

TESE

1193

UNIVERSIDADE FEDERAL DE ITAJUBÁ

PROJETO SISTEMA DE RECUPERAÇÃO DE DADOS DO SENSOR  
MSS - MULTI SPECTRAL SCANNER PARA O SATÉLITE LANDSAT 1,2 e 3

ANDRE OTTOBONI DIAS

ITAJUBÁ - 23/09/2002



Ministério da Educação



**UNIFEI**

**Universidade Federal de Itajubá**

Programa de Pós-Graduação em Engenharia Elétrica

**PROJETO SISTEMA DE RECUPERAÇÃO  
DE DADOS DO SENSOR  
MSS – MULTI SPECTRAL SCANNER  
PARA O SATÉLITE LANDSAT 1,2 e 3**

**Área de concentração:**

Automação e Sistemas Elétricos  
Industriais

**Orientador:**

Germano Lambert Torres

**Co-orientador:**

Paulo Roberto Martins Serra

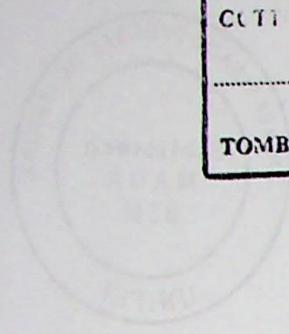
Dissertação apresentada à  
Escola Federal de Engenharia de  
Itajubá, para obtenção do título de  
Mestre em Engenharia Elétrica.

**André Ottoboni Dias**

Matricula : 10501

ITAJUBÁ - 23/ 09/ 2002

CLASS. 528.8(043-2)  
CUT R. 2541 p  
TOMBO. 1193



UNIBR

Universidade Federal de Brasília  
Programa de Pós-Graduação em Engenharia de Software

PROJETO SISTEMA DE RECUPERAÇÃO  
DE DADOS DO SENSOR  
MSS - MULTISPECTRAL SCANNER  
PARA O SATELITE LANDSAT 1, 2 e 3

Área de Engenharia de Software  
Laboratório de Engenharia de Software  
UNB - Brasília - DF  
1993

BRASIL  
UNIVERSIDADE FEDERAL DE BRASÍLIA  
UNIBR

## Agradecimentos

Ao professor Dr. Germano Lambert Torres pela orientação e condução deste trabalho .

Ao Eng. Paulo Roberto Martins Serra pela contribuição e apoio dado, incentivando a continuidade deste trabalho.

A todos os funcionários do DGI-INPE (Departamento de Geração de Imagem), e da UNIFEI (Universidade Federal de Itajubá) que direta ou indiretamente contribuirão para realização deste trabalho.

À minha esposa e filhos que compreenderam a importância e objetivo do meu trabalho

E finalmente a Deus que guia nossas vidas e nossos passos e sem ele, nada seria possível!

1.4	Organização do Trabalho	12
<b>CAPÍTULO II: O satélite LANDSAT</b>		
2.1	Descrição do Satélite Landat	14
2.1.1	Características do Satélite Landat	16
2.1.2	Orbita do Satélite Landat	18
2.2	Sistema Remotador MSS (Multi-Spectral Scanner System)	21
2.2.1	Resolução de Resolução do Sistema MSS	24
2.2.2	Descrição Física do Sensor Remotador Sensor MSS	25
2.2.3	Princípio de Funcionamento	26
2.2.4	Formação da Imagem	26
2.2.5	Alinhamento das Imagens de Sensoriamento Remoto MSS	27
2.3	Princípio das imagens Landsat	28
2.3.1	Formas Principais de Exatidão de Posição do NPE. Questões	28
2.4	Laboratório de Processamento e Geração de Imagens DGI	28
<b>CAPÍTULO III: O Projeto MSS</b>		
3.1	Sistema Análogo. Descrição de Processamento dos Dados -APD	31
3.2	Algoritmo de Projeto MSS	32
3.3	Formas para Implementação do Projeto	35
3.4	Identificação e Modelagem dos Problemas	36
3.5	Descrição do Sistema AMPEX PRISON	36
3.6	Características Físicas de cada um dos Sensores	36
3.6.1	Algoritmo de Correção da Formatação dos Dados MSS	37
3.7	Análise de Precisão da diferença de Fase (DFA) de Precisão angular (PRON)	37

Índice.....	2
Resumo.....	5
Abstract.....	6
<b>CAPÍTULO 1: Introdução.....</b>	<b>7</b>
1.1 Aspectos Gerais do Sensoriamento Remoto.....	7
1.2 Sensores.....	7
1.3 Tipos de Sensores utilizado no Sensoriamento Remoto.....	8
1.4 Radiômetro Imageador [Scanner].....	8
1.5 Razão da Escolha do Trabalho de Dissertação.....	9
1.6 Organização do Trabalho.....	10
<b>CAPÍTULO 2: O Satélite LANDSAT.....</b>	<b>11</b>
2.1 Descrição do Satélite Landsat.....	11
2.1.1 Características do Satélite Landsat.....	11
2.1.2 Orbita do Satélite Landsat.....	12
2.2 Sistema Imageador MSS (Multi-Spectral Scanner System).....	13
2.2.1 Seqüência de Varredura do Espelho MSS.....	14
2.2.2 Disposição Física do Plano Focal dos Sensores MSS.....	15
2.2.3 Formação do Pixel.....	15
2.2.4 Formação da Imagem.....	15
2.2.5 Utilização das Imagens de Sensoriamento Remoto MSS.....	17
2.3 Recepção das Imagens Landsat.....	18
2.3.1 Funções Principais da Estação de Rastreo do INPE Cuiabá.....	19
2.4 Laboratório de Processamento e Geração de Imagens DGI.....	20
<b>CAPÍTULO 3: O Projeto MSS.....</b>	<b>21</b>
3.1. Sistema Antigo Desativado de Processamento dos Dados –MSS.....	21
3.2 Surgimento do Projeto.MSS.....	22
3.3 Filosofia para Implementação do Projeto.....	23
3.4 Identificação e Modelagem dos Problemas.....	23
3.5 Descrição do Gravador AMPEX FR1928.....	24
3.6 Formato dos dados digitais de Saída do Gravador.....	25
3.6.1 Diagrama Temporal do Formato dos Dados MSS.....	26
3.7 Análise do Problema da diferença de Fases (Skill) do Gravador AMPEX FR1928...	27

<b>CAPÍTULO 4: Hardware do Projeto MSS</b> .....	<b>28</b>
4.1 Filosofia de Implementação do Hardware.....	28
4.1.1 Especificando a Interface de Aquisição de Dados .....	28
4.1.2 Arquitetura do Hardware - Interface PCI_DIO-32HS.....	29
4.1.3 Protocolo ou Modos de Aquisição - Interface PCI_DIO-32HS.....	30
4.1.4 Definindo o Protocolo de Aquisição da Interface PCI_DIO-32HS.....	32
4.1.5 Descrição do Protocolo Burst Mode da Interface. PCI_DIO-32HS.....	32
4.2 Interface Formatador de Dados MSS - IFDM.....	33
4.2.1 Descrição da Memória FIFO.....	34
4.2.2 Descrevendo o Hardware da Interface IFDM .....	35
4.2.3 Descrição da Lógica de Controle da Interface IFDM e sinais de Handshake.....	36
4.2.4 Fotos do Hardware – Interface IFDM.....	39
4.3 Simulador TPG – (Test Pattern Generation).....	40
4.3.1 Formato dos Padrões Gerados pelo TPG – (Test Pattern Generation).....	41
<b>CAPÍTULO 5: Software de Aquisição de Dados MSS</b> .....	<b>42</b>
5.1 Fases do Desenvolvimento do Software do Projeto MSS.....	42
5.2 Software de Aquisição Dados Brutos.....	43
5.3 Formato dos Dados Brutos Gravados pelo Software de Aquisição.....	44
5.4 Definições das Funções da Interface PCI-DIO-32HS.....	45
5.5 Descrição do Software de Aquisição de Dados.....	46
5.6. Software de Extração dos Dados dos sensores a Partir dos Dados Brutos Obtidos.	48
5.6.1 Relatório dos dados Gravados – Report.....	49
5.7 Formação das Imagens Espectrais ou Bandas do Sistema MSS.....	50
5.7.1 Software de Composição das Bandas sem correção.....	52
5.7.2 Exemplo de Composição de Imagens MSS.....	54
5.8. Correção Geométrica e Radiométrica dos Dados.....	54
5.8.1 Correção Geométrica dos Dados.....	54
5.8.2 Correção Radiométrica dos Dados.....	55
<b>CAPÍTULO 6: Resultados Práticos</b> .....	<b>58</b>
6.1 Fotos do Hardware do Sistema de Aquisição .....	58
6.2 Dados de Imagens Adquiridas pelo Sistema.....	60

6.3	Evolução das Imagens ou Cenas Obtidas do Banco de Dados .....	61
6.4	Análise dos Resultados.....	69
<b>CAPÍTULO 7: Plano de Operacionalidade do Projeto MSS.....</b>		<b>70</b>
7.1	Levantamento da Base de Dados .....	70
7.2	... Identificação dos Campos da Planilha de Controle .....	71
<b>CAPÍTULO 8: Trabalhos Futuros Utilizando o Projeto MSS.....</b>		<b>72</b>
<b>CAPÍTULO 9: Conclusões Finais.....</b>		<b>73</b>
Referências Bibliográficas.....		74
Imagens RGB – Coloridas adquiridas pelo projeto MSS .....		75
Anexos A _ Esquema Elétrico do Hardware da Interface –IFDM.....		A1
Anexos B _ Listagem do Software de Aquisição – Visual C++.....		B1
Anexos C - Listagem do Software de Formação das Bandas – Basic.....		C1

## Resumo

O presente trabalho trata-se, de um desenvolvimento de uma interface digital de conversão de sinais de imagem do sensor MSS (Multi Spectral Scanner), da série de Satélites LANDSAT I,II e III, que estão armazenadas em fitas magnética digitais de alta densidade HDDT (High Density Digital Tape) no banco de dados do DGI (Departamento de Geração de Imagens), do INPE-CP (Instituto Nacional de Pesquisas Espaciais Cachoeira Paulista – SP). Este acervo atualmente se encontra-se na forma de película fotográfica já reveladas, o sistema proposto deverá ter a capacidade de transferir os dados armazenamento nas fitas de alta densidade para um novo banco de dados na forma digital, em mídia atualizada tais como fitas DLT's ou DAT e CDROM

O Instituto Nacional de Pesquisas Espaciais [INPE], é responsável pela recepção, arquivamento, processamento e disseminação das imagens de satélites de sensoriamento remoto no Brasil, adquirindo sistematicamente dados dos satélites da série LANDSAT [1] desde 1973. Neste período constituiu uma base de dados contendo um importante acervo de imagens de satélite sobre o território brasileiro e a maior parte dos países da América do Sul.

As atividades de recepção, arquivamento, processamento e disseminação destas imagens foram iniciadas pelo INPE logo após o lançamento do satélite LANDSAT-1 [1]

Todas as imagens adquiridas pelos sensores dos satélites LANDSAT 1,2 e 3, cobrindo o período de 1973 a 1984 estão armazenadas no INPE em Cachoeira Paulista . Esse acervo está adquirindo um valor cada vez maior, tendo em vista o crescente interesse em estudos da evolução do meio ambiente.

Os dados armazenados desde 1973, gravados em fitas digitais magnéticas de alta densidade (HDDT's), com o tempo vem se deteriorando-se, correndo-se o risco de não poderem ser reproduzidas. É portanto imperativo, o salvamento e transferência, destes dados em outro equipamento ou meio de tecnologia mais atual e com maior capacidade de armazenamento que viabilize esta transferência de dados.

Este Trabalho visa desenvolver meios para viabilizar a automização de equipamentos para o tratamento, armazenagem e processamento das informações existentes no INPE.

## ABSTRACT

The current work, describes the system project of digital conversion image tapes of MSS (Multi Spectral Scanner) from the satellites 1,2,3 series. The High density Magnetics Digital Tapes (HDDT's) are stored at INPE-DGI data base. Nowadays, such base contain at DGI is only film form, but in digital form.

The purpose of this project, is to transcribe the HDDT's tape to a new archive compatible media likes digital tapes (DLT) or the CDROM.

INPE is in charge of the reception and archive many the satellite remote sensing images covering Brazil since 1973. During this time INPE built the first data base covering brazilian territory. All images aquired from Landsat Satellite1,2,3 serie cover the period between 1973 at 1984 and are stored in a data base at INPE-CP. This historic data base has a great value due to cientific community for studies and research evolution environment.

The database was stored in HDDT tapes since 1973, recorded in magnetic tapes (HDDT's) trought of this time has been deterioration. This does mean that the data stored may be not reproduce. For this reason must be such in another new media with high capacity of storage such DLT tapes or CDROM.

This work aims to develop a way to be able a equipament automatization to process and stored existent data information at INPE.

## CAPÍTULO 1 : Introdução

### 1.1 Aspectos Gerais do Sensoriamento Remoto

Os satélites tornam-se, dia a dia, mais importantes em nossas vidas e suas finalidades e objetivos vão gradualmente passando da pesquisa pura á aplicada dentro do vasto campo da pesquisa espacial e do meio ambiente.

O sensoriamento remoto é uma técnica, que é utilizada como mais uma ferramenta para o estudo dos problemas da ciência, ligadas principalmente ao recursos naturais . A agricultura que lidam com recursos naturais, vem se beneficiando com a técnica do sensoriamento remoto, pois se utilizam desta para estudar e solucionar problemas que necessitam ser tratados harmônicamente e repetitivamente dentro de uma visão global.

Com a utilização de dispositivos sensores, colocados a bordo de aviões e satélites, estes são utilizados para obter informações de objetos e fenômenos na superfície da terra, mediante a detecção das variações do fluxo de radiações eletromagnéticas de elementos e objetos situados na superfície terrestre.

Considerando-se que as propriedades dos alvos naturais se manifestam de forma característica no fluxo de radiação eletromagnética que refletem e emitem, tais imagens, quando obtidas em faixas espectrais adequadas, permitem que um máximo de discriminação entre os alvos e sua vizinhança seja conseguido um meio rápido, econômico e eficiente para detecção daqueles na área analisada.

O sensoriamento remoto apresenta três importantes característica que são: **rapidez**, **repetitividade** e poder de análise **multiespectral** [2].

Dentro deste conceito pode-se conceituar o sensoriamento remoto como um conjunto de atividades, cujo objetivo na caracterização das propriedades de alvos naturais, através da detecção, registro e análise do fluxo de energia radiante, por eles refletido ou emitido.

### 1.2 - Sensores

A Radiação Eletromagnética [REM] refletida ou emitida por um alvo qualquer, desde que convenientemente registrada e analisada, contém uma grande quantidade de informações . Assim a REM refletida ou emitida por um alvo, sob certas circunstâncias, após incidir sobre o olho humano pode ser transformada em impulso nervoso, que é interpretado pelo cerebro, gerando a sensação de visão. Entretanto o olho humano é um sensor remoto capaz de responder a uma faixa extremamente estreita do espectro eletromagnético (aprox. 400 a 700 nm).

Com o desenvolvimento tecnológico, o homem conseguiu ampliar a sua capacidade de "sentir" a REM, desde comprimentos de onda extremamente curtos [raios cósmicos] até

comprimentos de onda de centenas de metros, construindo dispositivos sensores que operam em grandes parte do espectro eletromagnético .

Assim pode –se dizer que um sensor remoto é um dispositivo capaz de responder á REM de um determinada faixa do espectro eletromagnético, registrá-la e gerar um produto numa forma adequada para ser interpretada pelo usuário.

Existem duas grandes categorias de sensores remotos : ativos e passivos . Os sensores ativos são aqueles que possuem fonte própria de energia, registrando a energia por eles emitida e refletida pelo alvo . Nesta categoria estão incluídos os radares e os lasers, por exemplo . Os sensores passivos não possuem fonte própria de energia, utilizando apenas a energia emitida ou proveniente de outras fontes [ex.,o sol] e refletida pelo o alvo.

### 1.3 Tipos de Sensores utilizados em Sensoriamento Remoto

Existem vários tipos de sensores utilizado, no sensoriamento remoto, mas como o trabalho e relativo ao sensor da série de satélites LANDSAT [1], este trabalho será focado em descrever o radiometro imageador (scanner) utilizado pelo satélite LANDSAT [1] .

Os tipos mais comuns de sensores são:

*Radiômetros*

*Sensores Fotográficos*

*Radiometro Imageador (Scanner)*

*Radar*

### 1.4 Radiometro Imageador (Scanner)

Se no caminho ótico do coletor de um radiômetro instalado a bordo de uma plataforma [avião, satélite, etc.], for colocado um espelho, que gira ou oscila na direção perpendicular ao deslocamento da plataforma, o elemento de resolução no terreno se deslocará, gerando um linha de varredura . Se o movimento do espelho for sincronizado com a velocidade da plataforma, a cada rotação ou oscilação do espelho, uma nova linha adjacente de varredura será gerada. Desta forma, pode –se obter uma imagem do terreno situado abaixo da plataforma.

Na ilustração da fig. 1.4, apresenta –se o esquema de um radiômetro imageador, neste caso cada linha de varredura pode –ser armazenada através de um gravador de fita magnética ou gravada sobre um filme fotográfico . No caso da gravação sobre o filme, a energia radiante proveniente de cada elemento de resolução no terreno é convertida, através do detetor e o sistema de processamento eletrônico, num sinal elétrico, que por sua vez, modula a intensidade luminosa de uma pequena lâmpada . A luz dessa lâmpada varre o filme fotográfico por meio de um pequeno espelho, que gira fixo ao eixo do espelho principal. O deslocamento do filme é feito de tal forma que, a cada rotação ou oscilação do espelho uma nova linha de varredura é gravada sobre ele.

Na maioria dos radiômetros, a área do detetor define o campo instantâneo de visada (resolução) do sistema, se um filtro é colocado na trajetória da radiação para restringir esta radiação a um intervalo de comprimento de onda escolhidos, pode-se transformar um radiômetro imageador em um multiespectral, desta maneira o sistema seria capaz de observar a mesma cena em várias faixas espectrais através de separadores de feixe [prisma, redes de difração, etc..]. Desta forma detetores separados, localizados num conjunto de detetores, observam o mesmo elemento de resolução da cena em diferentes faixas espectrais.

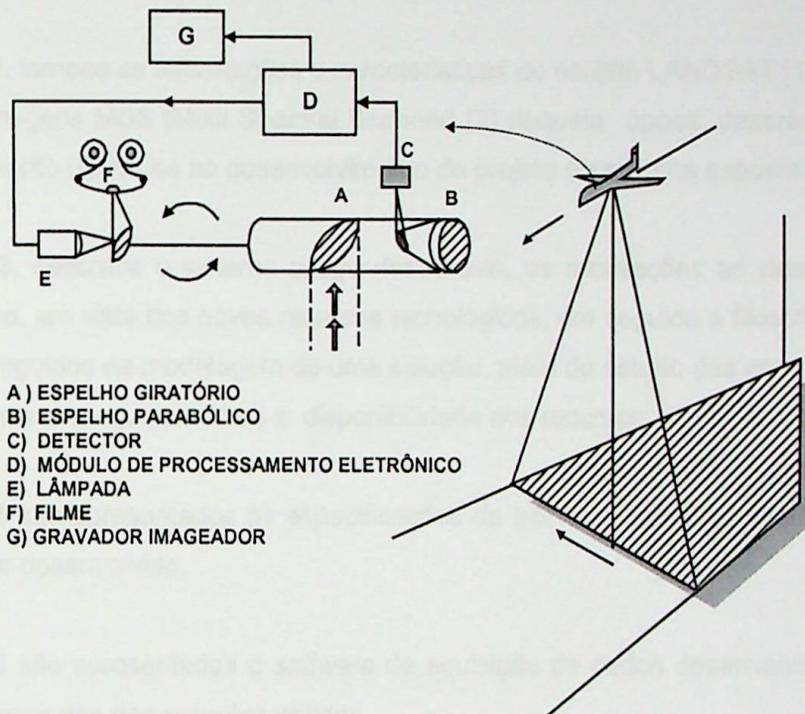


Fig 1.4 Ilustração de um Radiômetro Imageador a Bordo de uma Aeronave.[2]

### 1.5 Razão da Escolha do Trabalho de Dissertação

O tema desta dissertação foi escolhido devido á oportunidade de desenvolvimento de um projeto de engenharia na área de automação e aquisição de dados que oferecia em atender a uma necessidade crescente da área em que o autor está alocado (DGI – Departamento de geração de Imagens de Satélites), ao mesmo tempo este trabalho recupera de volta o acervo de imagens de satélites históricas da superfície do solo brasileiro, que se encontrava em estado de decomposição . Com o novo sistema, permitirá que se crie uma nova base de dados de imagens digitais geográficas de satélites, cobrindo o período de 1973 á 1984, que será disponibilizado pelo INPE via rede *ethernet*, para comunidade científica brasileira, para estudos da evolução do meio ambiente terrestre brasileiro.

## 1.6 Organização do Trabalho

Este Trabalho foi dividido em nove capítulos, onde são abordados aspectos do projeto do sistema de recuperação de dados de imagens do Satélite LANDSAT 1, 2 e 3 [2], desde a proposta inicial ao desenvolvimento, até os resultados finais alcançados.

O capítulo 1 fornece uma noção do que é o sensoriamento remoto, quais os tipos de sensores utilizáveis em sensoriamento .

O capítulo 2, fornece as informações e características do satélite LANDSAT [1] [2], que coletava os dados de imagens MSS [Multi Spectral Scanner] [2] daquela época, descrevendo o formato dos dados que serão utilizados no desenvolvimento do projeto do sistema exposto neste trabalho.

O capítulo 3, descreve o sistema antigo desativado, as motivações ao desenvolvimento de um sistema novo, em vista dos novos recursos tecnológicos, em seguida a filosofia de implementação do projeto seguidos da modelagem de uma solução, além do estudo das complexidades técnicas e dos problemas dos equipamentos e disponibilidade dos recursos .

O capítulo 4 são apresentados as especificações da interface utilizada, seguido da apresentação do *Hardware* desenvolvido.

O capítulo 5 são apresentados o *software* de aquisição de dados desenvolvido, e a formação de imagens a partir dos dados brutos obtidos.

O capítulo 6 descreve os resultados práticos obtidos, com os dados simulados e reais de imagens obtidas de regiões da superfície terrestre brasileira da época de 1977.

O capítulo 7 mostra a partir do projeto pronto, um plano de operacionalidade do sistema, descrevendo os procedimentos de classificação e execução do salvamento das imagens contidas no banco de dados.

O capítulo 8, os trabalhos a ser realizados futuramente utilizando o projeto MSS, onde descreve a inclusão deste banco de dados no CDSR ( Centro de Dados de Sensoriamento Remoto) do INPE.

E finalmente no capítulo 9, encontram-se as conclusões finais, com análise dos objetivos atingidos.

## CAPITULO 2: O Satélite LANDSAT

### 2.1 Descrição do Satélite Landsat

O sistema LANDSAT [1] foi desenvolvido com objetivo de permitir a aquisição de dados espaciais, espectrais e temporais sobre a superfície da terra, de forma global, sinóptica e repetitiva. Os satélites Norte Americanos LANDSAT [1] constituem a primeira série de satélites de Sensoriamento Remoto, que teve início em 23 de julho de 1972 com o lançamento do ERTS (*Earth Resources Technology Satellite*), posteriormente chamado de LANDSAT-1. Este programa científico, sob responsabilidade da NASA, lançou as bases para o Sensoriamento Remoto Espacial. Deslocando-se em uma órbita quase polar e heliossíncrona, os satélites LANDSAT [1] permitem que os dados sobre uma mesma região da terra sejam adquiridos a cada 18 dias, em condições semelhantes de iluminação.

#### 2.1.1 Característica do Satélite LANDSAT

O LANDSAT [1] foram instalados os seguintes instrumentos ilustrado na Fig.[2.1.1] :

Sistema de Varredura Multiespectral [MSS]

Vidicom de Feixe de Retorno [RBV]

Gravadores de Fitas de Banda Larga [WBRT]

Sistema de Coleta de Dados para recepção de Plataformas de Coletas de Dados [PCD].

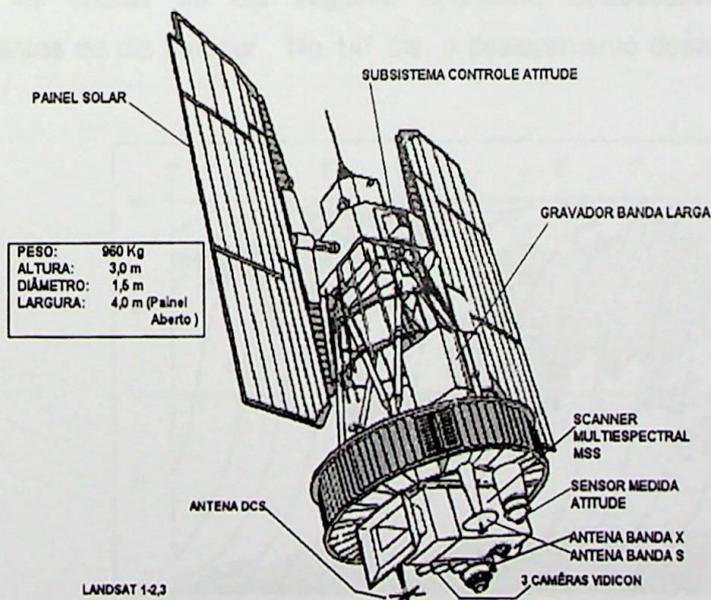
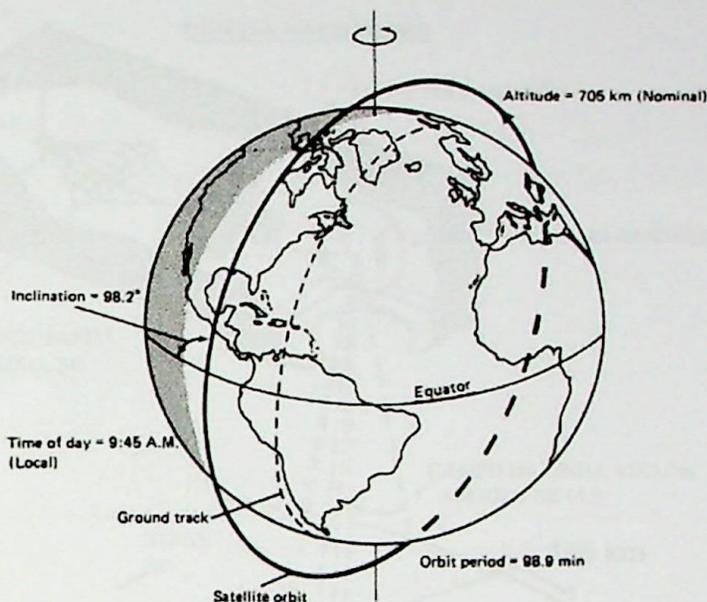


Fig 2.1.1 Instrumentos a bordo Satélite do Landsat [1]

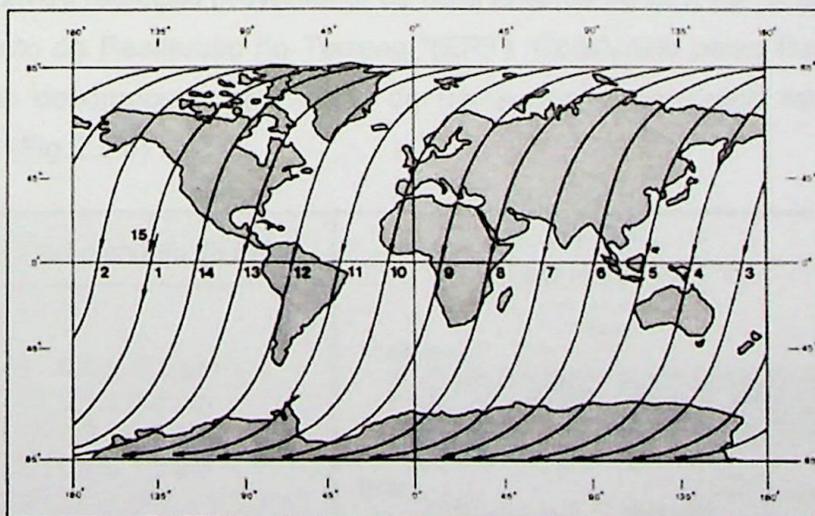
## 2.1.2 Órbita do Satélite LANDSAT [1]

O satélite LANDSAT [2] desloca-se em órbita geocêntrica, circular e polar com períodos de 98,9 (min) e a 705 (km) de altitude [Fig. 2.1.2.a]. A órbita é heliossícrona, isto é, o ângulo sol-terra-satélite permanece constante ( $98,2^\circ$ ), o que garante condições semelhantes de iluminação, ao longo do ano, na área imageada.



**Fig 2.1.2.a - Ilustração Órbita do Satélite LANDSAT [1].**

Deslocando-se do norte para sul, na parte iluminada, o satélite cruza o equador a cada 98,9 minutos, neste intervalo de tempo, a Terra gira 2760 Km (no equador), que é a distância entre as projeções de duas órbitas consecutivas. Ao fim de 24 (h), 14 órbitas são realizadas [Fig.2.1.2.b] entretanto, as órbitas do dia seguinte aparecem deslocadas 160 km em relação às correspondentes do dia anterior. No 14º dia, o deslocamento desaparece e recomeça um novo ciclo.



**Fig. 2.1.2.b – Diagrama das Órbitas de Passagens Diárias.**

## 2.2 Sistema Imageador -MSS [Multi-Spectral Scanner]

O Sistema Imageador MSS [Fig. 2.2.a], cuja a varredura é perpendicular á trajetória, é conseguida por meio de um espelho oscilante, com um campo de visada de  $14,9^\circ$ . O sistema permite a obtenção de linhas de varredura, na superfície da terra de aproximadamente 185 km.

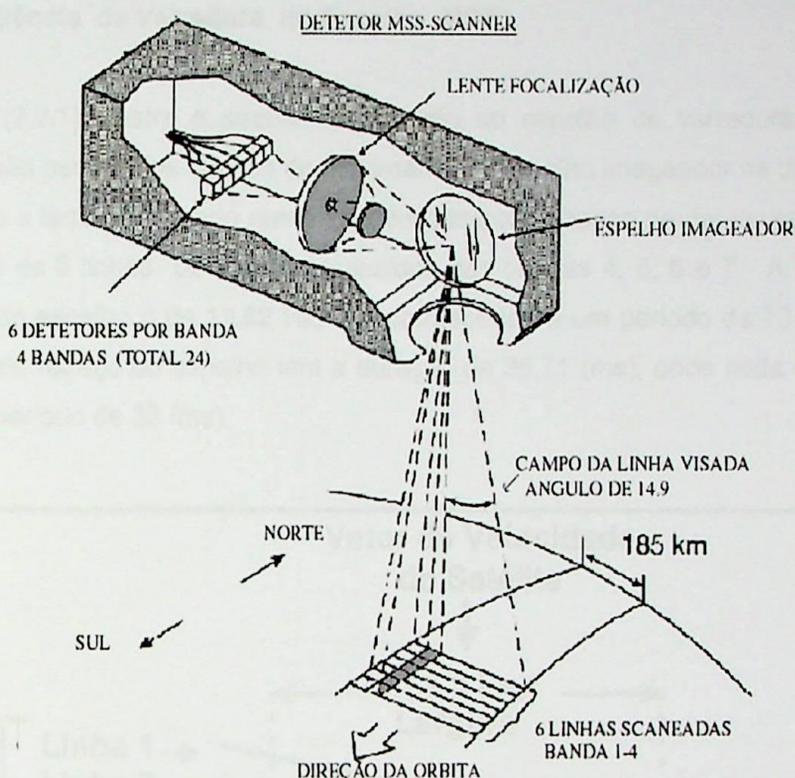


Fig. 2.2.a – Sistema Imageador de Varredura- MSS (Multi Spectral Scanner)[1]

A radiação proveniente da cena, após a reflexão no espaço oscilante, é focalizada pelo sistema óptico através de lentes, sobre uma matriz de 24 terminais de fibras ópticas, de tal forma que, cada terminal recebe o fluxo de radiação proveniente de uma área de  $79\text{ m} \times 79$  na superfície, área esta denominada “Elemento de Resolução no Terreno “(ERT). Conduzido pelas fibras ópticas, o fluxo de radiação é então decomposto, por meio de filtros, em 4 conjuntos nas seguintes faixas espectrais ou canais (Fig.2.2.b) :

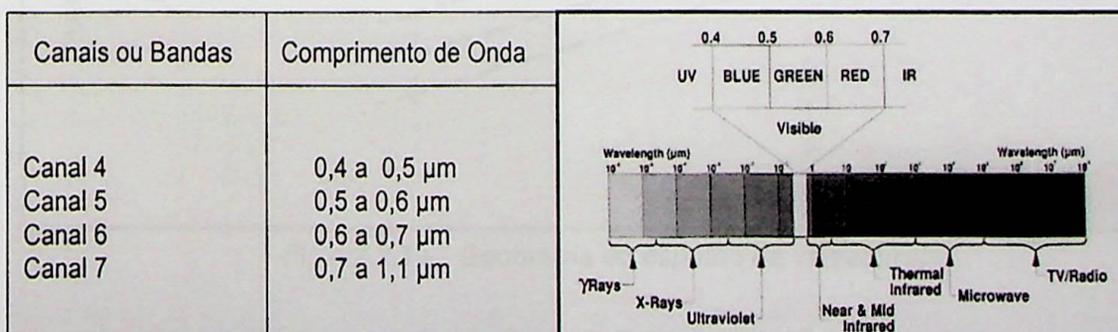


Fig 2.2.b. Diagrama das Bandas ou Canais no Diagrama Espectral [2]

Este fluxo ao atingir os detetores, em número de 6 por canal, a radiação é então convertida em um sinal elétrico de intensidade proporcional à radiância da porção da superfície incluída no ERT correspondente. O sinal que pode variar entre 0 e 4 volts, é então quantizado entre níveis que assumem valores de 0 a 63, sendo em seguida transmitido à terra, ou eventualmente gravado para transmissões posteriores.

### 2.2.1 Seqüência de Varredura do Espelho MSS

Na figura (2.2.1), ilustra a seqüência padrão do espelho de varredura para um detetor, os detetores são percorridos através do movimento do espelho imageador na direção oeste para leste em relação a terra, resultando numa imagem para cada banco de detetores, no que resulta numa informação de 6 linhas de vídeos produzidos nas bandas 4, 5, 6 e 7. A freqüência nominal da varredura do espelho é de 13.62 Hz, que corresponde a um período de 73,42 (ms), o período de varredura do retraço do espelho tem a duração de 36,71 (ms), onde cada cena MSS é adquirida durante o período de 33 (ms).

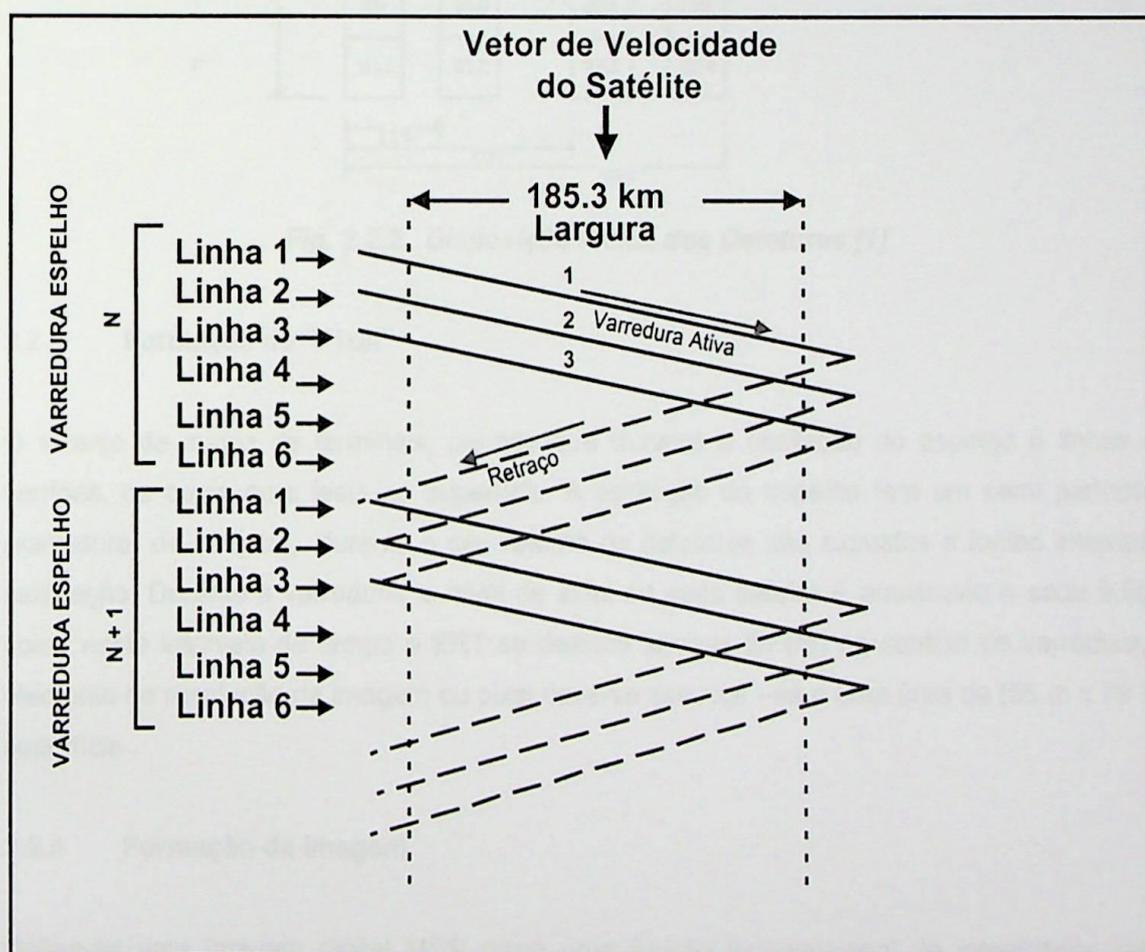


Fig - 2.2.1 - Geometria do espelho de varredura[1]

### 2.2.2 Disposição Física do Plano Focal dos Sensores MSS

A figura (2.2.2), ilustra a disposição física do plano focal dos detetores individualmente, ou seja a radiação produzida pela luz, é guiada e amostrada no plano focal para cada detetor ou sensor individualmente numa seqüência ou ordem programada sensores [S1....S24], e labels [A,B...F], formando se assim a banda espectral .

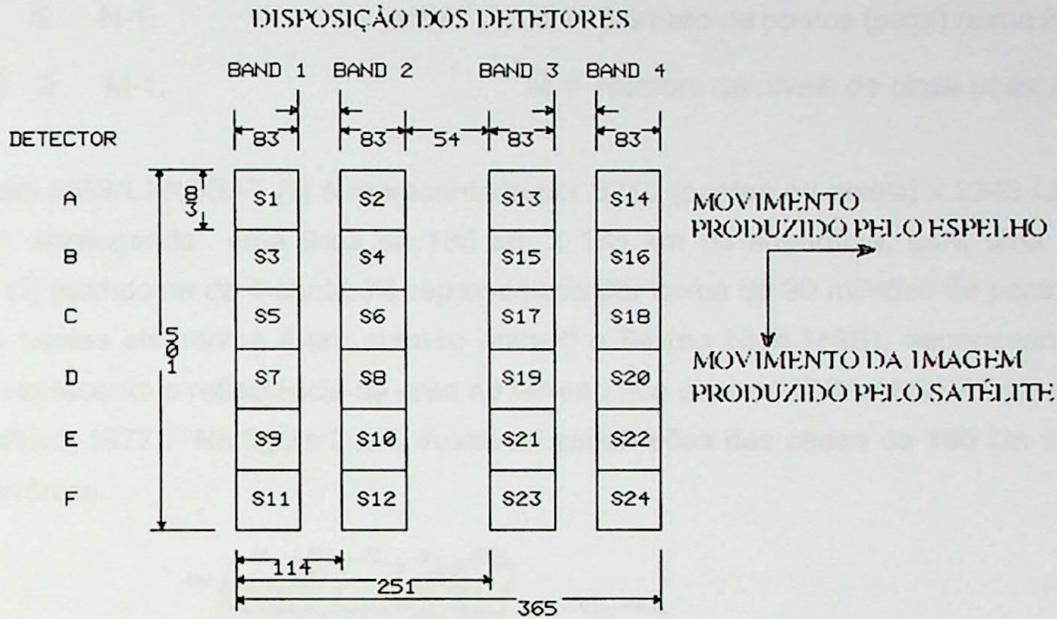


Fig. 2.2.2 Disposição Física dos Detetores [1]

### 2.2.3 Formação do "Pixel"

O arranjo da matriz de terminais, permite que durante a oscilação do espelho 6 linhas sejam varridas, de oeste para leste na superfície. A oscilação do espelho tem um semi período ativo (varredura) de 33 (ms), durante o seu retorno os detetores são expostos a fontes internas para calibração. Durante a varredura, o nível de sinal de cada detetor é amostrado a cada 9,95(  $\mu$ s); como neste intervalo de tempo o ERT se desloca apenas 56 (m) no sentido da varredura, cada elemento de resolução da imagem ou pixel deve-se associar -se a uma área de (56 m x 79 m), na superfície .

### 2.2.4 Formação da Imagem

Define-se uma imagem digital MSS como uma função bidimensional de intensidade luminosa  $f(x,y)$ , onde  $x$  e  $y$  são coordenadas espaciais e  $f$  em qualquer  $(x, y)$  é proporcional ao brilho ou ao nível de cinza da imagem naquele ponto . Uma imagem digital é uma imagem  $f(x,y)$  discretizada tanto nas coordenadas espaciais quanto ao brilho, seja estas grandezas só assumem valores inteiros (Rosenfeld, 1969;Duda and Hart, 1973;Dutra et alii, 1981) e não-negativos .

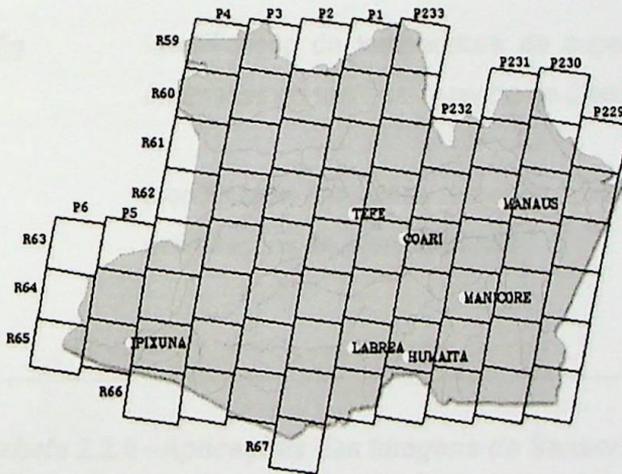
Se considerar a imagem digital como uma matriz, onde cada elemento da imagem será um elemento da matriz, ou seja, os índices de linhas e colunas identificam especialmente um ponto, e o valor do elemento correspondente na matriz identifica seu nível de cinza. Embora o tamanho desta matriz possa variar, geralmente ela possui 512 linhas x 512 colunas, podendo cada elemento assumir 256 níveis de cinza (Dutra et Alli, 1981)

Uma imagem digital pode –se então ser representada por:

$$0 \leq x,y \leq N-1, \quad \text{onde: } N = \text{Número de pontos (pixel) numa linha}$$

$$0 \leq f(x,y) \leq M-1, \quad M = \text{Número de níveis de cinza possível}$$

Uma imagem MSS/LANDSAT [1] é representada por 3240 (pontos ou pixels) x 2340 Linhas . para cada canal, abrangendo uma área de 185 km x 185 km na superfície, para uma única cena LANDSAT [1] (composta de 4 canais) é representada por cerca de 30 milhões de pontos ou pixels e cada um destes elementos é um numero entre 0 e 64 (no caso MSS), denominado níveis de cinza, que representa a reflectância de área no terreno que para o MSS/LANDSAT [1] é de 56 m x 79 m (Sheffield, 1977). Na figura 2.2.4 ilustra as subdivisões das cenas de 185 km x 185 km da região amazônica.



**Fig 2.2.4 - Subdivisões das cenas ou imagens da Região Amazonica [2]**

### 2.2.5 Utilização das Imagens de Sensoriamento Remoto MSS.

As imagens geradas pelo sensor MSS, são utilizadas em inúmeras aplicações e possibilidades do estudo ambiental da terra, ilustrado na tabela 2.2.5 abaixo :

<b>Campo de Atuação</b>	<b>Aplicações</b>
<u>Agricultura</u>	Identificação de tipos de cultura e plantio
<u>Geologia</u>	Identificação de recursos minerais, solo e relevo
<u>Recursos Florestais</u>	Identificação de níveis de vegetação, áreas de florestamento, áreas de desmatamento, discriminação de espécie e queimadas
<u>Hidrologia</u>	Identificação de bacias, curso e entalhes de rios e nascentes, drenagem e áreas aquáticas .
<u>Oceanografia</u>	Identificação da temperatura da superfície do mar, circulação do mar e anomalias do mar (ex. mancha de óleo)
<u>Urbanismo</u>	Identificação de áreas urbanas, planejamento de vias de transporte e delimitações de áreas urbanas

**Tabela 2.2.5 –Aplicações das Imagens de Sensoriamento Remoto - MSS**

### 2.3 Recepção das Imagens Landsat [1]

O Sistema de recepção de dados do LANDSAT [2] do INPE , encontra-se instalado em Cuiabá –MT, devido ao fato de ser um local que mais se aproxima do centro geográfico da América do Sul e permite o recobrimento da maioria dos países da América Sul (Fig. 2.3.a)

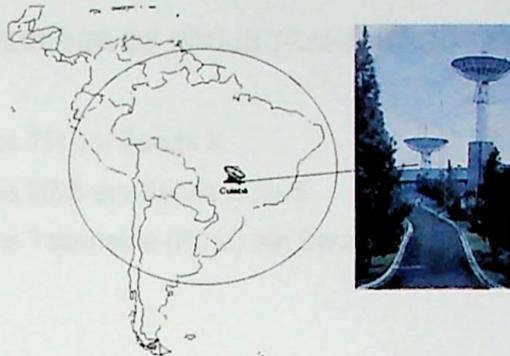


Fig. 2.3.a) Área de Cobertura da Estação de Cuiaba-MT [5]

O sistema da estação solo Cuiabá, através de sua antena de rastreamento, tem a função de rastrear e receber e gravar os sinais transmitidos pelo satélite em bandas S ou X, bem como os dados de telemetria, modulação por código de pulso PCM (*Pulse Code Modulation*) tais como altitude, ao longo de sua órbita. No diagrama da Fig 2.3.b, é ilustrado, como é gravado os sinais recebidos pelo satélites.

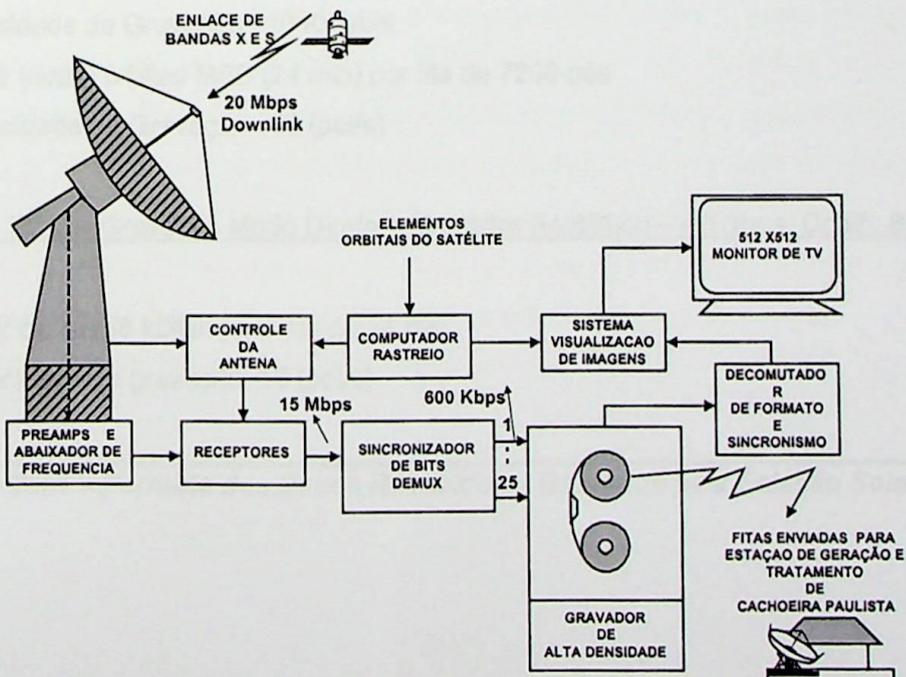


Fig 2.3.b) Diagrama de Recepção e Gravação dos Dados MSS da Estação[5]

### 2.3.1 Funções Principais da Estação de Rastreamento do INPE – Cuiabá –MT

A tabela 2.3.1 ilustrada abaixo descreve os formatos dos dados recebidos e gravados pela estação solo.

#### RASTREAR E RECEBER OS SINAIS TRANSMITIDOS EM BANDA S OU X

85 Mbit/sec – Dados TM em Banda X	(LANDSAT 5 )
15 Mbit/sec – Dados MSS em Banda X ou S	(LANDSAT 3,4 e 5 )
8 kbit/sec -- Dados Telemetria (PCM) em Banda S	(LANDSAT 3,4 ou 5)

#### GRAVAR E REPRODUZIR OS SINAIS TM, MSS E PCM – EM TEMPO REAL

##### Dados TM ou MSS em Gravador Digital de Alta densidade – 28 Trilhas – “Serial”

- Densidade de Gravação: 33 kbit/pol.
- Até 2 (duas) órbitas TM (15min) ou 6 (seis) órbitas MSS (80 min) por fita 9200 pés
- Velocidade de Gravação de 127,4 pol/seg para TM e 22,6 pol/seg para MSS

##### Dados MSS Multiplexados em Gravador Paralelo –28 Trilhas – Codificação MILLER

- Densidade de Gravação : 10 Kbit/pol
- Até 2 (duas) órbitas MSS (24 min) por fita de 7200 pés
- Velocidade de Gravação : 60 (pol/s)

##### Dados PCM – Gravação Modo Direto – Gravador Analógico – 4 Trilhas Codif.. Bi-OL

- Até 2 órbitas (8 kbits/s) por fita de 24 pés
- Velocidade de gravação: 15 (pol/s)

**Tabela 2.3.1 – Formato dos Dados Recebidos e Gravados pela Estação Solo de Cuiabá.**

## 2.4 Laboratório de Processamento e Geração de Imagens -DGI.

Em 1973, foi construído em Cachoeira Paulista -SP o Laboratório de Produção de Imagens (hoje Divisão de Geração de Imagens – DGI), destinado a processar os dados de satélites de sensoriamento remoto, cuja recepção era iniciada pela estação solo de recepção de Cuiabá.

Na área do INPE de Cachoeira Paulista, o DGI figura como a mais antiga das instalações e conta hoje com a seguinte organização:

- Secretaria.
- Laboratórios.
- Processamento Eletrônico.
- Processamento Fotográfico.
- Controle de Qualidade.
- Atendimento aos Usuários.
- Planejamento e Controle de Produção e Materiais.
- Grupo de Apoio e Desenvolvimento de Software.
- Engenharia e Manutenção de Equipamentos.

No início dessas atividades, o DGI adquiriu sua primeira estação de recepção e processamento, fornecida pela empresa Bendix Aerospace para recepção e processamento dos dados MSS [2.2] do Satélite LANDSAT-1. Este sistema posteriormente foi modificado para possibilitar o processamento dos Satélites LANDSAT 2, 3 e 4.

A fitas de alta densidade gravadas pela estação de Cuiabá –MT, eram transportadas via aérea para o DGI [2.4] em Cachoeira Paulista, onde os dados contidos foram processados e sofreram correções radiométricas e geométricas, transformando-se em imagens fotográficas ou cópias de fitas magnéticas de computador. Estas operações ocorram no local chamado de Laboratório Eletrônico, onde neste local a geração da matriz do Fotográfico, realizada através de um Gravador especial de imagens por feixe de elétrons, fabricado pela IGI (Image Graphics Elétrons), modelo EBR-2127, que utilizava –se de filme especial sensível a elétrons . Após esta etapa, o negativo era revelado e ampliado em papel fotográfico, e distribuído para os usuários.



*Fig.2.3 Vista do Laboratório Eletrônico*

## CAPÍTULO 3: O Projeto MSS

### 3.1 Sistema Antigo Desativado de Processamento dos Dados -MSS

O sistema antigo de processamento de dados de imagens MSS [2.4] do satélite LANDSAT[2], utilizado pelo DGI [2] na década de 70, era constituído pelos seguintes equipamentos :

Gravador de Alta Densidade.....(AMPEX FR1928)

Gerador de Padrões de Teste MSS.....(TPG)

Interface de Dados BENDIX .....(MDI)

Gerador de Time Code .....(TCG)

Computador Main Frame.....(DEC - PDP11)

A fig.3.1 lustra o diagrama em blocos da interconecção dos equipamentos de leitura dos dados em fitas Magnéticas de alta densidade HTD's:

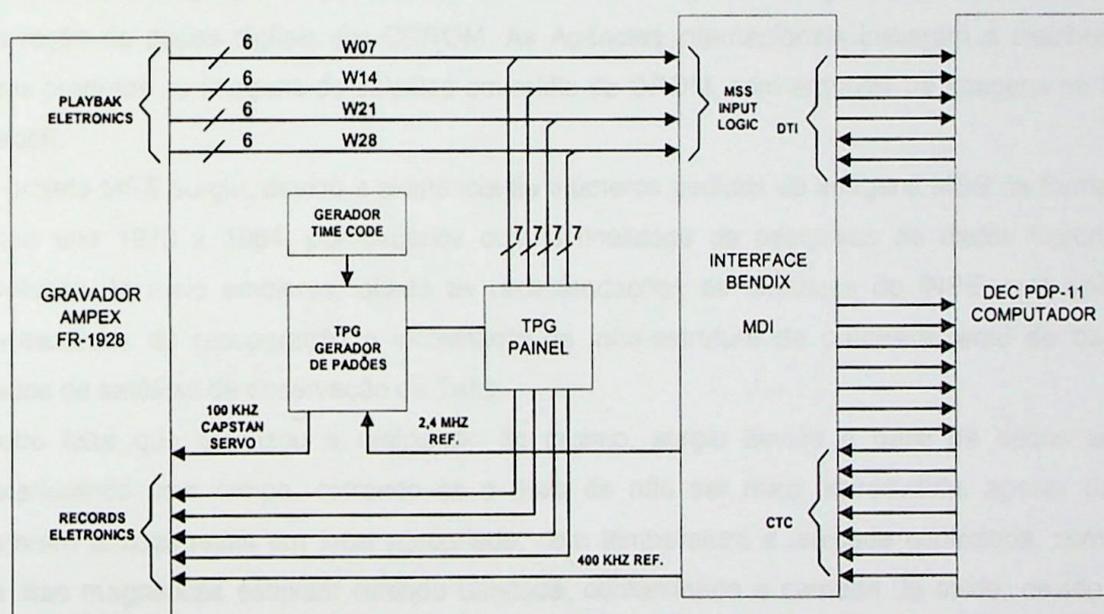


Fig 3.1) Diagrama em Blocos do Sistema antigo de Processamento de Imagens MSS.[3]

As fitas magnéticas de alta densidade recebidas pelo DGI, eram reproduzidas pelo gravador Ampex FR -1928 no formato de 24 trilhas paralelas, através de uma interface especial MDI – (Miller Code Data Interface) instalada entre o computador PDP-11 e o gravador Ampex. O MDI tinha função de corrigir e formatar os dados brutos enviados pelo gravador, de tal forma que somente o sinal de imagem era extraídos e enviados para o gravador de imagens por feixe de elétrons, na qual gerava-se o negativo em filme fotográfico, em seguida encaminhado para revelação em papel no laboratório fotográfico, e depois arquivados os negativos. Com a

implementação deste processo criou-se o primeiro banco de dados de imagens na forma de película fotográfica.

Em 1984, com o lançamento do Satélite LANDSAT- 4, mais evoluído e com características diversas de seus antecessores, os sistema inicial de recepção e processamento foram substituídos para permitir a recepção de sinais a taxas mais elevadas e em formatos diferentes. Com a substituição do equipamento por um sistema mais moderno de processamento, restou-se somente partes dos equipamentos antigos tais como; gravador AMPEX e gerador de padrões além das fitas de alta densidade que também foram resguardadas.

Todas as imagens adquiridas pelos sensores dos satélite LANDSAT 1,2 –3 cobrindo o período de 1973 a 1984 estão armazenadas no INPE em Cachoeira Paulista na forma de fitas de alta densidade, e uma parte arquivados na forma de película fotográfica.

### 3.2 Surgimento do Projeto MSS [3]

Com a evolução das pesquisas em sensoriamento remoto e com a ferramentas computacionais mais avançadas, os produtos ou imagens de satélites passaram a ser trabalhadas em computadores, agregando-se valores na forma de um produto digital a partir da inovação da gravação de dados digitais em CDROM. As Agências internacionais iniciaram a distribuição de seus produtos ou imagens de satélites em mídia de CROM, com arquivos de imagens no formato Geotiff.

O projeto MSS surgiu, devido a existência de inúmeros pedidos de imagens MSS na forma digital entre ano 1973 a 1984, por usuários com a finalidade de pesquisas de dados históricos e evolução do meio ambiente, aliado as recomendações de diretrizes do INPE, que enfatiza a necessidade da recuperação e incremento da infra-estrutura de processamento do banco de dados de satélites de observação da Terra.

Outro fator que enfatizou a realização do projeto, surgiu devido a base de dados estar se deteriorando com tempo, correndo-se o risco de não ser mais reproduzida, apesar das fitas estarem armazenadas em local apropriado, com temperatura e umidade controlada, com tempo as fitas magnéticas estavam retendo umidade, contaminado a camada de oxido, devido a esta deterioração continua com o tempo, era imperativo, a necessidade urgêncial de transferir este dados desta mídia , para uma mídia nova e atual.

### 3.3 Filosofia para Implementação do Projeto

Para atender a necessidade de transferir os dados gravados em fitas de Alta densidade HDDT para outro tipo de mídia, adotou-se que deveria ser utilizado a tecnologia mais atual possível, encontrada no mercado, ou seja deveria-se encontrar algum equipamento ou partes de equipamentos prontos para fazer esta transferência de dados, pois agilizaria a realização do projeto.

Como estamos na era dos microcomputadores, foi adotado como sistema de aquisição dados um microcomputador tipo PC. A fig 3.3 ilustra o sistema proposto, onde contém um microcomputador com uma interface, capaz de adquirir dados digitais em uma Interface de barramento paralelo, como uma taxa de aquisição de 700 Kbits/s, superior a taxa de 600 Kbits/s, da atual taxa de reprodução dos dados do gravador AMPEX FR1928, além de uma unidade de armazenamento de dados de grande capacidade.

Outra necessidade do sistema, seria o desenvolvimento de um circuito dedicado (interface de 24 canais), que sincroniza-se os sinais de reprodução do gravador com a interface, solucionando o problema de diferenças de fases (Skew), além de adequar os níveis de sinais de reprodução do gravador. Uma outra função desta interface, seria também, gerar os sinais de controle e ativação da transferência dos dados necessários a placa de interface de aquisição instalada no computador.

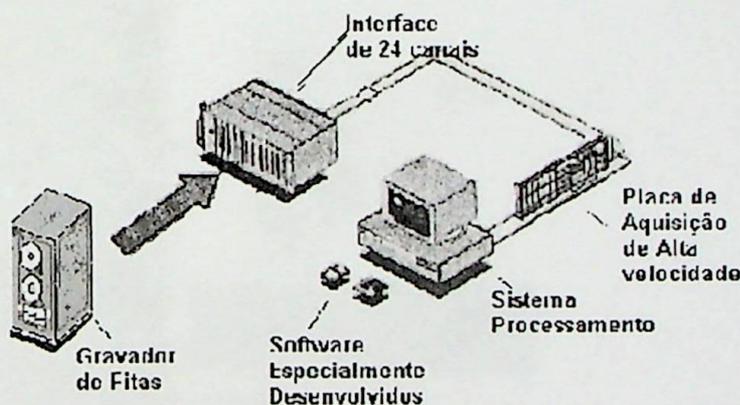


Fig 3.3 Sistema novo Proposto

#### • 3.4 Identificação e Modelagem dos Problemas

Para dar início ao projeto, foram realizados levantamentos dos equipamentos eletrônicos e da documentação existente destes equipamentos remanescentes daquela época, para verificar as condições de funcionamento e uso destes. Foram feitas também análises para avaliar as condições de reprodução das fitas magnéticas de alta densidade armazenadas no filmoteca do INPE.

Descobriu-se que os equipamentos estavam dispersos em localidades diferentes e que necessitavam de manutenções e reposição de peças para o uso destes, quanto as fitas magnéticas, descobriu-se que necessitavam de adotar um procedimento especial para o uso

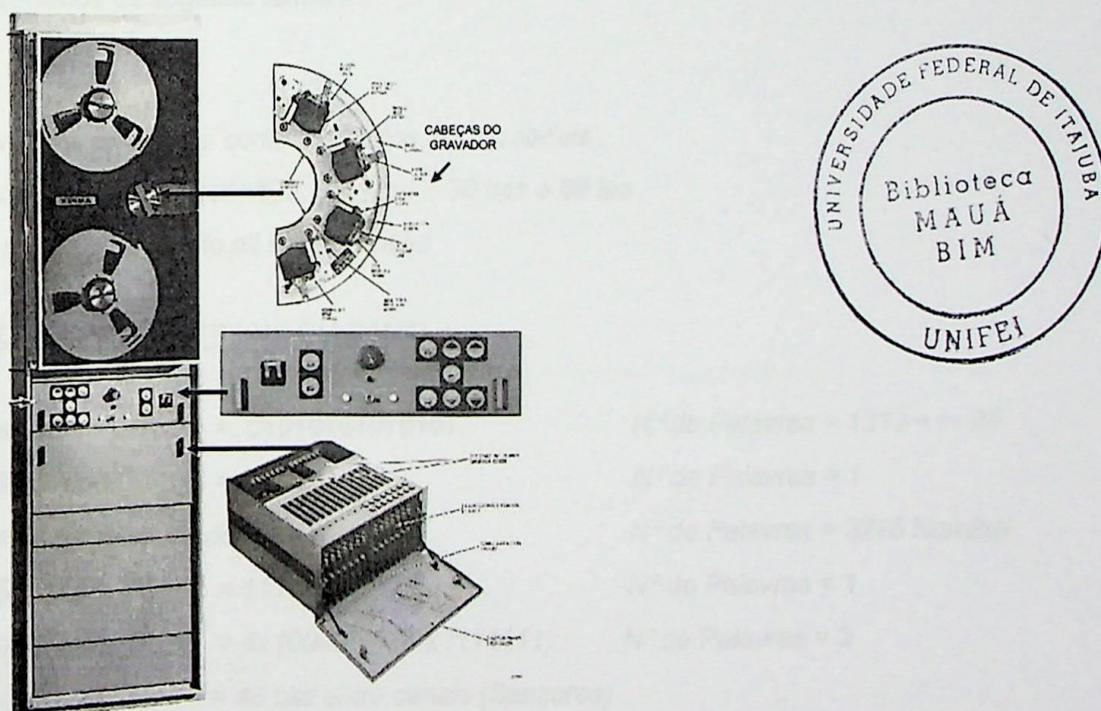
destas, pois devido ao tempo de armazenamento as fitas magnéticas encontravam –se com teor de umidade, necessitando –se, de um processo de secagem, ou seja deveriam permanecer em um forno na temperatura de 54 graus, durante 24 horas antes de ser utilizadas.

### • 3.5 Descrição do Gravador AMPEX FR 1928

Para gravação e reprodução dos dados do Satélite LANDSAT [2], o INPE utilizou–se de um gravador de Fitas Magnéticas de Alta Densidade marca AMPEX – FR1928 .

O Gravador AMPEX -FR1928 utiliza fitas magnéticas de alta densidade de 7200 pés, possui 24 canais de saída para reprodução, e 24 canais de entrada para gravação, sendo neste caso somente os canais de reprodução. foram utilizados Os canais de gravação e reprodução estão divididas pelas cabeças de gravação e reprodução, sendo 2 cabeças de gravação contendo cada uma 12 canais, divididos em canais pares e ímpares, com 2 cabeças de reprodução contendo cada uma 12 canais, divididos em canais pares e ímpares .

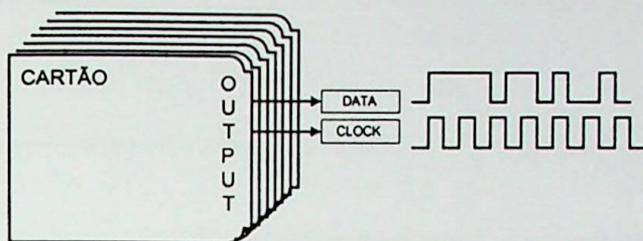
O gravador possui um sistema de servo motores que controlam a velocidade de gravação e reprodução, sendo que a velocidade de rotação do servo motor, possui 7 velocidades selecionáveis, que são 120, 60, 30, 7-1/2, 3-3/4, 1 – 7/8 (ips)



**Fig 3.5 .a. Gravador Ampex Modelo - FR1928 [6]**

O gravador possui 4 Racks ou Bays, onde são distribuídos os cartões com código Miller (Miller Code), que são responsável pelo tratamento do sinais que saem do pré-amplificador enviado pela cabeça do gravador, que perfazem um total de 24 cartões de reprodução dos canais, um para cada canal. Cada cartão é dotado de potenciômetros no qual permiti-se ajustar o ganho , jitter e

set point do mostrador analógico do painel (VU), os cartões possuem uma saída para dados e outra saída para clock, os dados são validos e sincronizados na borda de subida do clock., os dois sinais são ligados a conectores BNC, com impedância de saída de 75 ohms . (fig 3.5b)



**Fig 3.5.b - Sinais de Saídas**

### • 3.6 Formato dos dados digitais de Saída do Gravador

Para o desenvolvimento do projeto, um dado muito importante e que possibilitou a especificação e projeto da interface formatadora dos dados(IFDM) de saída do gravador, foi as especificações do padrão ou formato do pacote de dados (*frame*) de saída, ou seja a forma em que os dados MSS [2] são reproduzidos serialmente por cada sensor ou canal do gravador AMPEX FR1928, estes dados são especificados da seguinte forma :

#### SAÍDA :

- 24 Sensores ou canais contendo dados NRZ-L seriais .
- Taxa de dados por canal =602,505 + ou - 30 bps e 60 lps
- 6 bits por palavra sendo bit MSB primeiro

#### FORMATO PARA CADA TRILHA OU CANAL :

- *Preâmbulo cod.* =..010101010101. *Nº de Palavras = 1313 + / - 20*
- *Line Start cod.* = 111001 *Nº de Palavras = 1*
- *Dados de Imagem /Sensor* *Nº de Palavras = 3215 Nominal*
- *MMFS cod.* = 111000 *Nº de Palavras = 1*
- *End Line Code (ELC)* = 4x (000000), 4 x (111111) *Nº de Palavras = 3*
- *Data Skew* = 48 bits entre canais (Sensores)
- *Dados de Calibração* = 1024 em sweep alternados

### 3.6.1 Diagrama Temporal do Formato dos Dados MSS.

A figura abaixo mostra um diagrama temporal do formato dos dados, onde especifica a duração do formato dos dados .

#### FORMATADOS DOS DADOS MSS

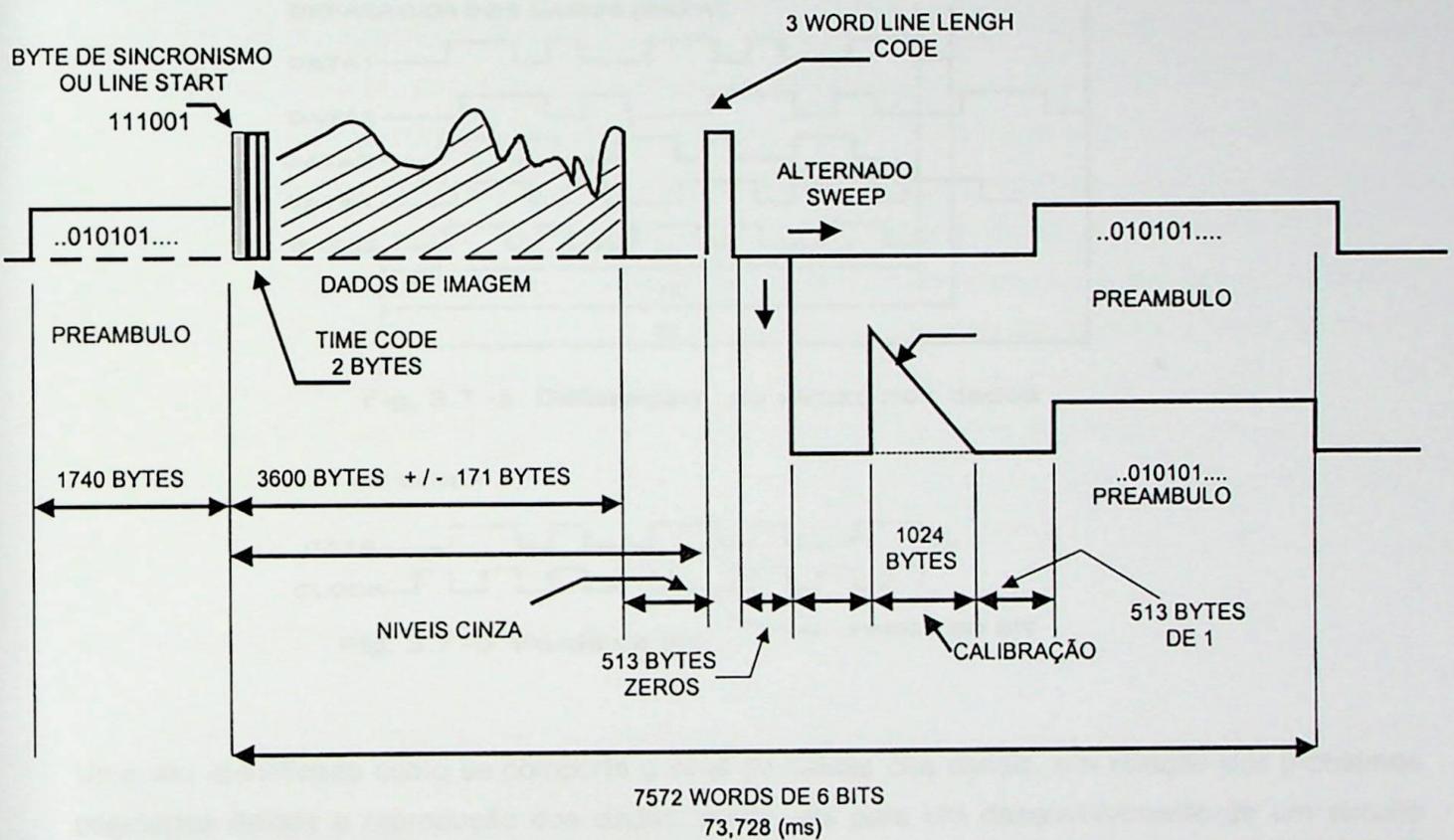


fig 3.6.1 – Diagrama do Formato dos dados MSS.

### 3.7 Análise do Problema da diferença de Fases (Skew) do Gravador AMPEX FR1928.

Como foi demonstrado, o gravador AMPEX FR-1928 disponibiliza 24 saídas seriais de clock de 600 kbits/s em cada um dos 24 canais e 24 saídas de dados sincronizado na subida do (clock), que são gerados a partir da leitura da fita de alta densidade. Estes dados apesar de terem sido gravados na mesma taxa de 600 kbits/s, quando reproduzidas pelo gravador apresentam-se na saída com características de "clock" independentes, apresentando diferenças de fases entre os canais (Skew) ou até mesmo eventuais perdas de dados (Sinc Fault), e no caso mais grave, perda de bits no clock (Bit Sleep). As figuras 3.7.a e 3.7.b. ilustram estes casos.

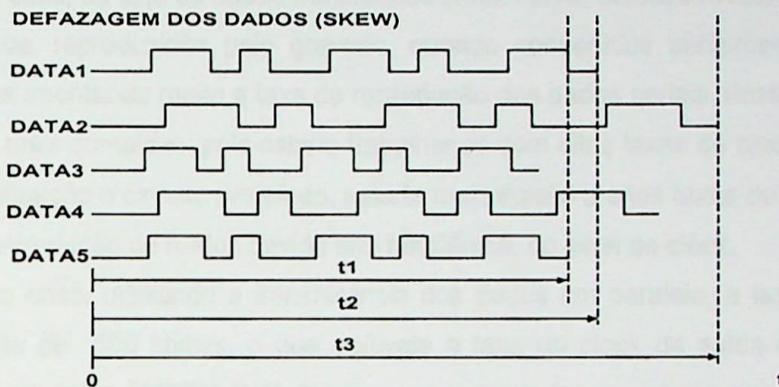


Fig. 3.7 -a Defasagem ou atraso dos dados

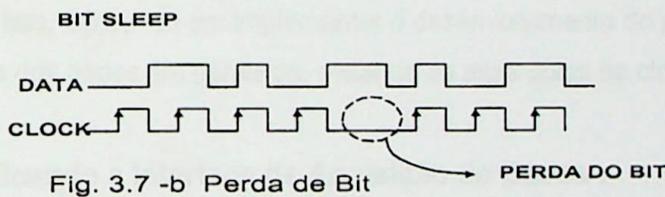


Fig. 3.7 -b Perda de Bit

Uma vez identificado como se comporta o sinal de saídas dos canais, em relação aos problemas originados devido a reprodução dos dados, partiu-se para um desenvolvimento de um circuito dedicado, que viabilize o interfaceamento dos dados entre o computador e a placa de aquisição dos dados, ou seja um circuito dedicado a formatar os dados, de forma a ser projetado para solucionar o problema de sincronismo existente, além de gerar dos sinais de transferência para a placa de aquisição PCI\_DIO-32HS instalada no computador.

Um estudo mais apurado foi criado para verificar qual a melhor estratégia adotada para a implementação de um circuito eletrônico que integraria esta interface, de tal forma a garantir a aquisição dos dados úteis, mesmo em uma eventual ausência de dados em alguns dos canais.

## CAPÍTULO 4: Hardware do Projeto MSS

### 4.1 Filosofia de Implementação do Hardware.

Para a implementação do Hardware do projeto, um fator de muita importância, foi a especificação de como os dados reproduzidos pelo gravador AMPEX FR1928 deveriam ser transferidos para o computador. Ou seja, definir a forma de transferência de dados, que apresenta-se maior confiabilidade para o sistema, neste caso haveria opção de transferência dos dados na forma serial ou na forma paralela.

Para o primeiro caso, ou seja os dados transferidos serialmente; estudos revelaram que o frame ou pacote de dados, reproduzidos pelo gravado, quando convertidos serialmente apresentava-se muito extenso, aumentando muito a taxa de reprodução dos dados seriais, neste caso o circuito de aquisição seria mais complexo, pois estaria trabalhando com altas taxas de clock em torno de 14.4 Mbits/s, nesta situação o circuito projetado, estaria mais sujeito a altas taxas de erro BER (Bit Error Rate) além da introdução de ruídos devida alta frequência do sinal de clock.

Para o segundo caso, utilizando a transferência dos dados em paralelo, a taxa de transferência dos dados, seria de 600 kbits/s, o que equivale a taxa do clock de saída de cada canal de reprodução do gravador AMPEX [3.5]. Neste caso o único inconveniente gerado neste modo de transferência, estaria no aumento do número de cabos de saída ( 24 cabos em vez de 2 cabos). Baseado neste fato, optou-se em implementar o desenvolvimento do projeto utilizando o processo de transferência dos dados em paralelos, evitando as altas taxas de clock.

#### 4.1.1 Especificando a Interface de Aquisição de Dados .

Após a definição ou método de transferência de dados para o computador, o próximo passo deveria ser necessariamente especificar a interface de aquisição deste dados paralelos, para ser instalada no barramento PCI de um computador tipo PC. Com a tecnologia atual mais avançada da indústria de semicondutores e placas de aquisição, pesquisas realizadas no mercado apontaram para diversos fornecedores de interfaces de diferentes tipos de sistemas de aquisição de dados de uso geral para microcomputadores PC. Através desta pesquisa de mercado nos diversos tipos de fabricantes, descobriu-se que a Empresa National Instruments, oferecia uma interface PCI para computador PC, com 32 bits de I/O dados paralelos, com alta velocidade de transferência de dados podendo chegar até 76 Mbytes/s . Como o gravador reproduz 24 sensores na taxa de 600 kbits/s cada um, logo a taxa total de reprodução ficaria em torno de  $24 \times 600 \text{ kbits/s} / 8 = 1.44 \text{ Mbytes/s}$ , neste caso mesmo adotando o processo de aquisição de dados serialmente, a taxa de reprodução dos dados ficaria ainda abaixo da taxa máxima da placa de aquisição que é de 76 Mbytes/s.

#### 4.1.2 Arquitetura do Hardware - Interface PCI\_DIO-32HS

No diagrama em blocos da interface PCI\_DIO\_32HS ilustrado na Fig. 4.1.2, descreve a interface PCI\_DIO\_32HS (National Instruments) de 32 bits programáveis para Entrada ou Saída, compatível com barramentos PCI de 32 ou 16 bits, utilizando tecnologia Plug and Play, sendo totalmente configurada todos os canais de DMA pelo software do sistema operacional, sem necessidade de configuração de jumpers no hardware. A placa contém o Chip National Instruments DAQ-DIO, onde prove de dois canais digitais independentes sendo selecionáveis da seguinte forma :

- Seleção de Grupos (8, 16, ou 32 bits)
- Fifos Internas para Buffer da Placa
- Seis protocolos diferentes de Handshaking.
- Sistema de Detecção de Start e Stop Trigger e Pattern
- 24 mA saídas
- Latches e Drives para Entrada ou Saída
- Oscilador 20 Mhz
- Transferência via DMA

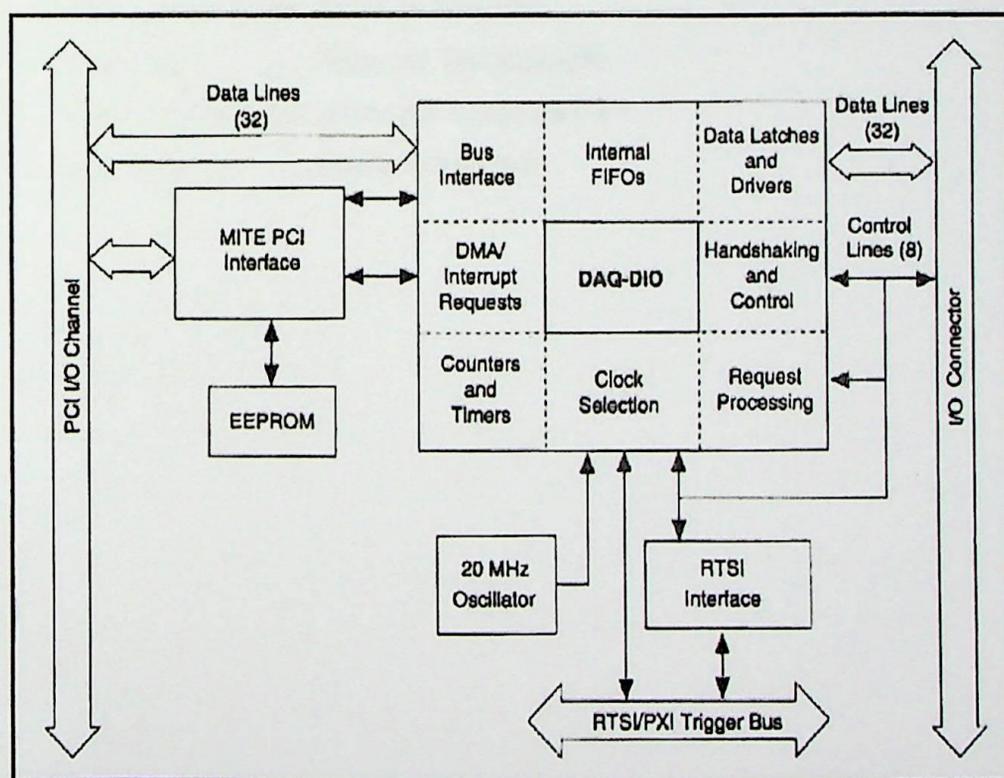


Fig- 4.1.2 – Diagrama em Bloco da Interface de Aquisição de Dados PCI- DIO32HS [8]

O Hardware acompanha um CD de instalação (NI\_DAQ), com o driver para maioria dos software programação tais como:

- LaBVIEW
- ComponentWorks
- Visual C++
- Delphi
- Boorland C++
- Etc..

• **4.1.3 Protocolo ou Modos de Aquisição - Interface PCI\_DIO-32HS**

Além das 32 linhas de I/O, a placa fornece sinais adicionais de controle para a transferencia dos dados, que são designados respectivamente por: ACK,REQ, PLCK . e utilizam os protocolos de transferência de dados como:

- 8255 Emulation.....Emulação do Chip 8255.
- Level ACK..... ACK por nível.
- Leading-Edge Pulse...Pulso de transição .
- Long Pulse.....Pulso por Temporização.
- Trainling-Edge Pulse ..Pulso por subida de 0 a 1.
- Burst Mode.....Frame Progamado..

Na Tabela 4.1.3, apresenta os tipos de protocolos de aquisição de dados, utilizando a Interface PCI-DIO-32HS, ilustrando as especificações de transferencias de cada protocolo.

Protocolo	Taxa de Transferência p/ cabos de Comprimento diferentes. (MS/s)		Polaridade dos Sinais : REQ e ACK	REQ Transição	Programação do Atraso por Software	Protocolo Complementares
	1 metro	2 a 5 metro				
Procolos Assincronos	5	2.67	Ativo em Baixo	Trailing	Entre Transferencias	Leading_Edge Pulso
Sinal de ACK por nível (Level ACK)	3.3	2.5	Programável	Leading	Antes de ACK	Level ack
Pulso Pela Borda (Laeding Edge Pulse)	3.3	2.5	Programável	Leading	Antes de Ack	Leading-Edge Pulso
Pulso Longo (Long Pulse)	3.3	2.5	Programável	Leading	Pela Largura do Pulso	Long Pulso 8255 Emulation
Trailing Edge Pulse	1.8	1.5	Programável	Trailing	Pela Largura do Pulso	Trailing_Edge Pulso
Protocolos Sincronos						
Burst (Sicronizado por Clock Interno)	20	10	Programável	Pelo Nível	Pela velocidade Do Clock	Burst

**Tabela 4.1.3 Protocolos de Aquisição utilizando a Interface PCI-DIO-32HS [8]**

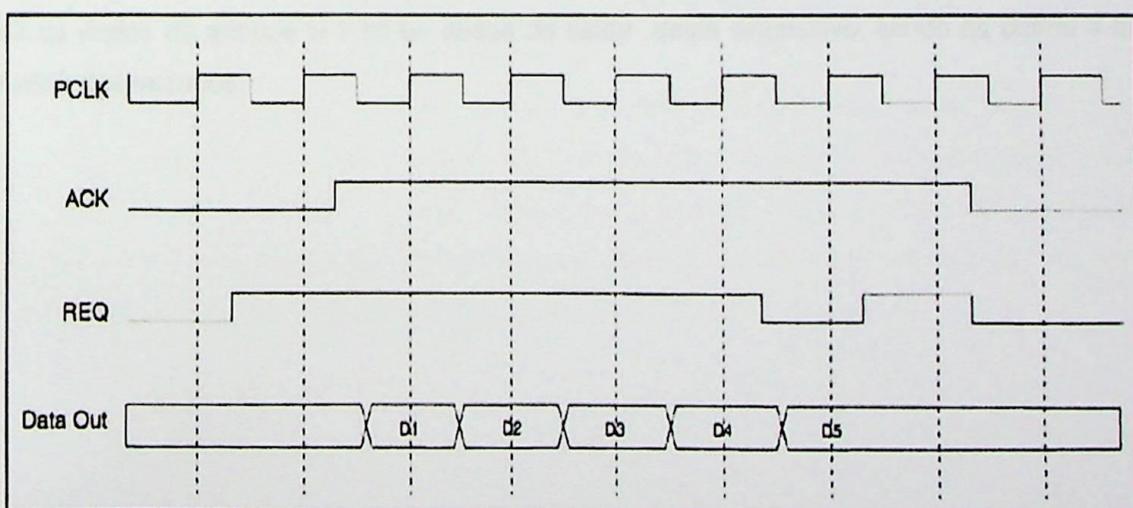
#### 4.1.4 Definindo o Protocolo de Aquisição da Interface PCI\_DIO\_32HS

Os protocolos de comunicação da interface de aquisição dos dados são subdivididos em protocolos assíncronos e síncronos, a escolha do protocolo síncrono (Bursts Mode) foi o que mais se adaptou ao interfaceamento entre os sinais da saída do gravador e a entrada da interface de aquisição PCI\_DIO\_32HS, devido ao fato que a reprodução dos sinais do gravador AMPEX, apresentarem diferenças de fases no sinal de clock de reprodução. O protocolo síncrono faz com que fixe-se um sinal clock de referência que pode ser programado através do software para transferência simultânea de todos os dados dos canais pela transição de subida (edge positivo) do sinal Clock de referência.

- **4.1.5 Descrição do Protocolo Burst Mode da Interface PCI\_DIO\_32HS**

O modo BURTS de aquisição é um protocolo de aquisição de dados síncrono. Neste modo, todos os dados são transmitidos ou recebidos compartilhando através de um mesmo sinal de clock disponibilizado pela interface chamado de (PLCK), este mesmo sinal de PLCK pode ter sua frequência programada através do software e pode variar entre 7 MHz a 20 Mhz.

A cada ciclo do sinal de clock, a interface envia um sinal de ACK, colocando no estado de nível alto(+5V), se estiver pronto para transferência dos dados, isto é, se o dispositivo externo interfaceado, colocar o sinal de REQ em estado alto (+5V) para interface, indicando que está pronto para a transferência, caso isto ocorra os dados são transferidos para o barramento da placa, na subida do (PLCK). Após a transferência dos dados, os sinais de REQ e ACK devem ser desabilitados ou colocados para nível baixo, repetindo-se novamente para um novo ciclo de aquisição, como mostra o diagrama da fig 4.1.5.



**Fig 4.1.5. Diagrama dos sinais para Modo Burst para aquisição dos dados de entrada.**

Com base neste modo de aquisição, partiu-se para implementar o desenvolvimento de uma interface formatadora dos dados de imagens MSS, reproduzidos pelo gravador AMPEX, denominada pela sigla IFDM ( Interface Formatadora dos Dados MSS), capaz de gerar este sinais obedecendo os procedimentos do protocolo BURST de aquisição e em sincronismo com os sinais de reprodução do Gravador Ampex.[3.5].

#### 4.2 Interface Formatador de Dados MSS – IFDM

A interface formatadora dos dados MSS (IFDM), tem a finalidade principal, em formatar, sincronizar e adequar os dados MSS reproduzidos pela a saída do gravador AMPEX FR1928, para ser transferidos ao barramento de dados da Interface de aquisição PCI\_DIO\_32HS instalada no computador PC. A interface IFDM, também tem a função de corrigir o problema de diferenças de fases entre canais ou sensores reproduzidos pelo gravador ( Skew) .

A tecnologia desenvolvida, para o hardware da interface, baseou-se na criação de um Buffer intermediários de memória, capaz de armazenar os dados temporariamente até que todos os dados permanecerem em fase, para isto foi utilizado, circuitos Integrados tipo FIFO's (First IN and First OUT).

Como por definição o SKEW máximo do gravador é de 48 bits, foi especificado uma memória que mais se aproximava deste valor, onde foi escolhido o circuito integrado memória FIFO paralela CMOS (P/N IDT 72413) de 64 x 5 bits com sinalizadores por hardware, tais como Half –Full, ( Metade da Capacidade), Almost Full (cheia) e empty (vazia)

Como necessitava que o controle destas memórias fossem independentes para cada canal, se utilizou de um circuito integrado para cada canal de reprodução do gravador e utilizando somente 1 bit de dados de entrada e 1 bit de dados de saída deste dispositivo, sendo os outros 4 bits de dados desprezados.

### 4.2.1 Descrição da Memória FIFO IDT 72413

Especificação:

- First-In/First out duplo porte
- Taxa máxima de Transfêrencia de 45 MHz
- 64 x 5 bits
- Pinos indicadores de (Half-Full e Almost-Full e Empty)

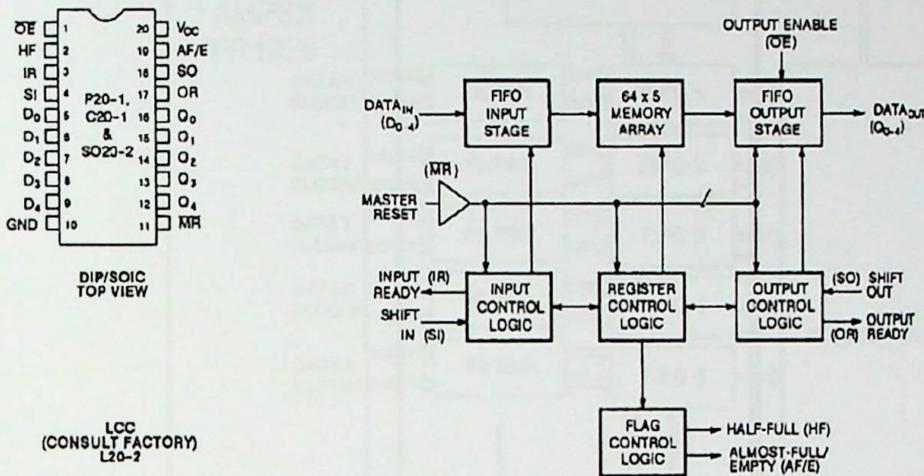


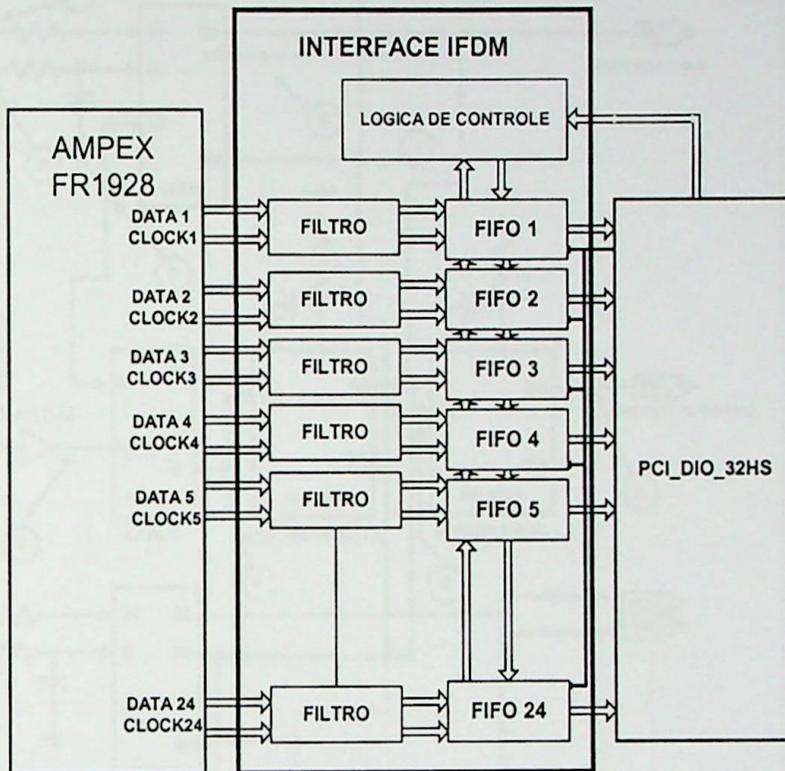
Fig 4.2.1 Diagrama da Memória FIFO [11]

Observando a Fig 4.2.1, o circuito integrado apresenta sinalizadores por hardware, onde são manipulados para controle, os sinalizadores mais importantes são os que controlam a entrada e saída de dados bem como os que indicam se a capacidade da memória, exemplo; Almost –Full, (AF), Empty (EF) e Half –Full (HF), ou seja, se a memória está completamente cheia, vazia ou até mesmo se está pela metade de sua capacidade.

Outro procedimento adotado, foi não permitir o estouro (overflow ) da capacidade da memória, pois neste caso haveria perda de dados de entrada,. Logo optou-se por utilizar, como controle o sinalizadores de Half Full (HF), ou seja sempre que a memória atingir a metade de sua capacidade de armazenamento ou seja 32 bits de dados, a interface de aquisição tem que obrigatoriamente retirar dados da memória FIFO, ou ocorre na perda de dados de entrada.

#### 4.2.2 Descrevendo o Hardware da IFDM

O diagrama em blocos da Fig. 4.2.2, ilustra as interligações entre o hardware da Interface IFDM, com os sinais do gravador de reprodução AMPEX e a interface PCI-DIO\_32HS

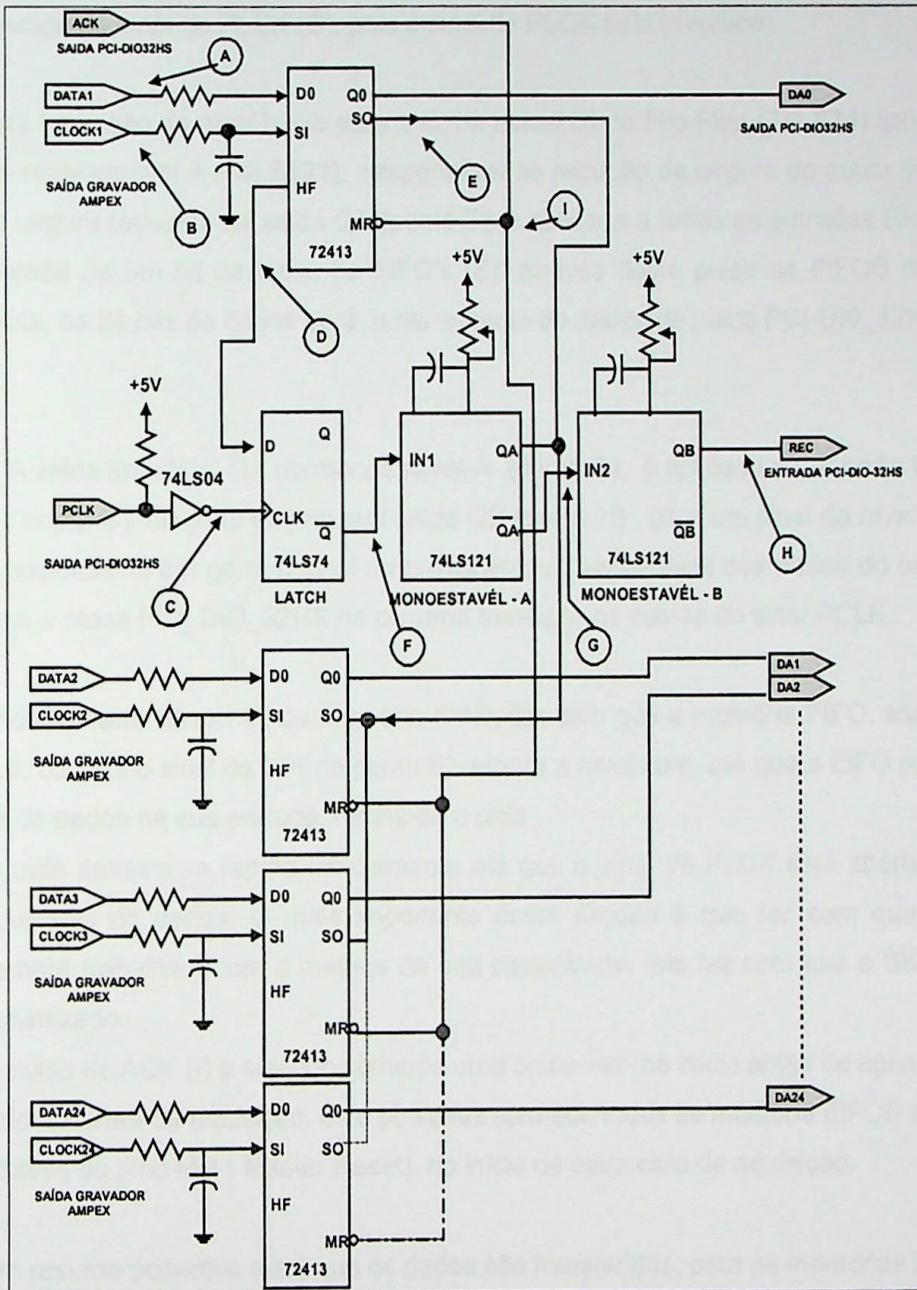


**Fig.4.2.2 Diagrama em Blocos do Hardware da IFDM (Interface Formatadora dos Dados)**

Observando-se o diagrama do bloco central, onde ilustrada o hardware da Interface IFDM, observa-se, que para cada canal de saída de dados do gravador é adicionada uma memória FIFO, que garante que haverá um buffer elástico de dados para cada sensor de reprodução do gravador AMPEX. O sinalizador Half – Full tem a função de manter o intervalo de tempo em que os dados de outros canais que venham-se atrasar para chegar até as FIFO's, evitando assim a defasagem dos sinais entre os canais (*skew*). Os dados somente são transferidos para a interface PCI quando o sinalizador Half– Full estiver ativo ou em nível alto, garantindo que sempre haverá dados para ser transferidos para interface, isto faz com que o atraso seja atenuado. O mesmo sinal de indicação de Half –Full é utilizado na placa de aquisição PCI\_DIO\_32HS, como sendo o sinal de REQ, ou seja o sinal que habilita a transferência de dados quando detectar a transição de 0 para 1 no sinal de PCLK da placa de aquisição.

### 4.2.3 Descrição da Lógica de Controle da Interface IFDM e sinais de Handshake .

O circuito de controle ilustrado na fig 4.2.3-a abaixo, mostra detalhado o funcionamento da lógica de controle sincronizado pelo sinal de reprodução do canal 1 do gravador AMPEX FR1928.



**Fig.4.2.3.a – Circuito de Controle dos Sinais de Aquisição de Dados**

O circuito foi implementado para realizar a lógica de controle entre a interface IFDM e a interface PCI-DIO32HS gerando os sinais necessários para o funcionamento do protocolo de aquisição BURST, para transferência dos dados para a Interface PCI.

De acordo com o circuito, da figura 4.2.3.a, os dados digitais MSS reproduzidos pela saída do gravador AMPEX (Pontos A e B), são continuamente transferidos seqüencialmente, para a memória FIFO através da subida positiva do sinal de clock, onde é aplicado na entrada Shift IN

(SI), até que alcance a metade da capacidade da FIFO, ou seja 32 bits de dados . Quando a memória FIFO atingir a metade de sua capacidade, ou seja o pino 2, da Memória FIFO ou Half Full, é levado ao nível alto, ou 5 Volts (pto. D), sinalizando a condição que haja dados em todas as memórias FIFO's. O mesmo sinal é aplicado na entrada de dados do FLIP-FLOP tipo D, o FLIP-FLOP somente mudará de estado na sua saída(Q) gerando uma transição, quando ocorrer uma descida do sinal de PLCK (C), pois o sinal de PLCK está invertido .

Esta transição de nível baixo para alto, na saída(Q) do Flip-Flop (74LS74) (ponto E) é aplicada em um monoestável A (74LS121), responsável na redução da largura do pulso de entrada, este pulso de largura reduzida na saída QA (ponto E), é aplicada a todas as entradas SHIFT-OUT, ou seja na retirada de um bit de todas as FIFO's (E). através deste pulso as FIFOS disponibilizam na sua saída, os 24 bits de dados para o barramento de dados da placa PCI-DIO\_32HS.

A saída invertida  $\overline{QA}$  do monoestável-A (ponto G), é aplicado na entrada IN2 do monoestável - B ( ponto G), fazendo que na sua saída QB (ponto H) , gere um sinal de nível alto REQ pulso, que é responsável em gerar o sinal de gravação ou transferencia dos dados do barramento da FIFOS, para a placa PCI\_DIO\_32HS na próxima transição de subida do sinal PCLK.

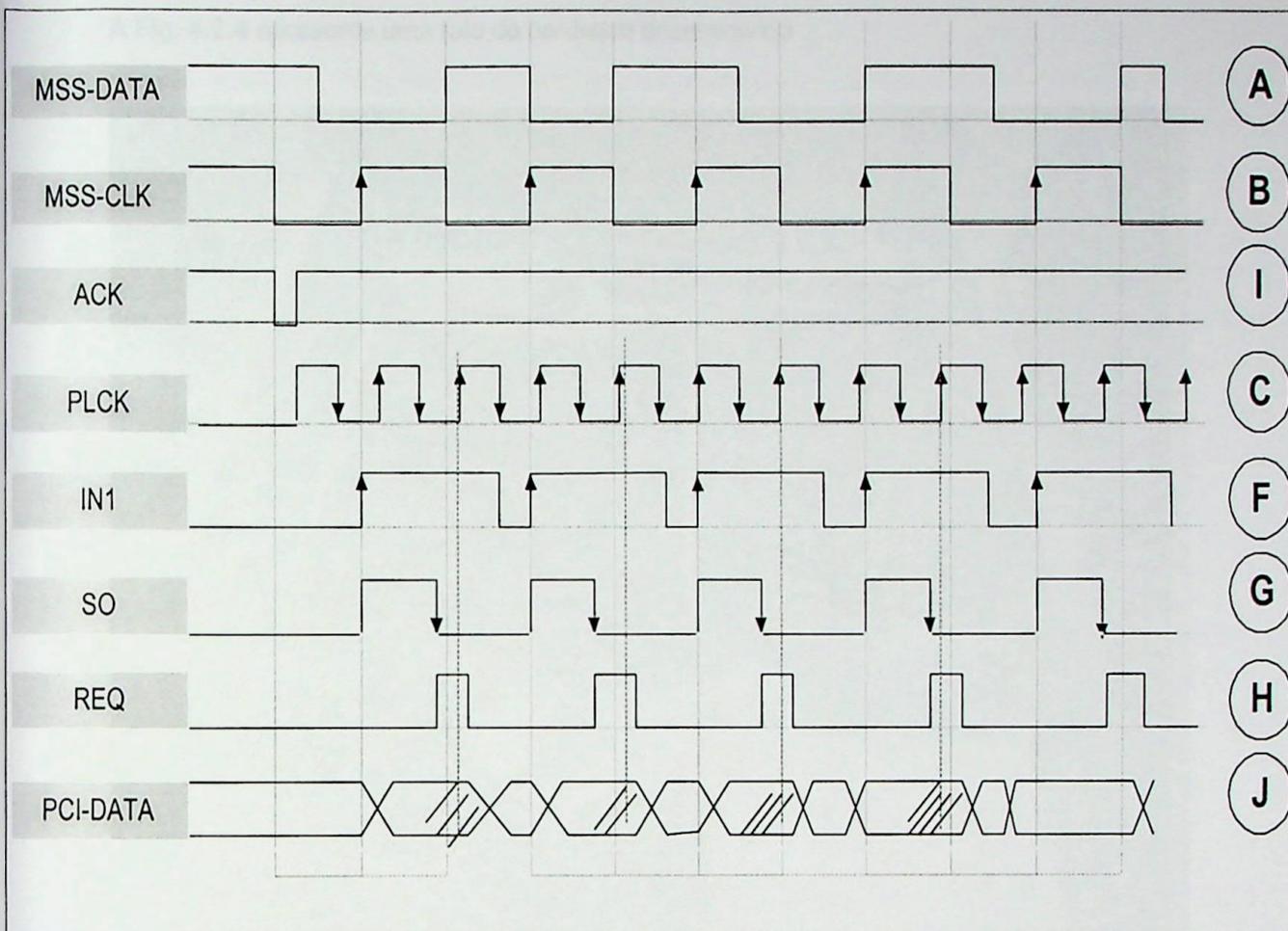
Após retirada de um bit da memória FIFO, faz com que a memória FIFO, saia do estado de Half Full, ou seja o sinal de HF( no ponto E) retorna a nível zero, até que a FIFO receba novamente um bit de dados na sua entrada, repetindo o ciclo .

O ciclo sempre se repete infinitamente até que o sinal de PLCK seja abortado pelo software de aquisição de dados. O mais importante deste circuito é que faz com que as memória FIFOS sempre trabalhem com a metade de sua capacidade. Isto faz com que o SKEW ou o atraso seja minimizado.

O pulso de ACK (I) é enviado somente uma única vez no início antes de aplicar o sinal de PLCK, pelo software de aquisição, este pulso faz com que todas as memória FIFOS sejam zeradas, através do pino MR ( Master Reset), no início de cada ciclo de aquisição.

Em resumo podemos dizer que os dados são transferidos, para as memórias FIFOs , na transição Negativa do sinal de PLCK, e são transferidos para o barramento de dados da interface PCI\_DIO , na transição positiva do sinal do PLCK, estando o sinal de Half – Full no estado de nível alto.

A fig. 4.2.3-b apresenta as formas de ondas geradas pelos sinais das Interfaces IFDM e PCI-DIO.



**Fig.4.2.3..b) Diagrama Temporal do Sinais de Controle entre Interface IFDM e PCI-DIO32HS**

*Fig. 4.2.4 Parte do Hardware (Parte da Interface IFDM)*

#### 4.2.4 Foto do Hardware \_ Interface IFDM

A Fig. 4.2.4 apresenta uma foto do hardware desenvolvido



*Fig. 4.2.4 Foto do Hardware (Placa da Interface IFDM)*

• 4.3 Simulador TPG

O simulador TPG (Test Pattern Generation) fornecido pela Empresa BENDIX Aerospace, foi uma ferramenta essencial no desenvolvimento e calibração do Sistema MSS, pois apresenta a possibilidade de simular os dados de imagens padrão em tempo real, com formato de imagens fixas e padronizadas . Este dados nunca sofrem variações de skew ou perda de bit, pois são gerados eletronicamente a partir de oscilador fixo interno. Além dos padrões pré-selecionados por uma chave seletora, pode-se simular falta de sincronismo em uma ou todas as 4 bandas espectrais através das chaves de comutação no painel (SYNC LOSS). Existem outras funções que são úteis para análise de simulação dos dados, como variação do código da palavra de LLC (Line Length Code) variação da escala de níveis de Cinza (Gray Level). A figura 4.3 ilustra o painel do simulador TPG com seus modos de operação selecionados por chaves e botões.

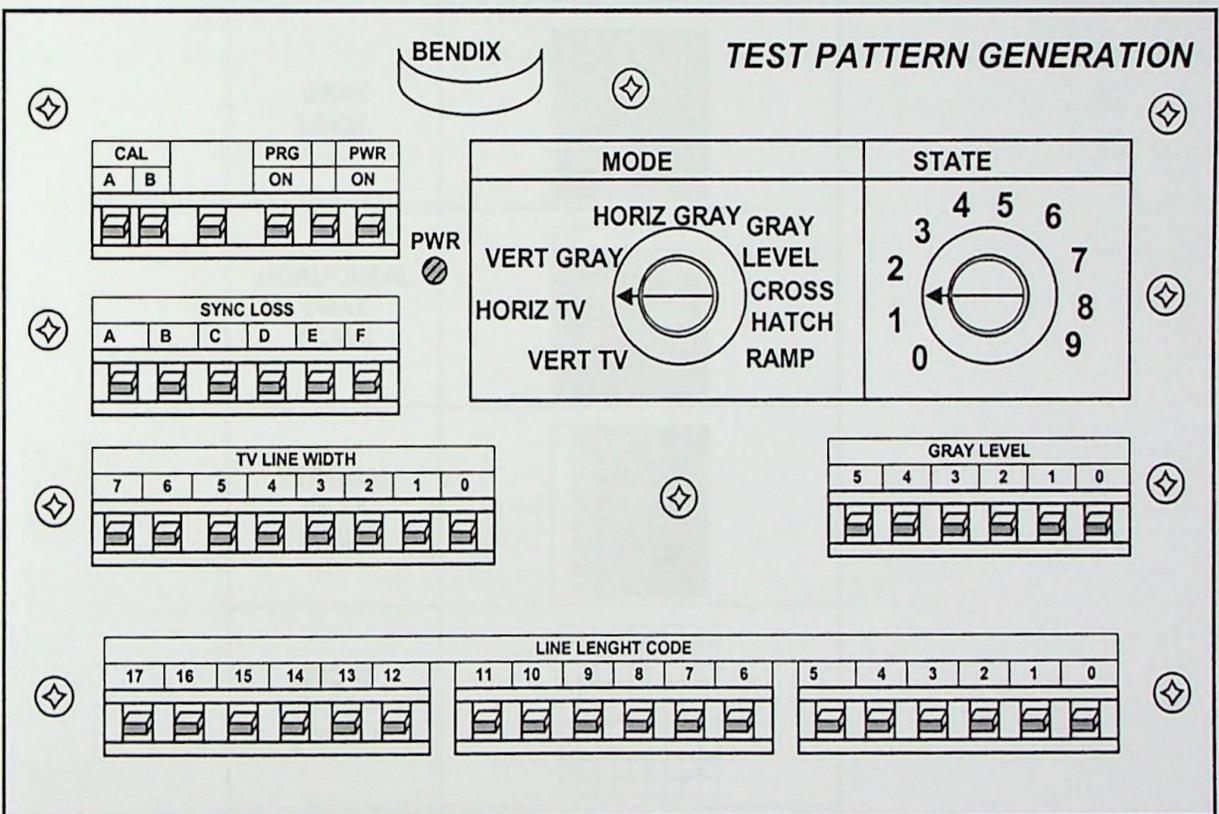


Fig-4.3) Figura do Painel Físico do Simulador TPG [3]

### 4.3.1 Formato dos Padrões Gerados pelo TPG – (Test Pattern Generation)

Os 7 modos ou imagens padrão gerados pelo TPG, são indicados no diagram da Fig 4.3.1 abaixo:

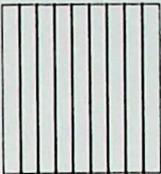
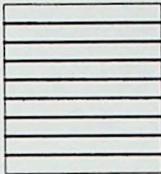
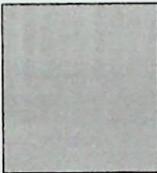
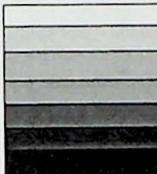
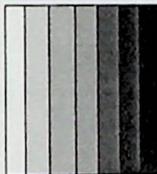
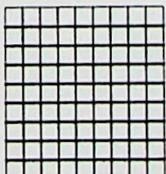
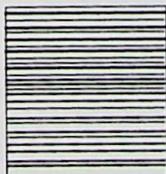
MODO	IMAGE
VERTICAL TV LINES	
HORIZONTAL TV LINES	
GRAY LEVEL	
HORIZONTAL GRAY SCALE	
VERTICAL GRAY SCALE	
CROSS HATCH ( GRID )	
HORIZONTAL GRAY WEDGE (RAMP)	

Fig. -4.3.1) Diagrama dos Formatos de Imagens Padrões [3]

## CAPITULO 5: Software do Projeto MSS

### 5.1 Fases do Desenvolvimento do Software do Projeto MSS

O software de aquisição de dados do projeto de recuperação e transferência dos dados MSS, foi elaborado em três fases distintas, até chegar ao objetivo final, que a geração de produtos de imagens digitais MSS Landsat históricas, cobrindo o período entre 1973 á 1984, onde será disponibilizado aos usuários com intuito do estudo da evolução do meio ambiente. No entanto o objetivo primordial do projeto, é resguardar a base de dados de imagens MSS, que está armazenada a mais de 30 anos em fitas magnéticas de alta densidade. A prioridade inicial no entanto, foi primeiramente em desenvolver um software de aquisição e transferência dos dados brutos MSS armazenados em fitas de alta densidade HDDT para Fitas digitais de alta capacidade de armazenamento tipo DLT, ou seja uma copia fiel dos dados brutos em HDDT para DLT, denominado Fase 1 do software do projeto (Fig. 5.1).

Uma vez concluída a Fase1 do projeto, estando a base de dados brutos resguardadas, o próximo passo seria em desenvolver um software a partir do dados brutos armazenados na composição das imagens no formato GeoTiff das 4 bandas espectrais com suas respectivas correções geométrica e radiométrica a partir dos dados orbitais do satélite, denominada a Fase 2 do projeto.(Fig. 5.1). e finalmente por ultimo a Fase 3 (Fig 5.1), que seria a catalogação das imagens no Centro de Dados de Imagens de Sensoriamento Remoto (CDSR) do DGI – INPE, podendo ser distribuídas aos usuários via Internet, FTP ou até mesmo em CDROM através de pedidos realizados pelos usuários.

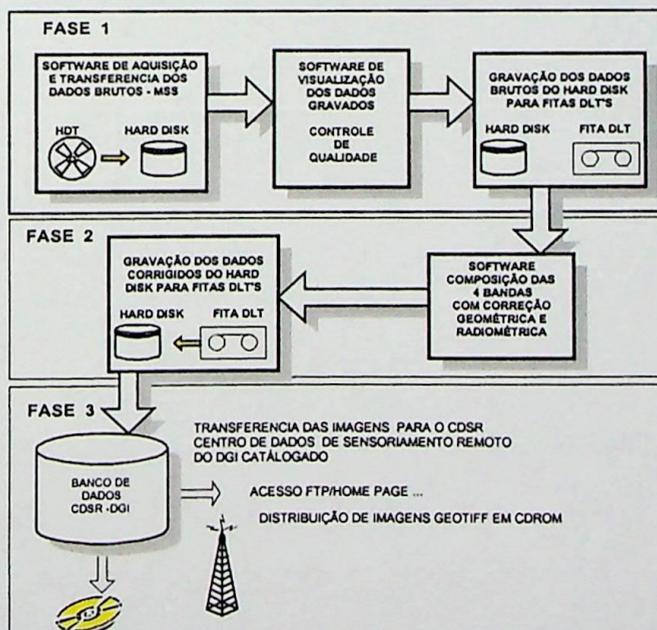


Fig 5.1) Diagrama em Blocos das Fases de desenvolvimento do Software

## 5.2 Software de Aquisição dos Dados Brutos

Software de aquisição de dados do projeto MSS, tem a função de coletar continuamente os dados em tempo real através da placa de interface PCI-DIO-32HS [4.1.2], durante a presença do sinal de clock do sinal saída do gravador AMPEX- FR1928 [3.5], ou seja o tempo de aquisição dos dados varia de acordo com o tempo de reprodução de cada orbita do satélite gravado nas fitas magnéticas de alta densidade, este tempo pode variar entre 8 a 10 minutos de reprodução das fitas, sendo que cada fita magnética pode possuir uma ou mais orbitas intercaladas.

O Software de aquisição ilustra através de uma tela de monitorização do estado dos sensores durante a transferencia dos dados (Fig 5.2), onde-se pode monitorar a qualidade dos dados gravados, através de sinalizadores luminosos (LED's Virtuais), que são amostrados em tempo real na tela do computador. Os sinalizadores, representam o sinal de cada canal ou sensor reproduzidos pela a saída do gravador AMPEX, através destes sinalizadores pode-se visualizar a condição de reprodução de cada sensor. Se o sensor reproduzido pelo gravador AMPEX está em sincronismo (sinalizador é representado pela cor verde) ou se o sensor está fora de sincronismo (sinalizador é representado pela cor vermelha). Estando o sinalizador ou "LED virtual" na cor vermelha (condição fora de sincronismo), deve se fazer ajustes na calibração do Gravador AMPEX, através de resistores ajustáveis (trimpots), localizados nos cartões de reprodução do gravador, até que o sinalizador retorne a condição de sincronismo novamente ( sinalizador na cor verde). Outros dados também pode ser visualizados pela tela de aquisição, tais como, o código de tempo da fita ( Time Code), Comprimento de cada Linha de varredura (Line Length ),e a taxa de transferencia dos dados em (Kbits/s).

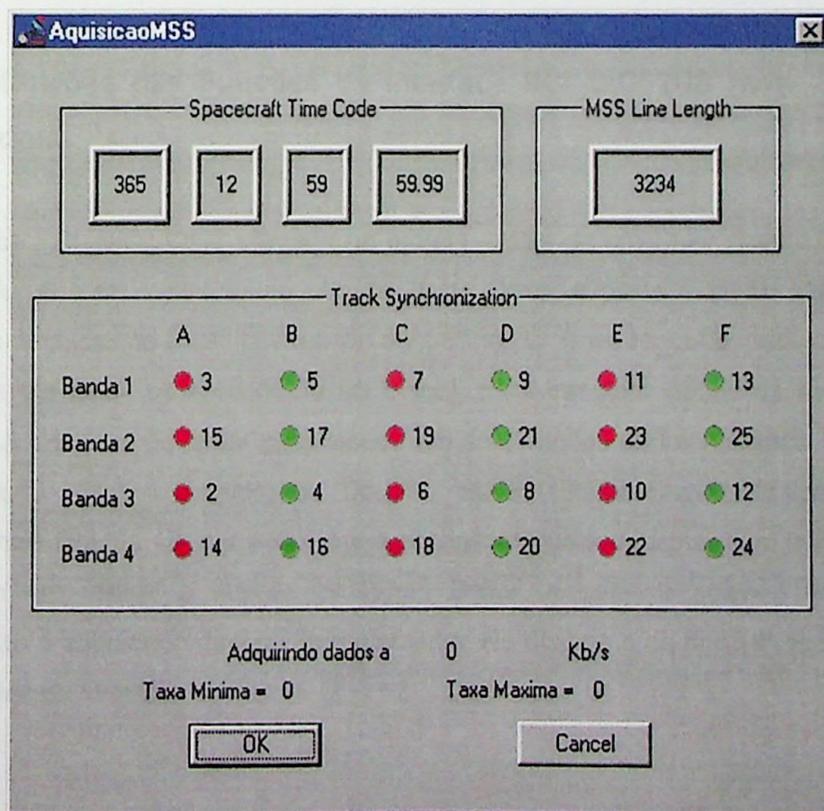
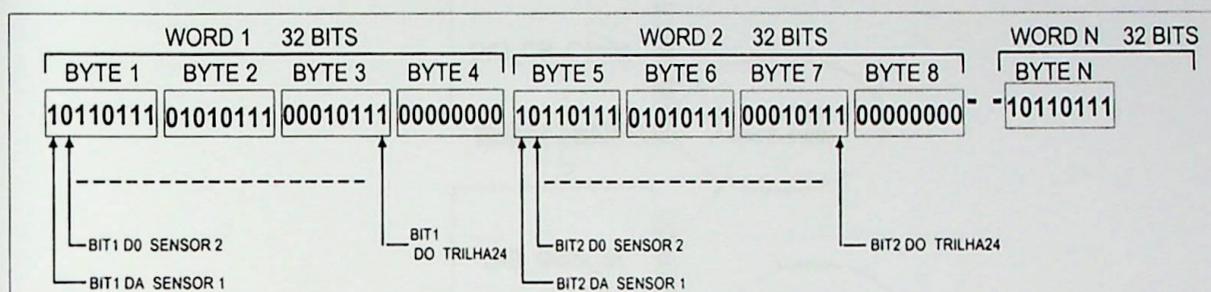


Fig. 5.2) Tela de Visualização e Controle do Software de Aquisição

### 5.3 Formato dos Dados Brutos Gravados pelo Software de Aquisição .

O software de aquisição dos dados Brutos foi desenvolvido na linguagem Visual C++, o software tem a função de fazer a leitura dos dados MSS da saída da interface IFDM, utilizando a interface PCI\_DIO\_32HS no modo BURST de Aquisição. Os dados continuamente recebidos via interface PCI\_DIO\_32HS são transferidos e escritos em um único arquivo no disco do computador durante o período de aquisição dos dados , através de aquisição de 4 bytes consecutivos ou 32 bit, sendo que o quarto byte é desprezado (Fig 5.3), onde cada bit de dados, é representado pelo bit de cada sensor consecutivamente.



*Fig. 5.3) Formato dos Dados Brutos Gravados no Hard -Disk*

O tamanho da palavra ou byte de cada sensor é de 6 bits, logo para a geração de 1 bytes de dados MSS com palavra de 6 bits para cada sensor, são necessários a aquisição de 6 words (32 bits) ou 24 bytes de 8 bits.

### 5.4 Definições das Funções da Interface PCI\_DIO32HS para transferencia dos dados .

A Interface PCI\_DIO\_32HS utilizada no projeto, utiliza de um Diver chamado NI\_DAO, fornecido pela National Instruments, onde através deste Drive, a placa é capaz de transferir os dados utilizando técnicas de DMA, interrupção ou polling. O Drive NI\_DAO, pode utilizar a transferencia de dados via DMA (Acesso direto ao Disco), para transferir dados da memória acima de 16 Mbytes. A interface pode ser programada em dois modos de transferencia de dados, tais como: Single Buffer (Buffer simples) ou Double \_Buffer (Buffer Duplo). Utilizando-se o modo de transferencia Double -Buffer o sistema é configurado para trabalhar com buffer circular, a grande vantagem em utilizar a técnica de Duplo Buffer, é que o sistema esta simultaneamente escrevendo e adquirindo dados interruptamente. No diagrama da fig (5.4) ilustrada a programação das funções da Interface .

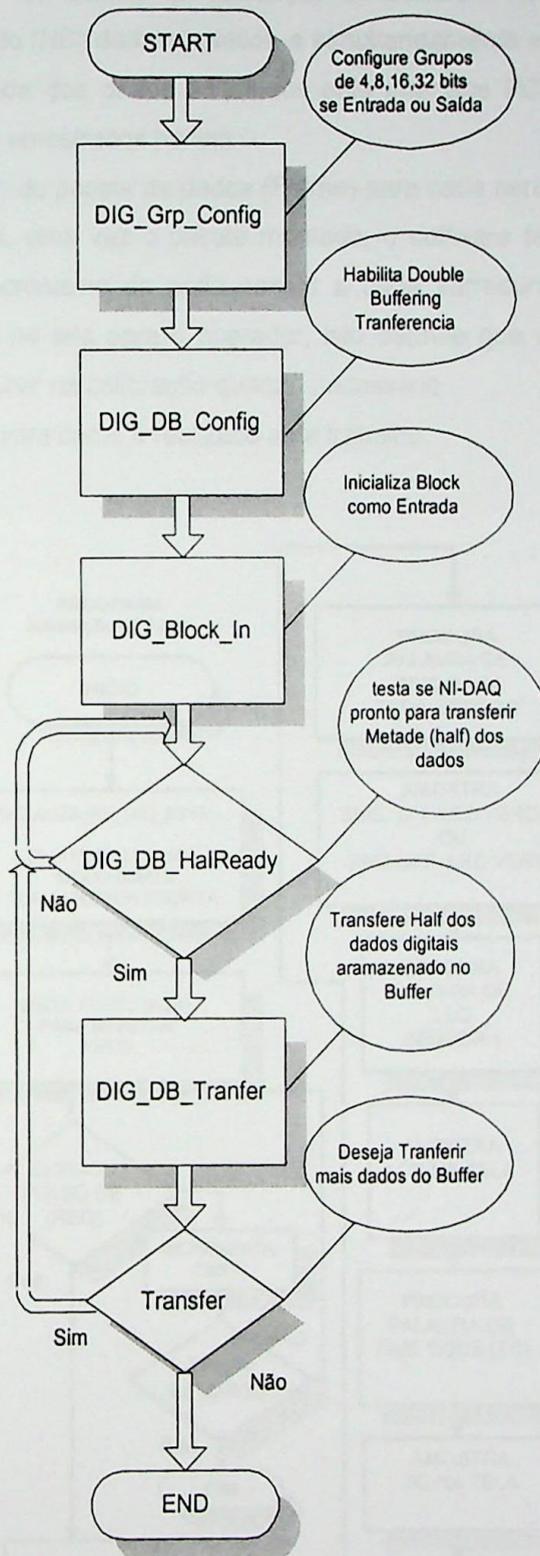


Fig. 5.4) Fluxograma Modo Double – Buffer de Transferencia dos Dados

### 5.5 Descrição do Software de Aquisição.

O Software de Aquisição de dados, tem a função de transferir os dados reproduzidos pelo gravador para o disco rígido (HD) do computador, e simultaneamente a função de monitoramento em tempo real da qualidade dos dados adquiridos pela interface PCI\_DIO\_32HS, através dos sinalizadores dos sensores amostrados na tela .

O software faz a montagem do pacote de dados (Frame) para cada sensor MSS, reproduzido pelo saída do gravador AMPEX, uma vez o pacote montado, o software tem a responsabilidade em procurar a palavra de sincronismo de cada sensor a cada varredura ou linha de imagem, e amostrar simultaneamente na tela para o operador, isto permite que operador possa analisar a qualidade dos dados, e intervir na calibração quando necessário .

O fluxograma da Fig. 5.5 ilustra como é realizado este trabalho.

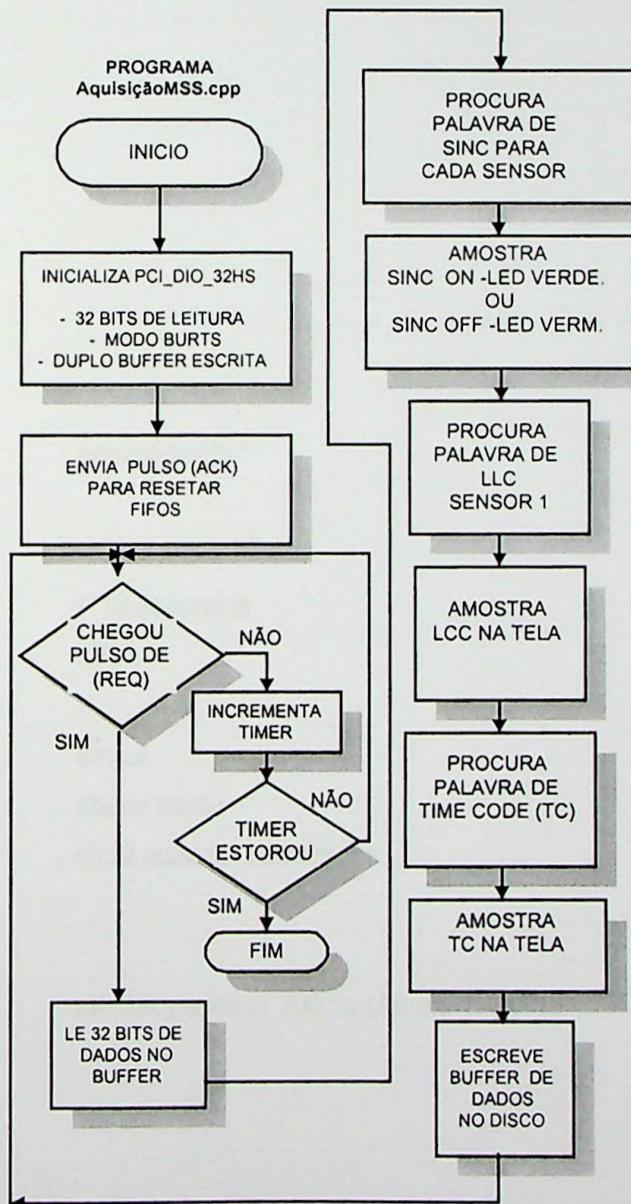


Fig- 5.5) Fluxograma de Operação do Software de Aquisição

## Arquivos fontes do Programa de Aquisição dos dados Brutos em Visual C++

- AcquireMSS . cpp            Fonte em Visual C++
- AcquireMSS . dsw
- AcquireMSS . dsp
- AquisicaoMSS . h
- AquisicaoMSSDIg . cpp
- AquisicaoMSSDIg . h
- AcquireMSS.EXE –            Aplicativo

Para o programa de aquisição o software necessita de configurar alguns parâmetros para o drive NI-DAQ da placa de aquisição PCI\_DIO-32HS, como mostrado abaixo.

```

ulCount = 32768;
iLoopCount = 0;
iStatus = 0;
iDevice = 1;
iGroup = 1;
iRetVal = 0;
ignoreWarning = 0;
iDBModeOFF = 0;
OldDataStop = 0;            //desabilitado
iPartialTransfer = 0;
iHalfReady;
i32 ITimeout = 180;        // 10 segundos
iGroupSize = 4;            //32 bits
iPort = 0;
Dir = 0;                    //input
Signal = 3;                //burst mode
iEdge = 0;                 // not used for burst mode
iReqPol = 0;
iAckPol = 0;
iAckDelayTime = 7;        //PCLK period of 700 ns (Aprox. 1.4 MHz)
iDBModeON = 1;

```

### 5.6 Software de Extração dos Dados dos Sensores a partir dos dados Brutos obtidos para Análise da qualidade dos dados gravados

O software de Extração dos dados dos sensores a partir dos dados brutos, foi desenvolvido com objetivo principal de analisar a qualidade e consistência dos dados brutos obtidos gravados no disco rígido do computador. Este software consiste em ler o arquivo bruto obtido, e extrair os somente os dados de imagens correspondentes aos 24 sensores do sensor MSS.

Depois de executado, este software cria um diretório na raiz do computador , chamado "/mssdados", onde são gerados 24 arquivos de imagens de vídeo dos sensores, representados pelos nomes de Video02.dat até Video25.dat, ilustrado no diagrama da fig.5.6

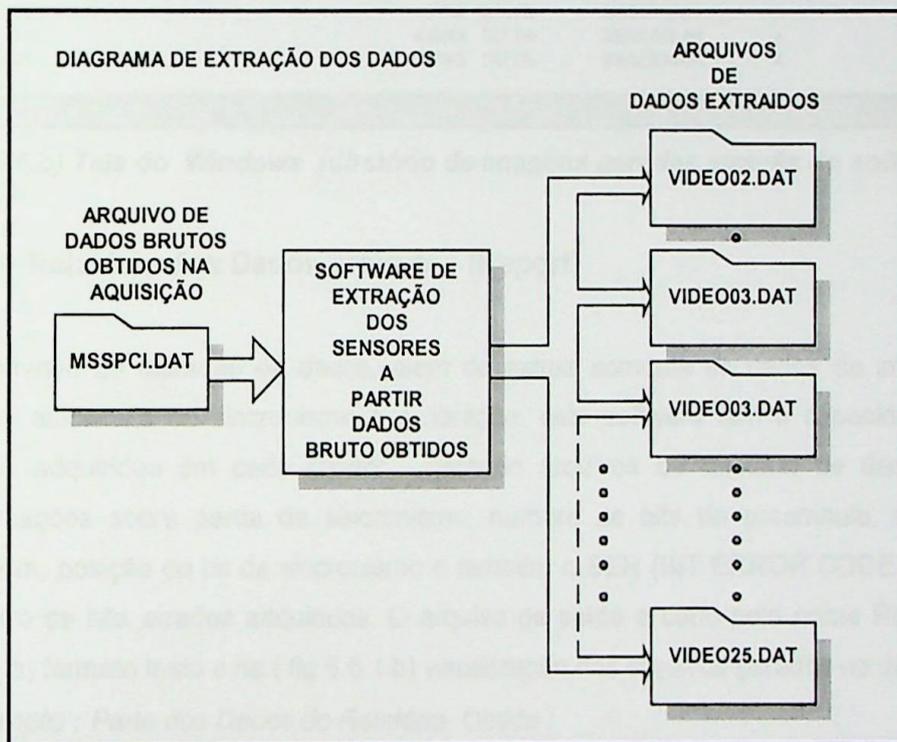


Fig 5.6) Diagrama em Blocos do software extração dos dados MSS.

Na fig 5.6.b , ilustra a tela capturada do diretório c:\mssdados , onde-se pode visualizar os 24 arquivos de imagens gerados a través do software de extração de dados MSS.



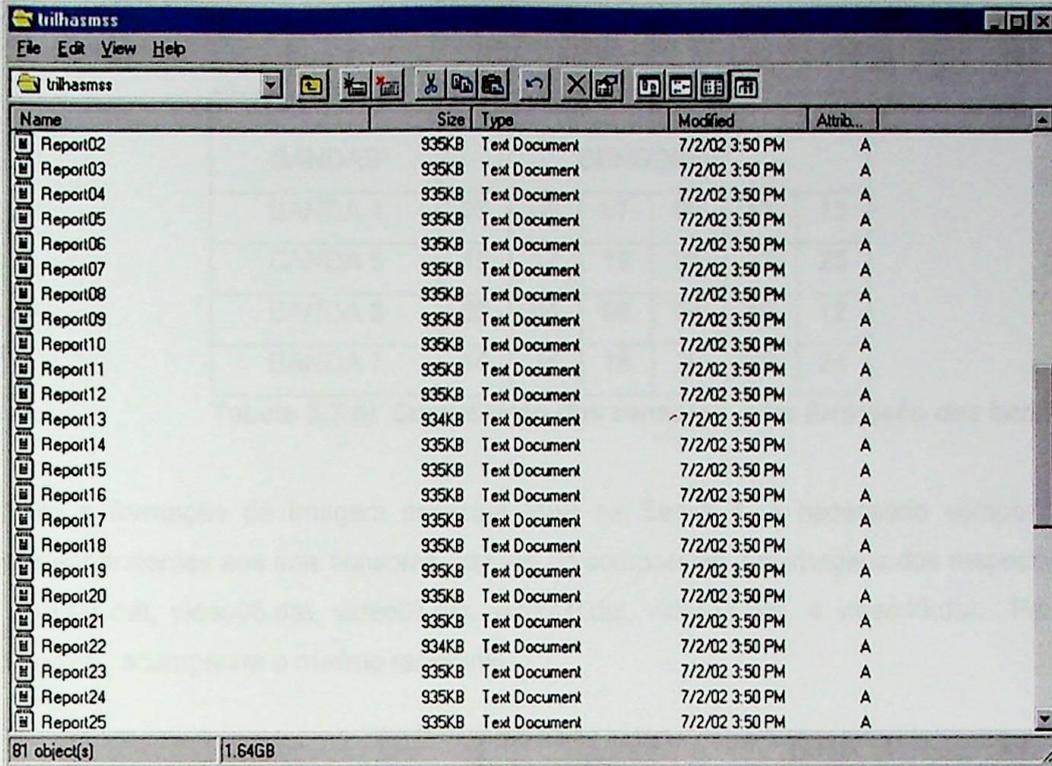


Fig (5.6.1.b) Tela do windows ilustrando o diretório dos dados de Report obtidos.

### 5.7 Formação das Imagens Espectrais ou Bandas do Sistema MSS

A câmera ou sistema imageador instalado no satélite Landsat 1,2 e 3 possui 4 canais ou bandas, com resolução espectral diferentes, conforme descrito no capítulo 2 que são :

<b>Bandas</b>	<b>Resolução</b>
Canal 4	0,4 a 0,5 $\mu\text{m}$
Canal 5	0,5 a 0,6 $\mu\text{m}$
Canal 6	0,6 a 0,7 $\mu\text{m}$
Canal 7	0,7 a 1,1 $\mu\text{m}$

Tabela 5.7.a) Bandas Espectrais com seus respectivos Comprimento de Onda.

Para composição de uma imagem de uma banda, deve-se utilizar de 6 sensores de imagens correspondentes a banda, para compor a imagem da banda ou canal.

A tabela 5.7.b ilustra os sensores utilizados para a formação das respectivas bandas.

BANDAS	SENSORES					
BANDA 4	03	05	07	09	11	13
BANDA 5	15	17	19	21	23	25
BANDA 6	02	04	06	08	10	12
BANDA 7	14	16	18	20	22	24

Tabela 5.7.b) *Composição dos sensores para formação das bandas*

Para a formação de Imagem espectral MSS na Banda 4, é necessário compor as imagens correspondentes aos seis sensores, através da composição das imagens dos respectivos arquivos video03.dat, video05.dat, video07.dat, video09.dat, video11.dat e video13.dat. Para as outras bandas, acompanha o mesmo raciocínio.

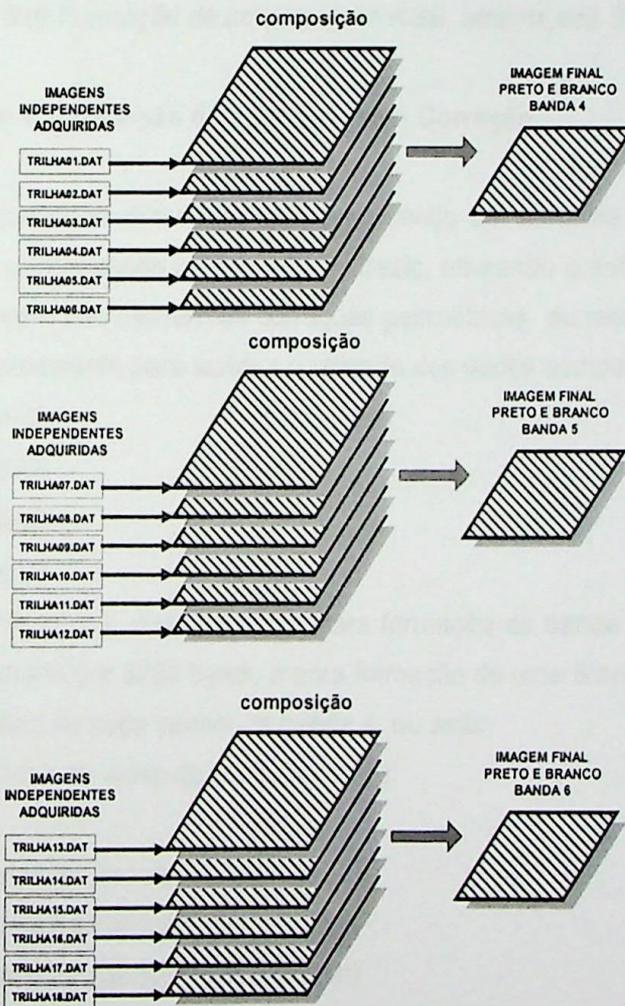


Fig. 5.7.c) *Ilustração da formação das Bandas Espectrais*

Desta forma formam-se as imagens espectrais das bandas respectivamente, mas caso seja necessário obter uma imagem espectral colorida tipo RGB, deve-se utilizar de uma composição de três bandas espectrais, para uma composição colorida, conforme ilustra o diagrama da Fig 5.5.d

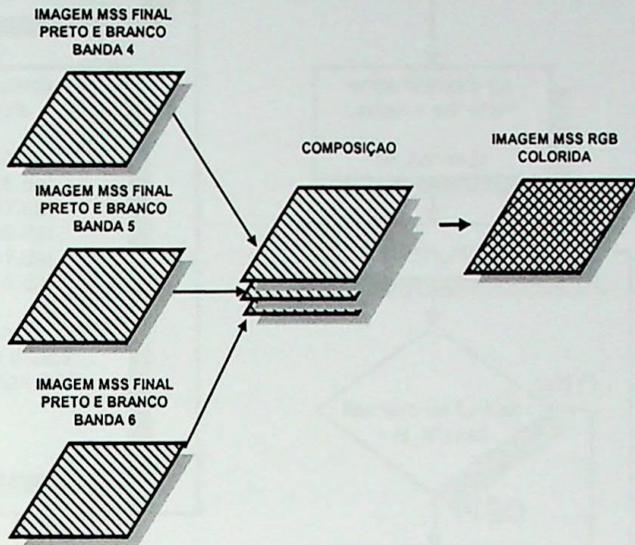


Fig. 5.5.d) Formação de uma Imagem RGB, através das 3 bandas.

### 5.7.1 Software de Composição das Bandas sem Correção.

Para analisar a qualidade dos dados foi desenvolvido um software de composição das bandas. Este software foi desenvolvido na linguagem Basic, utilizando o software de programação Quick Basic, Neste software não incluem as correções geométricas ou radiométricas [5.8], e somente foi desenvolvido simplesmente para testar a qualidade dos dados compostos, os arquivos fontes são :

- MSSBAND4.BAS
- MSSBAND5.BAS
- MSSBAND6.BAS
- MSSBAND7.BAS

O fluxograma da Fig 5.7.1, ilustra a rotina para formação da banda 4, cada linha de imagem de cada sensor é formado por 3280 bytes, e para formação de uma linha de vídeo para uma banda, é necessário ler 1 linha de cada sensor da banda 4, ou seja:

Para formar uma linha de vídeo da Banda 4 temos:

- 1 linha ou 3280 bytes do Sensor 3 (video03.dat)
- 1 linha ou 3280 bytes do Sensor 5 (video05.dat)
- 1 linha ou 3280 bytes do Sensor 7 (video05.dat)
- 1 linha ou 3280 bytes do Sensor 9 (video05.dat)
- 1 linha ou 3280 bytes do Sensor 11 (video11.dat)
- 1 linha ou 3280 bytes do Sensor 13 (video13.dat)

O Fluxograma da Fig.5.7.1 representa o software de composição da banda 4 ( MSSBAND4.BAS)

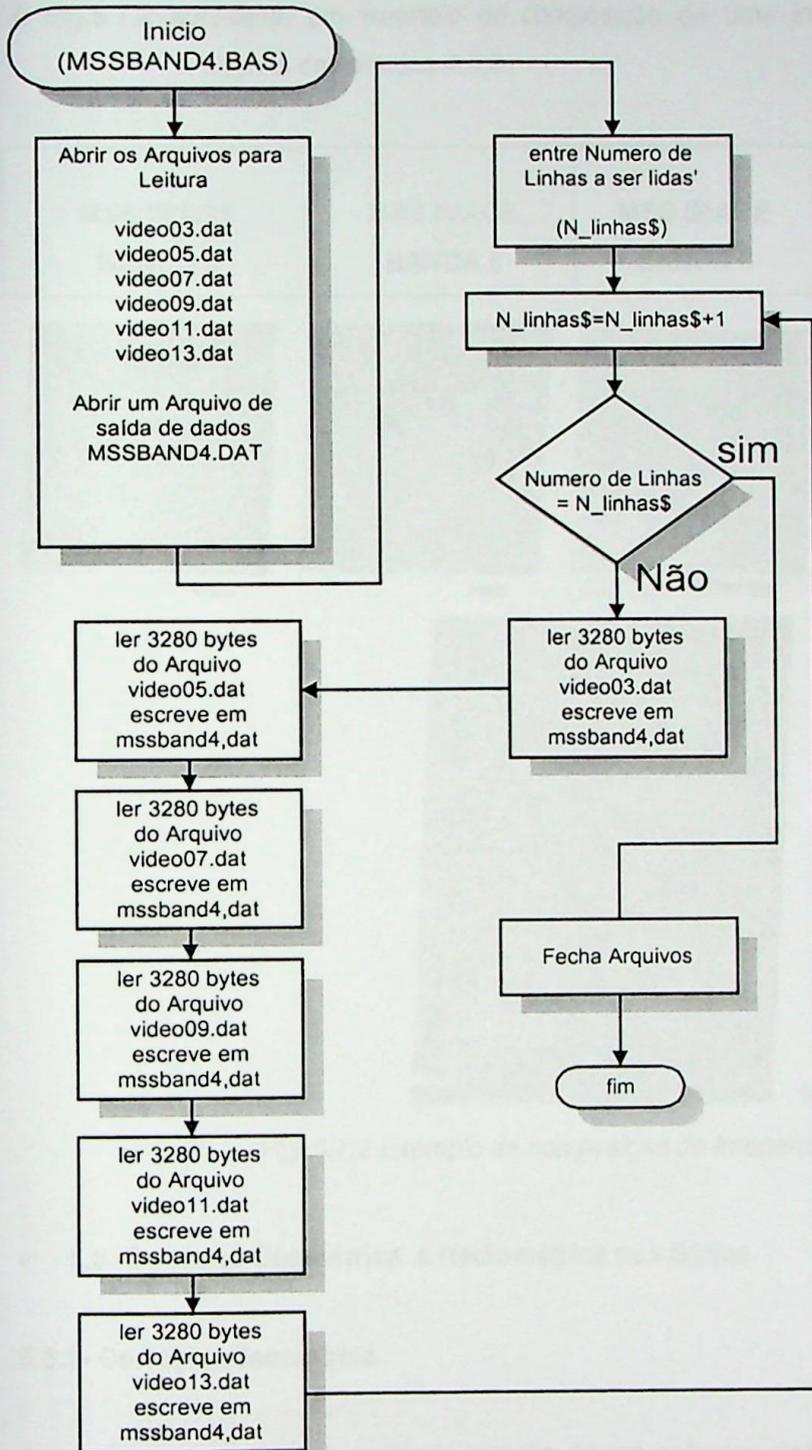


Fig 5.7.1 Fluxograma para o Programa MSSBAND4.BAS

### 5.7.2 Exemplo de Composição das Imagens MSS

A Fig.5.7.2 apresenta um exemplo de composição de uma imagem MSS-RGB, a partir das imagens das bandas 4,5,6

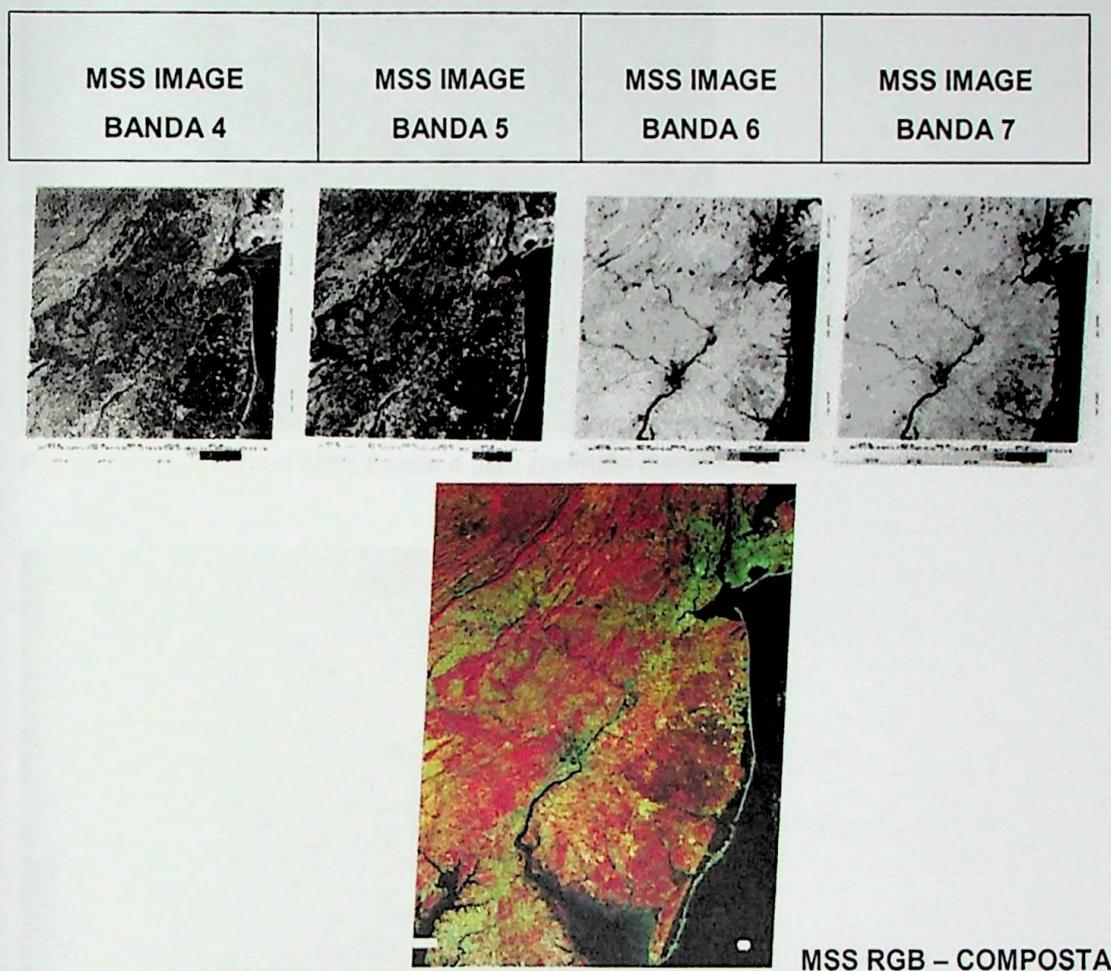


Fig. 5.7.2 Exemplo de composição de Imagens

## • 5.8 Correção Geométrica e Radiométrica dos Dados

### 5.8.1- Correção Geométrica

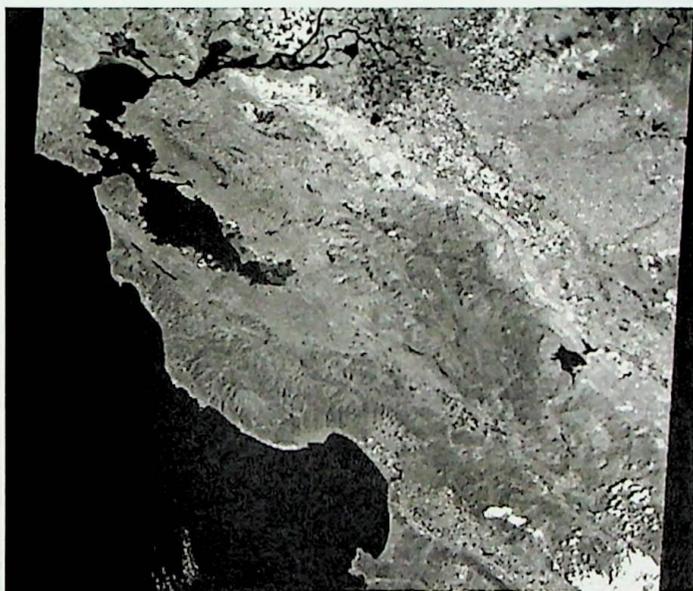
A correção geométrica é realizada através de um software dedicado para corrigir os efeitos de distorção nas imagens, que são causados, pelos seguintes casos:

- Rotação da Terra
- Variação de Velocidade do Espelho Escaneador (Mirror Scanning)
- Relação de aspecto da proporção de Pixels (Aspect Ratio Pixel)
- Diferença entre altura e largura

A Fig 5.8.1.a ilustra um exemplo de uma imagem MSS na banda 4 sem correção geométrica, onde se pode observar, que a imagem aparece esticada e retangular. A Fig 5.81.b ilustra a mesma imagem corrigida geometricamente após a aplicação do software de correção geométrica.



*Fig. 5.8.1.a - Imagem MSS banda 4 sem correção Geométrica*



*Fig. 5.8.1.b - Imagem MSS banda 4 corrigida Geométricamente.*

### **5.8.2- Correção Radiométrica**

As imagens orbitais obtidas pelos sensores podem, em um primeiro momento, ser muito escuras, dificultando até mesmo a identificação de um alvo na cena, devido a inconsistência ou erros nos valores de brilho e contraste da imagem. Vários fatores são responsáveis por este fenômeno, sendo que os principais são:

- Baixa inclinação solar quando da aquisição da imagem.
- Umidade elevada da região da cena em função de pluviosidade operante ou recente.
- Associação dos fatores anteriores.

Para a perfeita análise de uma cena com este problema é necessário a correção radiométrica da imagem, podendo essa correção ser obtida através de um procedimento de processamento chamado de "stretching", ou seja uma imagem de satélite é formada por uma população de "pixels", onde cada "pixel" varia sua intensidade radiométrica entre 0 e 63 tons de cinza. A distribuição estatística de uma população qualquer em função de um atributo considerado, tende a se comportar como uma curva normal. Em uma histograma que represente na linha horizontal o atributo radiométrico com 64 níveis, e na linha vertical a frequência de repetição do "pixel" tende a uma distribuição normal. Nas imagens escuras, a distribuição normal tem uma aproximação para valores tendentes a zero, se concentrando nos níveis de cinza mais baixos (CURRAN, 1985). A correção radiométrica das bandas individuais pode ser realizada através de ampliação de contraste linear das imagens com saturação de pontos de 2,5%.

A correção Radiométrica também utiliza-se de um procedimento de processamento, que visa a equalizar a resposta dos detetores de cada canal ou sensor do sistema imageador MSS, evitando o efeito de "Stripping - Lines" ou linhas de intensidade diferentes para alvos homogêneos,

No caso da imagem MSS temos 6 detetores por banda, cada um destes detetores necessita de uma calibração precisa com os outros detetores vizinhos de sua própria banda.

O método usado é adotado pela EROS Data Center - USA [7], utiliza a técnica de 2 passos para corrigir o efeito stripping.

PASSO 1: O dado para cada detector é normalizado utilizando a seguinte relação:

$$DNo_{(j,s)} = DNi_{(j,s)} (Sa / Si) + M_A - M_I (Sa / Si)$$

Onde:

$DNo_{(j,s)}$   $\Rightarrow$   $n^o$  digital de pixel de saída da linha  $j$  e amostra  $s$

$DNi_{(j,s)}$   $\Rightarrow$   $n^o$  digital de pixel de entrada da linha  $j$  e amostra  $s$

$Sa$   $\Rightarrow$  padrão de desvio entre cena.

$Si$   $\Rightarrow$  padrão de desvio de cada detector individual.

$M_A$   $\Rightarrow$  valor digital da cena.

$M_I$   $\Rightarrow$  valor digital de cada detector individual.

O primeiro passo não elimina totalmente o efeito striping, logo utilizamos o 2 passo, que é dado pela seguinte relação:

$$DNo_{(j,s)} = DNi_{(j,s)} - D$$

Onde :

$$D = LOCAVI - LOCAVI - 1$$

Sendo o termo *LOCAV* o valor ao longo de uma linha utilizando 75 pixels na mesma direção

Uma segunda aproximação, para eliminar o efeito do stripping é aplicação da Transformada de Fourier. A transformada de Fourier decompõe a imagem na direção das componentes de imagens, isto faz com que seja facilmente isolado e corrigido o efeito distorcido da imagem. A seqüências de Figuras 5.8.2 ilustra a correção de stripping lines na imagem:

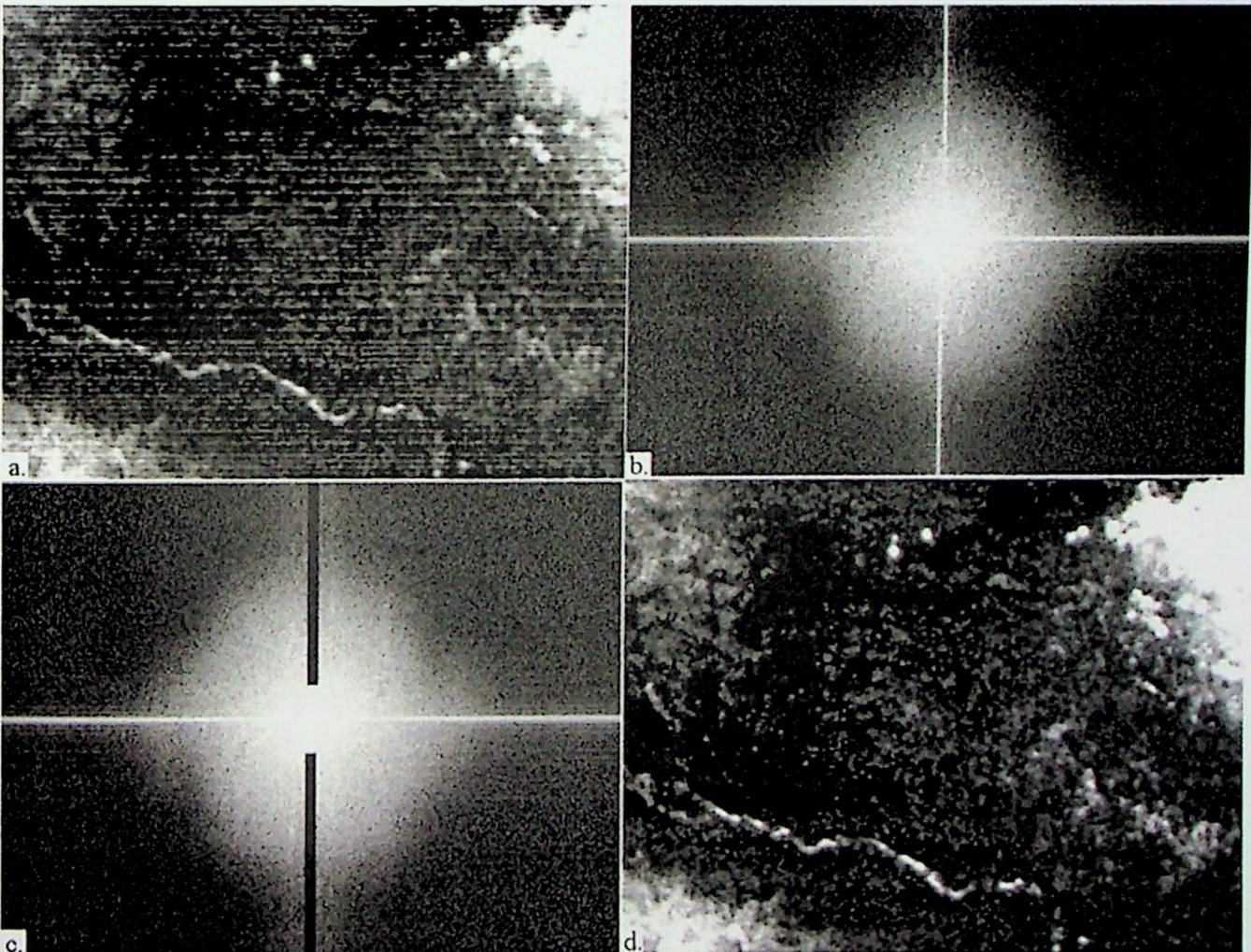


Fig 5.8.2 Correção da distorção da Imagem causada pelo efeito do Stripping Lines.

## CAPÍTULO 6: Resultados Práticos

### 6.1 Fotos do Hardware do Sistema de Aquisição

A seguir, a título de ilustração são apresentadas as fotos do hardware do sistema de aquisição, com a descrição dos seus subsistemas .

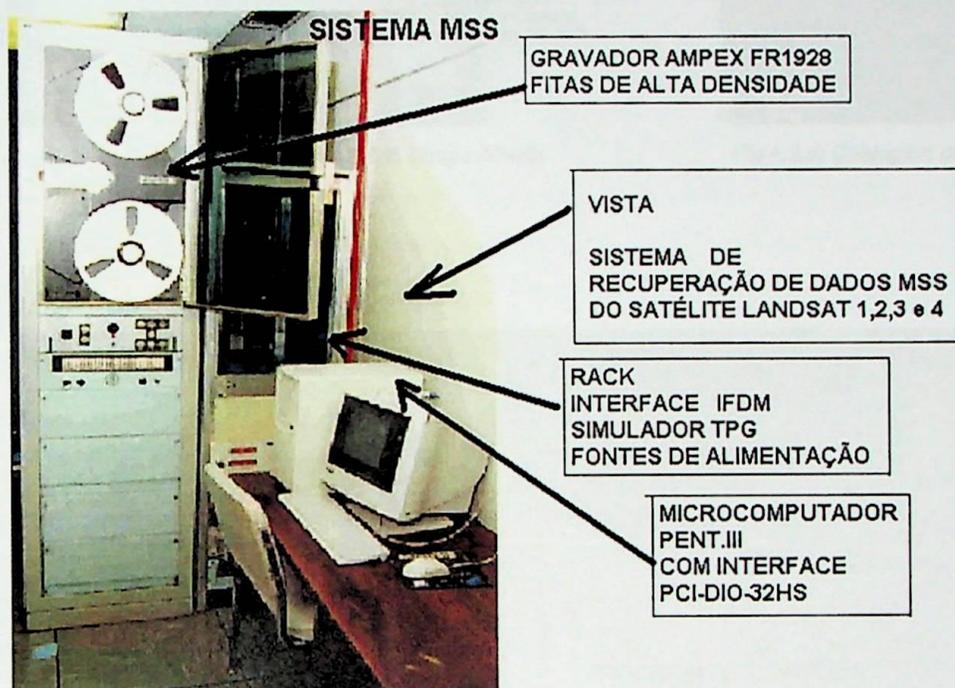
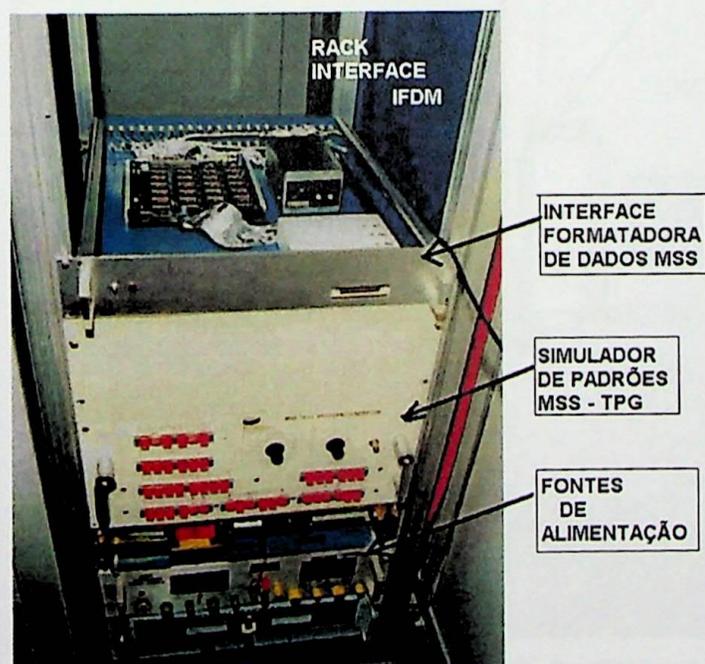


Fig. 6.1.a Vista Parcial do Sistema MSS



6.2 b Vista do Rack da Interface IFDM

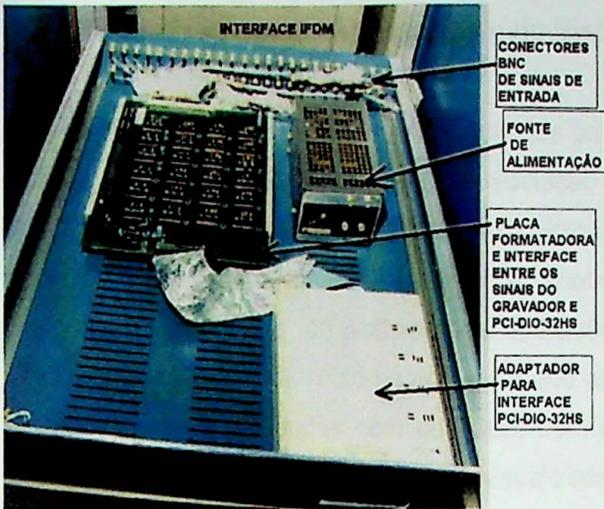


Fig 6.1.c) Vista Frontal Interface IFDM Tampa Aberta



Fig 6.2.d) Cablagem da Interface IFM

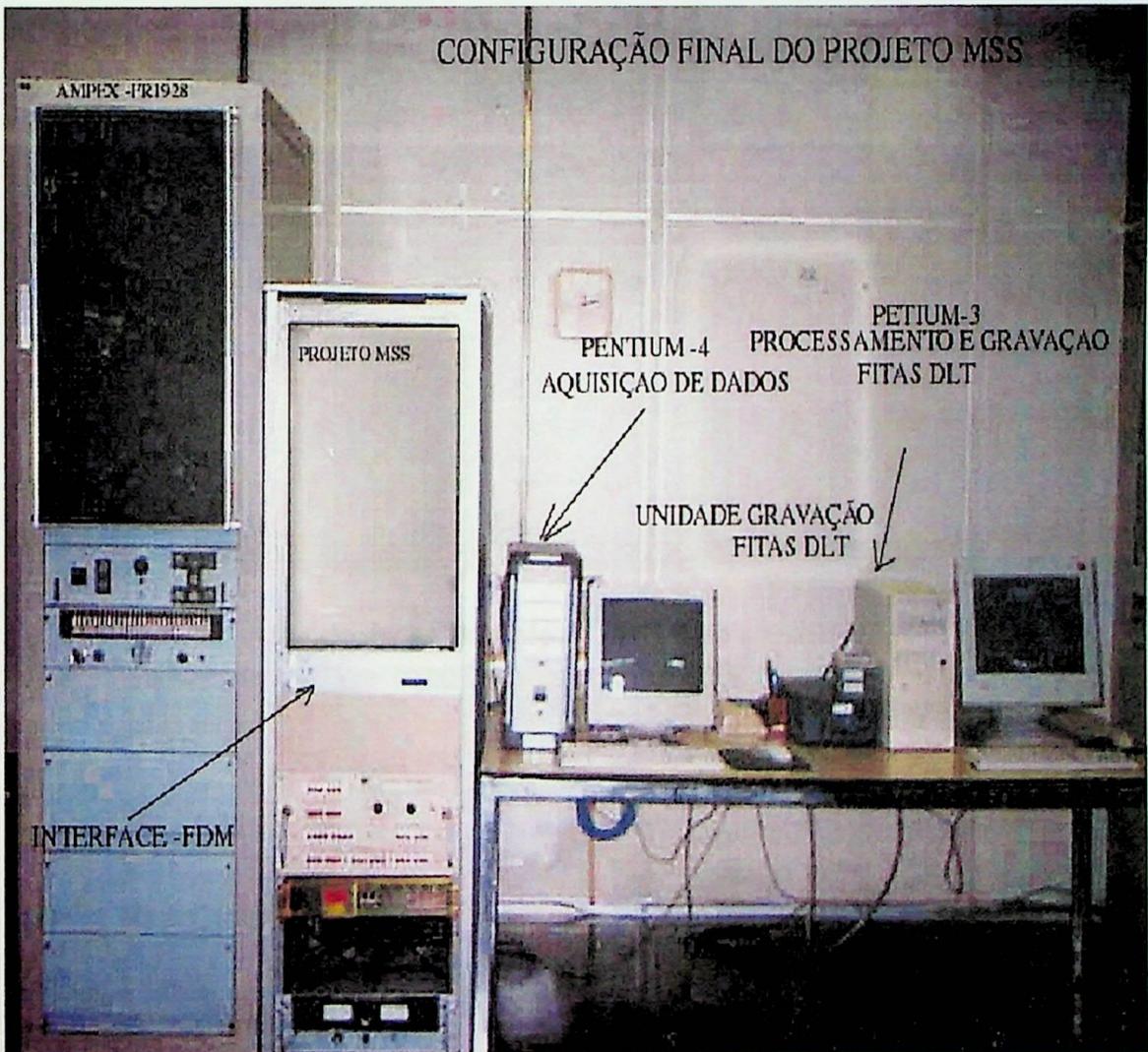
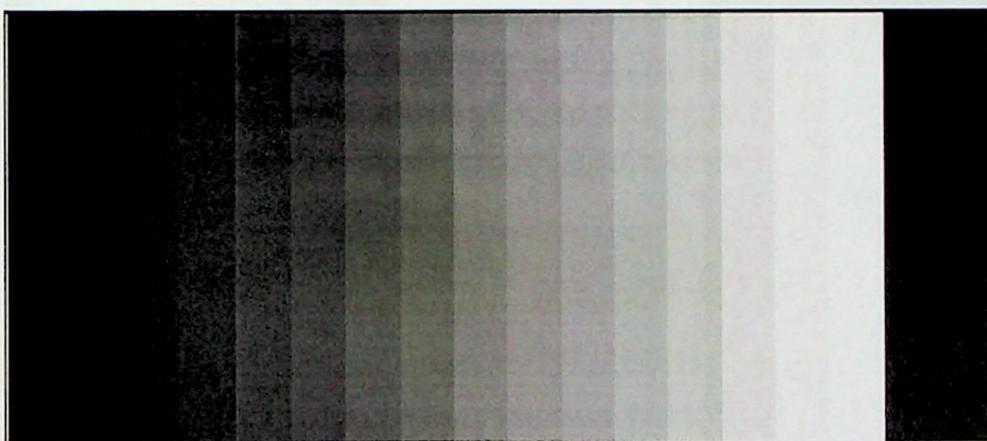


Fig.6.1.e) Vista da Configuração final do Projeto

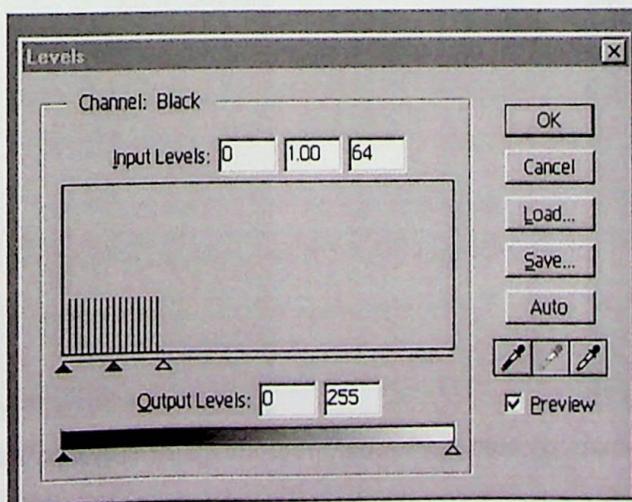
## 6.2 Dados de Imagens Adquiridas pelo Sistema de Aquisição MSS

As imagens ilustradas mostram os resultados práticos obtidos, mostrando a evolução do projeto quanto a observância da qualidade e resolução das imagens obtidas, ou seja mediante ajustes tanto do hardware como do software, possibilitou a chegar ao resultado final, quanto qualidade e resolução das imagens MSS, que é de 80 metros. Estas imagens foram extraídas dos dados brutos obtidos através do software de aquisição, essa imagens não contem nenhum tipo de correção geométrica ou radiométrica, visto que objetivo do projeto é de apenas confirmar que as imagens foram transferidas com qualidade sem nenhum tipo de ruído gerado pelo hardware do sistema. As imagens ilustradas forem visualizadas através do software editor de imagens Adobe Phothoshop, no formato RAW (Bruto).

A Imagem ilustrada na fig 6.2.a foi obtida pelo sistema de aquisição de dados MSS,e gerada a partir do simulador TPG na configuração de linhas verticais de cinza ( Vertical Gray), este primeiro comprova que o sistema de aquisição , adquiriu os dados gerados com sucesso, este tipo de teste porém no comprova o sistema, pois os dados gerados pelo simulador não apresenta distorções de skew ou bit sleep .



*Fig. 6.2.a) Image teste Adquirida pelo Sistema, gerada elo simulador TPG (Vertical Gray)*



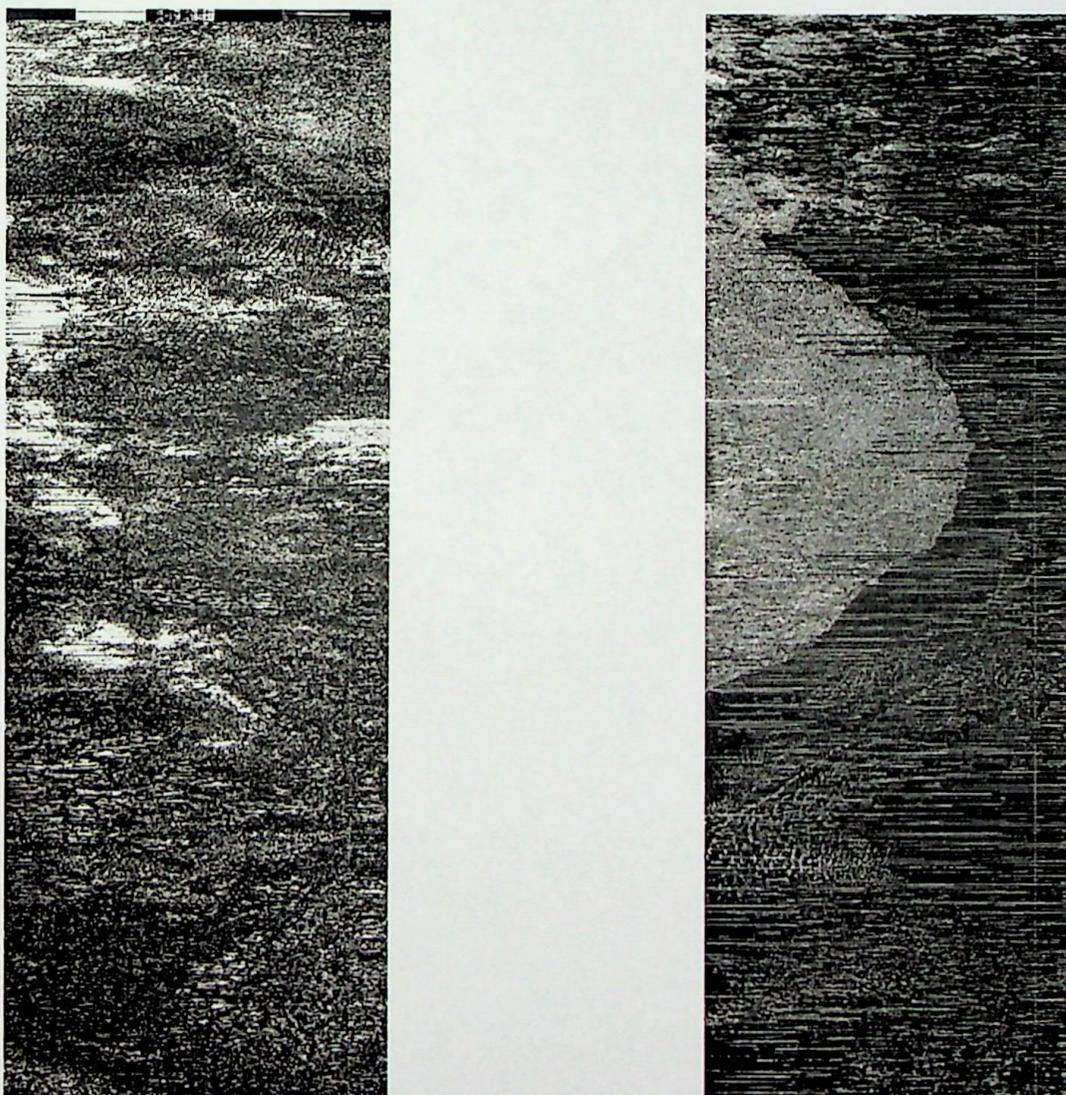
*Fig 6.2.b) Histograma do Programa Adobe Photoshop, indicado 64 níveis de cinza*

### 6.3) Evolução das Imagens ou Cenas Extraídas do Banco de Dados

Duas imagens mostradas na fig. 6.3.a , foram adquiridas pelo sistema desenvolvido, neste caso os dados são imagens das fita HDDT's, reproduzidos pelo gravador AMPEX FR1928. Durante este teste observou-se que as imagens apresentam baixíssima resolução, e um altíssimo nível de ruído com muita perda de informação, devido a total descalibração do sistema e do gravador AMPEX. A partir desta imagens obtidas, iniciaram-se os ajustes finais do hardware e do Software do sistema de recuperação de dados MSS.

Os ajustes apresentavam-se na forma de três variáveis independentes, que foram definidas respectivamente; ajuste da calibração do gravador FR1928, ajuste no Hardware da IFDM (interface Formatadora de Dados) e ajuste no Software de Aquisição.

As imagens da fig 6.3 analisadas correspondem somente a 1 sensor, no caso adotamos visualizar somente o sensor 24, e correspondem a duas cenas ou imagens de diferentes regiões do Brasil.



*Fig. 6.3.a)Primeiras Imagens teste, adquirida pelo Hardware e software desenvolvido.*

Após ajustes em quatro variáveis fundamental do projeto (Azimute da Cabeça do Gravador, Sincronismo dos cartoes Miller code , Software de Aquisição), gerou -se uma nova imagem ilustrada na Fig.6.3.b,:



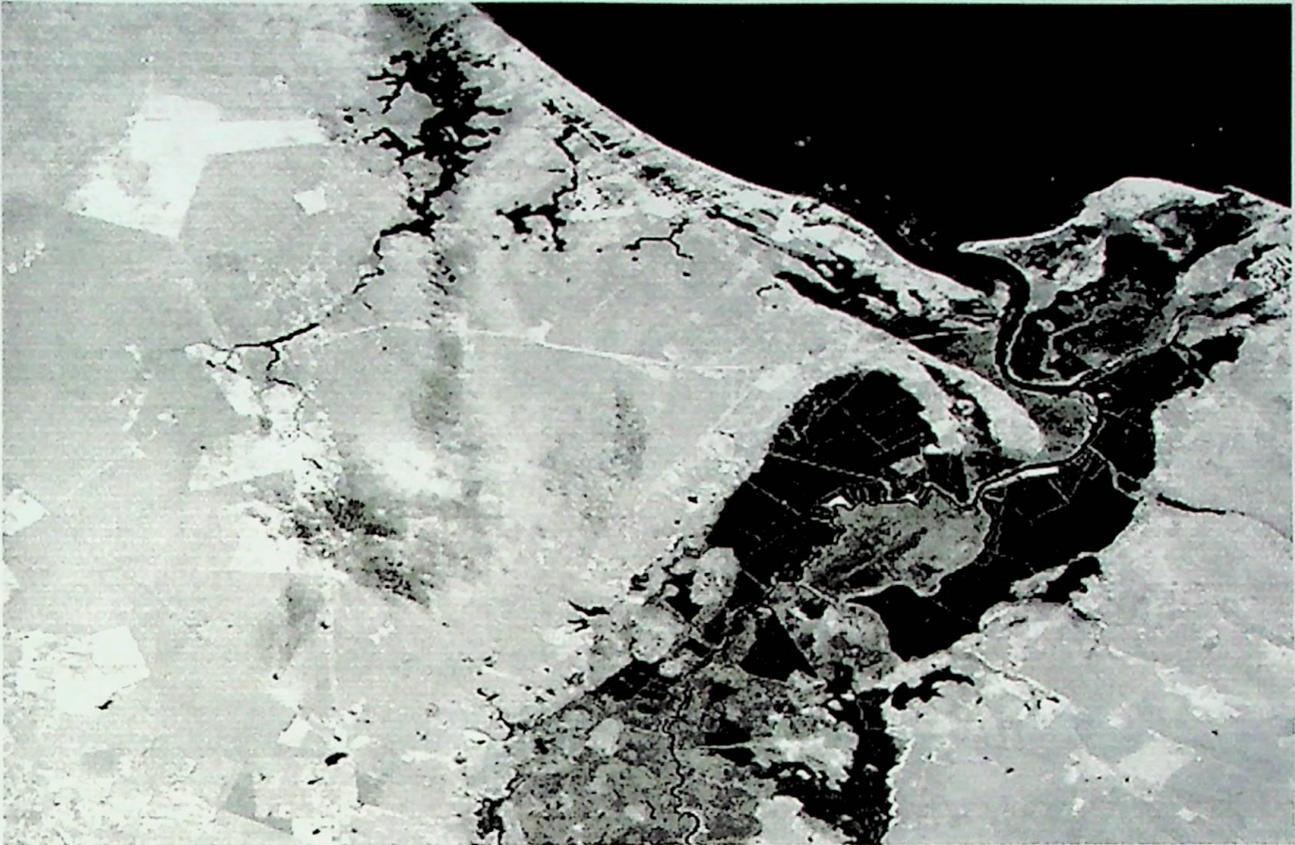
*Fig. 6.3.b) Image teste adquirida, pelo Hardware e software desenvolvido utilizando o Fitas do Banco de Dados data 1977. Região do Nordeste - Ilha de marajó.*

Observando a imagem (Fig 6.3.b), nota –se que a resolução aumentou razoavelmente, os pixels estão bem definidos, as linhas riscadas em branco na imagem é devido ao bit sleep, ou seja o problema originado pela perda de bit, necessitava de mais ajustes no hardware. do sistema e principalmente no Gravador AMPEX.

O próximo passo foi manter o software de aquisição inalterado e, ajustar o Hardware da IFDM, e o gravador AMPEX, onde resultou na solução do problema. Gerou-se uma nova imagem com a resolução final de 80 metros (fig 6.3.c), esta qualidade de resolução da imagem, foi percebida pela ampliação as imagens até 200 % . Uma outra confirmação da qualidade dos dados, mostrou-se através do desaparecimento do bit sleep .A seqüência de imagens ilustradas na Fig. 6.3, ilustra a região do rio grande do norte onde se concentram os grandes salinais.



**Fig 6.3.c) Imagem adquirida, pelo Hardware e software desenvolvido região litoral do Rio Grande do Norte ano de 1977 fator de aplicação ou zoom de 13 %**



*Fig 6.3.d ) Idem para ampliação de 50 %*



*Fig 6.3.e) Idem para ampliação de 100 %*



Fig 6.3.e )Idem para ampliação de 200 %

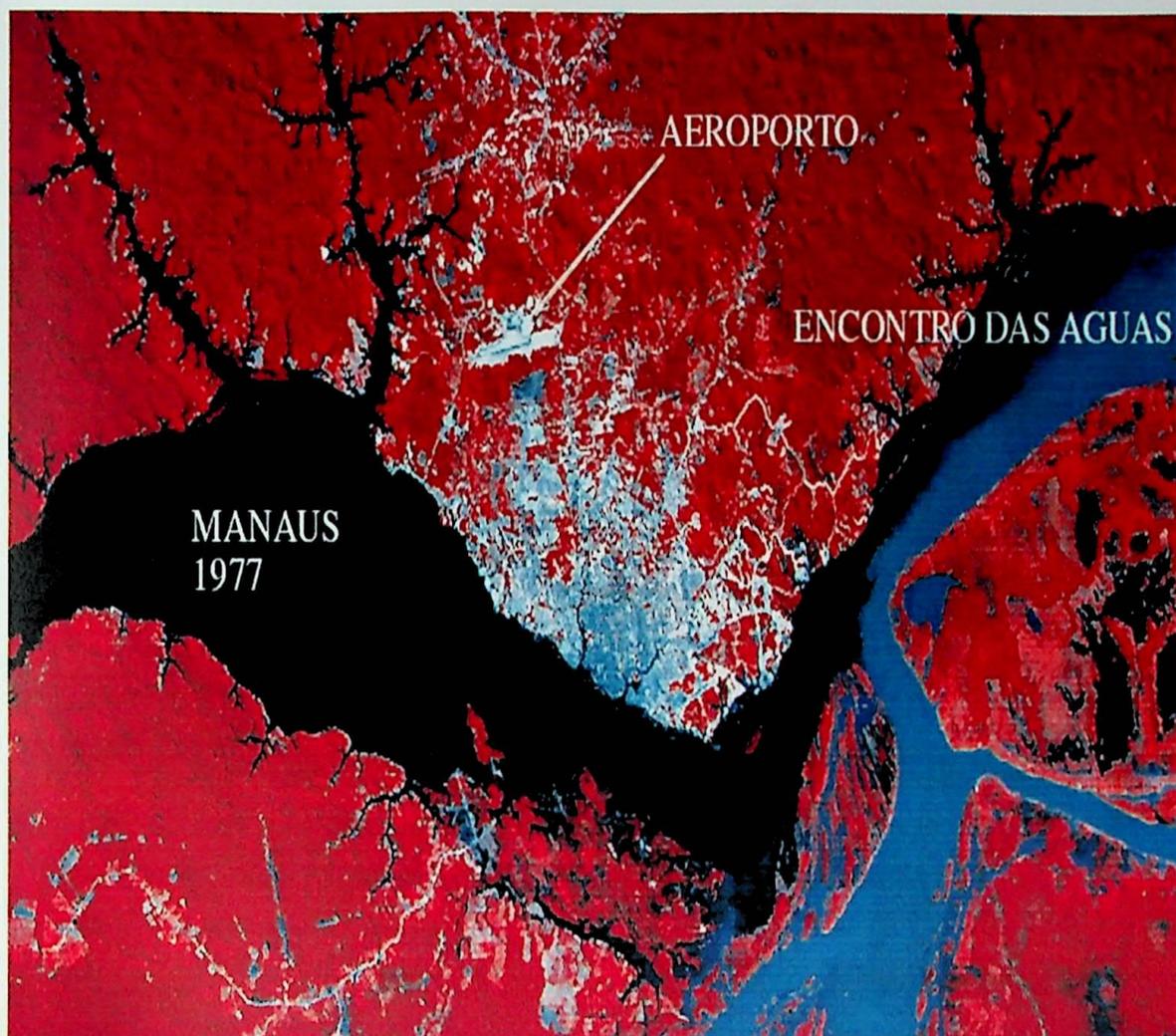


A imagem RGB composta pelas Bandas 3,4 e 7 ( Fig.6.3.e), foi adquirida pelo software e hardware do sistema de recuperação de dados desenvolvido. Um fator importante a observar nesta composição, é a capacidade de distinguir com mais clareza as informações de áreas urbanas, onde observa-se a cidade de Belém no estado do Pará.



**Fig. 6.3.e) Imagem teste Adquirida, pelo Hardware e software desenvolvido utilizando o Fitas do Banco de Dados data 1977.- Cidade de Belém.**

Uma outra composição RGB, adquirida também pelo sistema desenvolvido, (fig 6.3.f), ilustra a cidade de Manaus na região norte. Esta imagem confirma definitivamente a qualidade da resolução de 80 metros obtida, visto que pode-se observar detalhes na imagem com perfeita nitidez, no caso desta imagem observa-se a pista do aeroporto de Manaus, e o ponto onde ocorre o encontro das águas do Rio Negro e Solimões.



**Fig. 6.3.f – Image Adquirida pelo pelo Hardware e software desenvolvido ,utilizando o Fitas do Banco de Dados data 1977.  
Cidade de Manaus - Estado do Amazonas – Região Norte.**

A imagem MSS da Bandas- 4 (fig. 6.3.h), foi adquirida pelo sistema de recuperação de dados MSS desenvolvido, e corresponde a dados de imagem da região da Amazônia relativo ao Ano de 1977. A mesma cena ou imagem na (fig. 6.3. i), foi obtida do banco de dados do satélite Landsat -5, no ano de 1989, ou seja mais de dez anos depois. Através de uma comparação visual, demonstra os níveis de desflorestamento, com passar dos anos. Comparando as duas imagens podemos notar a expansão dos níveis de desflorestamento próximo a cidade de Manaus.



*Fig. 6.3.h – Image teste Adquirida pelo pelo Hardware e software desenvolvido utilizando o Fitas do Banco de Dados data 1977.Cidade de Manaus Estado do Amazonas – Região Norte.*



*Fig. 6.3.i – Image teste Adquirida pelo Hardware e software desenvolvido utilizando o Fitas do Banco de Dados data 1987*

## 6.4 Análise dos Resultados Obtidos

Os resultados obtidos são demonstrados através da grau de qualidade e resolução das imagens obtidas, o sistema MSS tem como o propósito fundamental, em transferir as imagens que estão armazenada em fitas magnéticas digitais de alta densidade (HDDT), para fitas digitais (DLT), de alta capacidade de armazenamento, a simples transferencia não comprova a qualidade dos dados. Devido a este fato foi necessário extrair os dados de imagens, a partir dos dados brutos obtidos e analisar a resolução e a qualidade das imagens obtidas, até que se comprovasse a resolução de 80 metros estipulada pelo satélites Landsat daquela época. Os dados acima foram analisados, por peritos em sensoriamento remoto do INPE, certificando a qualidade dos dados obtidos na aquisição. O sistema foi certificado, e entra agora na etapa de operação.

Como podemos observar as imagens de alguns sensores ilustradas na figuras anteriores, obtidas pela aquisição de dados foram visualizadas por um software comum tipo adobe photoshop, foram realizados o redimensionado do tamanho da imagem para um melhor visualização da imagem, demonstrando que os dados são coerentes, os resultados obtidos necessitam da aplicação de software especialistas para correções geométricas, radiométricas além de correção em eventual dado ruidoso , esta calibração das imagens, será realizada em uma segunda etapa do projeto, visto que, uma vez as fitas resguardadas em uma nova mídia, o fator tempo para realização da segunda etapa do projeto não é mais um caso crítico.

## CAPÍTULO 7: Plano de Operacionabilidade do Projeto MSS.

Para uma operacionabilidade ideal do sistema de recuperação de dados MSS, foram estabelecidas as seguintes metas r :

### • 7.1 Levantamento da Base de Dados

Sabemos que hoje o INPE possui armazenado no seu filмотeca, em torno de 2000 fitas magnéticas digitais de alta densidade HDDT's , sendo que cada fita de alta densidade possui em média em 3, orbitasdo continente brasileiro, e cada orbita em torno e 24 cenas de 185 km x 185 km, ou seja, em média, o INPE possui em torno de 250 mil cenas ou imagens de 185 km x 185 km, do território brasileiro do período dos anos de 1973 a 1984. Parte deste acervo está convertida em filme negativo e outra parte é desconhecida sendo que não existe nenhum produto deste tipo sob forma digital.

No início do projeto o INPE, realizou o levantamento desta base de dados, e estipulou inicialmente a transferencias dos dados de imagem contidas nas fitas HDDT's para DLT's, relativo ao ano de 1977.

Através de um levantamento das listagem das fitas HDDT's armazenadas no filмотeca do INPE, criou-se uma planilha de controle de salvamento dos dados, com as informações da ordem dos dados a ser salvos, ilustrado na fig.7.1

12										=
A	B	C	D	E	F	G	H	I	J	
<b>PROJETO MSS</b>										
1										
2										
3	<b>BASE - PUNTO</b>	<b>DATA</b>	<b>ÓRBITA</b>	<b>COBERTURA DE NUVEN</b>	<b>SATÉLITE</b>	<b>NÚMERO DA FITA</b>	<b>LOCALIZAÇÃO DA HDT</b>	<b>NÚMERO DA DLT</b>	<b>TIME CODE</b>	<b>OBSERVAÇÕES</b>
4	230 - 62									
5	230 - 63	29/10/77	14094	50%	Landsat II	12122	II - 10 - B	MSS001	11:19:12 - 11:25:49	Tripla 25 com problemas.
6	230 - 64	13/07/77	12588	60%	Landsat II	11096	II - 9 - B	MSS001	11:25:17 - 11:31:52	
7	230 - 65	13/07/77	12588	60%	Landsat II	11096	II - 9 - B	MSS001	11:25:17 - 11:31:52	
8	230 - 66	13/07/77	12588	100%	Landsat II	11096	II - 9 - B	MSS001	11:25:17 - 11:31:52	
9	230 - 67	19/02/77	10580	80%	Landsat II	9725	II - 8 - A	MSS001	11:32:30 - 11:39:04	
10	230 - 68	05/09/77	13341	50%	Landsat II	11606	II - 9 - D	MSS001	11:22:20 - 11:28:54	
11										
12	231 - 63	14/07/77	12602	60%	Landsat II	11118	II - 9 - B	MSS001	11:30:37 - 11:37:58	
13	231 - 64	14/07/77	12602	30%	Landsat II	11118	II - 9 - B	MSS001	11:30:37 - 11:37:58	
14	231 - 65	17/11/77	14359	30%	Landsat II	12319	II - 10 - B	MSS001	11:23:44 - 11:31:09	
15	231 - 66	17/11/77	14359	10%	Landsat II	12319	II - 10 - B	MSS001	11:23:44 - 11:31:09	
16	231 - 67	17/11/77	14359	30%	Landsat II	12319	II - 10 - B	MSS001	11:23:44 - 11:31:09	
17	231 - 68	19/08/77	13104	80%	Landsat II	11444	II - 9 - D	MSS001	11:28:37 - 11:36:01	
18	231 - 69	19/08/77	13104	40%	Landsat II	11444	II - 9 - D	MSS001	11:28:37 - 11:36:01	
19	231 - 70	01/08/77	12853	50%	Landsat II	11274	II - 9 - C	MSS001	11:29:37 - 11:37:02	
20	231 - 71	01/08/77	12853	40%	Landsat II	11274	II - 9 - C	MSS001	11:29:37 - 11:37:02	
21	231 - 72	01/08/77	12853	10%	Landsat II	11274	II - 9 - C	MSS001	11:29:37 - 11:37:02	
22	231 - 73	01/08/77	12853	20%	Landsat II	11274	II - 9 - C	MSS001	11:29:37 - 11:37:02	
23	231 - 74	01/08/77	12853	30%	Landsat II	11274	II - 9 - C	MSS001	11:29:37 - 11:37:02	
24	231 - 75	01/08/77	12853	0%	Landsat II	11274	II - 9 - C	MSS001	11:29:37 - 11:37:02	
25										
26	232 - 62	21/02/77	10608	80%	Landsat II	9768	II - 8 - A	MSS001	11:43:28 - 11:51:17	Tripla 25 com problemas.
27	232 - 63	07/09/77	13369	30%	Landsat II	11622	II - 9 - D	MSS001	11:33:14 - 11:41:05	
28	232 - 64	20/08/77	13118	20%	Landsat II	11452	II - 9 - D	MSS001	11:34:16 - 11:42:07	Fita enviada para os E.U.A.
29	232 - 65	20/08/77	13118	10%	Landsat II	11452	II - 9 - D	MSS001	11:34:16 - 11:42:07	
30	232 - 66	18/11/77	14373	0%	Landsat II	12327	II - 10 - B	MSS001	11:30:40 - 11:37:24	
31	232 - 67	18/11/77	14373	30%	Landsat II	12327	II - 10 - B	MSS001	11:30:40 - 11:37:24	
32	232 - 68	26/09/77	13620	90%	Landsat II	11800	II - 9 - E	MSS001	11:32:10 - 11:40:03	
33	232 - 69	15/07/77	12616	40%	Landsat II	11126	II - 9 - B	MSS001	11:36:17 - 11:44:04	
34	232 - 70	15/07/77	12616	30%	Landsat II	11126	II - 9 - B	MSS001	11:36:17 - 11:44:04	
35	232 - 71	15/07/77	12616	40%	Landsat II	11126	II - 9 - B	MSS001	11:36:17 - 11:44:04	

Fig.7.1 – Planilha de Controle de Salvamento de Dados MSS – Ano 1977

## 7.2 – Identificação dos Campos da Planilha de Controle.

- **Base Ponto:**

No primeiro campo ( Base Ponto ), indica a identificação das centenas de imagens enviadas pelo satélite, foi definido internacionalmente um sistema de referência, conhecido como WRS - World Reference System - que nada mais é do que uma grade, espaçada de uma cena, recobrando o globo terrestre. Os cruzamentos são identificados por 2 números: o número da órbita base (ou simplesmente "base") e o número do ponto. Por exemplo, a cidade de Brasília está localizada na cena de base 221 e ponto 71. Assim, para identificar a cena de interesse, basta informar os 2 números, a base e o ponto.

- **Data de Gravação da Passagem do Satélite:**

Como o satélite obtém as cenas repetitivamente, repassando no mesmo local a cada 16 dias, o campo ( data ) informar a data que foi gravada a órbita do satélite.

- **Órbita do Satélite**

Indica o número da órbita do satélite , gravado na fita de alta densidade HDDT.

- **Cobertura de nuvens:**

As imagens nem sempre mostram completamente a área, caso existam nuvens recobrando a região. O INPE classifica cada imagem pela porcentagem de nuvens que a recobrem. Uma cena livre de nuvens tem "cobertura zero", enquanto que outra, totalmente tomada pelas nuvens, tem "cobertura de 100%", ou "cobertura total".

- **Satélite**

Este campo informa qual o satélite que foi gravados os dados na fita HDDT.

- **Numero da Fita HDT**

Indica o numeração da fita HDDT

- **Localização da Fita HDT**

Este campo indica, a onde está localizada a fita HDT, ou seja em que armário e local está localizada a fita HDT no filmoteca do INPE.

- **Numero da DLT**

Este número indica o número da fita DLT (mídia atual) , que foi transferido os dados das fitas de alta densidade HDDT's.

- **Time Code**

Indica a hora inicial e a hora final de gravação dos dados da órbita .

## CAPÍTULO 8: Trabalhos Futuros utilizando o Projeto MSS.

Após a transferência da base de dados para a nova mídia digital (DLT), uma segunda etapa será aplicar um processamento digital nos dados brutos salvos, com este procedimento implementado, através de um software dedicado, o INPE estará disponibilizando um produto final, com imagens de satélites MSS históricas de 1973 a 1984, com correções geométrica e radiométricas, e geração de catálogo desta imagens .

Esta nova base de dados, deverá também ser disponibilizada, em um SBDG - Sistema de Banco de Dados Geográficos, e integrado no sistema do CDSR (Centro de Dados de Sensoriamento Remoto) do INPE., que visa para o futuro, atender as necessidades da área educacional do País, a comunidade científica na área de sensoriamento remoto e os usuários tradicionais do INPE.

## CAPÍTULO 9: Conclusões Finais

O Sistema MSS de Recuperação de Dados, é um sistema de Interfaceamento capaz de transferir dados de imagens de um gravador de fitas magnéticas de alta densidade, para uma outra mídia mais atualizada.

Neste trabalho foi apresentado o desenvolvimento de um sistema de interfaceamento implementado no INPE- DGI (Instituto Nacional de Pesquisas Espaciais- Departamento de Geração de Imagens), sendo o objetivo principal do projeto a criação de um Instrumento capaz de resguardar o acervo histórico de um a banco de dados de imagens geográficas . visando disponibilizar a comunidade e aos interesse em estudo dos recursos naturais e a evolução do meio ambiente.

O instrumento desenvolvido soluciona o problema de sincronismo existente, produzindo imagens no formato digital de melhor qualidade que as obtidas antigamente no processo fotográfico.

O circuito ficou relativamente simples, e de fácil manutenção, não necessitando técnicos especializados .

Como resultados práticos foram apresentados a imagens da floresta amazônica de 1977, estes dados comparados com os dados de imagens atuais mostram a viabilidade do estudo do desflorestamento ambiental do Brasil .

Como planejado o sistema esta em plena operação, sendo convertidas em torno de 3 fitas de alta densidade (18 orbitas ou 4464 cenas) por dia, devendo o acervo total, ser convertidas até o ano de 2004.

Todos os dados gravados são previamente processados para analise da qualidade dos mesmos, sendo somente dados brutos são armazenados em fitas DLT's .

Os dados de imagens brutas, estão agora armazenadas em nova mídia que podem durar até 50 anos, estes dados são utilizados pelos usuários aplicando-se a correção radiométrica e geométrica visando equalizar a resposta dos detetores de cada sensor, com isto possibilita maior resolução da imagem, corrigindo as distorções.

## Referências Bibliográficas

- [1] U.S. GEOLOGICAL SURVEY. *LANDSAT data users handbook*. Washington, DC, 1979.
- [2] STEFFEN, C. A. ; SOUZA de, R.C.M. Sistema LANDSAT. In: INSTITUTO DE PESQUISAS ESPACIAIS (INPE) – DEPARTAMENTO DE SENSORIAMENTO REMOTO.  
*Curso de treinamento: Introdução às Técnicas de Sensoriamento Remoto e Aplicações .*  
*Divisão de Transferência de Tecnologia*  
São José dos Campos, INPE, 1980. C.D.U:528.711.371.33 (INPE-1869-MD/004)
- [3] BENDIX AEROSPACE SYSTEM DIVISION. *INPERTS MSS data interface and Test Pattern Generator*, Michigan U.S.A, 1973.
- [4] BARBOSA, P. M. ;T. In: INSTITUTO DE PESQUISAS ESPACIAIS (INPE) – DEPARTAMENTO DE SENSORIAMENTO REMOTO.  
*Tratamento Automático de Imagens .* São José dos Campos, INPE, 1984. P. (INPE-3001-MD/024).
- [5] INSTITUTO DE PESQUISAS ESPACIAIS (INPE) . *Sistema LANDSAT: Estação brasileira de Cuiabá para recepção de Dados MSS .* São José dos Campos, INPE, 1984. P. (INPE-3151-TDL/168).
- [6] AMPEX CORPORATION ; FR1900 RECORDER/REPRODUCER. *Manual Maintenance Volume II* U.S.A, 1970.
- [7] USGS –Home Page . <http://edc.usgs.gov/>
- [8] NATIONAL INSTRUMENTS ; DAQ. DIO 6533 *User Manual –High –Speed Digital I/O Boards for PCI, PXI, CompactPCI, AT, EISA, or PCMCIA Bus System.* U.S.A 1997
- [10] MICROSOFT VISUAL C++ ; *Development System for Windows 95/ NT Version 4.* 1995
- [11] IDT DATA BOOK ; *Integrated Circuit .* U.S.A 1997

## Imagens RGB

IMAGEM RGB -MSS AQUIRIDA PELO SISTEMA DESENVOLVIDO

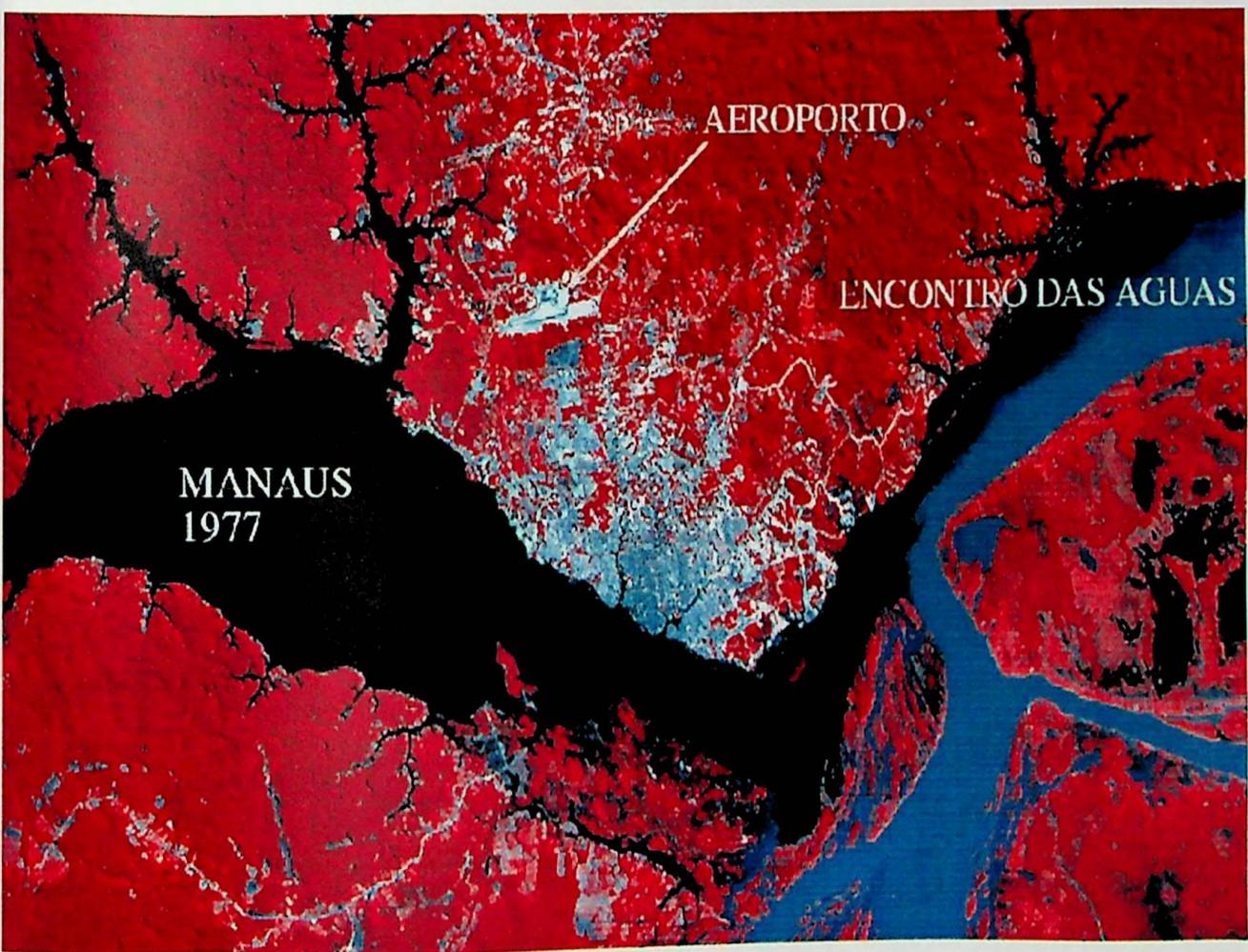


IMAGEM - MSS -LANDSAT 2 - CIDADE DE MANAUS -AM

IMAGEM MSS - ADQUIRIDA PELO SISTEMA DESENVOLVIDO

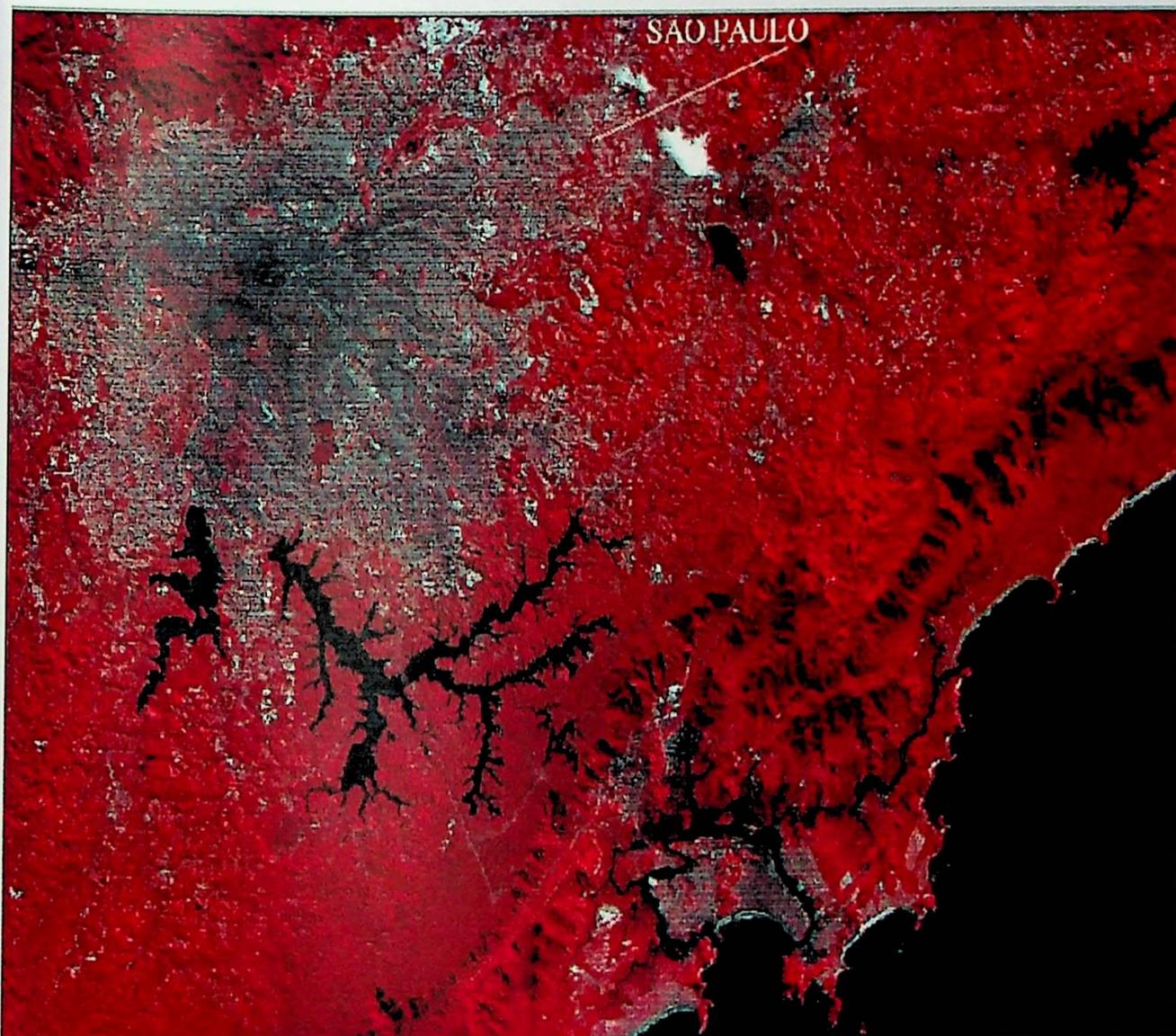


IMAGEM -MSS -LANDSAT2 -ANO: 1977 REGIÃO DE SÃO PAULO

IMAGEM MSS ADQUIRIDA PELO SISTEMA

IMAGEM -MSS ADQUIRIDA PELO SISTEMA DESENVOLVIDO

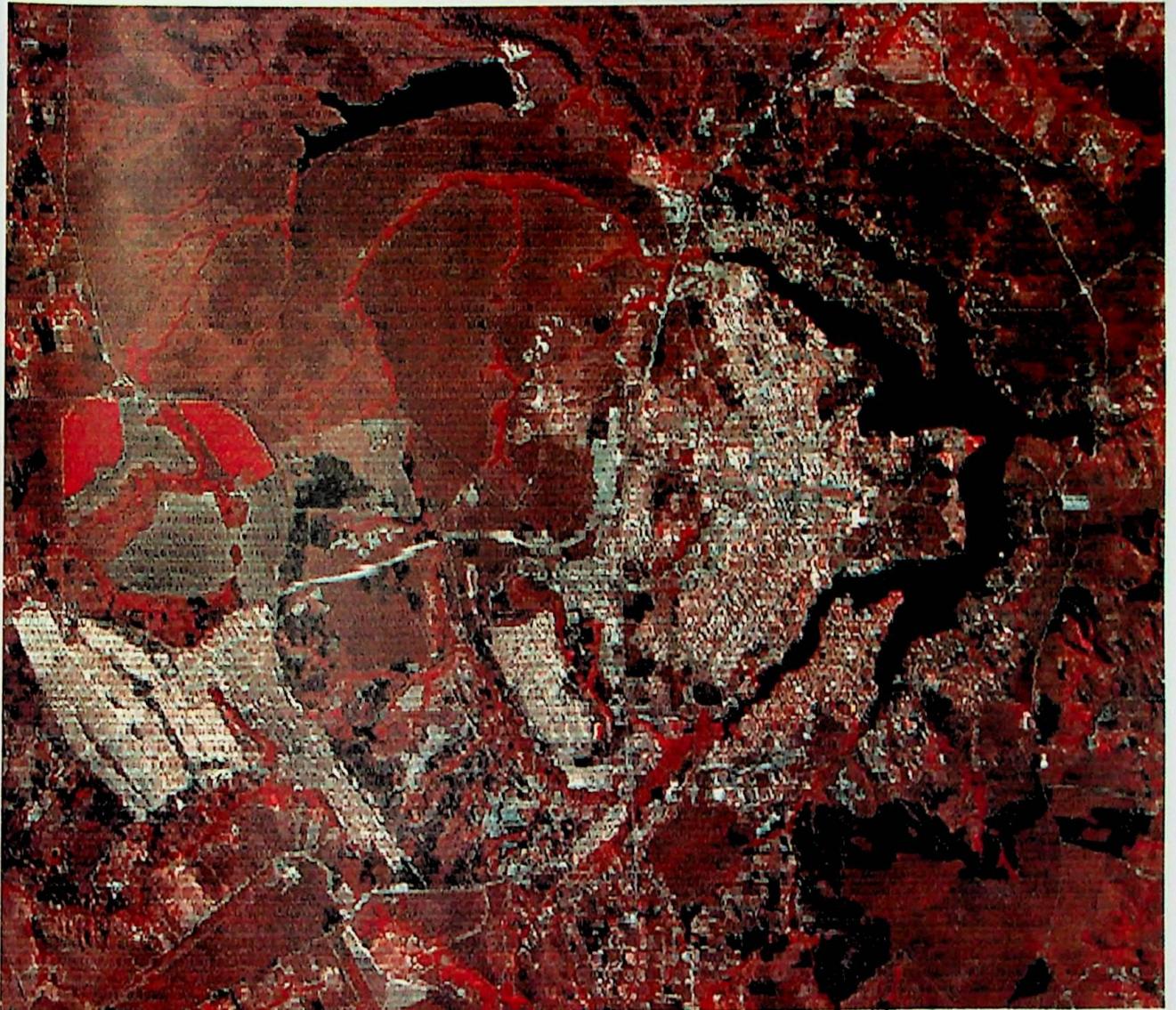


IMAGEM RGB MSS- LANDSAT2 ANO:1977 - CIDADE DE BRASÍLIA - DF

IMAGEM MSS-RGB COMPOSTA - ANO 1977 - MATO GROSSO

IMAGEM -MSS ADQUIRIDA PELO SISTEMA



IMAGEM MSS-RGB COMPOSTA - ANO 1977 - MATO GROSSO

IMAGEM MSS- ADQUIRIDA PELO SISTEMA



IMAGEM - MSS -LANDSAT 2 - ILHA DE MARAJÓ- PA

IMAGEM MSS- ADQUIRIDA PELO HARDWARE E SOFTWARE DESENVOLVIDO

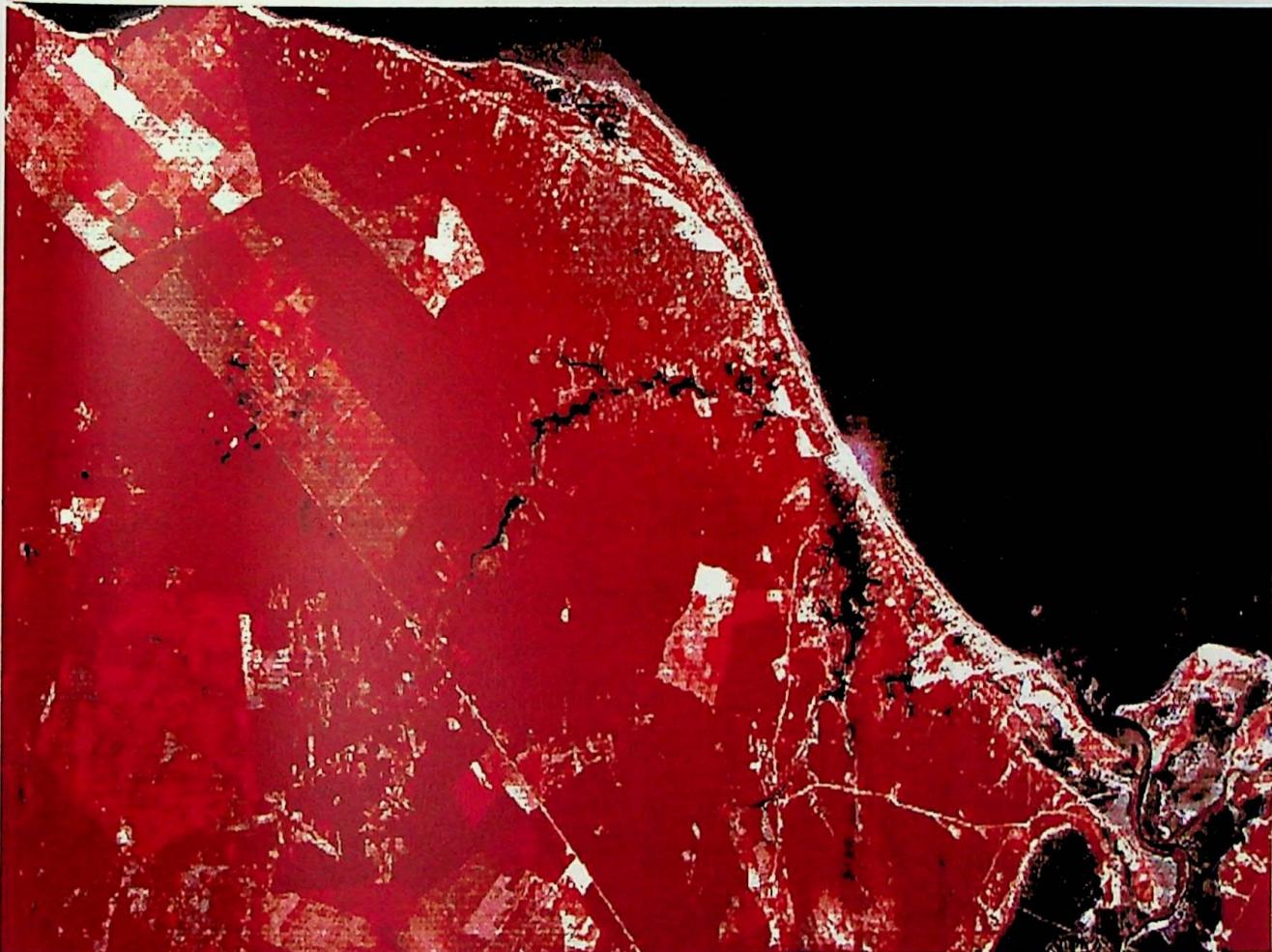


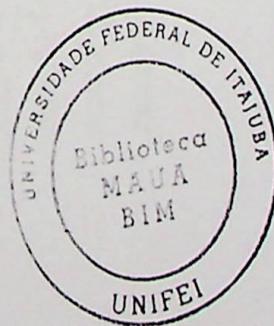
IMAGEM MSS-RGB COMPOSTA -BANDAS 4,6 E 7 - ANO 1977 - RIO GRANDE DO NORTE

AEROPORTO

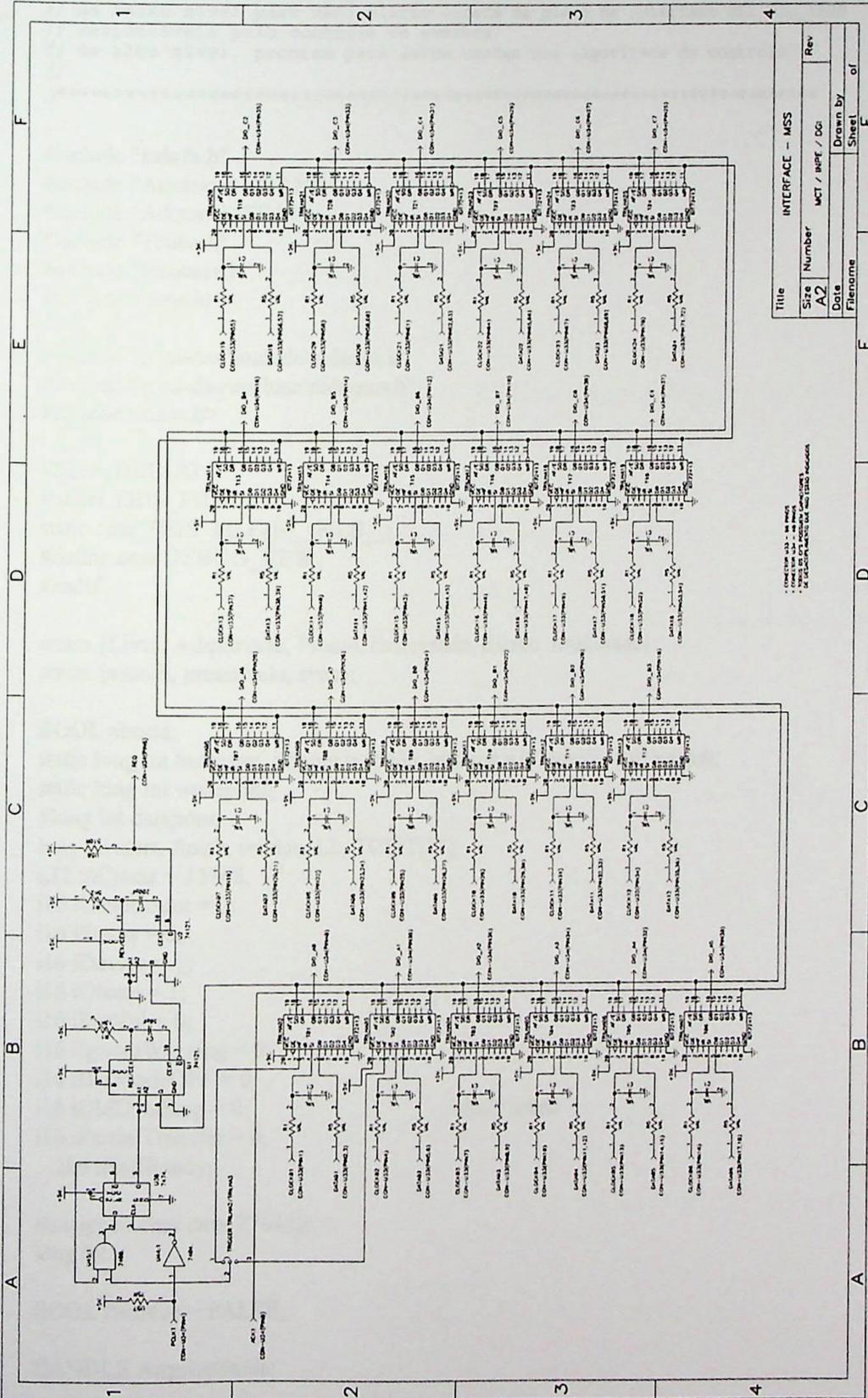
ENCONTRO DAS AGUAS

MANAUS  
1977

## Apêndices



APÊNDICE A – Esquema Elétrico – Interface IFDM – (Interface Formatadora dos Dados MSS)



## APÊNDICE B – Listagem do Software de Aquisição dos Dados MSS , gerados pelo Gravador – Visual C++

```

//*****
// Neste arquivo contém implementações dos métodos das classes:
// de baixo nível para manipulação direta da placa de interface PCI_DIO_32HS
// responsáveis pelo controle de eventos
// de alto nível, prontas para serem usadas nos algoritmos de controle
//
//*****

#include "stdafx.h"
#include "AquisicaoMSS.h"
#include "AcquireMSS.h"
#include "Trilha.h"
#include "resource.h"
#include <time.h>

#include "c:\ni-daq\include\nidaqex.h"
#include "c:\ni-daq\include\nidaqcns.h"
#include <time.h>

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

enum {Livre, Adquirindo, Pronto, Escrevendo, Escrito, Analisando};
enum {search, preambulo, sync};

BOOL aborta;
static long int bufcount, bitpont = 65536, bitcount = 0, bitinicio = 0, bitmask;
static long int writepont;
//long int datapont;
long int start, finish, iniciou, LL, TCBIT[24];
u32 ulCount = 32768;
i16 iLoopCount = 0;
i16 iStatus = 0;
i16 iDevice = 1;
i16 iGroup = 1;
i16 iRetVal = 0;
i16 iIgnoreWarning = 0;
i16 iDBModeOFF = 0;
i16 iOldDataStop = 0; //desabilitado
i16 iPartialTransfer = 0;
i16 iHalfReady;

//unsigned char data[27648];
long int i;

BOOL Escreve = FALSE;

HANDLE ArquivoSaida;

```

```

struct Escrita WriteThread[30];

i16 iHalfBufsToRead = 400; // Aprox. 80 segundos de dados

CEvent go, processa;
CAquisicaoMSSDIg* m_Principal;

UINT AdquireDados (LPVOID pParam)
{
    struct Escrita *buf = (struct Escrita *) pParam;
    iLoopCount = 0;
    while (!aborta)
    {
        if (!Escreve)
        {
            iHalfReady = 0;
            while (iHalfReady != 1)
            while (buf[writepont].bufstatus != Livre)
            {
                iStatus = DIG_DB_HalfReady(iDevice, iGroup, &iHalfReady);
                go.SetEvent();
                processa.SetEvent();
                Sleep(5);
            }
            buf[writepont].bufstatus = Adquirindo;

            // iStatus = DIG_DB_Transfer(iDevice, iGroup,
            // (i16*)WriteThread[writepont].bufpont, WriteThread[writepont].bufsize/sizeof(long int));

            // if ((iRetVal = NIDAQErrorHandler(iStatus, "DIG_Block_In",
            // iIgnoreWarning)) == IDNO) return 0;

            m_Principal->totalgeral += (ulCount/2);
            bitmask = 128;
            for (int i = 0; i < (long int)ulCount/2; i++)
            {
                (buf[writepont].bufpont)[i] = data[datapont]&bitmask? -1:0;
                bitmask /= 2;
                if (bitmask == 0)
                {
                    bitmask = 128;
                    datapont++;
                    if (datapont >= 27648) datapont = 0;
                }
            }
            /*
            }
            }
            else
            {
                while (buf[writepont].bufstatus != Livre)
                {
                    go.SetEvent();
                    Sleep(5);
                }
                buf[writepont].bufstatus = Adquirindo;
            }
        }
    }
}

```

```

        unsigned long int BytesWritten;
        ReadFile (buf[writepont].Arquivo, buf[writepont].bufpont,
buf[writepont].bufsize, &BytesWritten, NULL);
    }

    buf[writepont++].bufstatus = Pronto;
    go.SetEvent();
    Sleep(0);
    processa.SetEvent();

    if (writepont >= 30) writepont = 0;
//    if (iLoopCount%36 == 0)
//    {
        start = finish;
        finish = GetTickCount();
        m_Principal->total_time = max(finish - iniciou, 1);
        m_Principal->ratemedio = (double)m_Principal->totalgeral/(finish - start);
        m_Principal->totalgeral = 0;
        m_Principal->linelength = (LL - 2)/25 - 2;
        m_Principal->BCD[0] = ((TCBIT[22]*2 + TCBIT[20])*2 + TCBIT[18])*2 +
TCBIT[16];
        m_Principal->BCD[1] = ((TCBIT[14]*2 + TCBIT[12])*2 + TCBIT[23])*2 +
TCBIT[21];
        m_Principal->BCD[2] = ((TCBIT[19]*2 + TCBIT[17])*2 + TCBIT[15])*2 +
TCBIT[13];
        m_Principal->BCD[3] = ((TCBIT[10]*2 + TCBIT[8])*2 + TCBIT[6])*2 +
TCBIT[4];
        m_Principal->BCD[4] = ((TCBIT[2]*2 + TCBIT[0])*2 + TCBIT[11])*2 +
TCBIT[9];
        m_Principal->BCD[5] = ((TCBIT[7]*2 + TCBIT[5])*2 + TCBIT[3])*2 +
TCBIT[1];

        m_Principal->formato += 1;
        if (m_Principal->BCD[2] == 15) m_Principal->formato = 1;
        switch (m_Principal->formato)
        {
            case 1:
                m_Principal->segundo[0] = m_Principal->BCD[4];
                m_Principal->segundo[1] = m_Principal->BCD[5];
                m_Principal->minuto[1] = m_Principal->BCD[3];
                break;
            case 2:
                m_Principal->dia[0] = m_Principal->BCD[3];
                m_Principal->dia[1] = m_Principal->BCD[4];
                m_Principal->dia[2] = m_Principal->BCD[5];
                m_Principal->hora[0] = m_Principal->BCD[0];
                m_Principal->hora[1] = m_Principal->BCD[1];
                m_Principal->minuto[0] = m_Principal->BCD[2];
                break;
            case 3:
                m_Principal->segundo[0] = m_Principal->BCD[3];
                m_Principal->segundo[1] = m_Principal->BCD[4];
                m_Principal->minuto[1] = m_Principal->BCD[1];
                break;
        }
    }
}

```

```

        case 4:
            m_Principal->dia[0] = m_Principal->BCD[2];
            m_Principal->dia[1] = m_Principal->BCD[3];
            m_Principal->dia[2] = m_Principal->BCD[4];
            m_Principal->hora[0] = m_Principal->BCD[5];
            m_Principal->hora[1] = m_Principal->BCD[0];
            m_Principal->minuto[0] = m_Principal->BCD[1];
            break;
        }
        m_Principal->BufferReady.SetEvent();

//
    }
    ++iLoopCount;
    if (iLoopCount >= iHalfBufsToRead) aborta = TRUE;
}
    AfxMessageBox ("Aquisicao Terminada");

/* Clear the block operation. */
    if (!Escreve)
    {
// iStatus = DIG_Block_Clear(iDevice, iGroup);

/* Clear DB mode for the device. */

// iStatus = DIG_DB_Config(iDevice, iGroup, iDBModeOFF, iOldDataStop,
// iPartialTransfer);

/* Unconfigure group. */

// iStatus = DIG_Grp_Config(iDevice, iGroup, 0, 0, 0);
    }
    return 1;
}

UINT AnalisaDados (LPVOID pParam)
{
    int ipont = 0;
    struct Escrita *buf = (struct Escrita *) pParam;
    Trilha* trilhas[24];
    int track;
    for (track=0; track<24; track++)
        trilhas[track] = new Trilha(track);

    while (!aborta)
    {
        processa.Lock();
        while ((buf[ipont].bufstatus == Escrito))
        {
            buf[ipont].bufstatus = Analisando;
            for (track=0; track<24; track++)
            {
                m_Principal->TrackSync[track] = trilhas[track]->Analisa(buf[ipont]);
                TCBIT[track] = (trilhas[track]->TimeCode() == 63)? 1:0;
            }
        }
    }
}

```

```

        LL = trilhas[0]->LineLength();
        buf[ipont++].bufstatus = Livre;
        if (ipont>=30) ipont = 0;
    }
}
for (track=0; track<24; track++)
    delete trilhas[track];
return 1;
}

UINT EscreveDados (LPVOID pParam)
{
    struct Escrita *buf = (struct Escrita *) pParam;
    int pont = 0;
    Trilha* trilhas[24];
    int track;
    for (track=0; track<24; track++)
        trilhas[track] = new Trilha(track);
    unsigned long int BytesWritten;
    while (!aborta)
    {
        go.Lock();
        while (buf[pont].bufstatus == Pronto)
        {
            buf[pont].bufstatus = Escrevendo;
            if (Escreve)WriteFile (buf[pont].Arquivo, buf[pont].bufpont, buf[pont].bufsize,
// &BytesWritten, NULL);
// buf[pont].bufstatus = Escrito;
            processa.SetEvent();
            Sleep(0);
            for (track=0; track<24; track++)
            {
                m_Principal->TrackSync[track] = trilhas[track]->Analisa(buf[pont]);
                TCBIT[track] = (trilhas[track]->TimeCode() == 63)? 1:0;
            }
            LL = trilhas[0]->LineLength();
            buf[pont++].bufstatus = Livre;
            if (pont>=30) pont = 0;
// pont++;
// if (pont > 29) pont = 0;
        }
    }
    for (track=0; track<24; track++)
        delete trilhas[track];
    return 1;
}

UINT RecebeDados (LPVOID pParam)
{
    AdquireMSS Aquisicao;
    Aquisicao.Processa((CAquisicaoMSSDlg*) pParam);
    return 1;
}

```

```

//*****
// Parametização da Placa PCI-DIO para trabalhar com o Protocolo BURST
//*****
AcquireMSS::AcquireMSS()
{
    i32 lTimeout = 180;           // 10 segundos
    i16 iGroupSize = 4;         //32 bits
    i16 iPort = 0;
    i16 iDir = 0;               //input
    i16 iSignal = 3;           //burst mode
    i16 iEdge = 0;             // not used for burst mode
    i16 iReqPol = 0;
    i16 iAckPol = 0;
    i16 iAckDelayTime = 7;     //PCLK period of 700 ns (Aprox. 1.4 MHz)
    i16 iDBModeON = 1;

/*
    CFile ArquivoDados;
    ArquivoDados.Open ("c:\\mss\\msspci.dat", CFile::modeRead );
    ArquivoDados.Seek (10300, CFile::begin);
    ArquivoDados.Read (data, sizeof(data));
    ArquivoDados.Close();
    datapont = 0;*/

    /* This sets a timeout limit (#Sec * 18ticks/Sec) so that if there
    is something wrong, the program won't hang on the DIG_DB_Transfer
    call. */

// iStatus = Timeout_Config(iDevice, lTimeout);

// if ((iRetVal = NIDAQErrorHandler(iStatus, "Timeout_Config",
// iIgnoreWarning)) == IDNO) return;

    /* Configure group of ports as input, with handshaking. */

// iStatus = DIG_Grp_Config(iDevice, iGroup, iGroupSize, iPort,
// iDir);

// if ((iRetVal = NIDAQErrorHandler(iStatus, "DIG_Grp_Config",
// iIgnoreWarning)) == IDNO) return;

    /* Configure handshaking parameters for burst,
    active high request and 1.4 MHZ clock PCLK */

// iStatus = DIG_Grp_Mode(iDevice, iGroup, iSignal, iEdge, iReqPol,
// iAckPol, iAckDelayTime);

// if ((iRetVal = NIDAQErrorHandler(iStatus, "DIG_Block_PG_Config",
// iIgnoreWarning)) == IDNO) return;

    /* Turn ON software double-buffered mode, with half-buffer
    overwrite protection (iOldDataStop). */

```

```

// iStatus = DIG_DB_Config(iDevice, iGroup, iDBModeON, iOldDataStop,
// iPartialTransfer);

// if ((iRetVal = NIDAQErrorHandler(iStatus, "DIG_DB_Config",
// iIgnoreWarning)) == IDNO) return;

}

AdquireMSS::~AdquireMSS()
{

}

void AdquireMSS::Processa(CAquisicaoMSSDlg* Principal)
{
/*
* Local Variable Declarations:
*/

    m_Principal = Principal;

    static i16 piBuffer[65536] = {0};
// i16 iHalfReady;
u32 ulPtsTfr = 32768;

    CFileDialog PerguntaArquivo (TRUE, "dat", "Msspci.dat", OFN_HIDEREADONLY |
OFN_OVERWRITEPROMPT,
        "Arquivos de dados (*.dat)*.dat|Todos os Arquivos (*.*)|*.*||");
    if (PerguntaArquivo.DoModal() == IDOK)
    {
        ArquivoSaida = CreateFile (PerguntaArquivo.GetPathName( ),
            GENERIC_READ, 0, NULL, OPEN_EXISTING ,
FILE_FLAG_NO_BUFFERING, NULL);
        if (ArquivoSaida == INVALID_HANDLE_VALUE)
        {
            AfxMessageBox ("Erro na abertura");
            return;
        }
        Escreve = TRUE;
    }
    for (int i=0; i<30; i++)
    {
        WriteThread[i].bufstatus = Livre;
        WriteThread[i].Arquivo = ArquivoSaida;
        WriteThread[i].bufpont = (long int *)VirtualAlloc (NULL,
            ulCount*2, MEM_COMMIT, PAGE_READWRITE);
        if (WriteThread[i].bufpont == NULL)
        {
            AfxMessageBox ("Erro na alocação dos buffers");
            return;
        }
    }
}

```

```

        WriteThread[i].bufsize = ulCount*2;
    }

    /* Start the double-buffered pattern generation input. The
    'piBuffer' here is the circular acquisition buffer. NOTE: any
    warnings are ignored. */
    AfxMessageBox( "Clique OK quando estiver pronto");

    // iStatus = DIG_Block_In(iDevice, iGroup, piBuffer, ulCount);

    // if ((iRetVal = NIDAQErrorHandler(iStatus, "DIG_Block_In",
    // iIgnoreWarning)) == IDNO) return;

    iStatus = 0;

    CWinThread* T2 = AfxBeginThread (AdquireDados, WriteThread,
    THREAD_PRIORITY_HIGHEST);
    CWinThread* T0 = AfxBeginThread (EscreveDados, WriteThread,
    THREAD_PRIORITY_HIGHEST);
    // CWinThread* T1 = AfxBeginThread (AnalisaDados, WriteThread,
    THREAD_PRIORITY_HIGHEST);

    aborta = FALSE;
    go.ResetEvent();
    finish = GetTickCount();
    iniciou = finish;
    start = finish;
    bufcount = 0;
    writepont = 0;

    return;
}

void TerminaAquisicao()
{
    if (iStatus == 0) {

        AfxMessageBox(" Aquisicao de dados MSS terminada com sucesso");

    }

    else

        AfxMessageBox(" Aquisicao interrompida em virtude de erro");

    int tries;
    CloseHandle (ArquivoSaida);
    for (i=0; i<30; i++)
    {
    // free(WriteThread[i].bufpont);
        tries = 0;
        while (WriteThread[i].bufstatus != Livre &&
            tries < 100)
        {
            go.SetEvent();
            processa.SetEvent();
        }
    }
}

```

```
                Sleep(5);
                tries ++;
            }

            VirtualFree (WriteThread[i].bufpont, 0, MEM_RELEASE);
        }

        /* Disable timeouts. */
        /** iStatus = Timeout_Config(iDevice, -1);
            aborta = TRUE;
        */
    }
}
```

```
// Implementação da classe ou subrotina de Trilha
// Tem a função de localizar a palavra de sincronismo e separar aos dados de
// video para cada Sensor
//*****
//*****
// Trilha.cpp: implementation of the Trilha class.
//
```

```

////////////////////////////////////

#include "stdafx.h"
#include "AquisicaoMSS.h"
#include "Trilha.h"

#ifdef _DEBUG
#undef THIS_FILE
static char THIS_FILE[]=__FILE__;
#define new DEBUG_NEW
#endif

////////////////////////////////////
// Construction/Destruction
////////////////////////////////////

Trilha::Trilha(int track)
{
    numerotrilha = track;
    bitpont = 0;
    bitcount = 0;
    bitpos = 0;
    bitinicio = 0;
    numerosweep = 0;
    bitmask = 1<<track;
    estado = search;
    data = 0;
    pont = NULL;
    size = 0;
    insync = FALSE;
    EndOfBuffer = FALSE;
    CString NomeArquivo;
    NomeArquivo.Format ("Video%2.2i.dat",track+2);
    ArquivoTrilha = CreateFile (NomeArquivo,
        GENERIC_WRITE, 0, NULL, CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL,
NULL);
// NomeArquivo.Format ("Report%2.2i.txt", track+2);
// ReportTrilha = CreateFile (NomeArquivo,
// GENERIC_WRITE, 0, NULL, CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL,
NULL);
}

Trilha::~Trilha()
{
    CloseHandle (ArquivoTrilha);
    CloseHandle (ReportTrilha);
}

int Trilha::getbits (int numero)
{
    if (bitpos == 0) data = 0;
    while (bitpos < numero)
    {
        if (bitpont >= (long int)(size/sizeof(long int)))

```

```

        {
            bitpont -= size/sizeof(long int);
            EndOfBuffer = TRUE;
            return FALSE;
        }
        data = data << 1;
        data |= (pont[bitpont++]&bitmask)? 1:0;
        bitpos++;
    }
    bitcount += numero;
    bitpos = 0;
    return TRUE;
}

int Trilha::getbits (int numero, long int &result)
{
    if (!getbits (numero)) return FALSE;
    result = data;
    return TRUE;
}

int Trilha::getbitsinv (int numero, long int &result)
{
    if (!getbits (numero)) return FALSE;
    long int temp = data;
    result = data & 1;
    for (int ibits = 1; ibits<numero; ibits++)
    {
        temp = temp >> 1;
        result = (result<<1) | (temp & 1);
    }
    return TRUE;
}

int Trilha::pulabits (int numero)
{
    bitpont += numero;
    bitcount += numero;
    return TRUE;
}

BOOL Trilha::Analisa (struct Escrita pParam)
{
    pont = pParam.bufpont;
    size = pParam.bufsize;
    /*
    int bufpontin = 0;
    for (int bufpontout=0; bufpontout<2048; bufpontout++)
    {
        databuf[bufpontout] = ((pont[bufpontin++]&bitmask)>>numerotrilha);
        for (int bit=1; bit<8; bit++)
        {
            databuf[bufpontout] = (databuf[bufpontout]
<<1)|((pont[bufpontin++]&bitmask)>>numerotrilha);
        }
    }
    */
}

```

```
// WriteFile (ArquivoTrilha, databuf, sizeof (databuf), &BytesWritten, NULL);

EndOfBuffer = FALSE;
while (!EndOfBuffer)
{
    switch (estado)
    {
        case (search):
            if (!getbits (4))
                {
                    estado=search;
                    break;
                }

            if (data == 0x0A )
                {
                    estado=preambuloAA;
                    break;
                    insync = TRUE;
                }

            else

                if (data == 0x05)
                    {
                        insync = FALSE;
                        estado=preambulo55;
                        break;
                    }
                else
                    {
                        estado=search;
                        break;
                    }

        case (preambuloAA):
            if (!getbits (2)) break;

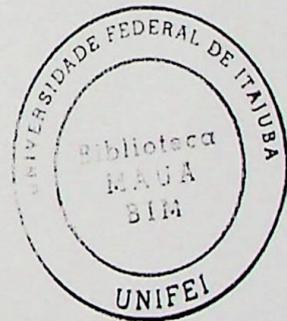
            if (data==0x02)
                {
                    bitspreambulo+=6;
                    estado = search;
                    break;
                }

            else
                {
                    if (data==0x03)
```

```
    {  
        if (!getbits (4))break;  
  
        if (data==0x0A)  
        {  
            estado=sync;  
            break;  
        }  
        else  
        {  
            estado=search;  
            break;  
        }  
    }  
    else  
    {  
        estado=search;  
        break;  
    }  
}
```

case (preambulo55):

```
    if (!getbits (2)) break;  
  
    if (data==0x01)  
    {  
        bitspreambulo+=6;  
        estado = search;  
        break;  
    }  
    else  
    {  
        if (data==0x03)  
        {  
            if (!getbits (4)) break;  
  
            if (data==0x0A)  
            {  
                estado=sync;  
                break;  
            }  
            else  
            {  
                estado=search;  
                break;  
            }  
        }  
    }  
}
```



```

    }
    }
    else
    {
        estado=search;
        break;
    }
}

case sync:
    estado =TC;
//    insync = TRUE;
    break;

case TC:
    if (!getbits(12, timecode))
    {
        estado = TC;
        break;
    }
    estado=PIXELS;
    ipix = 0;
case PIXELS:
    pulabits(1);
    for (i=ipix;i<3600;i++)
    {
        if (!getbits (6))
        {
            estado = PIXELS;
            ipix = i;
            break;
        }

        databuf[i] = (data&0xFF);
    }
    if (i<3600) break;
    WriteFile (ArquivoTrilha, databuf, sizeof (databuf), &BytesWritten, NULL);
    pulabits (171*6);
    estado=search;

case LL:
    if (!getbitsinv(18, linelength))
    {
        estado = LL;
        break;
    }
//    sprintf(LinhaReport, "\n%5i,%10i,%7i,%7i,%10i,%10i", numerosweep++,
bitinicio, bitspreambulo, linelength);
//    WriteFile(ReportTrilha, LinhaReport, 55, &BytesWritten, NULL);
//    pulabits (513*6);
    estado = search;

```

```

    }
}
return insync;
}

```

```

//*****
// Implementação da classe ou subrotina de Mensagem
// Tem a função de implementação da Tela de Aquisição, com suas
// respectivas informações a ser amostradas na tela.
//*****

// AquisicaoMSSDlg.cpp : implementation file
//

#include "stdafx.h"
#include "AquisicaoMSS.h"
#include "AquisicaoMSSDlg.h"
#include "AdquireMSS.h"

#ifdef _DEBUG

```

```

#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

////////////////////////////////////
// CAboutDlg dialog used for App About
UINT MostraProgresso (LPVOID pParam)
{
    CAquisicaoMSSDlg* Principal = (CAquisicaoMSSDlg*) pParam;
    int saveLL;
    char strrate [10];
    int itrack;
    long int inicio, ratecount;
    int segundo;
    BOOL SaveSync[24];
    CWnd* JanelaTaxa = Principal->GetDlgItem(IDC_TAXA);
    CWnd* JanelaDia = Principal->GetDlgItem(IDC_TCDIA);
    CWnd* JanelaHora = Principal->GetDlgItem(IDC_TCHORA);
    CWnd* JanelaMinuto = Principal->GetDlgItem(IDC_TCMINUTO);
    CWnd* JanelaSegundo = Principal->GetDlgItem(IDC_TCSEGUNDO);
    CWnd* JanelaLineLength = Principal->GetDlgItem(IDC_LINELENGTH);
    CStatic *Sync[24];
    Sync[0] = (CStatic*)Principal->GetDlgItem(IDC_Sync3A); //Trilha 2
    Sync[1] = (CStatic*)Principal->GetDlgItem(IDC_Sync1A); //Trilha 3
    Sync[2] = (CStatic*)Principal->GetDlgItem(IDC_Sync3B); //Trilha 4
    Sync[3] = (CStatic*)Principal->GetDlgItem(IDC_Sync1B); //Trilha 5
    Sync[4] = (CStatic*)Principal->GetDlgItem(IDC_Sync3C); //Trilha 6
    Sync[5] = (CStatic*)Principal->GetDlgItem(IDC_Sync1C); //Trilha 7
    Sync[6] = (CStatic*)Principal->GetDlgItem(IDC_Sync3D); //Trilha 8
    Sync[7] = (CStatic*)Principal->GetDlgItem(IDC_Sync1D); //Trilha 9
    Sync[8] = (CStatic*)Principal->GetDlgItem(IDC_Sync3E); //Trilha 10
    Sync[9] = (CStatic*)Principal->GetDlgItem(IDC_Sync1E); //Trilha 11
    Sync[10] = (CStatic*)Principal->GetDlgItem(IDC_Sync3F); //Trilha 12
    Sync[11] = (CStatic*)Principal->GetDlgItem(IDC_Sync1F); //Trilha 13
    Sync[12] = (CStatic*)Principal->GetDlgItem(IDC_Sync4A); //Trilha 14
    Sync[13] = (CStatic*)Principal->GetDlgItem(IDC_Sync2A); //Trilha 15
    Sync[14] = (CStatic*)Principal->GetDlgItem(IDC_Sync4B); //Trilha 16
    Sync[15] = (CStatic*)Principal->GetDlgItem(IDC_Sync2B); //Trilha 17
    Sync[16] = (CStatic*)Principal->GetDlgItem(IDC_Sync4C); //Trilha 18
    Sync[17] = (CStatic*)Principal->GetDlgItem(IDC_Sync2C); //Trilha 19
    Sync[18] = (CStatic*)Principal->GetDlgItem(IDC_Sync4D); //Trilha 20
    Sync[19] = (CStatic*)Principal->GetDlgItem(IDC_Sync2D); //Trilha 21
    Sync[20] = (CStatic*)Principal->GetDlgItem(IDC_Sync4E); //Trilha 22
    Sync[21] = (CStatic*)Principal->GetDlgItem(IDC_Sync2E); //Trilha 23
    Sync[22] = (CStatic*)Principal->GetDlgItem(IDC_Sync4F); //Trilha 24
    Sync[23] = (CStatic*)Principal->GetDlgItem(IDC_Sync2F); //Trilha 25
    CBitmap InSync, OutSync;
    InSync.LoadBitmap(IDB_BITMAP2);
    OutSync.LoadBitmap(IDB_BITMAP1);
    for (itrack=0; itrack<24; itrack++)
    {
        Principal->TrackSync[itrack] = FALSE;
        SaveSync[itrack] = FALSE;
    }
}

```

```

        Sync[itrack]->SetBitmap (Principal->TrackSync[itrack]? InSync:OutSync);
    }
    char str[10];

    if (Principal->aborta) return 0;

    JanelaDia->SetWindowText("000");
    JanelaTaxa->SetWindowText("0.0");
    JanelaHora->SetWindowText("00");
    JanelaMinuto->SetWindowText("00");
    JanelaSegundo->SetWindowText("00");
    JanelaLineLength->SetWindowText("0000");
    inicio = 0;
    segundo = 0;
    ratecount = 0;
    Principal->ratemedio = 0.;
    Principal->totalgeral = 0;
    Principal->linelength = 0;
    saveLL = Principal->linelength;

    while (!Principal->aborta)
    {
        Principal->BufferReady.Lock();
        if (inicio == 0)
        {
            inicio = GetTickCount();
        }
        for (itrack=0; itrack<24; itrack++)
        {
            if (Principal->TrackSync[itrack] != SaveSync[itrack])
            {
                SaveSync[itrack] = Principal->TrackSync[itrack];
                Sync[itrack]->SetBitmap (Principal->TrackSync[itrack]?
InSync:OutSync);
            }
        }

        if (Principal->linelength != saveLL)
        {
            sprintf (str, "%8i", Principal->linelength);
            saveLL = Principal->linelength;
            JanelaLineLength->SetWindowText(str);
        }

        Principal->ratemedio = (double)Principal->totalgeral/total_time;
        //
        // sprintf (strrate, "%6.1f", Principal->ratemedio);
        // JanelaTaxa->SetWindowText(strrate);
        /*
        if (segundo != (Principal->total_time/1000))
        {
            segundo = Principal->total_time/1000;
            sprintf (str, "%8i",segundo);
            JanelaSegundo->SetWindowText(str);
        }*/
        if ((Principal->formato==2)||((Principal->formato==4))

```

```

        {
            sprintf (str, "%1i%1i%1i", Principal->dia[0],Principal->dia[1],Principal-
>dia[2]);

            JanelaDia->SetWindowText(str);
            sprintf (str, "%1i%1i", Principal->hora[0],Principal->hora[1]);
            JanelaHora->SetWindowText(str);
            sprintf (str, "%1i%1i", Principal->minuto[0],Principal->minuto[1]);
            JanelaMinuto->SetWindowText(str);
            sprintf (str, "%1i%1i", Principal->segundo[0],Principal->segundo[1]);
            JanelaSegundo->SetWindowText(str);
        }

        sprintf (strrate, "%6.1f", Principal->ratemedio);
        JanelaTaxa->SetWindowText(strrate);

    }

    return 1;
}
class CAboutDlg : public CDialog
{
public:
    CAboutDlg();

// Dialog Data
//{{AFX_DATA(CAboutDlg)
enum { IDD = IDD_ABOUTBOX };
//}}AFX_DATA

// ClassWizard generated virtual function overrides
//{{AFX_VIRTUAL(CAboutDlg)
protected:
virtual void DoDataExchange(CDataExchange* pDX); // DDX/DDV support
//}}AFX_VIRTUAL

// Implementation
protected:
//{{AFX_MSG(CAboutDlg)
//}}AFX_MSG
DECLARE_MESSAGE_MAP()
};

CAboutDlg::CAboutDlg() : CDialog(CAboutDlg::IDD)
{
   //{{AFX_DATA_INIT(CAboutDlg)
   //}}AFX_DATA_INIT
}

void CAboutDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CAboutDlg)
   //}}AFX_DATA_MAP
}

```

```

BEGIN_MESSAGE_MAP(CAboutDlg, CDialog)
   //{{AFX_MSG_MAP(CAboutDlg)
        // No message handlers
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CAquisicaoMSSDlg dialog

CAquisicaoMSSDlg::CAquisicaoMSSDlg(CWnd* pParent /*=NULL*/)
    : CDialog(CAquisicaoMSSDlg::IDD, pParent)
{
   //{{AFX_DATA_INIT(CAquisicaoMSSDlg)
        // NOTE: the ClassWizard will add member initialization here
   //}}AFX_DATA_INIT
    // Note that LoadIcon does not require a subsequent DestroyIcon in Win32
    m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    formato = 0;
    aborta = FALSE;
}

void CAquisicaoMSSDlg::DoDataExchange(CDataExchange* pDX)
{
    CDialog::DoDataExchange(pDX);
   //{{AFX_DATA_MAP(CAquisicaoMSSDlg)
        // NOTE: the ClassWizard will add DDX and DDV calls here
   //}}AFX_DATA_MAP
}

BEGIN_MESSAGE_MAP(CAquisicaoMSSDlg, CDialog)
   //{{AFX_MSG_MAP(CAquisicaoMSSDlg)
        ON_WM_SYSCOMMAND()
        ON_WM_PAINT()
        ON_WM_QUERYDRAGICON()
   //}}AFX_MSG_MAP
END_MESSAGE_MAP()

////////////////////////////////////
// CAquisicaoMSSDlg message handlers

BOOL CAquisicaoMSSDlg::OnInitDialog()
{
    CDialog::OnInitDialog();

    // Add "About..." menu item to system menu.

    // IDM_ABOUTBOX must be in the system command range.
    ASSERT((IDM_ABOUTBOX & 0xFFF0) == IDM_ABOUTBOX);
    ASSERT(IDM_ABOUTBOX < 0xF000);

    CMenu* pSysMenu = GetSystemMenu(FALSE);
    if (pSysMenu != NULL)
    {

```

```

        CString strAboutMenu;
        strAboutMenu.LoadString(IDS_ABOUTBOX);
        if (!strAboutMenu.IsEmpty())
        {
            pSysMenu->AppendMenu(MF_SEPARATOR);
            pSysMenu->AppendMenu(MF_STRING, IDM_ABOUTBOX, strAboutMenu);
        }
    }

    // Set the icon for this dialog. The framework does this automatically
    // when the application's main window is not a dialog
    SetIcon(m_hIcon, TRUE);           // Set big icon
    SetIcon(m_hIcon, FALSE);        // Set small icon

    // TODO: Add extra initialization here

    return TRUE; // return TRUE unless you set the focus to a control
}

void CAquisicaoMSSDlg::OnSysCommand(UINT nID, LPARAM lParam)
{
    if ((nID & 0xFFF0) == IDM_ABOUTBOX)
    {
        CAboutDlg dlgAbout;
        dlgAbout.DoModal();
    }
    else
    {
        CDialog::OnSysCommand(nID, lParam);
    }
}

// If you add a minimize button to your dialog, you will need the code below
// to draw the icon. For MFC applications using the document/view model,
// this is automatically done for you by the framework.

void CAquisicaoMSSDlg::OnPaint()
{
    if (IsIconic())
    {
        CPaintDC dc(this); // device context for painting

        SendMessage(WM_ICONERASEBKGND, (LPARAM) dc.GetSafeHdc(), 0);

        // Center icon in client rectangle
        int cxIcon = GetSystemMetrics(SM_CXICON);
        int cyIcon = GetSystemMetrics(SM_CYICON);
        CRect rect;
        GetClientRect(&rect);
        int x = (rect.Width() - cxIcon + 1) / 2;
        int y = (rect.Height() - cyIcon + 1) / 2;

        // Draw the icon
        dc.DrawIcon(x, y, m_hIcon);
    }
}

```

```
    }
    else
    {
        CDialog::OnPaint();
    }
}

// The system calls this to obtain the cursor to display while the user drags
// the minimized window.
HCURSOR CAquisicaoMSSDlg::OnQueryDragIcon()
{
    return (HCURSOR) m_hIcon;
}

void CAquisicaoMSSDlg::OnOK()
{
    // ::SetPriorityClass(::GetCurrentProcess(), REALTIME_PRIORITY_CLASS);
    BufferReady.ResetEvent();
    // AfxGetApp()->SetThreadPriority (THREAD_PRIORITY_TIME_CRITICAL);
    CWinThread* pShowThread = AfxBeginThread (MostraProgresso, this,
    THREAD_PRIORITY_NORMAL);
    AdquireMSS Aquisicao;
    Aquisicao.Processa(this);
    // CWinThread* pThread = AfxBeginThread (RecebeDados, this,
    THREAD_PRIORITY_TIME_CRITICAL);

    // CDialog::OnOK();
}

void CAquisicaoMSSDlg::OnCancel()
{
    char strrate[10];
    aborta = TRUE;
    BufferReady.SetEvent();
    sprintf (strrate, "%6.1f", ratemedio);
    AfxMessageBox(CString("Taxa media =")+strrate);
    // JanelaTaxaMinima->SetWindowText(strrate);
    CDialog::OnCancel();
}
```

## APÊNDICE C – Listagem Software de Composição das Bandas – Quick Basic

```

; *****.
* Este Programa , escrito em linguagem basic, descreve a formação ou a
* composição da Banda 7, a partir da leitura das linhas de video dos sensores 14,16,18,20,22 e 24
*****
CLS
DIM byte1(1000)
' Define record fields.
TYPE TestRecord

    Namefield AS STRING * 1
    'ScoreField AS SINGLE
END TYPE
' Define a variable of the user type.
DIM Rec AS TestRecord

' G$ = ".DAT"
DIM FileBuffer AS TestRecord
' Calculate number of records in the file.
' Read and print contents of each record.

LOCATE 9, 10: INPUT "ENTRE N. LINHAS="; NR&
LOCATE 10, 10: INPUT "OFFSET      ="; F%

'HT$ = N$ + G$
OPEN "video14.dat" FOR RANDOM AS #1 LEN = LEN(FileBuffer)
OPEN "video16.dat" FOR RANDOM AS #2 LEN = LEN(FileBuffer)
OPEN "video18.dat" FOR RANDOM AS #3 LEN = LEN(FileBuffer)
OPEN "video20.dat" FOR RANDOM AS #4 LEN = LEN(FileBuffer)
OPEN "video22.dat" FOR RANDOM AS #5 LEN = LEN(FileBuffer)
OPEN "video24.dat" FOR RANDOM AS #6 LEN = LEN(FileBuffer)
OPEN "O", 7, "MSSBAND7.DAT"
'MAX = LOF(1) / LEN(FileBuffer)
FOR G& = 1 TO NR&
LOCATE 1, 1: PRINT G&

FOR MS& = 1 TO 3280
Q = Q + 1
GET #1, Q, FileBuffer
byte1 = ASC(FileBuffer.Namefield)
IF G& > F% THEN PRINT #7, CHR$(byte1);
NEXT MS&

FOR MS& = 1 TO 3280
R = R + 1
GET #2, R, FileBuffer
byte1 = ASC(FileBuffer.Namefield)
IF G& > F% THEN PRINT #7, CHR$(byte1);
NEXT MS&

FOR MS& = 1 TO 3280
S = S + 1

```

```
GET #3, S, FileBuffer
byte1 = ASC(FileBuffer.Namefield)
IF G& > F% THEN PRINT #7, CHR$(byte1);
NEXT MS&
```

```
FOR MS& = 1 TO 3280
T = T + 1
GET #4, T, FileBuffer
byte1 = ASC(FileBuffer.Namefield)
IF G& > F% THEN PRINT #7, CHR$(byte1);
NEXT MS&
```

```
FOR MS& = 1 TO 3280
U = U + 1
GET #5, U, FileBuffer
byte1 = ASC(FileBuffer.Namefield)
IF G& > F% THEN PRINT #7, CHR$(byte1);
NEXT MS&
```

```
FOR MS& = 1 TO 3280
V = V + 1
GET #6, V, FileBuffer
byte1 = ASC(FileBuffer.Namefield)
IF G& > F% THEN PRINT #7, CHR$(byte1);
NEXT MS&
NEXT G&
STOP
CLOSE
```

```
END
```



Ministério da Educação  
**UNIVERSIDADE FEDERAL DE ITAJUBÁ**  
Criada pela Lei nº 10.435, de 24 de abril de 2002

## ANEXO II

### FOLHA DE JULGAMENTO DA BANCA EXAMINADORA

Título da Dissertação: "Projeto Sistema de Recuperação de Dados do Sensor MSS –  
Multispectral Scanner para o Satélite LANDSAT 1, 2 e 3"

Autor: André Ottoboni Dias

#### JULGAMENTO

Examinadores	Conceito	Rubrica
1º	A	---
2º	A	OK.
3º	A	---

Resultado Médio: Conceito: A, ou seja: APROVADO

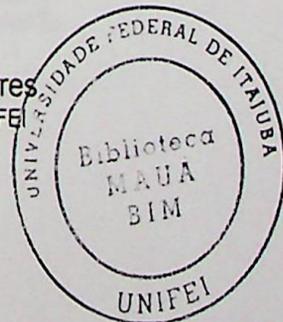
Observações: \_\_\_\_\_

Itajubá, 23 de setembro de 2002.

*Galdenoro Botura Junior*  
Prof. Galdenoro Botura Junior  
1º Examinador - UNESP-FEG

*Newton de Figueiredo Filho*  
Prof. Newton de Figueiredo Filho  
2º Examinador - UNIFEI

*Germano Lambert Torres*  
Prof. Germano Lambert Torres  
3º Examinador - (Orientador) - UNIFEI





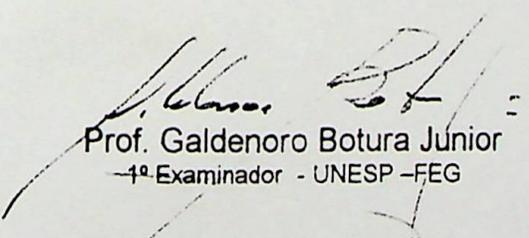
Ministério da Educação  
**UNIVERSIDADE FEDERAL DE ITAJUBÁ**  
Criada pela Lei nº 10.435, de 24 de abril de 2002

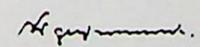
## **A N E X O I**

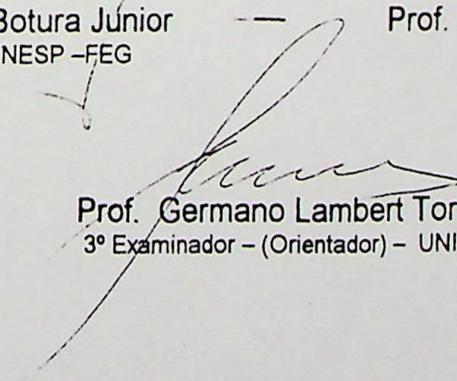
### **PRONUNCIAMENTO DA BANCA EXAMINADORA**

A Banca Examinadora, abaixo assinada, nomeada pela Portaria nº 316 de 23/09/2002, considerando o resultado do Julgamento da Prova de Defesa Pública da Dissertação de Mestrado intitulada: "**Projeto Sistema de Recuperação de Dados do Sensor MSS – Multispectral Scanner para o Satélite LANDSAT 1, 2 e 3**" apresenta pronunciamento no sentido de que o Coordenador dos Cursos de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Itajubá solicite ao DRA (Departamento de Registro Acadêmico) a expedição do título de **Mestre em Ciências em Engenharia Elétrica, na Área de Concentração Automação e Sistemas Elétricos Industriais**, satisfeitas as demais exigências regimentais, a **André Ottoboni Dias**.

Itajubá, 23 de setembro de 2002.

  
Prof. Galdenoro Botura Júnior  
1º Examinador - UNESP - FEG

  
Prof. Newton de Figueiredo Filho  
2º Examinador - UNIFEI

  
Prof. Germano Lambert Torres  
3º Examinador - (Orientador) - UNIFEI