloridos, o deslocamento de fase nas frequências próximas a frequência horizontal faz com que o quadro fique borrado, normalmente à direita. Uma série de outros efeitos eletrônicos, de natureza unidimensional, podem ocorrer bem como degradações devido a deficiências geométricas e de iluminação. Alguns destes tópicos são analisados em [ODG 81], abrangendo **shading**, **fator gamma**, **aberrações óticas** nas lentes e a **variação inerente aos componentes analógicos** utilizados no equipamento sensor.

Quando quadros sucessivos possuem variações abruptas de iluminação, a visualização pode ser prejudicada. Uma técnica utilizada em receptores de TV, a fim de minimizar este efeito, é realizar uma variação lenta na amplitude do sinal de vídeo, de acordo com a intensidade dos quadros anteriores, uma espécie de integral da luminosidade em um período de tempo.

Vários tipos de interferência, chamadas de interferência coerente, podem surgir dentro do sinal de vídeo proveniente de outras fontes. Um exemplo é a interferência de RF (radio frequência), que depende da relação de frequências entre o sinal interferente, a subportadora de cor e as frequências de varredura. Outros exemplos são encontrados em [LUT 89], como as barras horizontais que variam em posição e tamanho de acordo com o som, relacionadas ao sinal de áudio, ou a ondulação de 120Hz proveniente da tensão de alimentação. A interferência de fontes coerentes é muito mais visível que a interferência aleatória ou ruído, pois cria algum tipo de padrão que é repetido na mesma posição ou com lento deslocamento.

Alguns tipos de degradação podem ser modelados matematicamente, o que permite aplicar transformações na imagem que compensem ou atenuem os efeitos de acordo com o modelo utilizado. Este



tipo de enfoque é utilizado em algumas técnicas de enriquecimento de imagens e é a base de toda a área de restauração de imagens [EKS 84], [AND 77]. A seguir são dados dois exemplos simples de modelos. O primeiro modela o borramento ou *blurring* é utilizado nos processos de reforço de contorno ou *sharpening*. O exemplo subsequente introduz um modelo de ruído aditivo e algumas considerações do processo de *smoothing*.

O borramento, ou *blurring*, traduz-se em perda de definição, ou seja, redução da capacidade de delimitar o contorno entre várias regiões da imagem. Este é, em sua essência, uma média ou uma integração, onde as altas frequências são atenuadas. É intuitivo que o efeito possa ser corrigido com operadores diferenciais. Se uma imagem possui ruído e é borrada, as técnicas de *sharpening* devem considerar que o ruído geralmente contém consideráveis componentes de alta frequência espacial. Logo as técnicas devem ser restritas a imagens onde a relação sinal ruído é grande ou onde a redução de ruído já foi realizada, caso contrário os efeitos de ruído de alta frequência serão enfatizados. No sentido de salientar as características de borramento com orientação espacial arbitrária, um operador espacial isotrópico, isto é invariante na rotação, se faz necessário. Esta limitação é usada para derivar algumas propriedades que tais operadores devem possuir. Um dos operadores diferenciais isotrópicos mais utilizados é alguma forma de Laplaciano. O Laplaciano de uma imagem  $f$  pode ser representado como:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

Assumindo que o *blurring* é consequência de um processo modelado fundamentalmente como uma difusão, temos:

$$\frac{\partial g}{\partial t} = k \nabla^2 g$$

onde  $g$  é uma função de  $(x, y, t)$  e  $k > 0$ .

Em  $t=0$ ,  $g(x, y, 0)$  é a imagem sem degradação  $f(x, y)$ .

Em  $t > 0$  pode-se representar a imagem borrada  $g(x, y, t)$  como uma série de Taylor,

$$g_{x,y,t} = g_{x,y,0} + t \frac{dg_{x,y,t}}{dt} + \frac{t^2}{2} \frac{d^2 g_{x,y,t}}{dt^2} + \dots$$

Se considerarmos apenas os termos lineares, temos:

$$\begin{aligned} g_{x,y,0} &\rightarrow f \\ \frac{dg}{dt} &\rightarrow k \nabla^2 g \\ f &= g - kt \nabla^2 g \end{aligned}$$

Assim temos um mecanismo de primeira ordem para restaurar a imagem. É claro que o processo de difusão não é o fundamento de todos os fenômenos de *blurring*, porém existem métodos baseados no Laplaciano, expresso de forma discreta. Este mesmo operador diferencial pode ser encarado como um filtro de altas frequências espaciais e calculado através de um operador de convolução.

O *smoothing* está essencialmente ligado à remoção de ruído. Muitos métodos baseados em processos lineares para *smoothing* resultam em um grau inaceitável de borrimento da imagem. Minimizar as condições de borrimento implica em realizar otimizações globais para alguma "função custo" definida adequadamente. Na prática as operações locais são utilizadas, explorando o grau de liberdade oferecido nos problemas reais [OFF 85].

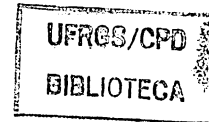
Uma imagem pode ser afetada, como citado anteriormente, por ruído de diferentes fontes, incluindo o ruído do sensor, ruído de quantização e ruído de canal. Este ruído pode ser ainda sistemático ou aleatório, sendo o primeiro facilmente tratável com técnicas de Fourier. O ruído em uma imagem possui, usualmente, componentes de alta frequência espacial em relação a imagem considerada. Assim um filtro espacial passa-baixa pode ser efetivo, embora não haja distinção entre imagem e ruído. Para obter-se uma maior eficiência, requer-se um conhecimento específico do modelo do ruído a ser tratado.

É fácil ver porque técnicas que utilizam cálculo de média reduzem ruído. Se  $s(i)$  representa um pixel livre de ruído e  $n(i)$  é o ruído correspondente, o cálculo da média fica:

$$\frac{s_1 + \dots + s_i + \dots + s_N}{N} + \frac{n_1 + \dots + n_i + \dots + n_N}{N}$$

O segundo termo da soma é uma variável aleatória com média zero e desvio padrão  $n/\sqrt{N}$ , onde  $n$  é o ruído RMS para um único pixel. Desta forma, tomando-se a média sobre um número de cópias independentes de uma dada imagem, a redução do ruído sem blurring é obtida (filtragem temporal). Se uma única imagem é disponível, então o ruído pode ser reduzido por uma média local, mas apenas se o ruído da imagem possui granularidade mais fina que os detalhes da mesma.

Os limites espaciais do processo de média local dependem da quantidade de redução de ruído desejada. Por exemplo, no caso anterior, a redução de um fator  $N$  envolve uma média sobre  $N^2$  pixels. Uma extensão das mesmas idéias, leva a média ponderada, com pesos que minimizem a média sobre detalhes de linhas e transições. Para isto deve-se usar um meio de localizar tais detalhes, implícita ou explicitamente.



## 2.3 Vídeo Digital

A fim de realizar o tratamento digital de imagens é necessário representá-la através de um padrão de bits. Na transformação de um sinal de vídeo analógico para sua forma digital, vários aspectos devem ser observados nos processos de quantização e amostragem, bem como os efeitos destes parâmetros sobre as imagens.

### 2.3.1 Amostragem e Quantização

O processo de converter sinais contínuos no tempo em um conjunto de valores discretos no tempo, é chamado de amostragem e a etapa de transformar a amplitude dos sinais em valores discretos é dita quantização. No caso de imagens, a função imagem  $f(x,y)$  deve ser digitalizada nas coordenadas espaciais  $(x,y)$ , chamada amostragem do sinal, e a intensidade (valor da função) deve ser convertida em vários tons de cinza, chamada de quantização.

Os processos de amostragem são inúmeros, sendo o mais simples a chamada amostragem uniforme [GON 87] onde uma imagem  $f(x,y)$  contínua é convertida em amostras igualmente espaçadas na forma de uma matriz  $M \times N$ . Nesta matriz cada elemento é um valor discreto que corresponde a intensidade luminosa do ponto. Nestes casos uma prática comum é que os valores de  $M$  e  $N$  sejam potências de 2 e muitas vezes faz-se  $M=N$ . A amostragem não uniforme pode melhorar a aparência da imagem, através de um processo adaptativo que depende das características da mesma. Em geral uma amostragem com menor granularidade é utilizada na vizinhança de uma transição e amostras mais espaçadas em regiões de luminosidade constante da imagem. Uma discussão ampla sobre os processos de amostragem uniforme e não uniforme utilizando matriz de pontos ou baseados em representação da imagem pelos coeficientes de uma função ortonormal pode ser encontrada em [PRA 78], [ROS 82], [HAL 79] e [MAS 84].

A aplicação do teorema da amostragem no caso bidimensional resulta que, uma função bidimensional  $f(x,y)$  pode ser teoricamente reconstruída se o intervalo da amostragem seguir as equações abaixo:

$$\Delta x' \leq \frac{1}{2W_u} \quad \Delta y' \leq \frac{1}{2W_v}$$

onde  $2W_u$  e  $2W_v$  representa a largura da função na direções  $u$  e  $v$  no domínio frequência. Porém quando a  $f(x,y)$  é espacialmente limitada por uma janela  $h(x,y)$ , a transformada da função amostrada é distorcida pela convolução da  $H(u,v)$  com  $S(u,v)*F(u,v)$  onde  $S(u,v)$  é a transformada de Fourier da matriz de impulsos de amostragem, e  $F$  é a transformada de Fourier de  $f(x,y)$ . Esta distorção origina-se do fato das imagens serem espacialmente limitadas e impede a reconstrução completa das imagens a partir de suas amostras.

Da mesma maneira a quantização pode ser uniforme e não uniforme [GON 87], [PRA 78], [ROS 82], [HAL 79] e [MAS 84]. Quantização uniforme significa que os níveis de discretização são igualmente espaçados entre 0 e L (valor máximo) na escala de tons. Normalmente o número de níveis também é feito uma potência de dois, que no caso de dois níveis passa a se chamar de binarização.

Uma classe de algoritmos baseia-se na quantidade de tons possíveis na representação da imagem. Alguns destes processos convertem representações de tons de cinza em uma imagem binária a fim de aumentar a velocidade de processamento. Entretanto a informação dos tons de cinza pode representar variações na iluminação, ponto de vista, propriedades de reflexão e permitir processos de medição quantitativos em uma imagem amostrada. No caso de imagens binárias, o limiar de quantização pode ser variável, variando de quadro a quadro ou localmente conforme a região da imagem. O quadro da figura 2.4, extraído de [SIL 87], mostra algumas aplicações e o método de quantização utilizado.

Limiar		Intensidade de Luz Variável	Superfície Contraste Var.	Iluminação Não Uniforme	Quantização Espacial
Binarização	Fixo				
	Automático	•	•		
	Local	•	•	•	
Tons de Cinza		•	•	•	•

Figura 2.4 Níveis de quantização e Aplicações

Para imagens em tons de cinza pode-se utilizar poucos níveis em transições abruptas e uma escala completa em regiões de lenta variação de luminosidade a fim de evitar falso contorno. Outra idéia similar é calcular a frequência de ocorrência dos níveis desejados e colocar níveis de quantização próximos na área de interesse e mais afastados nas outras áreas. A dificuldade de generalização destes métodos é que estes implicam em determinadas qualidades da imagem e um conhecimento prévio da mesma.

### 2.3.2 Efeitos da Digitalização

A bibliografia mostra a derivação de critérios de quantização ótima, que visa minimizar o erro entre a imagem original e a imagem quantizada. Este erro é referenciado como erro de quantização e no caso de quantização uniforme estes critérios não são significativos. Na prática o número de linhas, ou a resolução vertical é dada pela câmera utilizada, já que o sinal de vídeo é fornecido linha a linha pelo dispositivo de captura.

Uma vez que a resolução vertical é limitada, resta ainda determinar a resolução horizontal, composta por duas figuras de mérito: **número de pixel/linha** e **número de bits/pixel**. Quanto maior estes dois valores, maior a resolução da imagem digitalizada que se aproximará da imagem original. É evidente que estes parâmetros estão limitados na capacidade dos sensores, como visto anteriormente, bem como na

na capacidade de memória de armazenamento de imagens. O número de bits necessários para armazenar uma imagem digitalizada, fazendo M pixels/linha, N linhas e b bits/pixel é:

$$\text{Mem} = M \times N \times b \text{ bits}$$

Estes dois parâmetros e a taxa de amostragem devem ser observados para evitar distorção na imagem. Estes efeitos de distorção são conhecidos como falso contorno, *aliasing* e *pixellation*.

*Pixellation* é o efeito visual de uma baixa resolução; é também chamado de *checker-board* ou efeito de tabuleiro de xadrez. Este manifesta-se por uma aparência de blocos que formam a imagem. O grau de visibilidade depende da distância de visualização da imagem reproduzida. A relação entre a distância e o número de amostras de um quadro é a razão de visualização. Esta é a distância entre o observador e o dispositivo de exibição em múltiplos da altura do quadro. O gráfico da figura 2.5 mostra a relação necessária para uma boa visualização [LUT 89]. O gráfico assume que os pixels são quadrados (um pixel pode ocupar mais que uma linha de varredura).

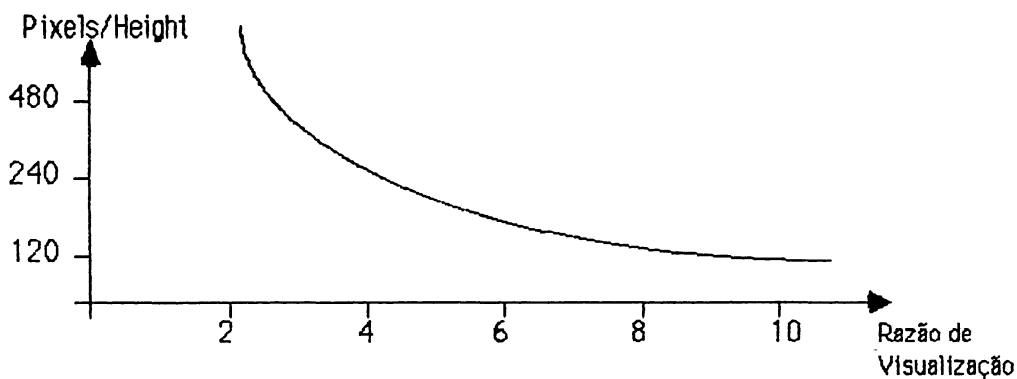


Figura 2.5 Razão de Visualização

A literatura apresenta um valor mínimo de 256 pixels/linha para imagens provenientes de uma câmera de TV. Resolução de 512 pixels é apontada como boa resolução e o ganho em qualidade de visualização não é

significativo com imagens digitalizadas com 1024 pixels/linha, no caso de imagens de TV.

O número de bits/pixel determina o número de tons ou cores possíveis para representação das imagens. Um fator importante a ser observado é que uma boa relação sinal/ruído em um sistema de vídeo é de 46dB, o qual produz 3% de ruído pico a pico. O número de níveis necessário para reproduzir um sinal com ruído é, no mínimo, igual ao ruído pico a pico. Esta condição resulta em 33 níveis para um sistema de 46dB de sinal/ruído. O fenômeno chamado de falso contorno ocorre porque níveis analógicos caem entre dois limiares de quantização e são colocados com o mesmo valor digital. Qualquer parte da imagem que variar lentamente será convertida em degraus quando quantizada. Considerando este efeito, uma quantização de 8bpp (bits por pixel) evita o problema de falso contorno [LUT 89], [EKS 84]. Não existe muita vantagem em utilizar mais do que 8 bits em um sinal que será exibido para visualização a menos que algum processo não linear, como a correção gamma, seja realizada após a digitalização. Um número maior de bits pode ser necessário se o sinal for processado para outras aplicações que não a de simples visualização.

*Aliasing* é o erro causado pela amostragem de uma imagem que contém componentes acima do limite de Nyquist para a taxa de amostragem já discutida anteriormente. Limite de Nyquist significa que a faixa de frequência ocupada pelo sinal deve ser menor ou igual à metade da frequência de amostragem. No momento da exibição, quando a imagem é convertida em dados analógicos e exibida no monitor, a largura de banda do sistema deve ser no mínimo igual ao limite de Nyquist para a taxa de amostragem utilizada. É claro que sempre ocorrerá algum tipo de suavização em transições abruptas, porém apenas na direção horizontal.

Outros tipos de efeitos de degradação podem ocorrer devido a natureza dos processadores digitais e da imagem amostrada. Por exemplo, a quantização pode introduzir ruído randômico, modelado com ruído aditivo [AND 77] e, para uma imagem ruidosa, uma técnica de binarização pode introduzir o já citado ruído do tipo *salt-and-pepper*. Como se trabalha com palavras de largura finita, erros podem ser introduzidos nas transformações de domínio devido a precisão limitada dos processadores digitais.



## 2.4 Exibição de Imagens

Dispositivos de exibição são aqueles que convertem uma imagem de sua representação numérica para uma forma perceptível ao olho humano. Nos sistemas baseados em CRTs, as posições horizontal e vertical do pixels são convertidas em tensões que são utilizadas para direcionar um feixe de elétrons [GRO 79]. A cada ponto direcionado a intensidade do feixe produz um elemento de imagem. Esta intensidade é modulada por uma tensão proporcional ao valor do pixel exibido.

Um sistema de exibição de dados digitais possui algumas características diferentes dos sistemas convencionais de TV. Estes aspectos podem ser considerados a fim de compreender a limitação de resolução e fidelidade destes equipamentos. Tais considerações podem ser encontradas em [PHI 85].

Devido ao fato que a distância de observação é, na maioria das vezes, muito menor que em sistemas de televisão o entrelaçamento não é recomendado, já que pode introduzir piscamento local ou *line flicker*. A percepção de contorno em um quadro é uma interação de três parâmetros: tamanho da malha (*spot size*), espaçamento do fósforo (*phosphor spacing*) e tamanho do pixel (*pixel size*). O espaçamento de ponto é também chamado de *pitch*. Quando o quadro possui mais de uma cor ativa localmente, erros de convergência afetarão a percepção do contorno. Além destes erros, a falta de foco (defocalização) influi na identificação de detalhes finos da imagem.

O tamanho da malha e espaçamento (*pitch*) são figuras características do tubo (CRT) ao passo que o número e tamanho dos pixels é determinado pelo controle de vídeo do computador e pela área de varredura.

*Spot size* (SS) é o diâmetro do feixe de elétrons medido no ponto onde sua intensidade cai a 50%. O SS visual é o diâmetro onde a intensidade cai a 5% e é em torno de 2x o *spot size* físico.

*Dotpitch* (DP) é um parâmetro geométrico do tubo e representa a distância entre os pontos de fósforo de mesma cor. Suas componentes

componentes ortogonais são chamadas de  $P_v$  e  $P_h$ , como mostrada na figura 2.6.

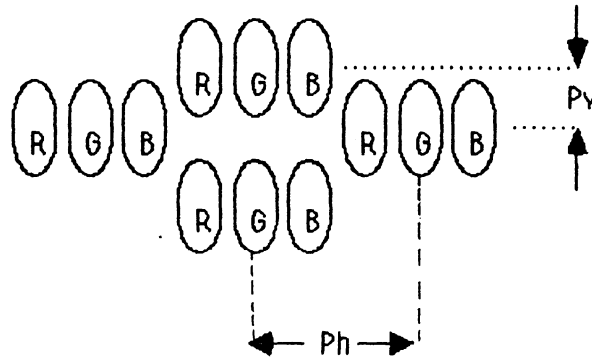


Figura 2.6 *Dot Pitch*  $P_v$  e  $P_h$

Para tecnologia com uma máscara de trilhas a teoria é a mesma que para máscara de pontos [PHI 85], exceto que a primeira não possui pontos de fósforo, que atuam como uma amostragem espacial na imagem de vídeo projetada.

É conhecido que todos os caracteres, alfanuméricos ou gráficos, exibidos por computadores digitais são formados por uma matriz de pixels. Estes pixels, gerados pelo sistema controlador de vídeo do computador não são parâmetros do tubo. O tamanho do pixel depende ainda da velocidade de varredura horizontal e da distância de linha (vertical). Em aplicações gráficas, a fim de evitar distorção, são preferidos pixels quadrados. A interação entre os parâmetros supracitados pode ser caracterizado nas relações *spot size vs. dot pitch*, *spot size vs. pixel size* e *dot pitch vs. pixel size*

#### \* *Spot Sizes vs. Dot Pitch*

O tamanho da malha (*spot size*) deve prevenir modulação de intensidade, caso contrário a luminosidade dependerá, aleatoriamente, da posição do spot relativamente ao espaço entre pontos. Desta forma o *pitch* deve ser menor que o *spot size*

\* *Spot Sizes Pixel Size*

Esta interação é considerada em termos de índice de modulação (m) e razão de contraste (c). Os dois valores descrevem o mesmo fenômeno, que pode ser visto na figura 2.7.

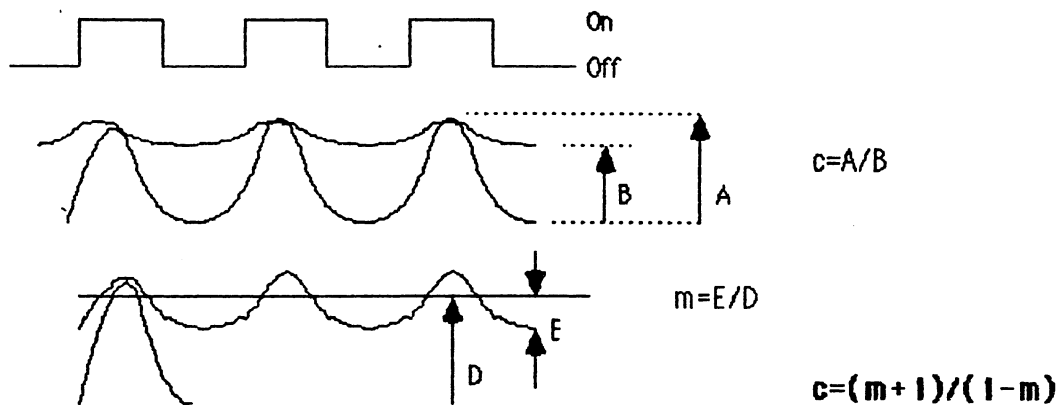


Figura 2.7 índice de Modulação (m) e Razão de Contraste (c)

A partir das expressões pode-se verificar, por exemplo, que um índice de modulação de 10% (-20dB) é equivalente a uma razão de contraste de 1.22. Para um padrão *on-off-on* exibido, 2 pixels vizinhos serão misturados. Pixels adjacentes em estado *on* (ligados) devem ser misturados para obter-se uma imagem homogênea. Assim o tamanho de pixels deve ser menor que o *spot size*

\* *Dot Pitch vs. Pixel Size*

Existem dois aspectos na relação entre o *dot pitch* e o tamanho do pixel. São eles o padrão de Moiré e o teorema de Shannon. Estes fenômenos ocorrem devido ao fato que a máscara de sombra do tubo atua como um filtro de amostragem espacial. Se o tamanho do pixel for quase igual ou múltiplo de  $P_h$  ou  $P_v$ , padrões de batimento espacial podem ocorrer na tela. O padrão de batimento em áreas iluminadas uniformemente são

chamados de " *scan Moiré* ", onde a distância da linha é quase igual ou múltiplo do espaçamento das trilhas verticais. O padrão de batimento em quadros que possuem faixas estreitas é chamado de vídeo Moiré, onde o espaçamento de faixas é quase igual ou múltiplo do espaço das trilhas horizontais. Deve-se assim evitar um tamanho vertical de pixel (distância de linha) perto de 1,2,3,.. vezes  $P_v$  e um tamanho horizontal múltiplo inteiro de  $P_h$ .

O teorema de Shannon para sinais amostrados diz que a largura de banda ocupada pelo sinal deve ser menor do que a metade da frequência de amostragem, caso contrário os detalhes finos serão perdidos. Transferindo o teorema para a máscara de sombra do tubo, vem que a distância mínima de pixel em um padrão *on-off-on* deve ser maior que 2 *itches*. Podemos representar esta última afirmativa nas inequações abaixo.

Altura pixel  $> P_v$

Largura pixel  $> P_h$

Na prática no mínimo 25% [ $P_{HI}$  85] deve ser adicionado ao tamanho do pixel para reduzir os efeitos de aliasing. Em aplicações de imagem esta regra pode ser violada, já que normalmente os quadros exibem regularidades que se estendem por mais do que um pixel. O "efeito escada" ao longo de linhas inclinadas pode ser reduzido ou mesmo eliminado quando utilizado pixels "sub-Shannon" (em tamanho). A figura 2.8 mostra a relação entre os parâmetros discutidos acima.

podem ser consideradas como sinais unidimensionais. Desta forma, conforme o enfoque dado ao tratamento e ao modelo de degradação usado, os algoritmos podem operar apenas com linhas, conjunto de linhas ou conjunto de quadros. Aqueles que trabalham a nível de linhas são referenciados como filtros horizontais e aqueles que tratam conjunto de linhas valem-se das propriedades de vizinhança dos pixels, herdada do modelo bidimensional de imagem. Na maioria das referências o tratamento bidimensional é realizado. Em alguns casos o sinal de transmissão de TV pode ser tratado de forma unidimensional, como encontrado em [EDE 89] e [PHI 86]. Outra técnica é utilizar a filtragem temporal, aproveitando a correlação entre quadros. Neste tipo de técnica são extraídos parâmetros de um quadro para processamento dos próximos quadros. É claro que para imagens móveis deve existir um detector de movimento que iniba os procedimentos de filtragem para grandes variações do quadro [EDE 89].

Além do número de pixels/linha a escolha da frequência de amostragem deve levar em conta sua relação com a frequência de linha. Esta relação torna-se importante na medida que é necessário utilizar componentes de armazenamento de linhas e quadros, que devem estar sincronizados com a frequência de amostragem. Se a taxa de amostragem, denotada por  $f_s$  for um múltiplo inteiro da frequência de linha  $f_l$ , evita-se o problema de sincronização do relógio a cada início de linha, caso contrário a posição dos pixels amostrados poderão variar de quadro para quadro, causando um efeito de piscamento nas transições, quando da captura de imagens contínuas.

Com base nos tópicos abordados neste capítulo foram escolhidos os valores dos parâmetros de nº pixels/linha e nº bits/pixel para o modelo de sistema utilizado ao longo do trabalho.

### ***Quantização Uniforme***

***512 pixels/linha***

***8 bits/pixel (bpp) - 256 tons***

***$f_s = M * f_l$ , com  $M$  inteiro***

Por exemplo para o padrão RS170 temos:

Frequência de linha: **15Khz**

Tempo de linha ativa: **54 us**

Frequência mínima:  **$1/(54/512) = 9.48\text{Mhz}$**

Frequência de amostragem =  **$9.48\text{Mhz} (632 * f1)$**

### 3 PROCESSAMENTO DE IMAGENS

Processamento de imagem consiste em qualquer operação que transforme uma representação de uma imagem em outra, a qual pode ser utilizada para visualizar algum detalhe específico, descrever um conjunto de objetos ou representar uma forma compacta de armazenamento desta imagem. Os algoritmos do processamento de imagens podem ser agrupados de acordo com o objetivo ao qual se destinam. Procedimentos para enriquecimento, restauração, codificação e transmissão de imagens já existem há algum tempo e hoje servem de núcleo para novas técnicas de visão computadorizada, que buscam a compreensão de imagens em cenas tri-dimensionais, modelando o comportamento do sistema de visualização humano.

Existem duas possíveis fontes de imagem: as capturadas através de sensores e as sintetizadas por algum método computacional. O tratamento de dados provenientes de sensores trabalha com cenas reais, procurando fidelidade ao quadro capturado ou extração de características e informações.

As técnicas de IP podem ser independentes do contexto, muitas vezes chamados de processamento de baixo nível, ou dependentes do contexto (orientada ao objetivo), também chamadas de processamento de alto nível. Os algoritmos usados podem ser classificados hierarquicamente em níveis de processamento. No nível mais baixo de processamento, o qual inclui muitos dos algoritmos convencionais de IP, a imagem é considerada como um conjunto de pixels, sem referência a estrutura ou objetos contidos na imagem. Estas operações atuam sobre a imagem, pixel a pixel, amostrada e organizada em uma matriz bidimensional. Exemplos destas operações são filtros, operadores de *smoothing*, codificação de imagens e transformação cosseno. Em procedimentos como enriquecimento de imagens, onde um dos fatores é a qualidade subjetiva dos dados, técnicas *AD HOC* são empregadas, bem como tratamento estatístico dos dados. Uma revisão de algumas técnicas pode ser encontrada em [OFF 85], [BAR 87], [DVO 88], [AMO 88].

Para tarefas mais complexas não basta tratar a imagem como um conjunto de pontos amostrados. O próximo nível de processamento consiste em extrair e manipular informações contidas na imagem. Estas informações podem ser descritas em termos de estruturas como bordas, limites, texturas ou formas que constituem objetos. Outra maneira de obtenção de tais dados é utilizar métodos de decomposição estatística da imagem.

Um terceiro nível de processamento atende tarefas mais complexas, em aplicações onde algum grau de interpretação e entendimento se faz necessário e, por conseguinte, incluem a utilização do conhecimento sobre o domínio utilizado. Tarefas que utilizam sistemas de inteligência são, por exemplo, descrição de cenas ou modelagem 3-D. Entretanto mesmo com estas tarefas que incluem um conhecimento significativo sobre a imagem, o nível mais baixo de processamento é ainda necessário. Esta afirmação é devido ao fato que sistemas baseados no conhecimento necessitam de um conjunto de qualidades confiáveis para trabalhar, função do processamento de baixo nível sobre as imagens. A figura 3.1 mostra a classificação hierárquica dos algoritmos de processamento de imagens, que resume-se em 3 níveis. Esta classificação é didática e alguns algoritmos caminham sobre todos os níveis, não restringindo-se a um nível particular.

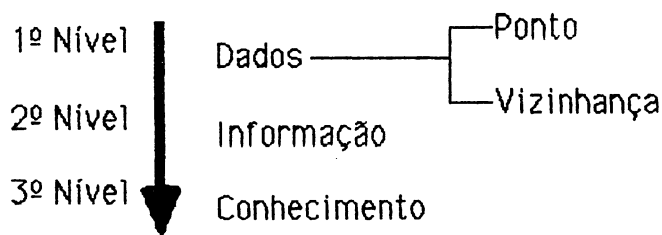


Figura 3.1 Classificação hierárquica dos algoritmos IP

Os algoritmos de primeiro nível podem ser ainda classificados em algoritmos de ponto ou de vizinhança e algumas tarefas baseadas em conhecimento podem orientar o processamento neste primeiro nível.



No tratamento de ponto os pixels utilizam apenas o seu valor local e não sua posição no quadro processado. Já nos algoritmos de vizinhança, o resultado depende da posição do pixel relativa aos seus vizinhos. Como exemplos de algoritmos de tratamento de ponto pode-se citar as transformações da escala de cinza para aumento do contraste e tratamento de pseudo-cores. Técnicas de filtragem bidimensional, extração de contorno e transformação de domínio (FFT, HADAMARD) e outras, consideram a natureza bidimensional de imagem, através da utilização dos valores dos pixels vizinhos na computação do resultado.

Operações entre pixels são extensivamente utilizadas em várias áreas do processamento de imagens. As operações aritméticas entre dois pixels  $p$  e  $q$  são descritas como abaixo,

adição:  $p + q$

subtração:  $p - q$

multiplicação:  $p \times q$  (ou  $p q$ )

divisão:  $p / q$

onde muitas vezes um dos termos é um operando constante, como por exemplo na multiplicação de imagens por um fator constante de brilho. As operações lógicas utilizadas em IP ( *Image Processing* ) são AND, OR e COMPLEMENTO, escritas como

AND:  $p \cdot q$  ( $p$  AND  $q$ )

OR:  $p + q$  ( $p$  OR  $q$ )

Complemento: NOT  $q$  ( $\bar{q}$ )

e podem ser combinadas para compor qualquer outra operação lógica, já que as citadas acima são funcionalmente completas.

Estas operações podem ser aplicadas de duas formas, de acordo com o tipo de algoritmo: pixel-a-pixel ou em operações orientadas a vizinhança. Por exemplo a soma de 2 imagens é uma operação do tipo pixel-a-pixel. As operações de vizinhança são ditas operações de máscara,

em função do próprio pixel e de seus vizinhos. Tais funções podem ser apenas a combinação linear destes pixels ou operações mais complexas, no sentido de envolver decisão durante a computação de valores. Seleccionando convenientemente os componentes das máscaras e aplicando-a a cada posição da imagem é possível realizar uma série de operações, que permitem redução de ruído, alargamento de regiões e detecção ou extração de determinadas propriedades [GON 87].

Uma operação de máscara que deriva das propriedades da transformação da imagem no domínio frequência é a convolução. Esta operação pode ser utilizada como bloco funcional de outros algoritmos, além de expressar matematicamente várias operações de acordo com a escolha das máscaras, e.g. LAPLACIANO, GRADIENTE e outras.

### 3.1 Componentes de um Sistema IP

Os elementos básicos de um sistema de processamento de imagens foram citados no capítulo 1: Digitalizador, processador, memória de imagem e sistema de exibição. Estes componentes são formados por um ou mais elementos interconectados de diversas formas. Existem variações na atribuição de funções de exibição entre algum hardware específico ou controle por microprocessador local. Vários exemplos são vistos em [COM 83], onde se mostra a utilização de máquinas paralelas e microprocessadores para realização do processamento. Variações em torno da organização em barramentos geraram produtos adaptados a padrões comerciais, o que traz vantagens por suportar interfaces de Software e Hardware padronizadas. Alguns sistemas utilizam este enfoque com mapeamento total ou parcial do espaço de memória de vídeo diretamente no barramento. Um exemplo é dado em [EKS 84], citando DeAnza Systems Displays e Vicom. Sistemas com barramentos especiais que permitem protocolos convencionais tem sido utilizados em aplicações dedicadas e de maior velocidade. Exemplos destes sistemas são o IKONAS e AYDIN [EKS 84]. Pode-se citar ainda um terceiro tipo de sistema com

maior velocidade. Exemplos destes sistemas são o IKONAS e AYDIN [EKS 84]. Pode-se citar ainda um terceiro tipo de sistema com controle de barramento interno, menos flexível que um barramento padronizado e serve como rede especializada na conexão entre a interface de exibição e outros subsistemas, como na figura 3.2.

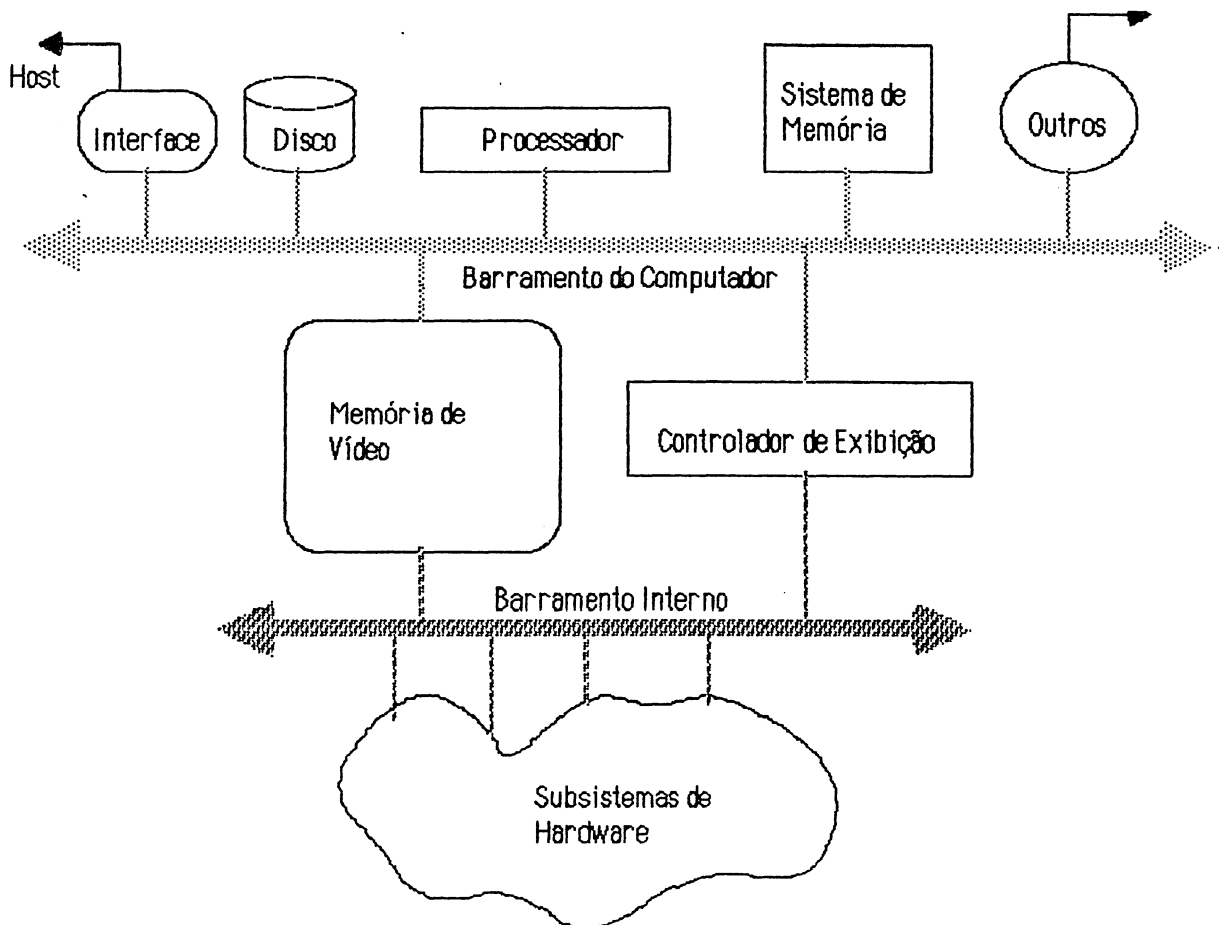


Figura 3.2 Arquitetura com Barramento Interno

Existem produtos compatíveis com Multibus e Q-Bus, placas para IBM PC modelos XT, AT e similares, bem como módulos VME orientados a tempo real além de estações dedicadas a análise e processamento de

imagens. Nestes sistemas existem módulos chamados ALUs (*Arithmetic Logic Units*), que realizam funções aritméticas e lógicas sobre os pixels, tais como soma, subtração, ou-exclusivo e outras. Existem também placas com funções específicas de convolução e histogramas e, nos sistemas mais especializados, módulos que realizam algoritmos de segmentação e técnicas de morfologia matemática [BAR 87]. Exemplos destas arquiteturas podem ser vistos em [MAH 86] que mostra a arquitetura da série IP-512, PC-Vision, séries 100 e 150 além da estação OEIL (Outil d'Etude d'Image Lyonnais).

Outro exemplo de sistema baseado em IBM-PC é encontrado em [MAT 87]. O blocodiagrama da figura 3.3 mostra o caminho dos dados dentro do sistema. Este *Vídeo Frame Grabber Digitizer* é capaz de operar em modo contínuo ou em captura de um único frame, e possibilita colocar planos adicionais de texto ou gráfico, chamado de capacidade de *keying*.

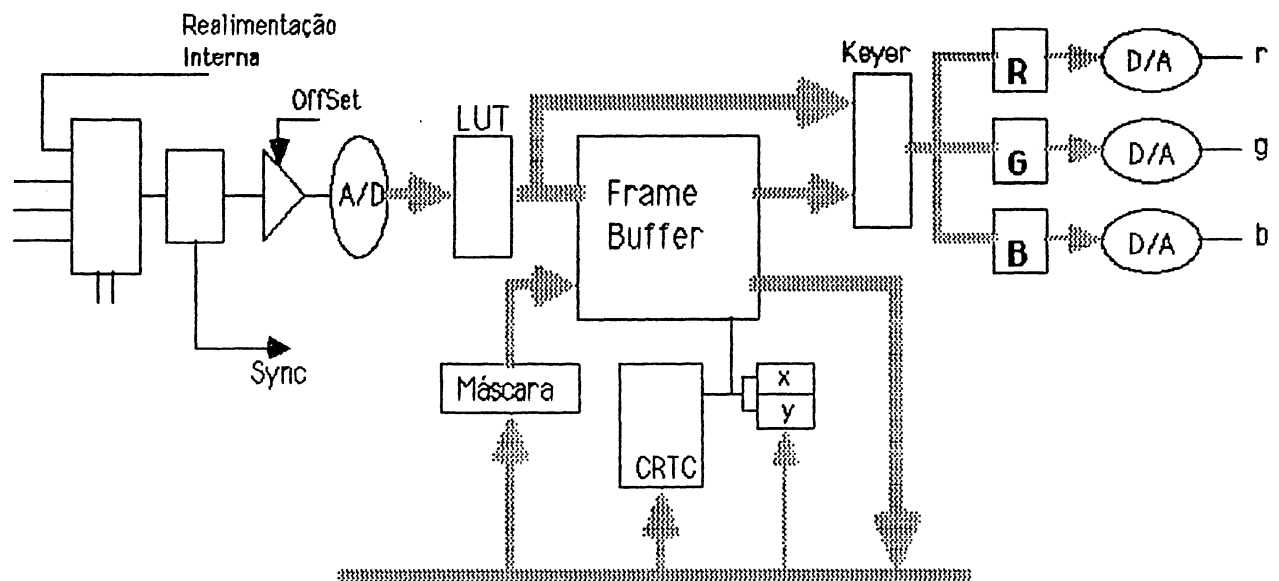


Figura 3.3 Digitalizador de Vídeo PIP

A qualquer nível de processamento podem existir módulos especializados que realizam as tarefas em hardware. No tratamento de primeiro nível, processadores de ponto e espaciais assumem diversas configurações, de acordo com a aplicação, e duas delas são mostradas na figura 3.4.

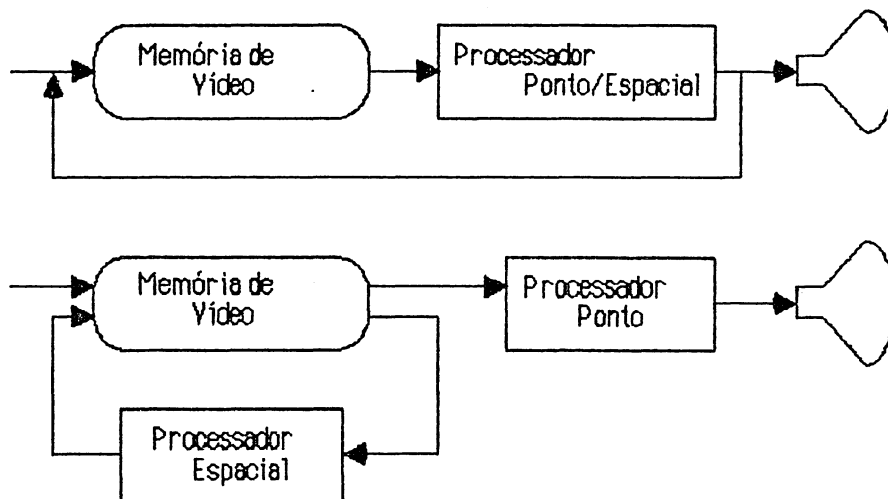


Figura 3.4 Configurações de Processadores Especializados de Vídeo

No que se refere a processadores de vídeo, é de senso comum que algumas tarefas exigem uma alta taxa de processamento, em especial os procedimentos de primeiro nível. Estas taxas não são alcançadas pela arquiteturas convencionais do tipo "Von Neumman". Explorando as características inerentes das classes de algoritmos IP, pode-se seleccionar um conjunto de operações que se repetem sobre os dados, pixel a pixel em alguns casos. Assim hoje são propostos e utilizados sistemas com componentes dedicados a DSP (*Digital Signal Processing*), que atingem alto poder computacional explorando o paralelismo potencial dos algoritmos. Pode-se citar alguns exemplos como o NEC uPD7281, um processador *Data Flow* que utiliza um pipeline circular em seu processamento. Uma família muito utilizada nas tarefas de processamento de sinais são os processadores Texas da família TMS 320. Este conjunto de componentes utiliza barramento de dados de 16 bits com 32 bits para

processamento interno, utilizando extensivamente a técnica de pipeline e uma arquitetura de **Harvard** modificada.

Processadores do tipo array exploram o paralelismo no processamento dos pixels, como o GRID (GEC Rectangular Image and Data Processor System) [OFF 85] e o GAPP (Geometric Arithmetic Parallel Processor) [NCR 85].

No processamento unidimensional, que está surgindo em receptores de TV, alguns fabricantes já desenvolveram componentes específicos como o SAA9010 [PHI 86], que realiza filtragem linear não recursiva, média de frame e congelamento de imagens. A descrição de uma série de funções e dos componentes que a implementam é encontrada em [EDE '89].

Alguns sistemas utilizam o enfoque dado anteriormente, ou seja, máquinas programáveis para processamento de imagens, vistos em [LOU 80], [CAD 80], [KUL 85].

Uma outra forma de abordagem para sistemas de processamento de imagens abandona o elemento central de processamento, normalmente programável, e procura identificar operadores comuns, os quais, devidamente agrupados, possam construir algoritmos mais complexos [OFF 85]. Uma vez identificados tais operadores, resta construir componentes especializados nestas funções e que atuem sobre um conjunto de dados de vídeo. A tradição em processamento de primeiro nível de imagens já identificou muitos operadores necessários a implementação de grande parte dos algoritmos, sendo possível integrar componentes específicos de alto desempenho. Várias arquiteturas tem sido investigadas para inúmeras funções de núcleo, como convolução, correlação e filtragem espacial; para que seja possível realizar o processamento em tempo real. Este tipo de enfoque pode ser claramente visto em [RUE 87] com um conjunto de processadores especializados que formam um sistema completo de IP, também discutido em [FOS 80]. Nas referências abaixo tem-se vários tipos de arquiteturas, cada uma situada ao operador implementado. Implementação de filtros de mediana bidimensional, rank order filter uni e bidimensional e operadores para detecção de bordas

podem ser vistos em [OFF 85], [WOO 81], [VAS 87], [RAM 88], [PER 87], [ARC 88], [FIS 81], [RUE 87], [KUN 80a].

Sistemas comerciais utilizam este enfoque, como por exemplo em [MAH 86] no sistema da série 150 da Tekelec Airtronic. Este é composto de 4 placas VME, com uma unidade de digitalização e restituição, uma memória de imagem, uma unidade aritmética e um sistema dedicado a convolução em tempo real. Os dados passam através do sistema em forma de pipeline mostrado no blocodiagrama da figura 3.5.

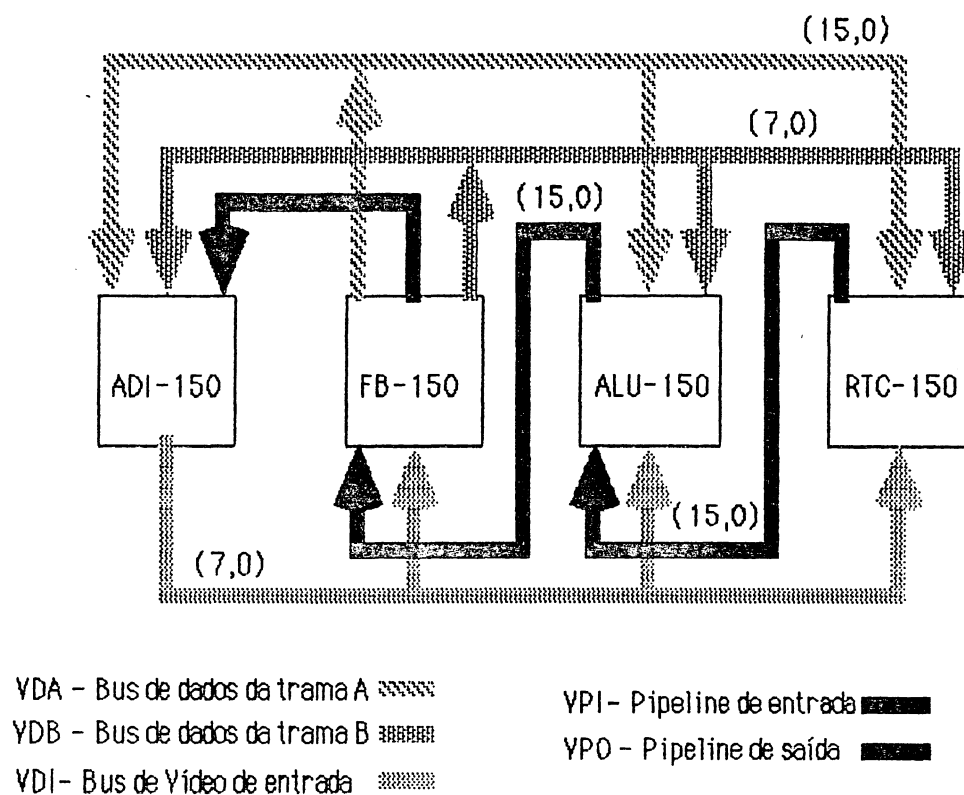


Figura 3.5 Arquitetura do Sistema Série 150

As principais características que são encontradas em todos os sistemas de IP são citadas abaixo.

- \* Captura de imagens a partir de um sensor capaz de amostrar imagens móveis: Câmeras de TV.



- \* Memória própria para o sinal de vídeo ( *framebuffer* ), trazendo independência do hospedeiro.
- \* Acesso do hospedeiro, ao controlador e a memória de vídeo, permitindo transferências com memória de massa e computação de algoritmos mais complexos.
- \* Mecanismos de sincronização na amostragem, exibição e computação dos dados em módulos especializados.
- \* Dispositivo de exibição com capacidade de resolução compatível com os algoritmos implementados.
- \* Possibilidade de colocação de módulos especializados para computação em hardware, eventualmente em tempo real.

Entre estes módulos podemos citar look-up tables de transformação de níveis ou tratamento de pseudo cores, capacidade de *keying* (planos de texto e gráficos sobre a imagem) e outros.

Para memória de vídeo existe uma tendência no uso de Vídeo-RAMs. Uma boa revisão das capacidades deste tipo de memória é vista em [NIC 88]. O shift register incluído neste tipo de memória facilita o controle de E/S de dados de vídeo, liberando o outro barramento de dados para acesso dos demais subsistemas.

### 3.2 Sistema A17

A seguir são dadas as linhas gerais para proposta de um sistema modular de IP, capaz de suportar a implementação de funções em software bem como suportar a adição de módulos especializados que realizam, de forma eficiente, algum algoritmo ou operação específica. Este sistema propõe-se a ser uma plataforma de experiências, tanto na pesquisa e teste de procedimentos de IP bem como na implementação VLSI de algoritmos para esta classe de processamento de sinais.



No projeto A17, cada sistema pode ser definido como um processo. Cada processo possui um produtor de pixels, fornecedor de dados, e um consumidor que utiliza ou armazena os resultados produzidos. Existem nesta modelização sinais que informam quando os pixels estão disponíveis para processamento e o momento em que os resultados são válidos na saída. Um sinal adicional informa à fonte de dados que o processo reconheceu a sinalização e está apto a iniciar o consumo dos dados na entrada. Adicionalmente cada processo pode possuir sinais específicos, como sinais de decodificação ou habilitação de dados na saída. O modelo deste ambiente de funcionamento, para cada processo, é mostrado na figura 3.6.

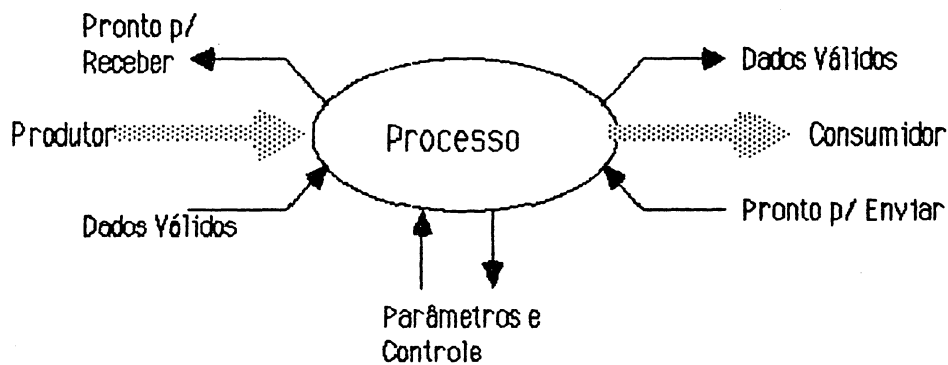


Figura 3.6 Modelo de Processo no Sistema A17

Desta forma o A17 pode ser descrito por uma interface com um computador para programação de parâmetros e um conjunto de processos com sinais de sincronismo. Neste modelo podemos incluir o sensor, *look-up tables*, memória de *frame*, dispositivo de exibição e componentes especializados. É claro que cada processo pode não utilizar todos os dados da interface, como é o caso do sensor e do dispositivo de exibição, ou ainda necessitar interfaces de adaptação entre um processo e outro de acordo com a forma de implementação.

Caminhos de recirculação de dados possibilitam o processamento iterativo sem armazenamento intermediário dos quadros de imagem. A

estrutura de interconexão deve satisfazer as especificações e a mais comum consiste em um barramento (único ou múltiplo), sendo possível utilizar-se outras redes em subsistemas específicos. Por exemplo, além do caminho normal dos dados, que permite um processamento em cadeia, podem existir outros dois barramentos. Um barramento de realimentação, que permite processamento cíclico e dá maior flexibilidade na ordem de colocação dos processos no percurso direto. O outro é um caminho que fornece dados em avanço para os processos, tipicamente dados oriundos diretamente do sensor. Este último visa, sobretudo, eliminar armazenamento temporário e tempo de espera de dados intermediários. A utilização destes dois barramentos auxiliares depende dos processos implementados, porém a sua disponibilidade permite uma fácil expansão do sistema.

A idéia é que cada processo, ou conjunto destes, seja implementado em um módulo, com uma interface que torne o sistema independente do computador hospedeiro, variando apenas a interface de comunicação. Na entrada deste sistema estão os módulos de aquisição (sensor e digitalizador), memória de frame, interface de exibição e interface com o computador de programação e controle. Existe ainda um grande trabalho na definição deste sistema, o padrão de armazenamento de imagens na memória de imagem e de massa, a interface com o usuário, a forma de identificação dos módulos e da configuração atual dos processos. Outros fatores importantes são a codificação dos parâmetros e dados, o protocolo de programação e o endereçamento dos módulos implementados em hardware. No entanto com o modelo proposto já é possível definir componentes mesmo antes de completar esta descrição, onde eventualmente interfaces especiais deverão ser construídas para que tais elementos tornem-se processos do ambiente A17.

O nome A17 vem da analogia ao processamento de imagens no cérebro humano, onde os cientistas acreditam que a região do cérebro chamada "zona 17" realiza as tarefas de tratamento e formação das imagens, preparando os dados para o complexo e desconhecido processo de evocação e entendimento.

### 3.3 Convolução Bidimensional

Dentro do enfoque de construção de operadores básicos que são blocos funcionais dos algoritmos de processamento de imagens, será descrito o algoritmo de convolução bidimensional e as suas principais características. Esta análise irá definir a forma de implementação de um bloco que realiza a função de convolução, chamado de CV2D.

A convolução bidimensional de duas funções  $f(x,y)$  e  $g(x,y)$  é expressa pela equação abaixo [GON 87]:

$$f(x, y) * g(x, y) = \iint f(\alpha, \beta) g(x - \alpha, y - \beta) d\alpha d\beta$$

O teorema da convolução em duas dimensões é dado pelas relações:

$$f(x, y) * g(x, y) \Leftrightarrow F(u, v) G(u, v)$$

$$f(x, y) g(x, y) \Leftrightarrow F(u, v) * G(u, v)$$

A convolução discreta em duas dimensões é formulada fazendo  $f(x,y)$  e  $g(x,y)$  discretas, como matrizes de dimensão  $A \times B$  e  $C \times D$  respectivamente. Como no caso unidimensional, estas matrizes são supostamente periódicas, com algum período  $M$  e  $N$  nas direções  $x$  e  $y$  respectivamente [GON 87], [ROS 82], [OPP 75]. Erros são evitados não permitindo a superposição dos períodos das funções pela escolha correta de  $M$  e  $N$  como abaixo.

$$M \geq A + C - 1$$

$$N \geq B+D-1$$

Estas sequências periódicas são formadas extendendo o domínio de  $f(x,y)$  e  $g(x,y)$  como segue:

$$f_e(x,y) \begin{cases} f(x,y) \rightarrow 0 \leq x \leq A-1 \text{ e } 0 \leq y \leq B-1 \\ 0 \rightarrow A \leq x \leq M-1 \text{ ou } B \leq y \leq N-1 \end{cases}$$

$$g_e(x,y) \begin{cases} g(x,y) \rightarrow 0 \leq x \leq C-1 \text{ e } 0 \leq y \leq D-1 \\ 0 \rightarrow C \leq x \leq M-1 \text{ ou } D \leq y \leq N-1 \end{cases}$$

Desta forma a convolução bidimensional discreta de  $f_e(x,y)$  e  $g_e(x,y)$  é dada pela equação abaixo, que representa uma média ponderada, onde os pesos são dados por uma das funções convoluidas.

$$f_e(x,y) * g_e(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_e(m,n) g_e(x-m, y-n)$$

Onde  $x=0,1,2 \dots M-1$  e  $y=0,1,2 \dots N-1$ .

Analogamente ao caso unidimensional o teorema da convolução supracitado é aplicável fazendo  $u=0,1,2 \dots M-1$  e  $v=0,1,2 \dots N-1$ .

Do ponto de vista prático, na maioria das vezes é mais eficiente calcular a convolução discreta no domínio frequência ao invés de a computar no domínio tempo. O procedimento consiste em calcular a transformada de Fourier de  $f_e$  e  $g_e$  (funções extendidas) utilizando FFT. As duas funções são então multiplicadas ponto a ponto e a transformada inversa de Fourier produz o resultado desejado. A decisão de computar no domínio tempo ou frequência é função do tamanho da máscara de convolução a ser utilizada.

Em relação a computação da equação descrita anteriormente algumas considerações devem ser feitas a fim de caracterizar sua implementação. São estas: Dimensão da máscara utilizada e domínio de processamento, número de operações e complexidade do algoritmo, condições de contorno e limites da imagem.

Máscaras de dimensão 3x3, 5x5 e até 8x8 são utilizadas em muitas operações, sendo mais comum a máscara 3x3. [KRU 73] mostra que operações com máscaras de quaisquer dimensões podem ser reduzidas a uma sequência de operações com um núcleo 3x3. Considera-se que a máscara caminha sobre o plano de imagem e, na maioria dos casos, as máscaras possuem valor fixo para toda imagem, ou seja não dependem da posição relativa dentro do quadro. Em [PRA 78] encontramos que para máscaras de dimensão 3x3 é computacionalmente mais econômico realizar o processamento diretamente no domínio tempo e não no domínio frequência.

É evidente que esta operação é computacionalmente cara, ainda mais se considerarmos o fato do processamento em tempo real. Considerando uma imagem  $N \times N$  e uma máscara  $k \times k$  temos  $O(2n^2k^2)$  operações aritméticas para convoluir um único frame. Como o volume de dados utilizados e gerados é também de  $O(n^2)$  para uma convolução, um grande número de operações deve ser possível e um igual número de acessos à memória. De uma maneira geral pode-se dizer que a velocidade de computação e de acesso à memória dependerá da distribuição de entrada e saída em relação a computação realizada, e.g. quantas vezes o mesmo dado deve ser buscado na memória. Para máscaras 3x3 com frames de 512 x 512 pixels cada pixel exige 9 multiplicações e 8 adições. Isto resulta um total de 2359296 multiplicações e 2097152 adições a serem realizadas em 1/30s (1 frame de TV).

Na maior parte das aplicações as máscaras utilizadas em operações que podem ser expressas através da convolução, são compostas apenas por números inteiros. Assim para processamento em tempo real necessitamos de uma máquina com desempenho equivalente a 161 MIPs. Este valor vem do fato que 512 pixels devem ser processados em 54 us, tempo ativo de linha no padrão RS170, compatível com a maior parte das câmeras utilizadas. Apenas como exemplo, o TRANSPUTER possui um

desempenho de 10 MIPS [MAY 87], muito abaixo da velocidade exigida. Pelo exposto acima, fica claro a necessidade de uma arquitetura específica que explore algum tipo de paralelismo no processamento dos pixels.

Ainda na descrição do algoritmo tem-se o problema do processamento das bordas da imagem. As primeiras e últimas linhas e colunas da imagem não possuem vizinhança para sua computação. Esta consideração não é muito importante, uma vez que sua influência é apenas nas bordas do quadro processado. Talvez com processamento recursivo, após várias iterações, os erros introduzidos pelo falso processamento das bordas possa causar efeitos indesejados na imagem. Neste caso considerar-se-á a vizinhança de pixels sem luminosidade (valor zero) para computação das linhas e colunas limite.

Outras considerações a serem feitas em relação ao processamento da convolução dizem respeito aos valores da máscara de convolução e ao tratamento de resultados negativos. Primeiramente a máscara utilizada pode colocar um nível de brilho adicional na imagem. Isto ocorre sempre que a soma dos valores da máscara utilizada for, em módulo, maior que a unidade. Normalmente o resultado de um pixel é dividido pela soma dos valores da máscara ou por um número próximo deste, a fim de minimizar o fator constante (nível DC) introduzido no cálculo. Este será chamado de "**fator de normalização**". Adicionalmente de acordo com o peso das máscaras, resultados negativos podem ser produzidos os quais, dependendo da utilização, podem ou não ter significado. Por exemplo nos algoritmos de detecção de transições o importante é o valor absoluto do resultado. De outro lado para resultados destinados à visualização, valores negativos não possuem significado uma vez que o valor do pixel é proporcional ao seu brilho e não existe brilho negativo. Cada sistema adota uma solução, como a de limitar o valor inferior ao nível de preto ou utilizar o valor absoluto do resultado, desconsiderando seu sinal como em [MAT 87]. Como opção de sistema, será permitido ao usuário escolher o tipo de tratamento de resultados negativos, codificados em complemento de dois. Este pode limitar o valor em zero ou obter na saída os valores negativos. Internamente as operações são realizadas em complemento de dois, e o usuário controla o formato de saída de dados. Isto dá uma maior

flexibilidade ao bloco funcional, viabilizando seu uso em uma quantidade maior de algoritmos.

Grande parte das máscaras utilizadas possuem valores que são potências de dois. Simulações realizadas mostram que algumas máscaras que não utilizam valores em potências de dois podem ser substituídas por estes com resultados satisfatórios. As imagens resultantes das simulações estão anexas a documentação do circuito.

Justifica-se desta forma a implementação da convolução bidimensional de acordo com a equação abaixo:

$$O(x, y) = \frac{1}{2^n} \sum_{i=0}^2 \sum_{j=0}^2 I_{x+i-1, y+j-1} \times h_{i,j} \left| \begin{array}{l} 0 \leq x \leq r \\ 0 \leq y \leq c \end{array} \right.$$

If (SAT=1 & O(x,y) < 0) then O(x,y)=0

Onde

h(i,j) - Coeficientes  
 c - nº colunas  
 r - nº linhas

Parâmetros Programáveis:

$h(i,j)$	$\Rightarrow$ [ $\pm 0,1,2,3,4,8,16$ ]
$n$	$\Rightarrow$ [ 0..15 ]
SAT	$\Rightarrow$ [ 0,1 ]
Pixels/linha	$\Rightarrow$ [ 1..512 ]

### 3.4 Conclusão

Neste capítulo foi mostrada uma visão geral do campo de processamento de imagens. Como principal resultado destaca-se o algoritmo de convolução bidimensional como um bloco operador que compõe vários algoritmos, tais como redução de ruído por *smoothing*, reforço de contorno e redução de efeitos de borramento. Selecionando os coeficientes que representem um Laplaciano, pode ser aplicado na preparação dos dados para algoritmos de detecção de bordas e segmentação bem como filtragem linear em geral. Por isso este operador foi escolhido como bloco a ser implementado, oferecendo uma ferramenta versátil para as tarefas de primeiro nível em IP. Como subproduto surgiram também algumas sugestões para um sistema de experimentação em IP, através da modelização da interface e dos processos a serem implementados. Esta interface é composta dos dados de entrada e saída e dos sinais de sincronização para ativação e recepção dos dados do processo.

A análise do algoritmo realizada no item 3.3 definiu alguns parâmetros de projeto, levando em conta o tipo de dados e as aplicações envolvidas. Tais decisões de projeto são sumarizadas abaixo:

- \* Processamento em tempo real;
- \* Operações com máscara de dimensão 3x3;
- \* Coeficientes de convolução seleccionáveis no conjunto [  $\pm 0,1,2,4,8,16$  ];



- \* Vizinhança da imagem com pixels de valor zero;
- \* Codificação dos resultados em complemento de dois;
- \* Utilização de resultados negativos ou limitados em zero;
- \* Fator de normalização entre 1 e 255;
- \* Número de linhas programável no intervalo [1..512], facilitando a sincronização de elementos externos introduzindo no entanto limitação no tamanho do quadro.

## 4 ARQUITETURA DE CONVOLUÇÃO

O algoritmo a ser implementado possui duas características fundamentais. Uma é que o valor final de um pixel independe do resultado da computação de seus vizinhos. Isto permite que o processamento dos dados seja realizado em paralelo, ou seja vários pixels podem ser calculados ao mesmo tempo. A outra característica é que o mesmo dado é utilizado no cálculo de vários pixels, que induz a uma utilização máxima de um dado dentro da máquina. Este tipo de otimização visa minimizar a largura de banda de memória (número de palavras acessadas por segundo) sem reduzir o desempenho.

O paralelismo potencial do algoritmo e a necessidade de computação em tempo real conduz a concepção de uma arquitetura paralela para realização da função proposta. No mapeamento estrutural de um algoritmo em uma arquitetura, vários aspectos devem ser considerados. Alguns deles são citados em [OFF 85] e [KUN 80b]: expressão do paralelismo da função a ser processada, comunicação entre processos, movimentação de dados, organização de memória, manipulação algébrica do problema e ordem de avaliação dos dados. Ainda deve ser levado em conta o formato (ou codificação) dos dados que pode conduzir a soluções mais otimizadas e avaliado o compromisso entre a distribuição do processamento no espaço ou no tempo.

Existem algumas propostas e implementações de arquiteturas de convolução uni e bidimensionais. A convolução em uma dimensão é citada uma vez que uma convolução bidimensional pode ser expressa pela soma de  $n$  convoluções unidimensionais, onde  $n$  é o tamanho de uma das dimensões da máscara de convolução. [DAN 84] mostra um apanhado sobre implementações serial e paralela de operadores de convolução, e outros exemplos são encontrados em [TEL 87], [ARA 87], [DOS 87], [KUN 81b], [KUN 81c], [BRI 89] e [SID 89]. [RUE 87] implementa a convolução utilizando coeficientes que são potências de dois e com a movimentação de coeficientes através dos processadores. [DEN 88] descreve o operador de convolução utilizado na tarefa de compressão de imagens, onde o tamanho dos dados é variável e o cálculo não é realizado em tempo real. Outras

propostas variando a dimensão da máscara, quantidade de armazenamento temporário e a movimentação dos dados pode ser visto em [SHU 88], [CAP 87] e [SID 89]. Através destes exemplos e pelas características do algoritmo, vê-se que este se adapta à implementação de forma sistólica [KUN 80a].

O princípio básico da computação sistólica é que o dado seja extensivamente utilizado pelo sistema, visando diminuir a largura da banda necessária para a memória. Alguns trabalhos tutoriais e exemplos de implementação podem ser vistos em [MOO 86], [COM 87].

A propriedade de recorrência da convolução pode ser modelada como processos pipeline, onde cada unidade computacional pode iniciar o processamento de um novo dado antes do resultado completo ser produzido. Em outras palavras, um processador pode iniciar a computação de um novo dado ao mesmo tempo que seu resultado é utilizado por outro processador. Este conceito, aliado ao fato que um único pixel de entrada produz diversos valores utilizados em diferentes cálculos mostra a viabilidade do processamento sistólico da convolução.

Algoritmos implementados em VLSI consomem dois tipos de recursos: tempo e espaço. A decisão de quais recursos utilizar é uma função do que é disponível e do que é necessário para realizar o algoritmo. A seleção do processamento distribuído no tempo ou no espaço depende dos recursos físicos disponíveis (e.g. área em silício) comparados com a complexidade das operações, precisão numérica e velocidade de computação exigida no problema. Assim o conjunto de passos que compõem o algoritmo pode ser implementado basicamente de 4 formas diferentes [DEN 85]:

- 1- Um Processador, arquitetura totalmente serial;
- 2- Vários processadores, com menor número de processadores do que passos de computação. Arquitetura multiplexada;
- 3- Igual número de processadores e operações a realizar: Conjuntos completos;

4- Mais processadores do que operações, paralelizando funções em aplicações de altíssima velocidade - Arquiteturas hiperparalelas.

#### 4.1 Concepção e Mapeamento de Arquiteturas

O processo de concepção de arquiteturas que implementam funções específicas é na verdade a descoberta da função de mapeamento do fluxo de dados e controle do algoritmo nos processadores e na comunicação dos processos na rede de interconexão física [OFF 85]. Alguns elementos foram em particular considerados na concepção da arquitetura CV2D. São eles: Compromisso de computação espaço x tempo, Característica do fluxo de dados, modelamento do problema, manipulação de frentes de onda e escolha de um modelo de arquitetura alvo. Estes envolvem não só as operações aritméticas mas também a comunicação dos sinais de dados e controle. Sem menos importância, o tipo de Interface de comunicação com outros subsistemas deve ser considerada.

Alguns aspectos gerais podem ser colocados no processo de concepção antes do detalhamento da arquitetura:

- a) Identificar como as entradas e saídas influem no projeto do sistema, e.g. a velocidade de variação das entradas e como estas são codificadas e o formato de saída necessário;
- b) Estabelecer o tempo disponível para realizar as operações necessárias sobre os dados. Isto determina se um algoritmo específico será implementado na forma serial ou paralela e, em casos extremos, limitar o tipo de tecnologia a ser utilizado. Da mesma forma, explora-se alguma forma de paralelismo que exista no algoritmo a ser implementado, bem como a granularidade deste (tarefas, funções, dentro de operações aritméticas) [HWA 84];
- c) Decidir como os vários tipos de informação do sistema devem ser codificadas. Algumas destas serão orientadas pela característica dos dados de entrada e saída, a menos que alguma espécie de conversão

seja utilizada. Para dados gerados internamente, a codificação será orientada pela forma que estes serão utilizados;

- d) Estudar as operações necessárias a fim de desenvolver um algoritmo que seja particularmente adaptado ao problema. Uma operação aritmética como multiplicação ou manipulação de dados, pode ser realizada de várias maneiras. Muitas vezes a grande oportunidade de simplificação do projeto de um sistema está no algoritmo, onde cada forma de implementação é considerada um novo algoritmo;
- e) Especificar uma estrutura de sistema composto por subsistemas e suas interconexões, de forma a atingir os requisitos iniciais do problema;
- f) Implementar os subsistemas. Em um sistema complexo, os procedimentos de projeto podem ser iterativamente repetidos até chegar-se ao ponto da implementação física.

#### 4.2 Materialização do Algoritmo

A principal característica dos chamados "arrays sistólicos" é a da computação ser realizada por múltiplos elementos de processamento (PEs), interconectados de maneira regular, a fim de explorar o potencial de pipeline e de multiprocessamento. Uma das maiores vantagens deste tipo de arquitetura é que cada dado de entrada pode ser utilizado várias vezes no mesmo acesso e assim uma alta taxa de saída de dados (*throughput*) é alcançada sem necessitar de grande largura de banda de memória. Outras vantagens são a capacidade de expansão modular, fluxo de dados e controle simples e regular e, normalmente, processadores simples e uniformes.

Os arrays sistólicos são classificados em semi-sistólicos, com comunicação de dados global, e arrays sistólicos puros, sem comunicação global [KUN 80a]. Nos arrays semi-sistólicos os dados são lidos da memória, difundidos e utilizados por um determinado número de PEs, idênticos ou não, concorrentemente. Como o dado lido deve ser distribuído a vários PEs

em cada ciclo, é necessário um barramento global e, eventualmente uma redução de velocidade de processamento a medida que aumenta o número de processadores. De outro lado os arrays sistólicos puros eliminam o uso de barramentos de distribuição e implementam os algoritmos em pipelines estendidos em diferentes direções. Vários dados seguem ao longo de diferentes "pipes" com velocidades iguais ou diferentes, que quando encontram-se são processados. Os PEs operam sincronamente e utilizam um ou mais relógios, de forma que todos os operandos a serem processados por um PE em cada passo computacional devem chegar simultaneamente a este processador. Esta forma de pipeline é característica do **processamento sistólico**.

Um dos problemas mais importantes de projeto no processamento sistólico é o desenvolvimento de uma metodologia de transformação de um algoritmo representado em alguma construção de alto nível em uma arquitetura sistólica, especificada pela temporização dos dados e pela interconexão dos elementos de processamento de forma a satisfazer as necessidades do problema. Muitas técnicas tem sido propostas e a maior parte delas segue um enfoque transformacional, i.e., arquiteturas sistólicas são derivadas através da transformação da descrição original do algoritmo. Estas podem ser caracterizadas pela forma que o algoritmo é descrito, quais os modelos formais utilizados, como são especificadas as arquiteturas sistólicas e que tipo de transformação é utilizada entre as representações. Uma boa revisão destes métodos é dado em [FOR 86] e [LI 85]. A seguir será descrito brevemente o método proposto por Cohen & Johnson [JOH 81], utilizado para derivar a arquitetura básica de convolução uni e bidimensional.

#### 4.2.1 Método de Choen & Johnson

Partindo de uma expressão matemática envolvendo variáveis com subscritos, as quais representam conceitualmente o sequenciamento (ou o caminho dos dados) e controle no tempo (ou no espaço) chega-se a uma estrutura de interconexão entre operadores aritméticos e elementos de memória.

O método inicia pela derivação de uma nova expressão, onde um operador  $Z$  é utilizado para modelar deslocamentos no tempo (elementos de armazenamento) ou deslocamentos no espaço (alocação de seqüências de dados em PEs). Técnicas de manipulação simbólica são utilizadas para transformar a expressão derivada em expressões equivalentes utilizando as propriedades do operador  $Z$  e dos operadores funcionais da expressão.

Seja uma seqüência de elementos  $\{x(i)\}$ , onde  $x(i-1)$  é o elemento que chega a um determinado ponto de rede uma unidade de tempo antes do elemento  $x(i)$ . O operador de atraso (*delay*) é definido por:

$$Z[x(i)] = x(i-1)$$

ou recursivamente

$$Z^n[x(i)] = Z[Z^{n-1}[x(i)]] = x(i-n)$$

Graficamente podemos representar o elemento de delay como na figura 4.1.

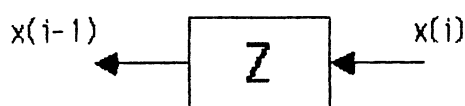


Figura 4.1 Representação Gráfica do Elemento de Delay

Os cálculos definidos pelas expressões podem ser formalmente manipulados para gerar uma rede computacional alternativa, produzindo os mesmos resultados, porém com diferentes características de desempenho. Estas manipulações podem ser resumidas nas seguintes propriedades do operador  $Z$ :

$$Z^n[f(x, y)] = f[Z^n(x), Z^n(y)]$$

$$Z^n(Cx) = CZ^n(x)$$

Um exemplo desta manipulação e implementação destas redes é a convolução unidimensional discreta para  $N$  pontos:

$$y(n) = \sum_{i=0}^{N-1} a(i)x(n-i)$$

que pode ser escrita como:

$$y(n) = \sum_{i=0}^{N-1} a(i)z^i x(n) \quad (a)$$

Uma vez que os coeficientes  $a(n)$  são, neste caso, independentes do tempo, eles são constantes para a variável manipulada e a expressão acima pode ser reescrita:

$$y(n) = \sum_{i=0}^{N-1} z^i [a(i)x(n)] \quad (b)$$

Para cada uma das expressões existe uma rede que a implementa. A equação (a) corresponde a figura 4.2a e 4.2b implementa a equação (b).



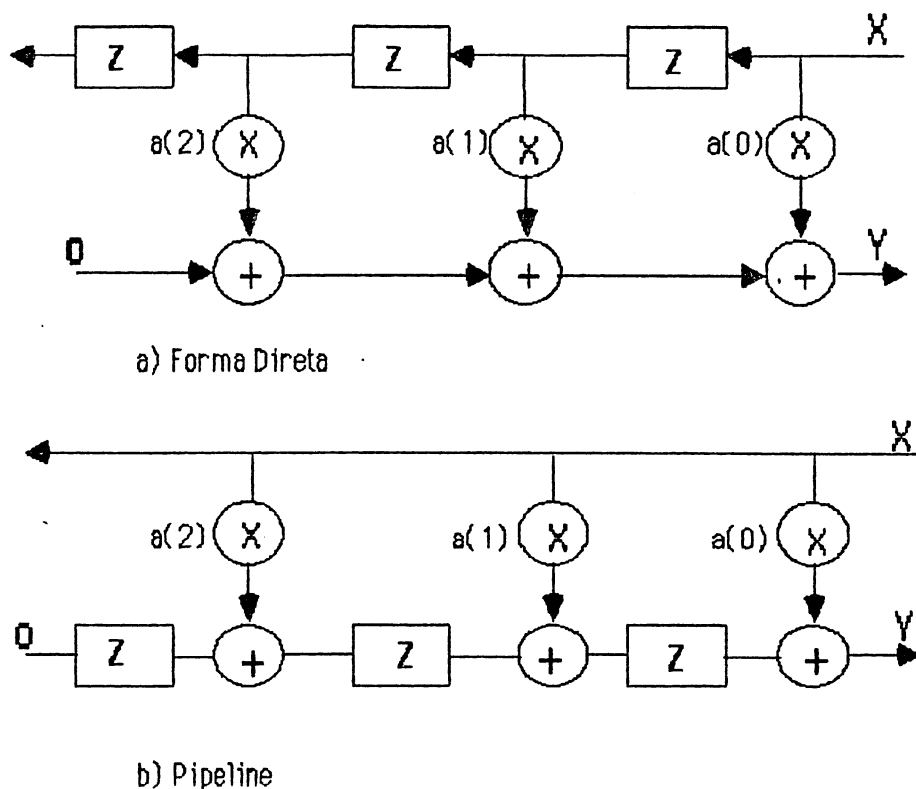


Figura 4.2 Implementação das Redes de Convolução

A implementação pipeline da figura 4.2b pode alcançar maior desempenho, por não haver necessidade do resultado de vários somadores encadeados em um período  $T$ . Outros exemplos que envolvem explicitação dos sinais de controle encontra-se em [JOH 81].

O número e a colocação dos PEs, bem como sua interconexão pode ser derivada das expressões. Os requisitos de temporização e armazenamento são inferidas da colocação dos elementos de atraso, que correspondem ao operador  $Z$ . Este método é aplicável a algoritmos que podem ser descritos através de expressões matemáticas relativamente simples e concisas, como é o caso da convolução. Através deste método podem ainda ser implementadas arquiteturas com características sequenciais e paralelas, otimizando os recursos de hardware para um dado problema. Os sinais de controle e os dados são tratados da mesma forma e

as otimizações são feitas de forma "AD HOC". Um tipo de otimização é alcançada através de técnicas conhecidas como "RETIMMING" a partir de uma arquitetura alvo proposta. [GLA 85] mostra as técnicas de "HOLD UP" e "SLOW DOWN" no caso unidimensional.

A manipulação adequada do modelo escolhido para representar o algoritmo desejado pode levar a diversas alternativas com diferentes características de *throughput*, *delay*, latência e forma de interconexão dos operadores. Fatores como a largura de banda do canal de entrada de dados do sistema podem ser otimizados, aumentando o número de operações sobre um mesmo dado de entrada do sistema.

#### 4.3 Frentes de Onda

A progressão de frentes de onda ("Wavefronts") de sequências de dados através de uma rede computacional é descrita pela aplicação de operadores para ordenar elementos  $x(i, j)$  onde  $i$  e  $j$  são índices de posição e tempo. Estes operadores podem ser utilizados para deslocar, rotacionar ou defasar frentes de onda. A combinação de duas frentes de onda de entrada para produzir uma sequência de dados pode também ser representada neste formalismo. Alguns exemplos de manipulação destas frentes de onda é dada em [JOH 81]. A manipulação adequada do formato das frentes de onda pode levar ao uso de operadores de diferentes velocidades, sem alterar o *throughput* do sistema. Este tratamento pode estar associado ao nível de partição de operadores (no caso de bits) ou das operações (no caso de palavras) de acordo com a velocidade dos operadores disponíveis.

#### 4.4 Fluxo de Dados

Na concepção da arquitetura deve ser considerado se os dados são apresentados continuamente na entrada, na ordem em que são usados para o

cálculo ou se existe a necessidade de alguma manipulação de memória ou armazenamento intermediário para execução do algoritmo. A análise é válida também para os dados de saída, no que diz respeito à utilização de armazenamento intermediário para ordenação dos resultados produzidos, recurso típico utilizado na manipulação de frentes de onda. A continuidade do fluxo de entrada é um fator importante no caso de otimizar a utilização de recursos em pipeline e no uso de operadores ou processos paralelos.

Características especiais dos dados devem ser consideradas na concepção de uma arquitetura dedicada. Por exemplo os limites, ou intervalo, dos valores dos dados de entrada e saída bem como a representação (codificação) destes pode produzir otimizações ou restrições nos operadores aritméticos e de comunicação (barramentos e redes) e portanto reduzir ou aumentar a complexidade do sistema.

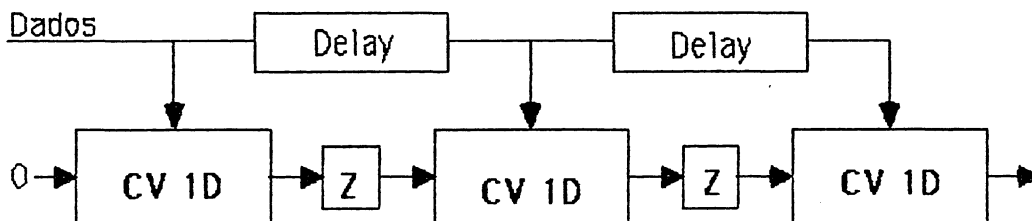
No caso da convolução, o sinal de vídeo, como já discutido, pode ser considerado como um sinal de uma dimensão, variável no tempo. Para realizar o tratamento bidimensional é necessário a utilização de linhas de atraso. A função básica destes elementos é aceitar o valor de um pixel na entrada e colocar na saída o valor do pixel espacialmente diretamente acima, i.e. atrasado de uma linha de vídeo. Representamos este elemento por  $Z^{-L}$  onde  $L$  é o tamanho, número de pixels, da linha. Esta linha de delay pode ser implementada através de um registrador de deslocamento que possua comprimento igual ao número de pixels da linha, o que consome uma quantidade razoável de espaço e potência. Tal função pode ainda ser implementada utilizando uma memória que cumpra a tarefa de armazenamento temporário. Esta memória pode ser interna ao componente que implementa o algoritmo [RUE 87] ou externa [SHU 88]. Se forem armazenados resultados intermediários, o problema continua sendo o adequado gerenciamento de memória.

Como o sinal de vídeo jamais apresenta valores negativos e a multiplicação é realizada por valores que são potências de dois, o projeto do multiplicador a ser utilizado pode ser simplificado significativamente.

#### 4.5 Conclusão

Uma análise de todas as arquiteturas possíveis para um dado problema é inviável. A escolha da estrutura básica a ser manipulada deve levar em conta as características gerais do modelo computacional (data flow, control flow ou data driven) bem como da forma de processamento (sistólico, array, multiprocessador) que mais se adaptem à representação do algoritmo e ao formato dos dados utilizados. Uma vez escolhido basta manipular a arquitetura básica, dados e operadores até chegar em uma solução que satisfaça a especificação, considerando tempo, precisão, área disponível, tecnologia e latência e outros fatores do problema.

A estrutura básica escolhida para implementação apóia-se no fato de que a convolução bidimensional pode ser vista como um somatório de convoluções unidimensionais e através das expressões escritas no item 4.2.1 esta pode ser representada, entre outras formas, como na figura 4.3, explicitando a presença das linhas de atraso. A convolução unidimensional pode ser derivada da mesma forma e está representada na figura 4.2b.



Delay = Intervalo de Linha + Latência das Unidades CV-1D

CV1D = Convolução Unidimensional

Figura 4.3 Implementação da Convolução Bidimensional

A estrutura acima foi adotada para implementação e será chamada de "**Arquitetura CV2D**" ou simplesmente **CV2D**, descrita em detalhe no próximo capítulo.

## 5 CV2D

Este capítulo dedica-se a descrever a arquitetura do processador de convolução CV2D. Primeiramente é dada uma visão resumida das principais características e soluções de projeto adotadas. O restante do capítulo dedica-se a descrição funcional e operacional do circuito, bem como a descrição dos blocos interno que compõem o CV2D.

### 5.1 Descrição Geral

O algoritmo de convolução bidimensional com núcleo de dimensão  $3 \times 3$ , no caso discreto pode ser representado pela equação abaixo, incluindo o índice de normalização. Os índices  $m$  e  $n$  são as dimensões do quadro processado, ditos  $m$  colunas e  $n$  linhas.  $I(x,y)$  é a imagem a ser processada,  $O(x,y)$  é o resultado ou a imagem processada e  $H(i,j)$  é chamada de máscara ou núcleo da convolução. Este núcleo pode ser descrito através de uma matriz, representada abaixo.

$$O(x, y) = \frac{1}{2^n} \sum_{i=0}^2 \sum_{j=0}^2 I_{x+i-1, y+j-1} \times h_{i,j} \left| \begin{array}{l} 0 \leq x \leq m \\ 0 \leq y \leq n \end{array} \right.$$

If (SAT=1 &  $O(x,y) < 0$ ) then  $O(x,y)=0$

$$h_{i,j} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix}$$

A partir da equação, pode-se verificar a influência dos pixels vizinhos no atual pixel produzido e observa-se claramente que são utilizados dados da linha anterior e posterior à atual linha de saída. Esta condição leva à necessidade de um sistema de armazenamento temporário de linhas, até que cada dado seja utilizado em todas as operações necessárias. Este sistema será chamado de "*Line Delay System*" e tem como função armazenar o pixel atualmente recebido e fornecer o valor do pixel espacialmente acima, ou seja, atrasado de uma linha. Durante o processamento dos limites, o CV2D considera toda vizinhança do quadro como pixels de valor zero (sem luminosidade). A convolução e o sistema de line delay podem ser representados por duas funções **f** e **g** e a função de transferência pode ser dada pela composição de **f** e **g**.

Parte significativa dos algoritmos de processamento de imagem utilizam máscaras ou núcleos de convolução com valores que são potências de dois com sinal negativo ou positivo. Exemplos podem ser vistos nos processos de *smoothing*, *sharpening*, processamento do Laplaciano e Gradiente. Desta forma os coeficientes utilizados pertencem ao conjunto  $\pm[0, 1, 2, 4, 8, 16]$ . De acordo com o valor e sinal dos coeficientes

coeficientes utilizados, o cálculo da convolução pode resultar em uma imagem com um nível DC acima do brilho da imagem original (sempre que o módulo da soma dos coeficientes for maior que a unidade). O CV2D permite que seja programada a faixa de brilho dos dados produzidos através do deslocamento dos bits do resultado. O número de bits a deslocar é programável no intervalo [0,16] e realizado com extensão de sinal, permitindo a utilização de resultados negativos. Ainda no controle de saída, pode-se modificar o tratamento de resultados negativos, transformando-os em pixels de valor zero. Esta opção pode ser utilizada sempre que os resultados produzidos forem destinados a exibição imediata.

O processamento é realizado durante a varredura de uma linha, ou seja, cada linha é processada durante o tempo da sua amostragem. Isto permite que o componente seja utilizado em aplicações de tempo real ou como acelerador em sistemas que utilizem o cálculo de convolução.

Os dados de entrada são elementos de imagem quantizados em 8 bits/pixel e, como não existem pixels de brilho negativo, pertencentes ao intervalo [0,255] com até 512 pixels/linha. Os resultados são produzidos em 16 bits representados em complemento de dois.

Devido a limitações de área, o sistema responsável pelo armazenamento temporário de linhas é implementado com uma memória externa, comandada através de um sinal fornecido para ativação deste sistema.

Além da programabilidade dos 9 coeficientes de convolução, o tratamento de números negativos (ajuste em zero) também é selecionado pelo usuário, assim como a quantidade de bits deslocados no ajuste de resultados e o número de pixels nas linhas processadas. Esta última característica permite uma maior flexibilidade na escolha da região da imagem a ser processada, de acordo com o objetivo do tratamento realizado.

O modelo do ambiente de funcionamento é mostrado na figura 5.1, onde existe um produtor e um consumidor de dados, um sistema de line delay e um sistema de programação de parâmetros. É claro que estes elementos podem ser aglutinados ou expandidos em um ou mais componentes ou sistemas físicos, de acordo com a disponibilidade ou

aplicação em questão. A implementação de maneira sistólica resulta em uma arquitetura regular, adequada a realização em silício e com alto desempenho. O coração da estrutura é uma cadeia de 9 elementos de processamento, responsável pelo cálculo da convolução.

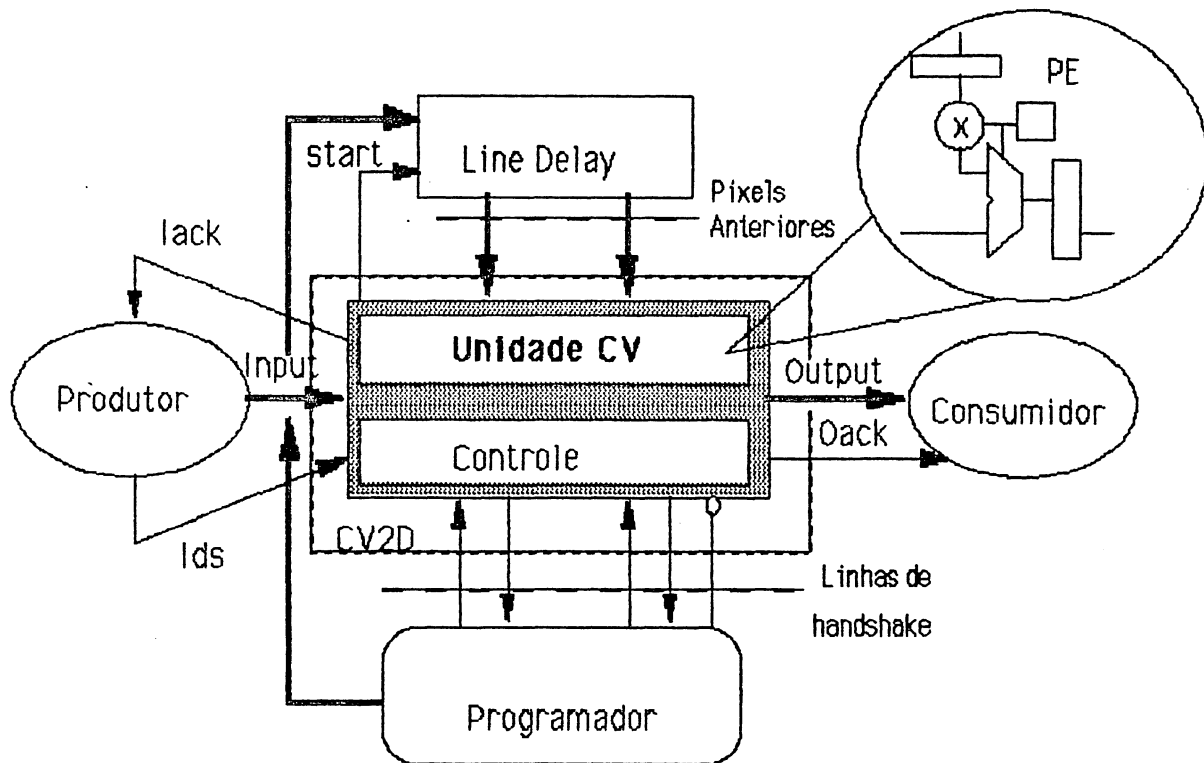


Figura 5.1 Modelo do Ambiente de Funcionamento do CV2D

## 5.2 CV2D - Descrição Funcional

Na figura 5.2 é dado o diagrama em blocos das unidades funcionais do chip e a seguir a descrição funcional dos pinos de entrada e saída. Cada bloco representa uma das funções executadas pelo sistema, como segue:

- \* Convolução - Realiza o cálculo da convolução bidimensional;
- \* Deslocamento - Processa o deslocamento dos dados de saída, com extensão de sinal;



- \* Programação - Controla a carga de parâmetros a serem armazenados nos registradores internos. Este possui linhas específicas para controle da sequência de carga dos valores;
- \* Sincronização - Realiza a sincronização entre os sinais de entrada e o processo de convolução.

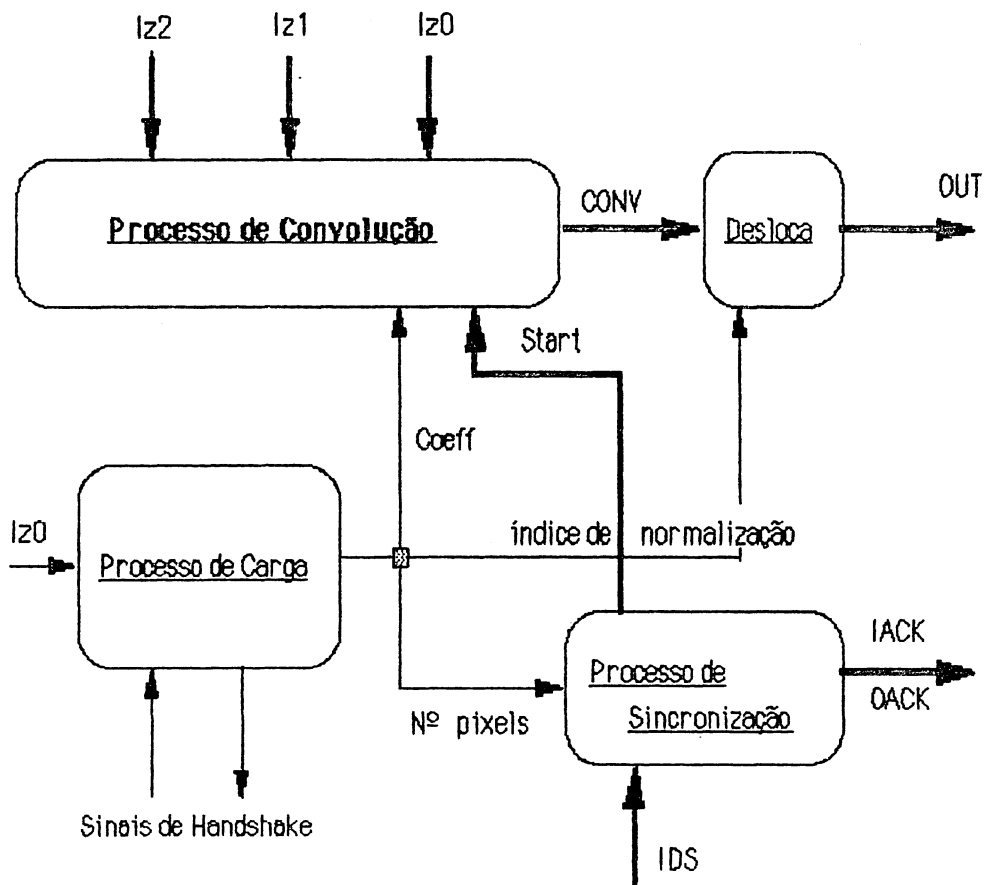


Figura 5.2 Diagrama em Blocos

O CV2D possui 54 pinos, realizando as funções descritas a seguir:

24 dados de entrada

16 dados de saída

10 sinais de controle

4 pinos de alimentação

#### Controle Geral: RST, $\overline{OE}$

RST (input) - Ativo em nível lógico um e coloca o sistema em estado de reset. Neste estado são zerados todos os registradores internos, inicializadas todas as unidades de controle e mantidos os pinos de dados de saída em alta impedância. Este sinal deve permanecer ativo no mínimo durante 15 $\mu$ s.

$\overline{OE}$  (input) - Ativo em nível lógico zero, controlando o estado dos pinos de saída de dados. Quando inativo, a saída é levada assincronamente ao estado alta impedância.

#### Entrada de Dados: IZ0[7:0] IZ1[7:0] IZ2[7:0]

Durante o processamento da convolução IZ0 é a entrada dos pixels da linha atual e serve como porta de entrada de parâmetros quando em modo de programação. IZ1 e IZ2 são os valores dos pixels fornecidos pelo sistema de line delay, respectivamente atrasados de 1 e 2 linhas. No modo de programação estes valores não são considerados.

#### Saída de Dados: OUT[15:0]

Fornecem o resultado da convolução após ajuste da faixa de exibição. Estes pinos quando em estado de RESET, ficam em alta impedância bem como quando desativado o sinal OE.

#### Controle de Sincronização: IDS, IACK, OACK

São sinais que fazem o protocolo de sincronização com os sistemas produtor e consumidor de pixels, e controlam a ativação do processo de convolução.

IDS - (Input Data Strobe) - Ativo em 'um'. Na primeira transição ativa indica que novos dados estão disponíveis para processamento e enquanto ativo indica que os dados nas entradas IZ são válidos.

IACK (Input Acknowledge) - Ativo em 'um'. Indica que o sinal IDS foi recebido e que o processo de convolução está ativo. Este mesmo sinal informa ao sistema de Line Delay que novos dados devem ser aceitos e armazenados a cada transição positiva do relógio. Os dados de entrada são considerados válidos somente após o sinal IACK estar ativo.

OACK (Output Acknowledge) - Ativo em 'um'. Indica que os dados de saída são válidos para utilização. Este sinal é utilizado a fim de sincronizar o sistema consumidor com o tempo de latência do pipeline interno. Este sinal é apenas função dos sinais IACK e da latência do pipe, não importando o estado do sinal de  $\overline{OE}$ .

A retirada do sinal IDS indica que os dados não serão mais válidos para processamento na próxima transição de relógio e deve permanecer inativo no mínimo durante 10 ciclos, respeitando o tempo de bloco, ou seja o tempo de inicialização da cadeia de cálculo para a próxima linha.

A seguir é mostrada a relação destes sinais através dos diagramas da figura 5.3.

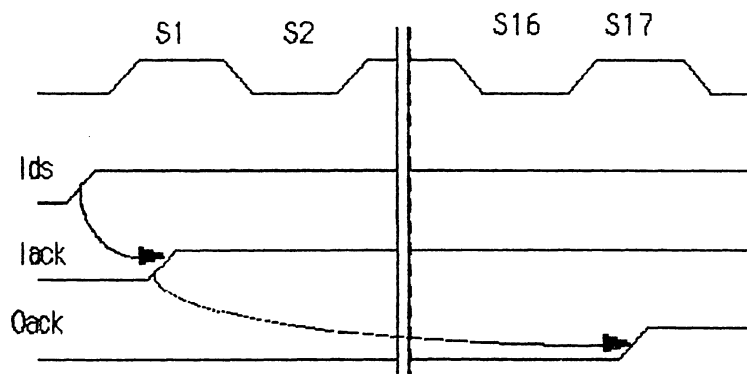


Figura 5.3 Forma de Onda de Sincronização

Controle de Programação:  $\overline{\text{LOAD}}$ ,  $\overline{\text{DONE}}$ ,  $\overline{\text{DS}}$ ,  $\overline{\text{DTACK}}$

Estes sinais controlam o protocolo assíncrono durante a programação de parâmetros. O protocolo faz com que a velocidade de carga seja determinada pelo programador e não pelo CV2D.

$\overline{\text{LOAD}}$  - (entrada) - Ativo em 'zero'. Ativa o processo de carga de parâmetros, preparando o caminho e sequência de carga dos registradores de programação.

$\overline{\text{DS}}$  - (Data Strobe - entrada) - Ativo em 'zero'. Informa, a cada novo dado, que o valor da porta IZO é válido para utilização.

$\overline{\text{DTACK}}$  - (Data acknowledge - saída) - Ativo em zero. Após efetuada a carga, o sistema responde ativando o sinal de DTACK, indicando que o sinal DS pode ser desativado e indicando que pode ser iniciado o ciclo de carga de um novo dado.

$\overline{\text{DONE}}$  (saída) - Ativo em 'zero'. Após realizada a carga dos 11 parâmetros, o sinal  $\overline{\text{DONE}}$  é ativado, indicando que a carga está completa e o processo de programação será desativado quando retirado o sinal  $\overline{\text{LOAD}}$ .

A relação dos sinais é mostrada nas formas de onda da figura 5.4.

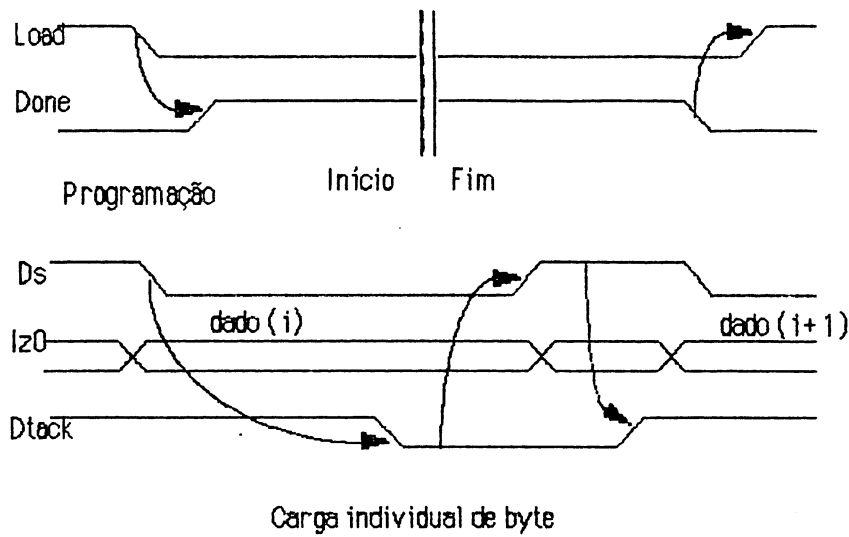


Figura 5.4 Formas de Onda de Programação

### 5.3 Descrição Operacional

Existem 3 estados possíveis do sistema CV2D: RESET, PROGRAMAÇÃO de parâmetros e computação da CONVOLUÇÃO. Abaixo são descritos a seqüência e formato dos dados utilizados pelo sistema em cada um dos estados.

RESET- Uma vez ativo o sinal de RESET, o CV2D coloca todos os registradores internos em zero e as linhas de dados de saída em alta impedância, não respondendo aos demais sinais de controle.

PROGRAMAÇÃO - Ao ativar o sinal  $\overline{\text{LOAD}}$  o CV2D entra no modo de programação, onde são carregados sequencialmente os 11 bytes correspondentes aos coeficientes de convolução, índice de deslocamento e número de pixels por linha. A ordem de programação é dada na tabela da figura 5.5. O formato dos parâmetros é mostrado na figura 5.6.

Byte	Parametro
# 01	h(00)
# 02	h(01)
# 03	h(02)
# 04	h(10)
# 05	h(11)
# 06	h(12)
# 07	h(20)
# 08	h(21)
# 09	h(22)
# 10	Pixel/Line
# 11	Displacement Index

Figura 5.5 Parâmetros de Programação

a) Para os valores de h somente um bit no campo CF pode estar ativo em um dado momento. A especificação de coeficiente zero tem precedência sobre o coeficiente selecionado. O sinal do coeficiente é dado pelo bit mais significativo, onde o valor '1' indica um valor negativo.

b) O índice de deslocamento, indica o número de bits a serem deslocados no resultado final. Um bit adicional controla o tratamento de números negativos, truncando em zero (valor um) ou estendendo o sinal para os bits deslocados (valor zero). O MSbit deste parâmetro contém o valor do bit menos significativo do número de pixels/linha.

c) O parâmetro pixels/linha consiste em 9 bits. Os 8 MSbits são carregados no parâmetro #11 e o LSB é programado junto ao byte #10.

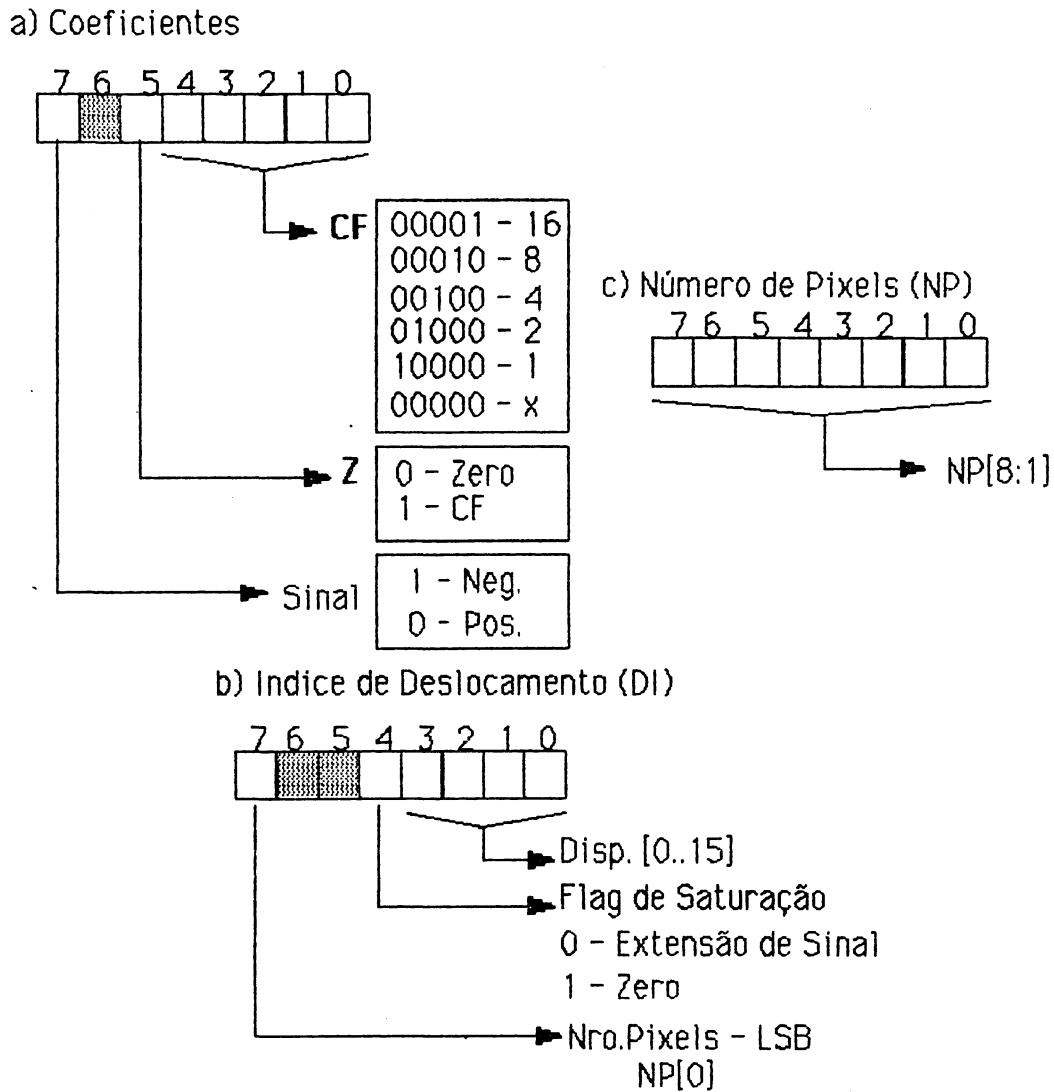


Figura 5.6 Formato dos Parâmetros de Programação

CONVOLUÇÃO - Quando ativo o sinal IDS o modo de convolução é selecionado e a cada novo ciclo é produzido um novo resultado em 16 bits deslocado à direita de acordo com o índice de deslocamento programado. O intervalo mínimo entre duas linhas consecutivas é de 10 ciclos, permitindo a inicialização da cadeia de cálculo da convolução. A tabela da figura 5.7 mostra um exemplo do resultado de saída antes e após o deslocamento, e podem ser vistos os efeitos de extensão de sinal e truncamento.

Índice Desl.	Convolução	Resultado
00000	010010110110110	010010110110110
00001	010010110110110	001001011011011
10011	011101101001000	000011101101001
10011	101110111010011	000000000000000

Figura 5.7 Efeito do Índice de Deslocamento

#### 5.4 Particionamento

O sistema é dividido logicamente em 4 processos que definem a arquitetura interna do chip. Cada processo corresponde a um bloco funcional descrito acima.

- \* Processo de CONVOLUÇÃO - Realiza o cálculo da convolução. Recebe 3 linhas de imagem e produz uma nova linha, resultado da convolução dos dados de entrada. Sua atividade é controlada por um sinal de sincronização.
- \* Processo de DESLOCAMENTO - Executa o ajuste dos resultados, realizando o deslocamento à direita com extensão de sinal ou truncamento de valores negativos
- \* Processo de PROGRAMAÇÃO - Controla a carga de parâmetros do sistema. Este processo implementa um protocolo assíncrono com o sistema de programação (LOAD, DS, DTACK, DONE).
- \* Processo de SINCRONIZAÇÃO - Responsável pela ativação do processo de convolução no início e final de cada linha, permitindo o tratamento de bordas. Este gera também sinais que permitem sincronização externa (IDS, IACK, OACK).

A implementação de cada processo foi realizada em 2 passos. Primeiro determina-se a arquitetura alvo com sinais de entrada e saída, e



são estabelecidos novos sinais na interface quando necessário. No passo seguinte são estabelecidos subprocessos, na forma de ciclos e tarefas, e também a implementação de cada um destes. Os sinais que comunicam as diferentes unidades podem ser classificados em duas grandes classes: Estáticos e Dinâmicos. Os dinâmicos são aqueles que variam na taxa de pixel e desta forma determinam o caminho crítico do sistema. Entre estes estão as linhas de dados, resultados e as linhas de ativação da cadeia de cálculo. Os estáticos permanecem estáveis durante a convolução, quais sejam todos os coeficientes, parâmetros e linhas de programação.

#### 5.4.1 Unidade de Convolução

Recebe 3 linhas de dados e produz uma nova linha, resultado da computação da convolução, e tem sua atividade controlada pelas linhas de sincronização.

Implementada de maneira sistólica, explora a característica de recorrência do algoritmo. Devido a topologia de conexão dos elementos de processamento são necessárias duas unidades de ajuste de fase (*skew*), que garantem a sequência correta dos dados de entrada nos processadores. A figura 5.8 mostra a arquitetura da unidade de convolução. O **MUX** colocado na entrada de dados permite a entrada de pixels com brilho nulo no processamento das bordas, através da introdução de zeros na cadeia. Os  $h(i,j)$  representam os coeficientes utilizados em cada **PE**.

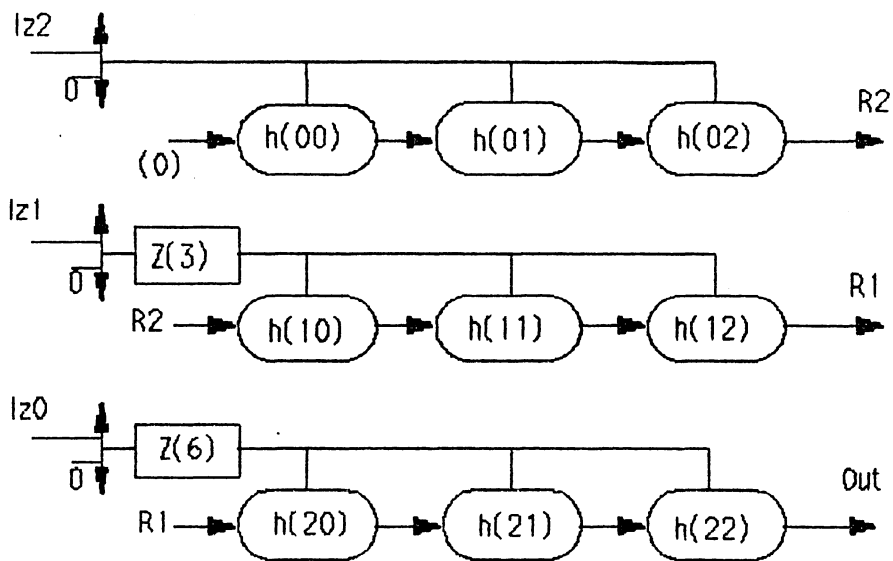


Figura 5.8 Unidade de Convolução

A unidade comunica-se através de 4 barramentos e um sinal de controle: três barramentos de 8 bits de dados, um barramento de 16 bits de resultado e um sinal de controle de entrada de dados.

#### 5.4.1.1. Elementos de Processamento

Os elementos de processamento (PE) realizam a função  $y = h * x + b$ .  $x$  é o dado de entrada,  $b$  recebe o resultado do estágio anterior e  $h$  é o coeficiente de convolução associado ao PE. Cada PE é composto de 16 módulos de 1 bit (*bit slice*) que realizam a função descrita em aritmética de ponto fixo. Dada a característica dos coeficientes serem potências de dois, a operação de multiplicação reduz-se a um deslocamento realizado por um *barrel shifter* na entrada de dados. O sinal do coeficiente é manipulado diretamente pela ALU que realiza a soma. Sendo os dados de entrada sempre positivos, duas operações são possíveis:

$$y = hx + b \text{ para coeficientes positivos}$$

$$y = hx - b \text{ caso contrário;}$$

com resultados codificados em complemento de dois. O latch de entrada garante que os dados estarão estáveis para cálculo no mínimo durante uma fase completa do relógio. A arquitetura do bitSlice é dada na figura 5.9.

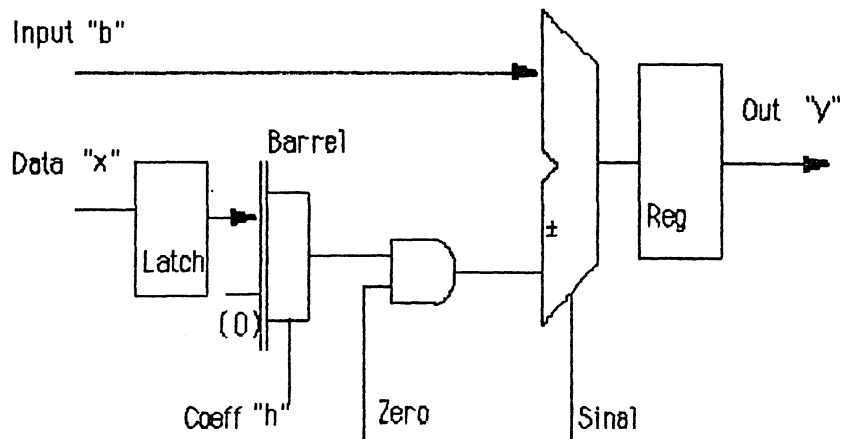


Figura 5.9 Arquitetura do Elemento de Processamento

O barramento interno de 16 bits é suficiente para prevenir a ocorrência de overflow durante a operação. ( $\text{bus} \geq (\text{length}(h*x) + \ln n) + 1$ , onde  $n$  é o número de PEs).

#### 5.4.1.2 Sincronização

O tempo de pipeline, ou seja, intervalo entre dois dados consecutivos de entrada/saída é unitário (1 ciclo). O tempo de bloco, definido como tempo mínimo entre duas linhas consecutivas é igual a 10 ciclos, tempo que permite propagar zeros a todos acumuladores da cadeia. A latência do pipe, dada pelo número de ciclos desde o primeiro dado válido para computação até a produção dos resultados, é de 8 ciclos para cada linha. Enquanto não ativada a unidade, esta processa pixels de valor nulo.

#### 5.4.1.3 Unidades de Ajuste de Fase

Destinadas a ajustar a temporização dos dados de entrada ao tempo de utilização pelos processadores. São compostas de 8 bitSlice, cada um formado pelo número de registradores necessários para o *skew* requerido. Existem duas unidades com skew de 3 e outra com 6 períodos respectivamente, que fornecem a frente de onda adequada ao conjunto de processadores.

#### 5.4.2 Unidade de Deslocamento

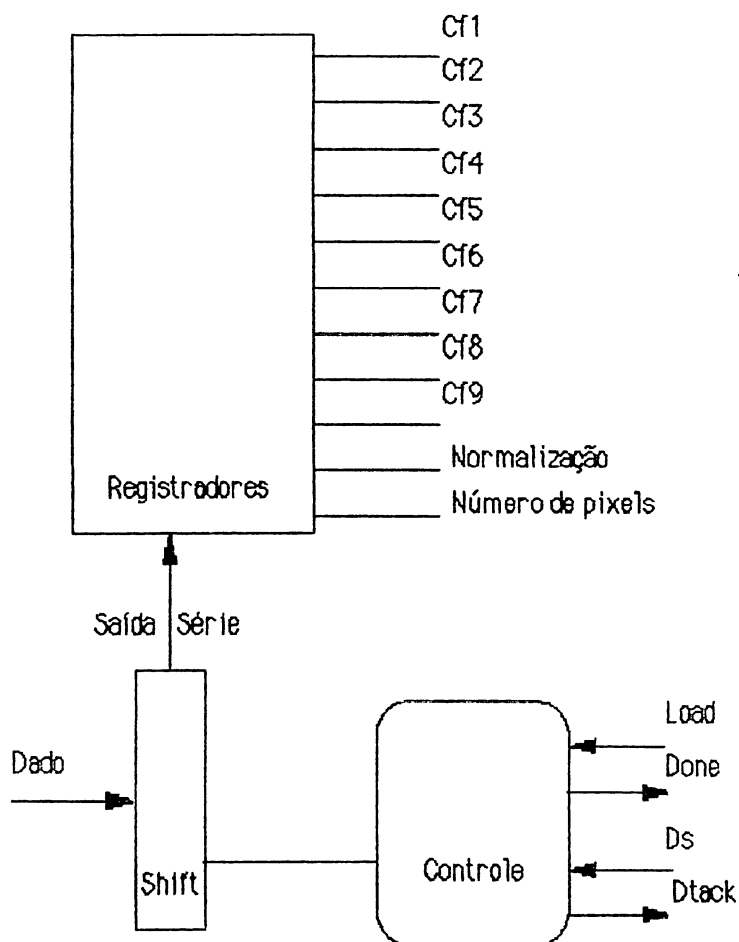
É implementada através de um *barrel shifter* com uma lógica de saturação em zero para resultados negativos. O controle é dado pela decodificação do índice de deslocamento, que gera os bits de comando do *barrel*. A ativação da lógica de saturação é determinada de acordo com o bit mais significativo dos dados de entrada e pelo bit [4] do índice de deslocamento. A interface é composta apenas por 2 barramentos de 16 bits (entrada e saída de dados) e pelo índice de deslocamento.

#### 5.4.3 Unidade de Programação

A arquitetura da interface de programação é dada na figura 5.10. A máquina de estados implementa o protocolo com o ambiente externo ao chip. O núcleo desta máquina é um PLA dinâmico de 13 entradas, 11 saídas e 22 mInterms. Três registradores de estado fazem o sequenciamento e um contador fornece o bit escrito e o número do byte ao qual pertence.

A leitura dos parâmetros é realizada na forma de palavra paralela e escrita serialmente. Desta forma os registradores que armazenam os

parâmetros são encadeados, tornando o banco de registradores um registrador deslocador de 88 bits. Esta característica possibilita que apenas uma linha de controle percorra todos os registradores, o que reduz o número de ligações e portanto área em silício. Este banco de registradores é formado por 3 registradores de 24 bits (coeficientes), um registrador de 9 bits (pixel/linha) e outro de 6 bits (índice de deslocamento). Cada registrador fica próximo a unidade que utiliza os dados, a fim de reduzir o comprimento das ligações.



UFRGS/CPD  
BIBLIOTECA

Figura 5.10 Unidade de Programação

A interface é composta pelas linhas de protocolo (LOAD, DS, DTACK, DONE), um barramento de entrada de dados, um barramento unifilar

de dados de saída e uma linha de controle para habilitação de carga dos registradores.

#### 5.4.4 Unidade de Sincronização

O núcleo desta unidade é uma máquina de estados e um registrador de deslocamento que conta os períodos de latência do pipe. A unidade controla a entrada de dados nos processadores de convolução, habilitados a partir do sinal IDS e desabilitados pela retirada deste ou ao ser atingido o número de pixels/linha estabelecido. A figura 5.11 mostra a arquitetura da unidade com sua máquina de controle, registrador deslocador e um contador do número de pixels processados. Sua interface é composta pelos sinais de sincronização externa (IDS, IACK, OACK), e fornece o sinal de controle de entrada de dados para a máquina de convolução.

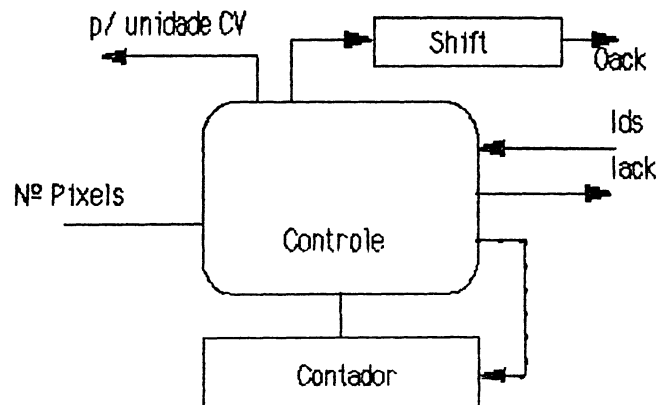


Figura 5.11 Unidade de Sincronização

#### 5.5 Detalhes

Alguns pontos do circuito possuem características peculiares que definem seu comportamento externo. Entre estes detalhes estão a colocação da cadeia de processadores no estado inicial (*RESET*),

distribuição dos coeficientes ao longo do circuito, o caminho crítico dos dados, a distribuição do sinal de relógio e o atraso de saída dos dados.

a) Inicialização da Cadeia de Processadores:

A inicialização dos registradores de cada elemento de processamento é realizada introduzindo dados de valor zero em cada um dos PEs. A seleção do valor zero é realizada através de um multiplexador colocado na entrada das unidades de *skew*, em cada linha de processadores. O sinal de seleção é gerado pela máquina de controle de sincronização. Este inibe a entrada de dados de acordo com o estado do sinal *IDS* ou quando o limite de pixels/linha é alcançado.

No caso de imagens em tempo real, o período de retraço da câmera é suficiente para inicialização de todos os PEs, o que evita a colocação de uma linha de *reset* que atinja todos os processadores.

b) Formato e Distribuição dos Coeficientes:

O formato dos coeficientes foi escolhido de tal forma a corresponder diretamente aos sinais de controle necessários aos PEs. Esta decisão elimina a necessidade de decodificadores e, conseqüentemente, reduz a área do circuito. A distribuição dos coeficientes é realizada serialmente, o que diminui o número de linhas de interconexão. Os registradores que armazenam os coeficientes são colocados próximos aos PEs, sendo necessário apenas as linhas de distribuição serial e uma linha de habilitação de carga.

c) Caminho Crítico:

Um ponto que determina a velocidade do sistema refere-se a ativação da unidade de convolução: Após a ativação de *IDS*, a unidade de sincronização deve habilitar a entrada da cadeia de convolução em tempo limitado, a fim que os dados estejam estáveis antes da próxima fase do relógio (quando fecha o *LATCH* de entrada dos PEs). A figura 5.12 mostra o caminho crítico e a relação temporal do sinais envolvidos.

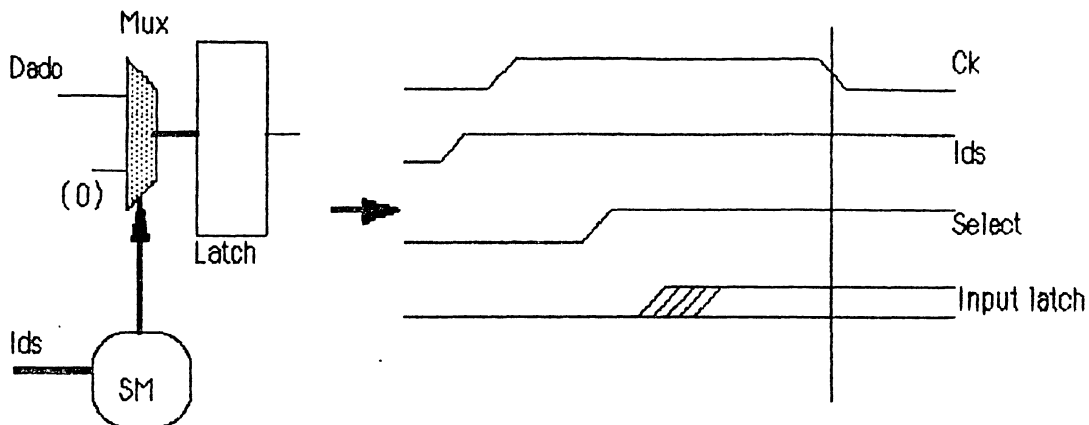


Figura 5.12 Caminho Crítico

## d) Atraso de Saída:

Após deixar o último PE, os dados devem passar através da unidade de deslocamento e lógica de saturação, ambas combinacionais. A lógica de saturação necessita a informação de sinal do resultado para sua atuação, aumentando o atraso no caminho dos resultados. Esta condição não limita a velocidade da máquina, porém a relação temporal entre o dado gerado e o relógio externo é fundamental para a correta utilização dos resultados. A fim de reduzir este efeito poder-se-ia colocar um registrador na saída de dados, diminuindo o atraso dos resultados em relação ao relógio externo. No entanto esta solução aumenta em um ciclo a latência do pipeline.

Existe ainda a defasagem do relógio externo em relação a ativação efetiva do circuito. Esta diferença é consequência da rede de distribuição de relógio, cuja função é fazer com que o sinal de relógio chegue praticamente ao mesmo tempo em todos os componentes do circuito e sempre após os dados estarem estáveis [GLA 85]. Esta rede permite que o circuito funcione sincronamente trocando dados através de seus diversos blocos.



## 6 IMPLEMENTAÇÃO

Segundo Ruetz [RUE 87] existem duas técnicas para um rápido desenvolvimento de circuitos integrados dedicados. A primeira é o projeto hierárquico e a outra é a utilização de ferramentas para produzir hardware a partir do nível mais baixo da hierarquia utilizada.

A hierarquia no projeto foi mostrada na descrição da arquitetura (capítulo 5), onde o circuito foi dividido em blocos funcionais e cada bloco em pequenos elementos que executam uma tarefa bem definida junto às unidades de controle. No diagrama esquemático do circuito está o nível mais baixo da hierarquia, ou seja as unidades funcionais que são implementadas diretamente sobre silício.

O CV2D foi concebido utilizando uma estação de trabalho composta de uma *Micro-Vax Station GPX* com software da *VLSI Technology (VTI)*. Durante o projeto foram utilizadas várias ferramentas que podem ser agrupadas em quatro classes: Compilação, Montagem, Edição e Verificação. As ferramentas de edição e verificação são bem conhecidas e citadas:

- Editor de Esquemático e Ícones
- Editor de Layout
- Simulador Elétrico (SPICE)
- Simulador MIX-MODE
- Verificador de "Netlist"
- Extrator de "Netlist"
- Comparador de "Netlist"
- Verificador de regras de projeto (DRC)

Este capítulo dedica-se à descrição da utilização das ferramentas de compilação e montagem utilizadas na concepção do circuito. As ferramentas são denominadas: Data Path Compiler (DPC), Logic Synthesizer (LS) e Chip Compiler (CC). Uma visão geral da sequência de teste

é dada no final deste capítulo e a temporização e distribuição de PADs no anexo 3.

### 6.1 Síntese do Fluxo de Dados

A síntese do fluxo de dados é realizada com auxílio da ferramenta chamada "Data Path Compiler" (DPC). O DPC é capaz de montar, a partir de uma descrição esquemática, um *bit-sliced data path* composto por células pré definidas e organizadas de acordo com a especificação do usuário. O projetista especifica a organização em termos de um determinado número de *slices* e a forma como estes são construídos. Cada slice é a composição de blocos funcionais básicos, tais como somadores, registradores e portas. O roteamento e distribuição das linhas de alimentação são manipulados pelo DPC a partir de um conjunto de parâmetros fornecidos pelo projetista. Junto ao layout são fornecidos a posição dos conectores, BB (*Bounding Box*), um modelo para simulação e um modelo a ser utilizado em esquemáticos de nível hierárquico superior. Esta ferramenta foi extensamente utilizada para geração das unidades funcionais do circuito, que são citadas abaixo, junto com as suas principais características. A área ocupada é dada para uma tecnologia de duplo poço (*twin tub*), canal de 2.0 $\mu$ .

#### 6.1.1 Cadeia de Processadores

O DPC gera layouts que crescem em comprimento de acordo com o número de unidades funcionais e na coordenada Y com a largura da palavra utilizada. Este layout é composto pela unidade funcional descrita, um conjunto de buffers para os sinais de controle e clock e pela rede de alimentação do DP. Os buffers de controle e clock objetivam diminuir a capacitância destas linhas para o restante do circuito exterior, bem como equalizar a distribuição do sinal de relógio no interior do DP. A planta-baixa (*floor plan*) tem a forma da figura 6.1 e os barramentos de VDD, VSS e linhas de reforço de alimentação podem ser determinadas através de

parâmetros ou estimados pelo compilador a partir da frequência de operação.

Para cada bit não existente no DP, o que é possível já que é permitido diferentes largura de palavra no mesmo bloco, o DPC cria "vazios" na estrutura. Nestes buracos são instaladas capacitâncias que representam a carga de uma célula do DP e também as linhas que permitem a propagação dos sinais de controle aos outros *slices*.

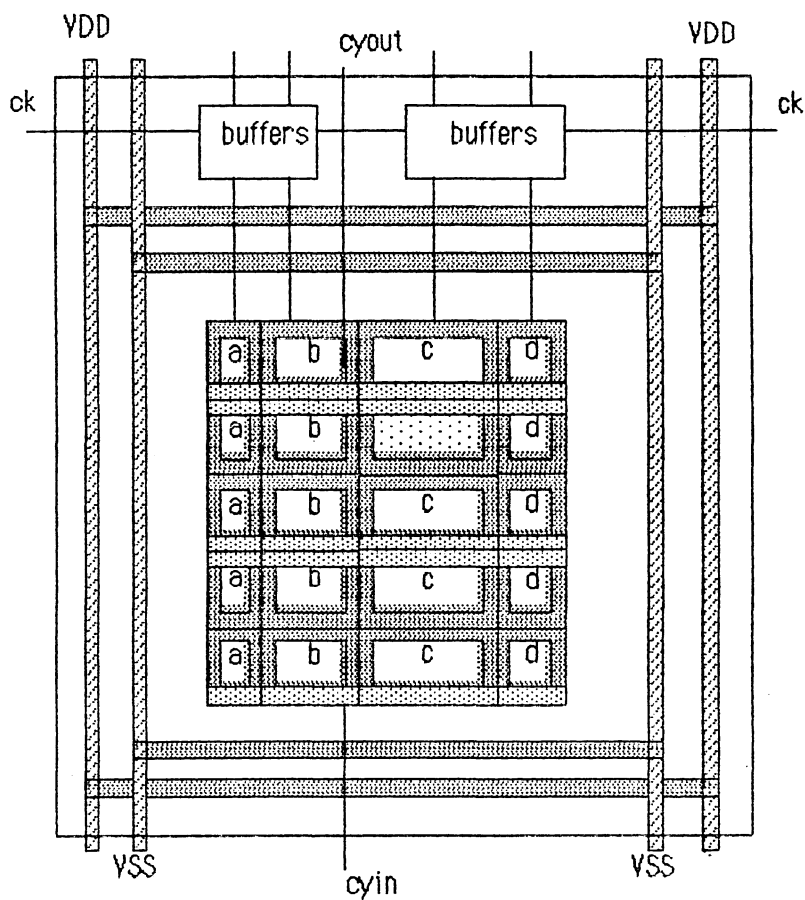


Figura 6.1 Layout gerado pelo DPC

A cadeia de processadores foi dividida em 3 partes. Cada uma destas, chamadas de "*ROW processor*", contém os processadores que compartilham os mesmos dados de entrada e a respectiva unidade de

*skew* . Cada PE (Processing Element) é formado por 16 bit slices e este instanciado três vezes na mesma especificação.

Área ocupada (*Row Processor*):  $3205.5 \times 2527 = 8.1 \times 10^6 \mu^2$

### 6.1.2 Registradores de Deslocamento

Os registradores de deslocamento da unidade de programação, onde são armazenados os parâmetros carregados, estão implementados com o DPC e, a fim de serem posicionados próximos às unidades que os utilizam, foram divididos em 3 registradores de 24 bits e 3 registradores de 9, 8 e 7 bits respectivamente. Cada deslocador é especificado para o compilador como um *slice* composto de apenas uma unidade funcional (flip flop), variando apenas a forma de conexão para cada instância da unidade.

Área ocupada (shift 24 bits):  $178.5 \times 2779 = 4.96 \times 10^6 \mu^2$

### 6.1.3 Unidade de Deslocamento

Esta unidade contém um *barrel shifter* e uma porta para controle de ajuste em zero para cada bit de dado. São especificadas para o DPC as ligações comuns a mais que um slice, e.g. o sinal de controle a todos os bits da unidade.

Área ocupada:  $594 \times 1964 = 1.16 \times 10^6 \mu^2$

## 6.2 Síntese das Máquinas de Controle

As máquinas de estados que compõem o controle são geradas a partir da ferramenta chamada Logic Syntetizer ou State Machine Compiler. A parte combinacional das máquinas pode utilizar tanto lógica aleatória como PLAs. O projetista especifica o comportamento dos blocos,

como PLAs. O projetista especifica o comportamento dos blocos, combinacionais ou sequenciais, através de uma linguagem textual. Nesta descrição podem figurar equações booleanas, transição de estados, limitações de tempo de resposta e outros aspectos físicos como carga e capacidade de drivers. A ferramenta fornece uma lista dos elementos que implementam as funções (chamada de *netlist*) e/ou o código do PLA que realiza a função. A *netlist* gerada descreve uma arquitetura fixa, que define uma máquina segundo o modelo de MEALY. As saídas variam conforme as entradas e estado atual e a transição de estado se dá de forma síncrona. É possível ainda colocar latches nos sinais de entrada e saída e FFs RS nos sinais de saída, sincronizando-os com o relógio. A arquitetura é mostrada na figura 6.2 e utilizando um PLA o projetista deve colocar os registradores de estado manualmente, caso estes existirem.

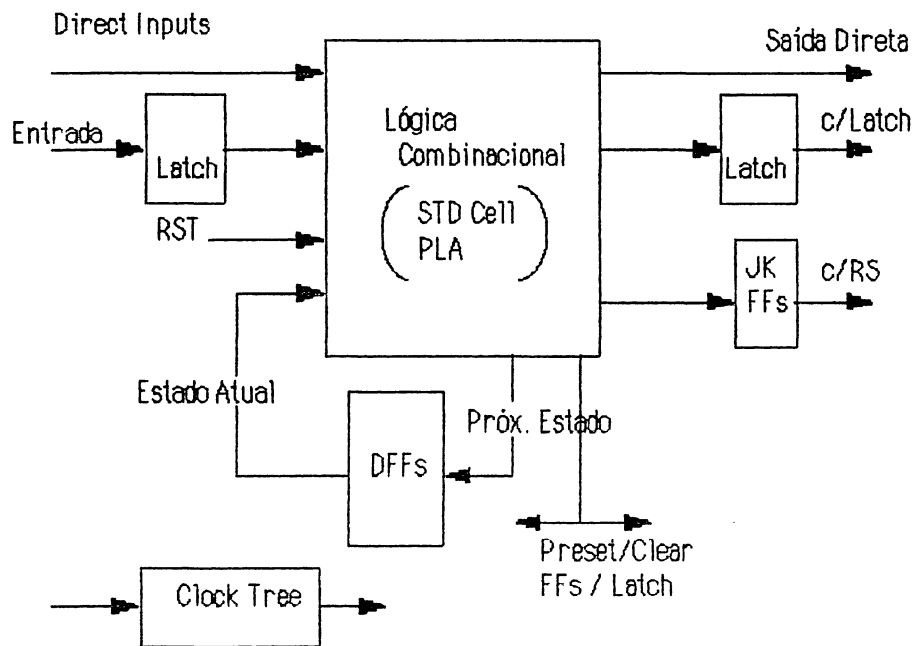


Figura 6.2 Arquitetura Alvo das Máquinas de Estado

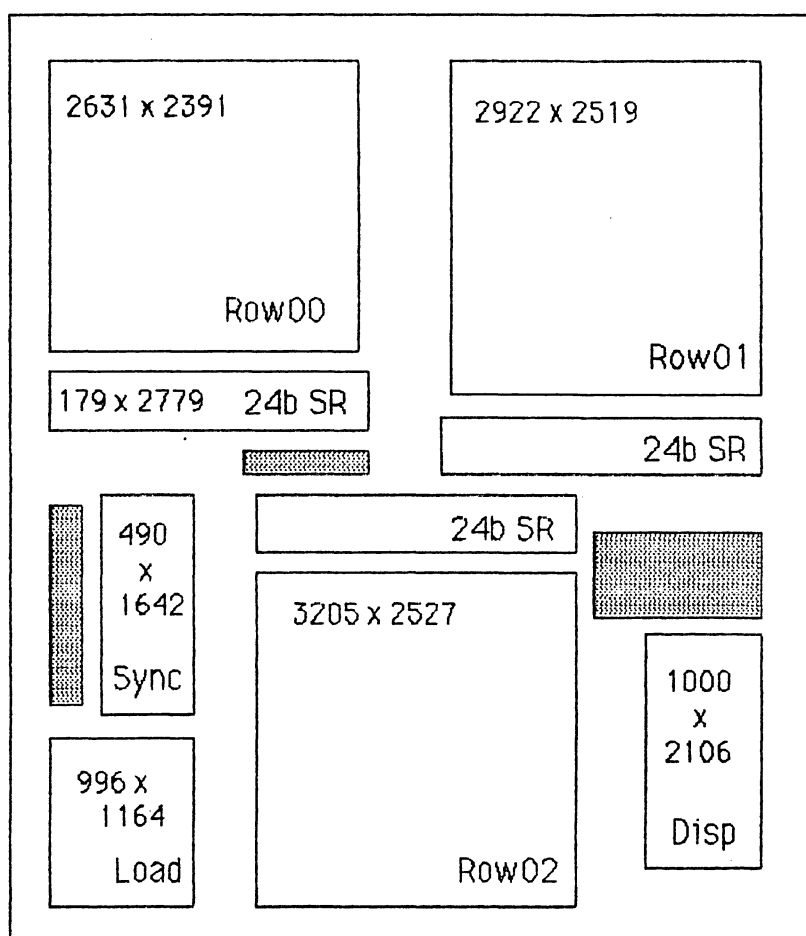
Esta ferramenta foi utilizada para implementar as duas unidades de controle do circuito, sincronização e programação, e o decodificador da unidade de deslocamento.

- \* Controle de Programação: Implantação baseada em um PLA dinâmico, evitando problemas de glitches. O PLA gerado possui 13 entradas, 11 saídas e 22 mintermos ocupando uma área de  $647 \times 477 = 3.08 \times 10^5 \mu^2$ .
- \* Controle de Sincronização: O controle de sincronização foi completamente implementado com a técnica de *standard cells*, uma vez que utiliza sinais de rápida transição e possui poucas saídas. O pequeno número de sinais gerados torna fácil o controle de glitches.

### 6.3 Posicionamento e Interconexão

O CHIP COMPILER é uma ferramenta de posicionamento (*placement*) e roteamento (*routing*) iterativo, baseada em roteamento de canais em *sliced floor plans*. No posicionamento apenas a colocação de *standard cells* é realizada automaticamente, sendo especificado pelo usuário o local e o formato das áreas que conterão as células. Com esta ferramenta podem ser utilizadas partes de circuito desenhadas de forma não automática que sigam determinadas regras na colocação dos conectores. O projetista pode orientar a ferramenta através de parâmetros, como por exemplo a posição preferencial de algum componente *standard cell*, posição de conectores e caminhos preferenciais de roteamento.

A ferramenta foi utilizada de maneira hierárquica. O circuito foi particionado em 5 grandes blocos físicos, correspondentes às unidades funcionais. A não utilização de hierarquia pode gerar layouts mais otimizados, porém o tempo de processamento torna-se proibitivo com o aumento do número de ligações. A planat baixa (*floor plan*) geral e a área de cada bloco é dada na figura 6.3, onde as unidades de medida são em microns.



■ Logica

CV2D - Planta Baixa

Figura 6.3 CV2D - Planta Baixa ( *FloorPlan* )

#### 6.4 Testes

A sequência de testes do protótipo CV2D dar-se-á basicamente em 4 etapas. Detalhes das sequências de comandos utilizadas serão apresentados na documentação de teste anexa ao projeto da placa onde funcionará o protótipo.

As ferramentas necessárias ao teste do protótipo são simples, e todas encontradas no laboratório do CPGCC/UFRGS. São elas: Fonte de

alimentação com limitação de corrente e um miliamperímetro a fim de medir consumo. Um osciloscópio de 100Mhz, utilizado para verificação de formas de onda, medição acurada de tempo e níveis de tensão. Finalmente um analisador lógico, DAS 9000 ou TEK1241, utilizado para excitação e amostragem dos sinais de resposta.

Os passos de teste são citados, resumidamente, abaixo:

- \* Teste de consumo, alimentando o circuito com uma fonte de tensão com limitação de corrente. Com a utilização de um miliamperímetro pode ser monitorado o consumo de corrente, verificando eventuais curto circuitos e evitando danificação do protótipo.
- \* Teste de protocolo. Este se dá com duas sequências diferentes, aproveitando o fato que as máquinas de estado que controlam a carga de parâmetros e sincronização de dados são independentes.
  - Carga de Parâmetros: Excitar com o padrão de comunicação e observar os sinais de resposta e sua temporização. Em um primeiro passo, os sinais de estímulo são os mesmos das simulações e após, outros testes poderão ser realizados.
  - Sincronização: De acordo com os coeficientes carregados e o número de pixels/linha, acionar as linhas de protocolo e observar o comportamento do circuito e seus dados de saída, também com o mesmo padrão de simulação anexo à documentação do circuito.
- \* Teste de Dados. Carregar uma sequência de coeficientes, colocar um padrão de dados de entrada e observar o padrão de saída. Estes devem obedecer aos valores esperados pelos cálculos existentes nas simulações do circuito.

Estas 3 etapas podem ser realizadas em dois passos. Primeiramente com um relógio de baixa frequência, e.g. 1 MHz. Este tipo de teste não apresenta dificuldade, uma vez que o cadenciamento do circuito provém sempre de um oscilador externo. Em uma segunda etapa, aplicar as mesmas sequências com a frequência de operação normal (10Mhz) e testar as condições de temporização após realizado o teste funcional da primeira etapa. É claro que o teste realizado desta forma não cobre todas as possibilidades, porém não se dispõe hoje de um gerador de vetores de teste



ou mesmo um avaliador dos vetores aplicados a fim de determinar a cobertura dos mesmos.

O quarto passo citado é a construção de uma placa de teste, que contenha o protótipo, unidades de line delay e uma interface com equipamento de controle. Em conjunto com um digitalizador e um dispositivo de exibição (vídeo) pode se observar o efeito visual sobre as imagens processadas e armazenar resultados para posterior análise.

## 7 CONCLUSÕES

" Duvidar de tudo ou tudo crer  
são duas soluções igualmente cômodas,  
que dispensam, ambas, de refletir. "  
Henrique Poincaré (1854-1912).  
A Ciência e a Hipótese.

Nos capítulos anteriores foi especificado o problema e também descrita a realização do projeto CV2D. A abordagem foi feita em 4 etapas, sumarizadas a seguir, seguindo a ordem da cadeia de processamento de sinais: geração, aquisição, tratamento e utilização.

- a. Análise do sinal a ser processado, sua geração e principais características. Foram abordados os aspectos e consequências da digitalização do sinal de vídeo, considerando a resolução e definindo os parâmetros de número de bits/pixel e pixels/linha, bem como a frequência de amostragem necessária ao sistema. Ainda nesta etapa, considerações sobre os dispositivos de exibição foram úteis para detecção das limitações inerentes ao sistema;
- b. Caracterização do sistema de processamento de imagens, identificando os principais componentes e as topologias utilizadas. Foram mostrados alguns operadores comumente utilizados nos sistemas de IP e definido um operador a ser implementado. O algoritmo de convolução foi o escolhido devido a sua grande utilidade nos algoritmos de primeiro nível de tratamento de imagens. Uma vez definido o operador, foi realizada a caracterização deste, observando suas limitações, forma de cálculo, tipos e número de dados utilizados e os resultados a serem produzidos. Foram também levados em conta os requisitos temporais para funcionamento em tempo real. Ao final desta análise, foram definidas as condições de contorno do projeto e caracterizado funcionalmente o operador implementado, condições estas descritas no capítulo 3;
- c. Análise das várias formas de implementação do operador funcionalmente proposto. Foi realizada uma revisão de algumas

técnicas de implementação, representação do paralelismo potencial e fluxo de dados. Concluída a análise, foi escolhido um modelo arquitetural alvo: Processamento Sistólico. A partir do modelo, foi derivada uma estrutura de implementação que satisfaz a especificação funcional do problema. A implementação em cascata através de três unidades de convolução unidimensional fica representada por uma linha de processadores;

- d. Foi projetada a arquitetura, detalhando os elementos de processamento e colocando as estruturas de sincronização necessárias. Neste ponto foi definida a interface de circuito através das linhas de programação e sincronização externa. O capítulo 5 descreve a arquitetura projetada.

### 7.1 Condições de Desenvolvimento do Projeto

O objeto de implementação situa-se em um campo pouco explorado no CPGCC: processamento em baixo nível de imagens. Desta forma, além de um período de estudo da área a fim de reconhecer os principais algoritmos, sentiu-se uma carência de sensibilidade na aplicação dos mesmos. Por exemplo, quando uma operação de equalização de histograma ou filtragem, é aplicada sobre a imagem quais os resultados visuais esperados ou qual o efeito da variação dos parâmetros utilizados.

Outro aspecto a salientar é a necessidade da definição de um ambiente de funcionamento para o processador implementado, uma vez que não existe, no CPGCC, um sistema de processamento onde o circuito possa ser instalado. A busca de uma definição geral, que não limite a colocação do componente em algum sistema, tomou uma parte significativa do tempo de especificação do sistema.

No projeto de um circuito integrado é necessário um conjunto de ferramentas que permita sua realização. Tais ferramentas, citadas no capítulo 6, estiveram a disposição na Université Catholique de Louvain (UCL) durante os 6 meses de desenvolvimento do projeto nesta universidade. Junto à estação de projeto de CIs, esteve disponível um

sistema de digitalização e exibição de imagens, o que permitiu observar os resultados dos algoritmos sobre imagens reais bem como a influência dos coeficientes de convolução sobre a imagem processada. Após o conhecimento prévio das ferramentas de CAD e definida a estrutura do circuito a ser implementado (máquina de estados e o fluxo de dados), um prazo de dois meses é suficiente para o seu projeto físico.

## 7.2 Melhorias

Após uma avaliação do projeto realizado, algumas características adicionais no sistema seriam bem situadas dentro da aplicação em processamento de imagens e no campo de projeto de circuitos integrados VLSI. Tais melhorias são abaixo citadas:

a. **Função Módulo.** A colocação de uma opção de módulo dos resultados seria útil tanto na visualização dos resultados negativos, como para otimizar a aplicação do operador de convolução nos algoritmos de detecção de bordas e cálculo de gradientes;

b. **Programação do Número de Pixels/Linha.** O limite fixado em um número de pixels por linha, apesar de programável, impede a utilização direta em imagens com maior resolução horizontal. A opção de sincronizar apenas pelos sinais externos de controle, desconsiderando o valor do contador de número de pixels, eliminaria tal restrição, dando maior flexibilidade ao componente;

c. **Varredura Entrelaçada.** Atualmente os sistemas com varredura entrelaçada são os mais comuns, tanto em câmeras quanto em receptores de TV. Esta técnica de varredura não é, no entanto, a mais adequada a imagens de alta resolução. Uma alteração a fim de suportar este sistema de exploração ampliaria o espectro de utilização do operador de convolução, aplicando-se este a sinais de câmeras mais simples, e portanto mais baratas, possibilitando ainda aplicações de melhoria de imagem em receptores de TV;

d. **Arquitetura das Máquinas de Estado.** As máquinas de controle do CV2D possuem arquitetura fixa, definida pelo tipo de

ferramenta de concepção utilizada. Estas máquinas de estado poderiam ser otimizadas, explorando outras formas de implementação, suplantando as limitações impostas pela estação de trabalho utilizada.

### 7.2.1 Testabilidade

Uma avaliação crítica mostra que o circuito possui uma baixa observabilidade e baixa controlabilidade dos blocos internos ao circuito, o que compromete a testabilidade deste.

As principais razões de não serem colocadas estruturas adicionais para teste foram a limitação em área e o tempo disponível para o projeto físico. Além disso não estava disponível um simulador de falhas, que permitisse avaliar a cobertura de alguma estrutura proposta. Os conceitos de testabilidade e as várias técnicas e estruturas de teste são encontrados em [TSU 88]. As técnicas não são discutidas, no entanto são tecidas considerações e apresentadas sugestões que poderiam ser incluídas em uma segunda versão do circuito.

\* **Adição de pinos de teste**, uma vez que atualmente o circuito é do tipo *core limited* e são disponíveis encapsulamentos com mais de 64 pinos.

\* **Conexão do último bit do banco de registradores de parâmetros a um pino de saída**. Estes registradores já estão conectados serialmente e tal medida poderia ser utilizada para verificar o conteúdo do banco de registradores e avaliar o funcionamento da máquina de estados de carga de parâmetros.

\* **Ligação dos registradores dos PEs em *Scan Path***. Esta alternativa é viável desde que não provoque um atraso significativo dos dados, já que estes registradores fazem parte do caminho crítico.

\* **Análise de assinatura**. Os principais pontos das máquinas de estado poderiam ser testados através de um analisador de assinatura, capaz de condensar a informação de tais pontos e colocar os resultados de

resultados de maneira serial, adicionando-se apenas um *PAD* de saída. Este tipo de estrutura ocupa pouca área. Porém a escolha de um polinômio de realimentação com razoável cobertura de falhas fica prejudicada se não for utilizado um simulador de falhas.

Os pontos a serem avaliados seriam os sinais de controle gerados pelas máquinas de estado, o que permitiria verificar seu funcionamento.

\* **LFSRs para geração de vetores de teste.** A geração interna de vetores de teste para as máquinas de estado ou cadeia de processadores pode ser realizada através de LFSRs. Novamente a escolha do polinômio fica vinculada a uma razoável simulação de falhas.

Desta forma os blocos internos do circuito poderiam ser testados através desta estrutura em conjunto com analisadores de assinatura, mostrado na figura 7.1, utilizando um pino único de habilitação de teste e uma saída para cada analisador.

As modificações sugeridas ocupam pouca área, se comparadas com a superfície total dos elementos de processamento, e adicionam, também, poucos pinos para realização de testes.

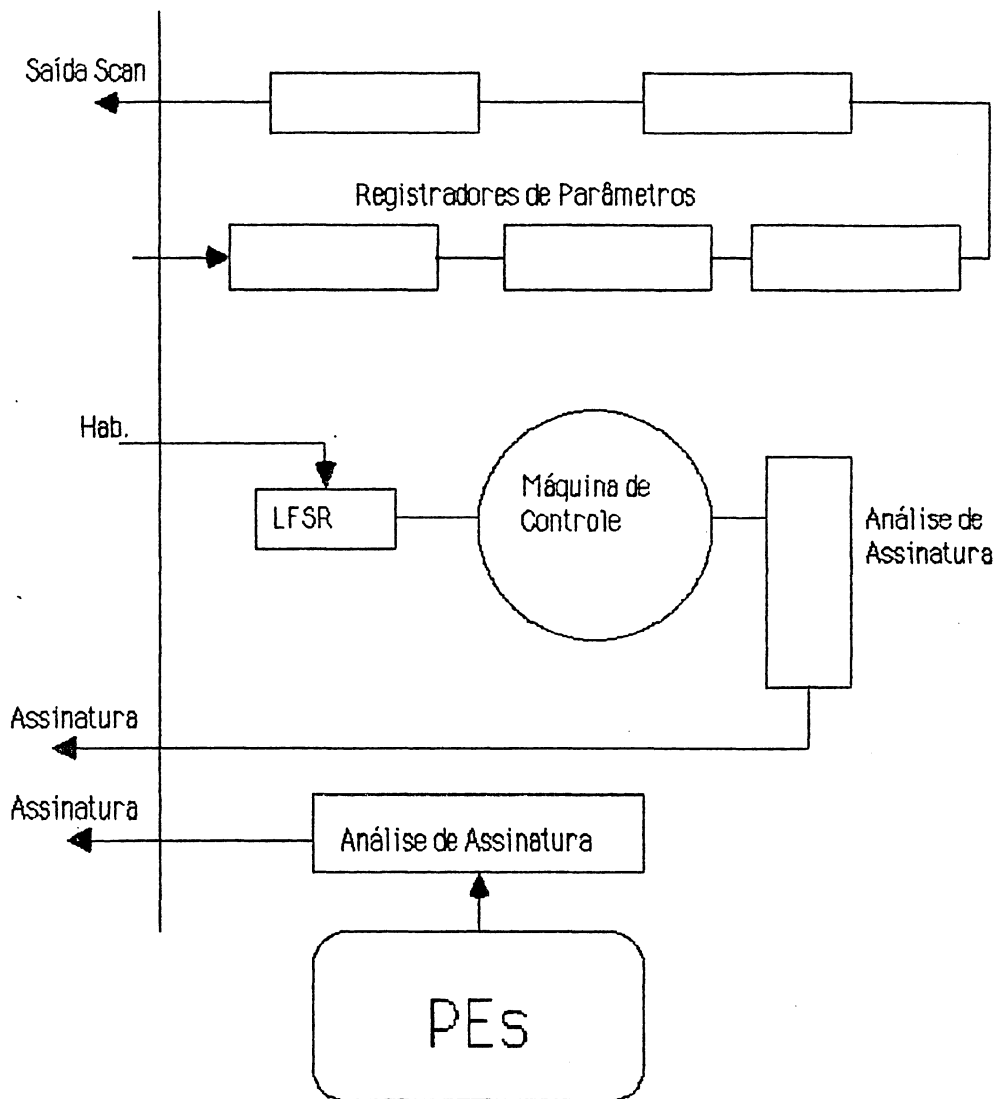


Figura 7.1 Estruturas de Teste

### 7.3 Resultados

Um dos resultados práticos deste trabalho é um simulador funcional do circuito, que executa as operações tal qual o circuito projetado e através dos resultados obtidos com a simulação pode ser verificada a validade do uso de máscaras exclusivamente com potências de dois.

Foi estabelecido um protocolo assíncrono de programação, descrito no capítulo 5, baseado nos ciclos de leitura e escrita da família MC68000 da Motorola. Além disso foi definido um ambiente de funcionamento e o modelo de processo no capítulo 3, que possibilita o desenvolvimento de outros sistemas e sua integração futura sem conhecimento em detalhe do circuito ora realizado.

Hoje (outubro 89) o CPGCC possui algumas estações gráficas PROCEDA que possuem um processador dedicado ao processamento de dados e outro processador para exibição e gerência da tela gráfica. Esta estação aliada a um digitalizador permite a realização de uma plataforma de pesquisa em processamento de imagem. A figura 7.2 mostra a arquitetura do sistema e indica a colocação dos módulos de software de IP e dos componentes dedicados ao processamento em tempo real.

Nota-se a tendência e a necessidade de estruturas assíncronas, devido à velocidade de processamento, onde trocam-se os processadores sistólicos pelos processadores de frente de onda " *wavefront processor*" [KUN 87], [KUN 84].

Como último resultado prático destaca-se o componente CV2D, atualmente em fase de fabricação do protótipo, e sumariamente descrito abaixo. Os resultados das simulações elétricas e funcionais estão na documentação do circuito, a ser publicada após os testes do protótipo.

Nome: CV2D
Função: Convolução Bidimensional de Imagens
área: 64 mm <sup>2</sup>
Tecnologia: CMOS, Twin-Tub, 2.0μ
Complexidade Média: 25K transistores
Consumo: 400mW @ 10MHz



A contribuição deste trabalho reside, além da formação pessoal, na experiência com software profissional durante o estágio na Universidade Católica de Louvain. Tal experiência permitiu uma avaliação das ferramentas construídas no grupo de microeletrônica do CPGCC. Além disso, fica a proposta de um sistema de processamento de imagens que pode ser desenvolvido dentro da realidade nacional. A implementação deste contribuiria para o ensino e a pesquisa na área de processamento de sinais e imagens.

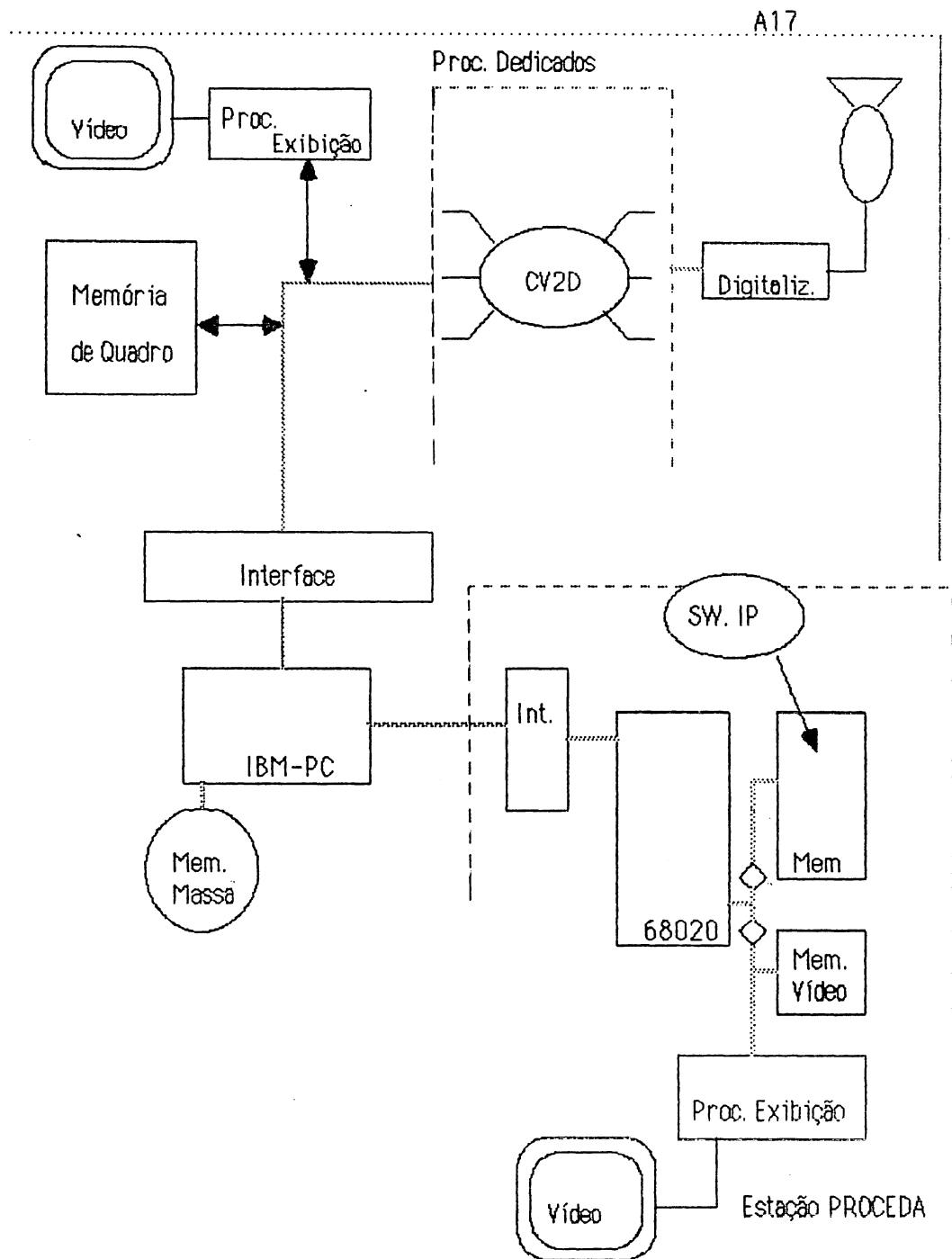


Figura 7.2 Arquitetura PGCC

O seguimento imediato deste trabalho são o teste do protótipo CV2D, validando as idéias que originaram o projeto e a construção de uma placa de interface com um computador hospedeiro. Após estas tarefas,

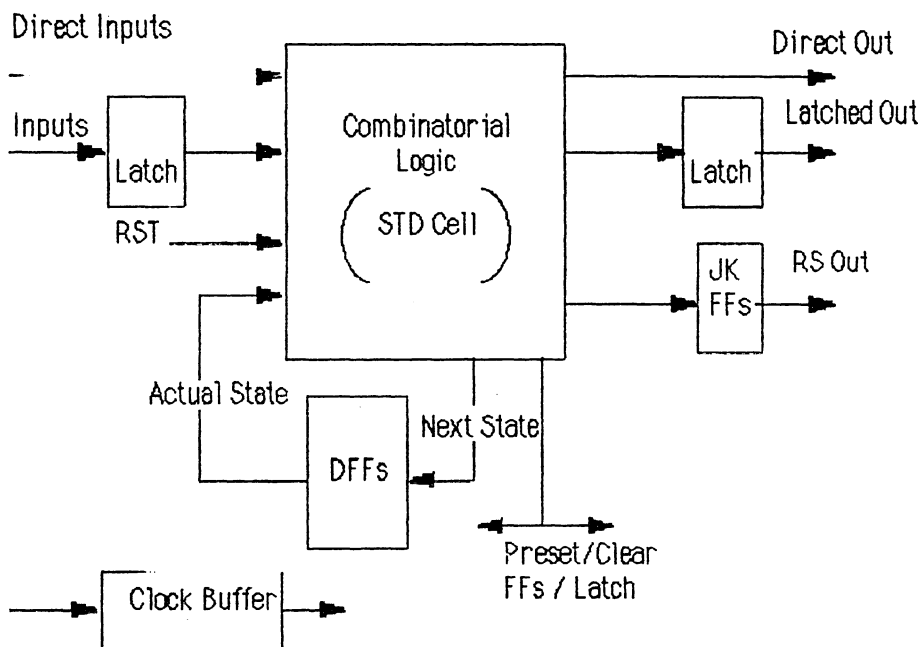
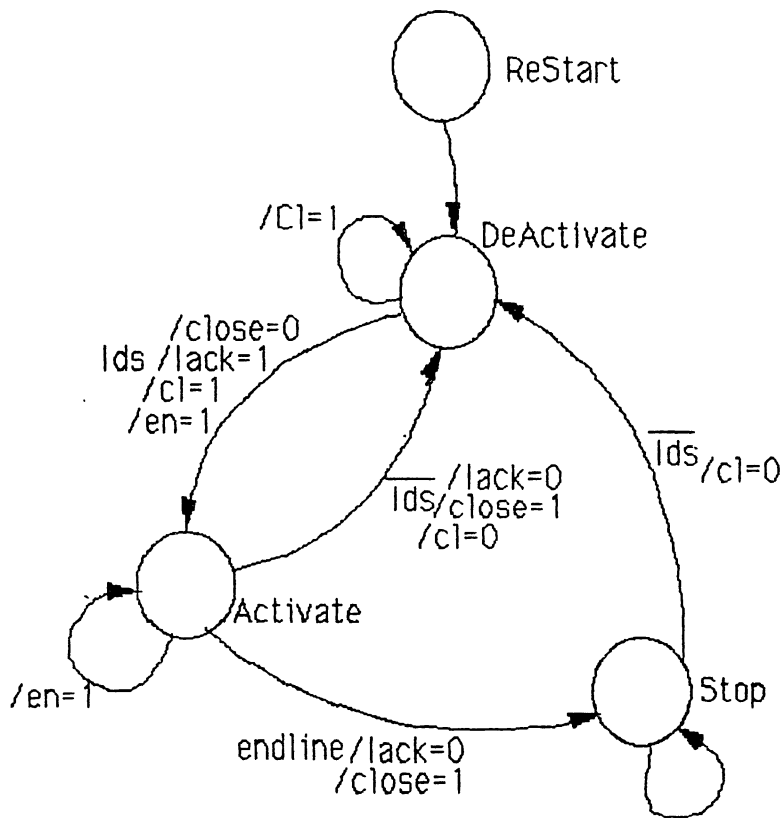
prossegue o desenvolvimento do sistema de IP, com a realização da arquitetura PGCC da figura 7.2, implementando um conjunto de rotinas básicas que permita a experimentação em processamento de imagens. Inicia-se, após, a definição dos pontos abordados no capítulo 3, permitindo a especificação de novos componentes para o sistema de processamento A17. Estas tarefas já estão sendo iniciadas, sendo objeto de outra dissertação de mestrado neste curso.

Fica a esperança de que o esforço no desenvolvimento e implementação tecnológica supere a fase das aplicações bélicas e do consumismo e passe a servir ao desenvolvimento do homem na sua totalidade. Que a direção do esforço mundial seja voltado ao bem estar da humanidade, à difusão da cultura e, principalmente, a uma eficiente distribuição de alimentos a todos os povos, contribuindo assim para a libertação pessoal e comunitária do homem. Talvez, com este objetivo, as pessoas voltem a acreditar no amor, na paz e na vida.

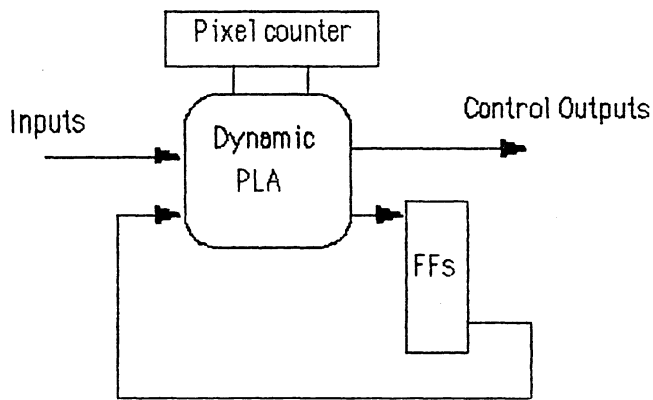
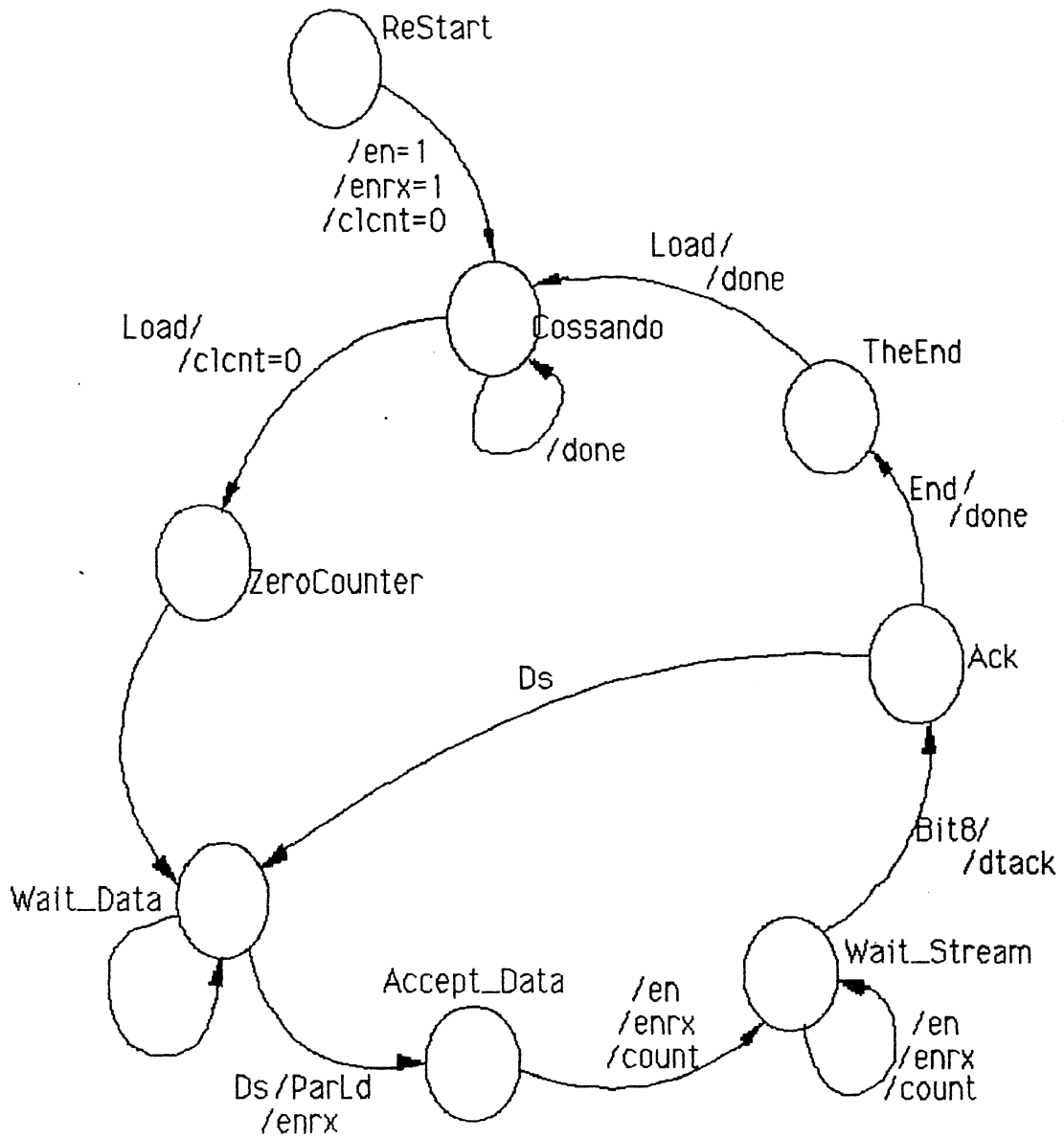
## ANEXO 1 – Diagramas de Estado

Neste apêndice são dados os diagramas de estado das máquinas de controle e a respectiva arquitetura alvo utilizada para implementação.

Nos desenhos dos diagramas, os círculos representam os estados e os arcos representam as transições. A cada estado está associado um nome e nos arcos estão indicadas as condições de transição e os sinais de saída, na forma " **condição/saída** ".



Arquitetura Alvo - Máquina SYNC



Arquitetura Alvo - Máquina LOAD

## **ANEXO 2 – Distribuição de PADS Tabela de Tempos**

A seguir é dada a distribuição de PADS no CV2D, onde o PAD de número 1 está centralizado na borda superior do chip. A pinagem não foi definida, uma vez que não se sabe qual o encapsulamento disponível para o protótipo.

A tabela de tempos correspondente aos diagramas foi extraída através de simulações e servirá de parâmetro para testes do circuito.

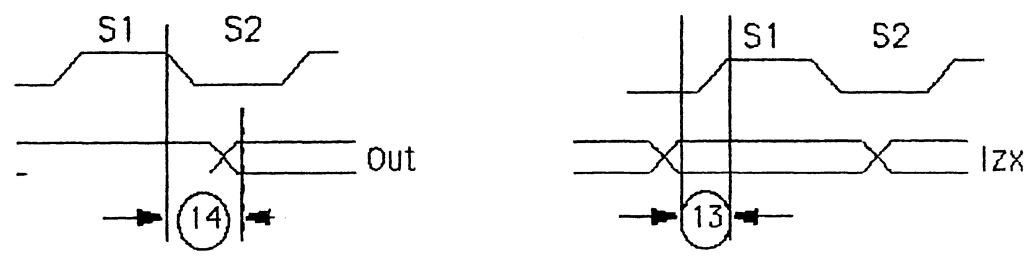
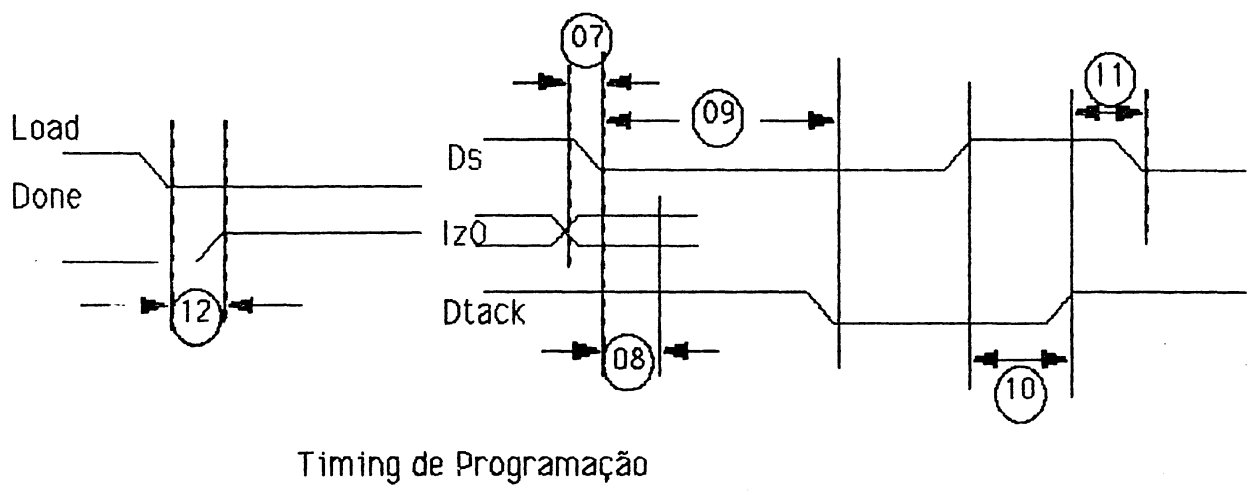
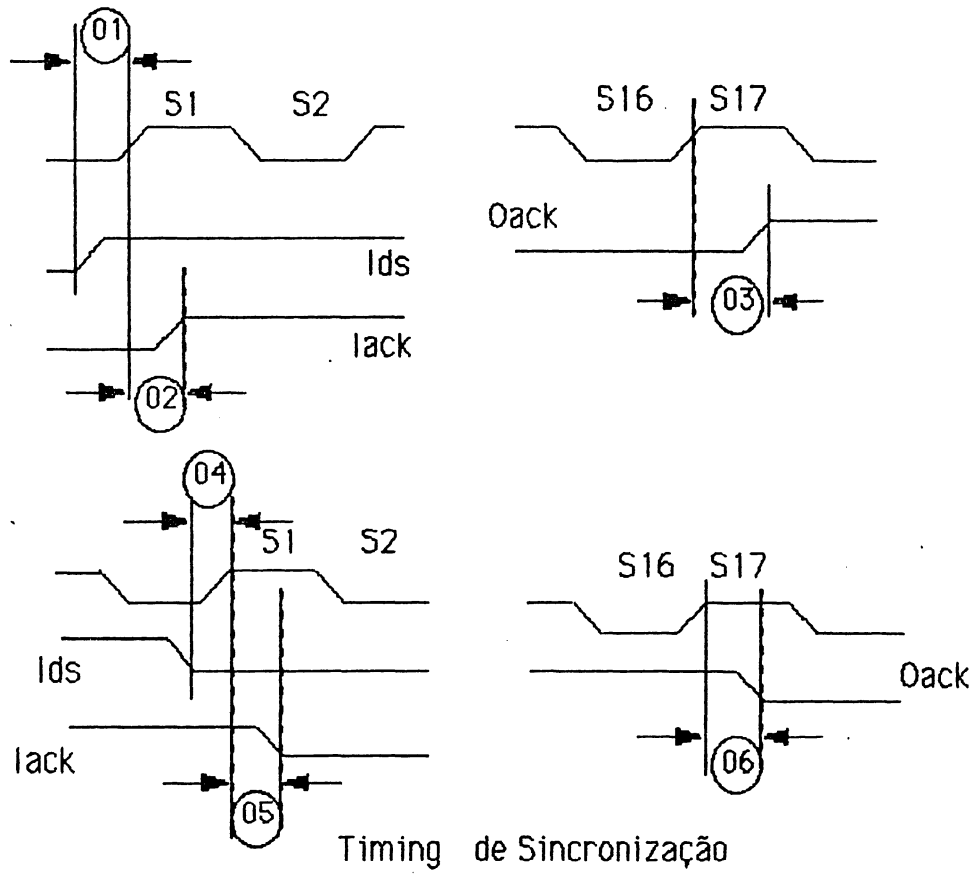
## Distribuição de PADs

PAD number #	Signal
01	Iz1[3]
02	Iz2[1]
03	$\overline{DS}$
04	Iz2[4]
05	Iz1[7]
06	$\overline{OE}$
Empty Corner Nw	
07	Iz2[6]
08	Iz0[4]
09	Iz2[3]
10	Iz1[1]
11	Iz1[0]
12	Iz1[5]
13	VDD-core
14	Out[1]
15	RST
16	Iz0[1]
17	OACK
18	IDS
19	Out[15]
Empty Corner Sw	
20	$\overline{DTACK}$
21	$\overline{DONE}$
22	$\overline{LOAD}$
23	Iz0[5]
24	IACK
25	Out[8]
26	Out[4]
27	Iz0[3]

PAD number #	Signal
28	Out[13]
29	Out[7]
30	Out[0]
31	Out[2]
32	Out[11]
33	Out[10]
Empty Corner Se	
34	Out[12]
35	Out[6]
36	Out[14]
37	Out[5]
38	Iz0[7]
39	Out[3]
40	Out[9]
41	VSS-core
42	Iz1[4]
43	Iz2[0]
44	Iz0[6]
45	Iz1[6]
46	VDD-padring
Empty Corner Ne	
47	Iz0[2]
48	Iz0[0]
49	Iz2[7]
50	VSS-padring
51	Iz1[2]
52	Iz2[2]
53	CK
54	Iz2[5]



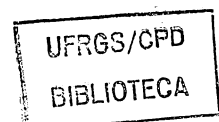
Formas de Onda



## Tabela de Tempos

Name	Min	Typ	Max
01 - Setup IDS high to Ck	2		
02 - Delay IACK high to Ck	45		
03 - Delay OACK high to Ck	34		
04 - Setup IDS low to Ck	0		
05 - Delay IACK low to Ck	34		
06 - Delay OACK low to Ck	34		
07 - Setup IZ0 to DS	0		
08 - Hold IZ0 to DS	100		
09 - DTACK low to DS low	1 $\mu$ s		1.1 $\mu$ s
10 - DTACK high to DS high	100		200
11 - DS low to DTACK high	0		
12 - DONE rise to LOAD fall	47		147
13 - Setup Data (Iz) to Ck	0		
14 - Dealy Data Out (OUT) to Ck fall	4		9

Nota: Todos os tempos estão expressos em ns, salvo quando especificada outra unidade.



## **ANEXO 3 - Informações Preliminares**

Este sumário das principais características do CV2D foi enviado para aprovação da VLSI Technology antes do envio da fita com a descrição do circuito para fabricação.

## CY2D - A Single Chip Real Time Two Dimensional Convolver

### Functionality:

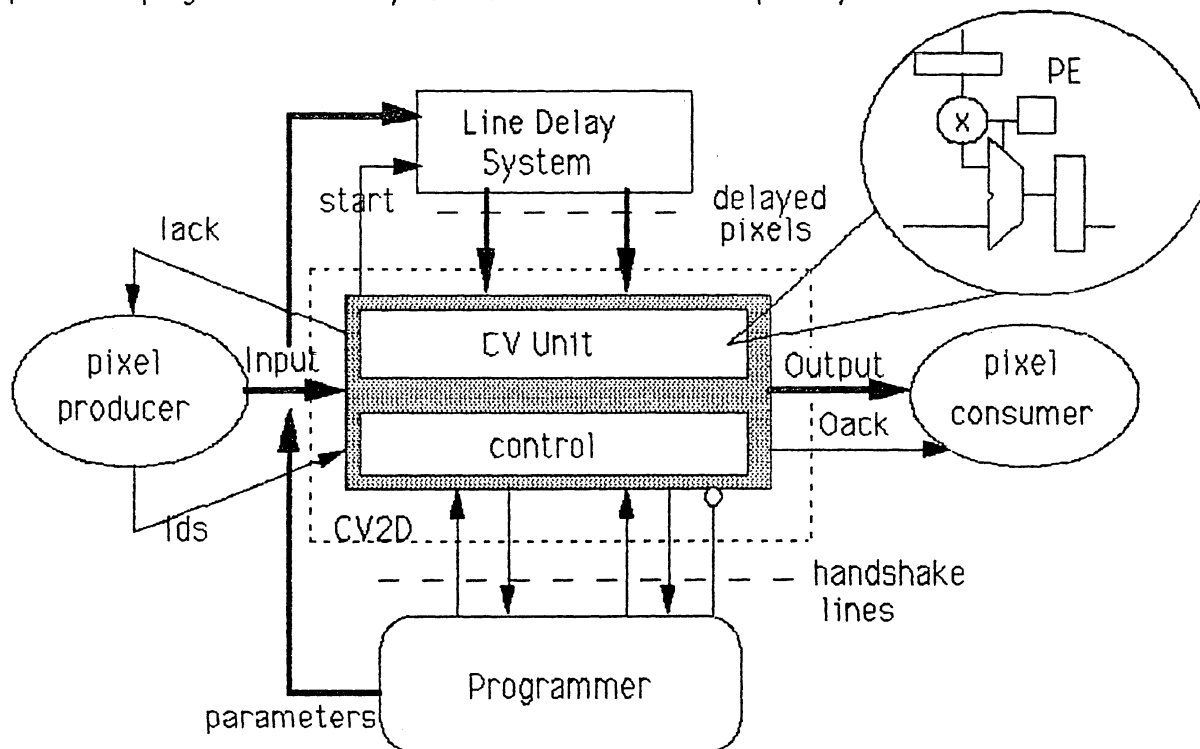
The CY2D computes, at pixel rate, the bidimensional convolution over 8bit/pixel quantized images up to 512 pixels/line. The chip was designed to be an interface for image processing systems performing linear filtering or convolution processing of a pixel stream. The convolution process needs two delayed lines and the storage of these lines is performed by means of a line delay system, actually implemented off chip.

### Characteristics:

- Technology: YTI ,CMOS 2.0  $\mu\text{m}$ , N well, Double metal.
- Frequency: 10 Mhz.
- Pincount: 51 pins.
- Internal computation in Systolic Fashion.
- 3 independent inputs, the incoming and 2 delayed lines.
- Selectable coefficients of convolution from the set  $[\pm 0, 1, 2, 4, 8, 16]$ .
- Interval between 2 input pixels or between 2 output pixels equals 1 cycle. This allows real time computation at 10 Mhz clock frequency with performances equivalent of 161 MIPs.
- Controlability of output data format. The results produced are 16-bit wide and the output range can be manipulated by a displacement parameter, with signal extension. Additionally negative results can be set to zero.
- 3 handshaking lines for synchronization give a simple and flexible interface with input and output hardware.
- There is also a signal to control the line delay system (start /stop).
- Output enable control allows interconnection on a bus system.
- The loading parameter interface is based on asynchronous handshaking, allowing connection to systems with different speeds with a minimum effort.
- The parameters programming consists in loading sequentially 11 bytes.

### Chip Architecture:

Figure 1 shows the CY2D and the environment model. Basically, the CVUnit performs the convolution process through a systolic chain and consists of 9 Processing Elements (PEs). This characteristic results in a highly regular architecture well suited for integration. The other block implements the asynchronous interface for parameter programming and synchronization with the actual pixel systems.



## REFERÊNCIAS BIBLIOGRÁFICAS

- [AMO 88] AMORESE, Philip; BLOOMFIELD, J. A Slew of Standards for Camera Systems. ESD: The electronic System Design Magazine, Boston, 18 (3): 94-98, Mar., 1988.
- [AND 77] ANDREWS, H.C. ; HUNT, B.R. Digital Image Restoration. Englewood Cliffs, Prentice-Hall, 1977.
- [ARA 87] ARAMBEPOLA, B.; CHEUNG, G.; PATEL, V.B. Castable One/Two Dimensional Digital Convolver. In: IEEE 1987 CUSTOM INTEGRATED CIRCUITS CONFERENCE. Portland, May 4-7, 1987. Proceedings. New York, IEEE, 1987. p. 311-314.
- [ARC 88] ARCE, Gonzalo R. et al. Theory and VLSI Implementation of Multilevel Median Filters. In: IEEE 1988 INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, Finland, June 7-9, 1988. Proceedings. New York, IEEE, 1988. p.2795-2798.
- [AWC 87a] AWCOCK, G.J.; STONE ,F.W.; THOMAS, R. Image Acquisition System. Electronics & Wireless World, London, 93 (4): 441-444, Apr., 1987.
- [AWC 87b] AWCOCK, G.J.; STONE ,F.W.; THOMAS, R. Image Acquisition System. Electronics & Wireless World, London, 93 (5): 531-534, May, 1987.
- [AWC 87c] AWCOCK, G.J.; STONE ,F.W.; THOMAS, R. Image Acquisition System. Electronics & Wireless World, London, 93 (6): 589-592, June, 1987.
- [BAR 87] BARRERA, Junior. Processamento Digital de Imagens. Sao Jose dos Campos, INPE , 1987.
- [BRI 89] BRIGHTFIELD, Steve. A One-Chip Real-Time Convolution Solution. ESD: The Electronic Systems Design Magazine, Boston, 17 (5): 47-54, May, 1989.

- [CAD 80] CADY,F.M; HODGSON, R.M. Microprocessor Based Interactive Image Processing System. IEE Proceedings - Special Issue on Image Restoration and Enhancement. London, 127 Pt.E (5): 197-202, Sept., 1980.
- [CAP 87] CAPPELLO, Peter R.; WU, Cheng-Wen. Computer Aided Design of VLSI FIR Filters. Proceedings of the IEEE, New York, 75(9): 1260-1271, Sept., 1987.
- [COM 83] COMPUTER, Los Angeles, v. 16, n.01, Jan., 1983. Computer Architectures for Image Processing.
- [COM 87] COMPUTER, Los Angeles, v.20, n.07, July, 1987. Special Issue on Systolic Arrays.
- [DAN 84] DANIELSSON, P.E. Serial/Parallel Convolvers. IEEE Transaction on Computers, New York, C-33(7): 652-667, July, 1984.
- [DEN 85] DENYER, Peter; RENSHAW, David. VLSI Signal Processing: A Bit-Serial Approach. London, Addison-Wesley, 1985.
- [DEN 88] DENAYER,T. VANZIELEGHEM,E. ; JESPERS,P.G.A. A Class of Multiprocessor for Real Time Image and Multidimensional Signal Processing. IEEE Journal of Solid State Circuits, New York, 23(3):630-638, June, 1988.
- [DOS 87] DOSH,K.; VARMAM,P. A Modular Systolic Architecture for Image Convolutions. In: ANNUAL INTERNATIONAL SYMPOSIUM ON COMPUTER ARCHITECTURE,14., Pitsburg, June 2-5, 1987. Proceedings. New York, IEEE, 1987. p. 56-63.
- [DVO 88] DVORAK, Petr. Edge Detection and its Using by the Automatic Processing of Cineangiographic Images of Left Ventricle. Porto Alegre, CPGCC- UFRGS, 1988. RP n.86
- [EDE 89] EDEN, A.W.M Van Den; VERHOECKX, N.A.M. Discrete-Time Signal Processing - an Introduction. Cambrige, Prentice-Hall, 1989.

- [EKS 84] EKSTROM, Michael P. Digital Image Processing Thecniques. Florida, Academic Press, 1984.
- [FIS 81] FISHER, Allan L. Systolic Algorithms for Runing Order Statistics in Signal and Image Processing. In: CMU CONFERENCE ON VLSI SYSTEMS AND COMPUTATIONS, Pittsburg, Oct. 19-21, 1981. VLSI Systems and Computations. Rockville, Computer Science Press, 1981. p. 265-272.
- [FOR 86] FORTES, Jose A.B., KING-SUN, Fu; BENJAMIN, W.Wah. Systematic Design Approaches for Algorithmically Specified Systolic Arrays. In: MILUTINOVIC, V.M. Computer Architecture - Concepts and Systems. Amsterdam, North-Holland, 1986. p.454-494.
- [FOS 80] FOSTER, M.J.; KUNG,H.T. Design of Special Purpose VLSI Chips: Example and Opinions. In: ANNUAL SYMPOSIUM ON COMPUTER ARCHITECTURE, 7., May 6-8, 1980. Proceedings. New York, IEEE, 1980. p. 300-307.
- [GLA 85] GLASSER, Lance A.; DOBBERPHUL, Daniel W. The Design and Analysis of VLSI Circuits. Reading, Addison-Wesley, 1985.
- [GON 87] GONZALEZ, R.C; WINTZ, Paul. Digital Image Processing. Reading, Addison-Wesley , 1987.
- [GRE 87] GREGORIAN, R.; TEMES, Gabor C. Analog MOS Integrated Circuits for Signal Processing. New York, John Wiley, 1987.
- [GRO 79] GROB, Bernard. Televisao básica, principios e reparacao. Rio de Janeiro, McGraw-Hill, 1979.
- [HAL 79] HALL, Ernest L. Computer Image Processing and Recognition. New York, Academic Press, 1979.
- [HAM 77] HAMMING, R.W. Digital Filters. New Jersey, Prentice-Hall, 1977.

- [HAW 84] HAWANG, K. and Briggs, F. Computer Architecture and Parallel Processing. New York, McGraw-Hill, 1984.
- [JAC 86] JACSON et al. An Approach to the Implementation of Digital Filters. IEEE Transactions on Audio and Electroacoustics, New York, AU-16(3): 413-421, Sept., 1968.
- [JOH 81] JOHNSON,L.; COHEN,D. A Matematical Approach to Modelling the Flow of Data and Control in Computational Networks. In: KUNG, H.T. et al. (eds) CMU Conference on VLSI Systems and Computation. Rockville, Computer Science, 1981. p.213-225.
- [KRU 73] KRUSE,B. A Parallel Picture Processing Machine. IEEE Transactions on Computers, New York, C-22(12):1075-1087, Dec., 1973.
- [KUL 81] KULPA; DERNALOWICZ; NOWICKI; BIELIK. CPO-2/K-202: A Universal Digital Image Analysis System. In: Bolco, L.; KULPA,Z. (eds) Digital Image Processing Systems. Berlin, Springer-Verlag, 1981. p.169-199. Lecture Notes in Computer Science 109
- [KUN 80a] KUNG,H.T. Special Purpose Devices for Signal and Image Processing: An Opportunity in VLSI. Rockville, Carnegie - Mellon University, July, 1980.CMU CS-80-132.
- [KUN 81b] KUNG, H.T. The Structure of Parallel Algorithms. Advances in Computer, New York, 19:65-112, 1981.
- [KUN 81a] KUNG, H.T. Why Systolic Architectures. Rockville, Carnegie-Mellon University, 1981. CMU-CS-81-148.
- [KUN 81b] KUNG,H.T.; SONG, S.W. A Systolic 2-D Convolution Chip. Rockville, Carnegie-Mellon University, 1981.CMU-CS-81-110.
- [KUN 81c] KUNG, H.T.; SPROULL, Bob; STEELE,Guy. CMU Conference on VLSI Systems and Computation. Rockville, Computer Science Press, 1981.



- [KUN 81d] KUNG,H.T. et al. A two Level Pipelined Systolic Array for Convolutions. In: CMU CONFERENCE ON VLSI SYSTEMS AND COMPUTATIONS. Pittsburg, Oct. 19-21, 1981. VLSI Systems and Computations. Rockville, Computer Science Press, 1981. p.255-264.
- [KUN 84] KUNG,S. On SuperComputing with Systolic/Wavefront Arrays Processors. Proceedings of the IEEE, New York, 72(07): 867-884, July 1984.
- [KUN 87] KUNG, S.Y.; LO,S.C.; JEAN,S.N.; HWANG,J.N. WaveFront Array Processors - Concept to Implementation. Computer, Los Angeles, 20(7):18-33, July 1987.
- [LEW 88] LEWIS, Geoff. Hig Definition Television. Electronics & Wireless World, London, 94(1625):226-229, Mar., 1988.
- [LI 85] LI, G.J.; WAH,B.W. The Design of Optimal Systolic Arrays. IEEE Transactions on Computers New York, C-34(01):66-77, Jan., 1985.
- [LOU 80] LOUGHEED,R.M; MCCRUBBREY,D.L. The Cytocomputer: A Pratical Pipelined Image Processor. In: ANNUAL SYMPOSIUM ON COMPUTER ARCHITECTURE, 7., May 6-8, 1980. Proceedings. New York, IEEE, 1980. p.271-277.
- [LUT 89] LUTHER, Arch C. Digital Video in the PC Environment. New York, McGraw-Hill, 1989.
- [MAH 86] MAHMOUDI,M. Cartes et Systems Modulaires de Traitment d'Images. Eletronic Industrielle, Paris (113):42-47, Oct., 1986.
- [MAS 84] MASCARENHAS, N.D.A; VELASCO, F.R.D. Processamento Digital de Imagens. Sao Paulo, Instituto de Matematica e Estatistica, 1984. Quarta Escola de Computacao.
- [MAT 87] MATROX. Electronic Systems. PIP Video Digitizer Board Hardware Manual. Quebec, 1987.

- [MAY 87] MAY, D.; SHEPHERD, R. The INMOS Transputer. In: PARALLEL Processing. New York, Pergamon Infotech, 1987. p.71-92. State of the Art Report.n.15:14.
- [MOO 87] MOORE, Will; MCCABE, Andrew; URQUHART, Roddy. In: INTERNATIONAL WORKSHOP ON SYSTOLIC ARRAYS, Oxford, 2-4, July 1986. Systolic Arrays. Bristol, Adam Hilger, 1987.
- [NIC 88] NICOUD, Jean-Daniel. Video Rams: Structure and Applications. IEEE Micro, New York, 8(1):8-27, Feb. 1988.
- [NCR 85] NCR CORPORATION. Detection of Edges and Gradients In Binary and Gray Scale Images with the GAPP Processor. Dayton, NCR, 1985. GAPP Application Note No.3.
- [ODG 81] ODGERS, Chistopher R. Criteria for Choosing a Camera for Use in a Video Digitizing System. Program of Computer Graphics. California, Cornell University c/ Hanna- Barbera Productions, June, 1981.
- [OFF 85] OFFEN,R.J. VLSI Image Processing. London, William Colling Sons, 1985.
- [OPP 75] OPPENHEIM, Alan V.; SHAFER, R.W. Digital Signal Processing. Englewood Cliffs, Prentice-Hall, 1975.
- [OPP 78] OPPENHEIM, Alan V. Applications of Digital Signal Processing. Englewood Cliffs, Prentice-Hall, 1978.
- [PEL 76] PELED, Abraham; LIU, Bede. Digital Signal Processing - Theory, Design and Implementation. New York, Jhon Wiley, 1976.
- [PER 87] PERLMAN,S.S. et Al. Adaptative Median Filtering for Impulse Noise Elimination in Real-Time TV Signals. IEEE Transactions on Communications, New York, COM-35(6):646-652, June, 1987.

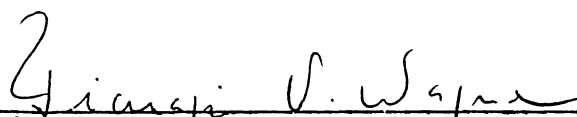
- [PHI 85] PHILIPS. A Medium Resolution 37cm Colour Data & Graphics Display (DGD) Operating at 16KHz and 50/60 Hz. Eindhoven, Philips EDS 8507, Oct., 1985. Project AP1102.
- [PHI 86] PHILIPS. SAA 9010. Picture Enhancement Processor. Data Sheet, Philips/ Signetics. 1986.
- [PRA 78] PRAT, William K. Digital Image Processing. New York, John Wiley. 1978.
- [RAB 75] RABINER, L.R.; GOLD, B. Theory and Applications of Digital Signal Processing. New Jersey, Prentice-Hall, 1977.
- [RAM 88] RAMACHANDRAN, K.; CORDELL, R.R. A 30 MHz Programmable CMOS Video FIR Filter and Correlator. In: IEEE 1988 INTERNATIONAL SYMPOSIUM ON CIRCUITS AND SYSTEMS, Espoo. June 7-9, 1988. Proceedings. New York, IEEE, 1988. p.705-708.
- [ROS 82] ROSENFELD, Azriel; KAK, Avinash C. Digital Picture Processing. New York, Academic Press, 1982. v.1
- [RUE 87] RUETZ, P.A.; BRODERSEN, R.W. Architectures and Design Techniques for Real Time Image Processing IC's. IEEE Journal of Solid State Circuits, New York, SC-22(2): 233-250, Apr., 1987.
- [RUE 89] RUETZ, P.A. The Architectures and Design of a 20MHz Real Time DSP Chip Set. IEEE Journal of Solid State Circuits, New York, 24(2), Apr., 1989.
- [SAL 77] SALAZAR, Andres C. Digital Signal Computers & Processors. New York, IEEE Press, 1977.
- [SHU 88] SHUKLA, Shiridhar B.; AGRAWAL, D.P. A Kernel-Independent, Pipelined Architecture for Real-Time 2-D Convolution. In: ANNUAL INTERNATIONAL SYMPOSIUM ON COMPUTER ARCHITECTURE, 15., Honolulu, May 30 - June 2, 1988. Proceedings. New York, IEEE, 1988. p.160-166.

- [SID 89] SID-AHMED, M.A. A Systolic Realization for 2-D Digital Filters. IEEE Transactions on Acoustics, Speech and Signal Processing, New York, 37(4):560-565, Apr., 1989.
- [SIL 87] SILVE, William M. Gray Level Processing in Machine Vision. ESD: The Electronic System Design Magazine, Boston, 17(5): 79-83, May 1987.
- [TEL 87] TELICHEVESKY, R. ; ROTMAN, A. Projeto de um Processador de Imagens em Tempo Real. In: CONGRESSO NACIONAL DE INFORMATICA, 20., Sao Paulo, 31 ago. - 06 set. 1987. Anais. São Paulo, SUCEsu, 1987.v.2. p. 1198-1207.
- [TSU 88] TSUI, Frank F. LSI/VLSI Testability Design. Singapore, McGraw-Hill, 1988.
- [VAS 87] VASANTHAVADA, N.; BAKER, R.; KANOPOULOS, N. A Monolithic Image Edge Detection Filter. In: IEEE 1987 CUSTOM INTEGRATED CIRCUITS CONFERENCE. Portland, May 4-7, 1987. Proceedings. New York, IEEE, 1987. p. 307-310.
- [WOO 81] WOODS, R.E.; GONZALEZ, R.C. Real Time Digital Image Enhancement. Proceedings of IEEE, New York, 69(5): 643-654, May, 1981.

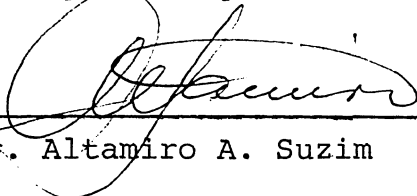
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
Pós-Graduação em Ciência da Computação

CV2D - Operador de convolução  
para processamento de imagens  
em tempo real

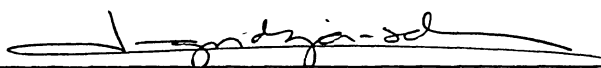
Dissertação apresentada aos Srs.



Prof. Tiaraju V. Wagner

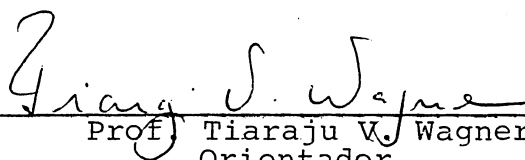


Prof. Dr. Altamiro A. Suzim



Profa. Dra. Ingrid E.S. Jansch Pôrto

Visto e permitida a impressão  
Porto Alegre, ...04../.07/.90..



Prof. Tiaraju V. Wagner  
Orientador



Profa. Ingrid E. S. Jansch Pôrto  
Coordenador do Curso de Pós-Gra-  
duação em Ciência da Computação