



**AUTARQUIA ASSOCIADA À UNIVERSIDADE
DE SÃO PAULO**

**AQUISIÇÃO DE DADOS EM ESPECTROSCOPIA
GAMA NUCLEAR**

DAVID BARG FILHO

**Dissertação apresentada como parte
dos requisitos para obtenção do Grau
de Mestre em Ciências na Área de
Tecnologia Nuclear - Aplicações.**

**Orientador:
Dr. Paulo R. Pascholati**

**São Paulo
2000**

INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
Autarquia associada à Universidade de São Paulo

AQUISIÇÃO DE DADOS EM
ESPECTROSCOPIA GAMA NUCLEAR

DAVID BÄRG FILHO



Dissertação apresentada como parte dos requisitos para obtenção do grau de Mestre em Ciências na Área de Tecnologia Nuclear.

Orientador:
Dr. Paulo R. Pascholati

SÃO PAULO

2000

Desculpem-me aqueles que não foram citados, mas vocês também foram muito importantes. Agora, há algumas pessoas que merecem agradecimentos específicos, tais como:

O Prof. Paulo R. Pascholati e família que me acolheram em sua casa dando-me condições de finalizar o presente trabalho.

O Prof. Vito R. Vanin por ter tanta paciência como meu chefe.

A “gangue” da aquisição formada pelo Renato Neves, o Vito Vanin e eu. Os Professores Paulo R. Pascholati, Vito R. Vanin e Joseph Max Cohenca por terem sido como meus “pais” nestes anos todos, através de seus conselhos, amizade, repreensões e preocupação para comigo.

O Rodrigo Moraes de Oliveira e o Paulo B. Rios pela amizade de longas conversas, pelo trabalho duro e pela companhia.

O Ruy M. de Castro e o Zwinglio O. Guimarães Filho que infelizmente tiveram de suportar a minha ira num momento delicado de suas vidas.

Desculpem-me.

A Ana Cecília que num momento delicado de minha vida, ofertou-me amparo.

A Rose Braga que mostrou-me uma luz no fim do túnel.

A Cláudia B. Schwantes pelo seu imprescindível apoio profissional.

A Denise Bender que embora tenha entrado a pouco em minha vida, tem me inspirado e dado força, vontade e um objetivo.

Agradeço ao CNPq e FAPESP pelo apoio material e ao IPEN e CNEN pela possibilidade de desenvolver este projeto.

Enfim, a todos que durante este trabalho me auxiliaram, tiraram dúvidas, resolveram problemas, apoiaram e principalmente estiveram ao meu lado quando passei pelo momento mais negro de minha vida, que infelizmente coincidiu com este período, eu quero agradecer e dizer que estão no meu coração.

Obrigado!

AQUISIÇÃO DE DADOS EM ESPECTROSCOPIA GAMA NUCLEAR

David Bãrg Filho

RESUMO

Apresenta-se o desenvolvimento de um sistema dedicado à aquisição de dados em espectroscopia gama nuclear compreendendo *hardware* e *software*. Este sistema utiliza o padrão CAMAC, microcomputadores IBM-PC, o sistema operacional de tempo real QNX, tipo Unix, e o padrão gráfico OSF/Motif na construção da interface com o usuário. Para a montagem deste sistema foi necessário a construção de um controlador de bastidor CAMAC no padrão ISA e um controlador para medidas multiparamétricas no padrão CAMAC bem como o desenvolvimento dos programas de computador necessários. Este sistema se encontra no momento pronto para utilização.

DATA AQUISITION IN NUCLEAR NUCLEAR GAMMA-RAY SPECTROSCOPY

David Bãrg Filho

ABSTRACT

The development of both the hardware and software of a dedicated nuclear gamma-ray spectroscopy acquisition data system is presented. It is based on the CAMAC-interface standard, IBM-PC computers, the Unix-like real-time QNX operating system, and the standard graphical user interface OSF/Motif. This development required the construction of a CAMAC crate controller with ISA interface, a CAMAC multiparameter adapter, and many computer programs. This system is ready.

SUMÁRIO

1 INTRODUÇÃO	2
2 AQUISIÇÃO DE DADOS NO L.A.L.....	5
2.1 AQUISIÇÃO DE DADOS COM O PDP-11	5
2.2 POR QUE SUBSTITUIR O SISTEMA DE AQUISIÇÃO DE DADOS?	7
2.3 ALTERNATIVAS AO SISTEMA DE AQUISIÇÃO DE DADOS LAL	8
2.4 CONTROLADOR DE BASTIDOR CAMAC.....	12
2.5 BARRAMENTOS DO MICROCOMPUTADOR PADRÃO IBM-PC	13
2.6 DATAWAY, O BARRAMENTO DO CAMAC.....	15
2.7 QUAL DOS BARRAMENTOS DEVEMOS ESTENDER?	17
2.8 DOMINANDO O BARRAMENTO ISA	18
2.9 ESPECIFICAÇÃO DO CONTROLADOR CAMAC-ISA	20
2.10 CONSTRUÇÃO DO CONTROLADOR CAMAC-ISA.....	22
2.11 ARQUITETURA DO NOVO SISTEMA DE AQUISIÇÃO.....	26
3 AQUISIÇÃO MULTIPARAMÉTRICA	29
3.1 O QUE É UMA AQUISIÇÃO MULTIPARAMÉTRICA.....	29
3.2 O MÓDULO MULTI.....	30
3.3 DESCRIÇÃO DO PROTÓTIPO.....	31
3.4 DESCRIÇÃO DO FUNCIONAMENTO DO PROTÓTIPO.....	33
3.5 TESTES E MEDIDA PRELIMINAR	37
3.6 UM DEFEITO SUTIL.....	43
3.7 ESTUDOS PARA A CONSTRUÇÃO DO MÓDULO DEFINITIVO.....	46
3.7.1 <i>Não-linearidade no sinal de tempo</i>	46
3.7.2 <i>Outras modificações necessárias</i>	52
3.8 CONSTRUÇÃO DO MULTI DEFINITIVO.....	54
3.9 TESTES PRELIMINARES	58
3.10 MEDIDAS COM FONTES RADIOATIVAS.....	60
3.11 MONTAGEM DO SISTEMA MULTIDETECTOR	65
4 PROGRAMAS DE AQUISIÇÃO DE DADOS	67
4.1 INTRODUÇÃO.....	67
4.2 REQUISITOS DO NOVO SISTEMA DE AQUISIÇÃO DE DADOS.....	68
4.3 DESCRIÇÃO GERAL DO SISTEMA DE AQUISIÇÃO UNIPARAMÉTRICO.....	70
4.4 A COMUNICAÇÃO ENTRE OS PROGRAMAS.....	74
4.5 AS INTERFACES DO SISTEMA DE AQUISIÇÃO.....	77
4.5.1 <i>A janela de controle de aquisição</i>	81
4.5.2 <i>As janelas de controle de visualização</i>	83
4.6 O PROGRAMA GERENTE.....	88
4.7 OS PROGRAMAS DE AQUISIÇÃO.....	90
4.7.1 <i>Similaridades entre os programas de aquisição de dados</i>	90
4.7.2 <i>Particularidades da linha de aquisição do ADC114</i>	94
4.7.3 <i>Particularidades da linha de aquisição do ADC NIM</i>	95
4.7.4 <i>Algumas pequenas diferenças entre os programas</i>	96
4.7.5 <i>Acesso simultâneo ao bastidor CAMAC</i>	98
5 CONCLUSÃO.....	102
REFERÊNCIAS BIBLIOGRÁFICAS.....	104
APÊNDICE 1	110

1 Introdução

Por volta do ano de 1973, o Laboratório do Acelerador Linear (LAL) recebeu um computador DIGITAL PDP-11 e uma equipe iniciou o desenvolvimento de um sistema de aquisição de dados automatizado utilizando este computador. Ao final daquela década o sistema atingiu a sua maturidade, sendo possível adquirir, visualizar e analisar dados de espectroscopia gama nuclear de uma forma totalmente controlada por computador. Para termos uma idéia do que isto significou, basta lembrar que o microcomputador pessoal IBM-PC surgiu apenas no ano de 1982.

O tempo e a tecnologia são completamente incontroláveis e no início dos anos noventa o panorama da computação havia se modificado completamente, tornando o sistema de aquisição de dados do LAL obsoleto devido à incompatibilidade com as novas tecnologias e padrões que surgiram, à dificuldade e preço da manutenção da máquina e ao seu desempenho. Neste período, também surgiu a necessidade e possibilidade de aquisição de dados multiparamétricos, mais especificamente de medidas envolvendo coincidências entre vários detetores. Portanto, a troca do computador PDP-11 e o desenvolvimento de um novo sistema de aquisição de dados que não só substituísse o antigo como também possibilitasse a execução de medidas multiparamétricas, tornou-se um objetivo do LAL.

Este trabalho descreve o desenvolvimento do sistema de aquisição de dados do LAL. Esta é uma tarefa que envolve muitas facetas diferentes, tais como a análise do sistema almejado, o planejamento da execução das tarefas necessárias, o desenvolvimento de *hardware* e *software* e a validação do sistema no sentido das informações físicas obtidas pelo mesmo. Todas estas facetas, embora sejam tão diferentes entre si são complementares e inter-

relacionadas, pois o resultado que o usuário terá acesso será a somatória das mesmas e portanto elas devem estar em completa harmonia para atender os requisitos e expectativas depositadas no mesmo.

Uma das facetas de todo este complexo sistema é o desenvolvimento de programas de computador que possibilitem ao usuário final executar o controle do *hardware* com o objetivo de efetuar a aquisição dos dados de seu interesse. Normalmente o programador apenas obtém do usuário as especificações e escreve um programa que atenda aos requisitos pedidos. Mas como veremos, quando se trata da montagem de um sistema de aquisição de tempo real, no qual é necessário o desenvolvimento de *hardware*, o programador necessita também participar da especificação e da construção do mesmo, pois neste caso ele será o usuário deste equipamento e deve procurar otimizar a interface entre o programa e a eletrônica. Além do mais, como o *hardware* construído deve se comunicar com o computador que efetuará o controle, o programador deve participar criando programas que permitam a execução dos testes necessários ao desenvolvimento e depuração do equipamento.

No capítulo 2, trataremos da especificação do novo sistema de aquisição de dados, da necessidade e do desenvolvimento de um controlador de bastidor CAMAC e da montagem física do sistema. Já no capítulo 3 serão analisadas as medidas multiparamétricas e o desenvolvimento de um módulo que permite a sincronia para verificação de coincidências de até 8 detetores. Nesse capítulo será mostrado como o desenvolvimento de um equipamento para aquisição de dados deve ser feito em conjunto pelo programador e pelo engenheiro eletrônico, com vistas a obter dados que possuam o sentido físico real. Finalizando o capítulo 4 apresenta os programas que foram desenvolvidos para o sistema de aquisição. Veremos a estrutura destes

programas, e será mostrada como o uso de interfaces gráficas mascara a complexidade do sistema para o usuário final.

Em cada um dos capítulos que compõem o presente trabalho, serão discutidas as várias tecnologias à disposição para o desenvolvimento ou montagem de um sistema de aquisição, tanto para o *hardware* quanto para o *software*. Junto às possibilidades apresentadas serão discutidos os motivos que levaram às escolhas tomadas para a construção do sistema de aquisição de dados do LAL.

2 Aquisição de dados no L.A.L.

2.1 Aquisição de dados com o PDP-11

O minicomputador DIGITAL PDP-11 (Vide figura 1) foi o primeiro computador do mundo de 16 bits. Uma de suas características inovadoras era possuir um barramento único chamado *UNIBUS*, onde eram conectados não só a memória e a unidade central de processamento (UCP) como também os discos, unidades de fita, etc.



Figura 1 - Foto do minicomputador Digital PDP-11, mostrando da esquerda para a direita; a unidade de fita, um disco removível de 2,2 Mbytes, a leitora/perfuradora de fita de papel, o terminal gráfico Tektronix e o console DECWriter. Vale lembrar que nesta época não havia terminal de vídeo alfanumérico e toda comunicação com o PDP-11 devia ser feita através da DECWriter ou do terminal Tektronix.

Baseado neste minicomputador foi desenvolvido o sistema de aquisição de dados do LAL, que compreendia o *hardware* e o *software*^[6] dedicados a esta tarefa. Para tanto, foi conectado ao UNIBUS um dispositivo chamado *Multichannel Branch Driver* (MBD), ou seja, um canal múltiplo de acesso direto à memória microprogramável. O MBD por sua vez era conectado a um controlador de bastidor CAMAC^[10] através de um barramento chamado *Branch Highway*. Os dados digitais provenientes dos *Analogic to Digital Converter* (ADC) adentravam ao sistema através de módulos *Input Register* conectados ao barramento *Dataway* do bastidor CAMAC. A figura 2 nos mostra um esquema simplificado deste sistema.

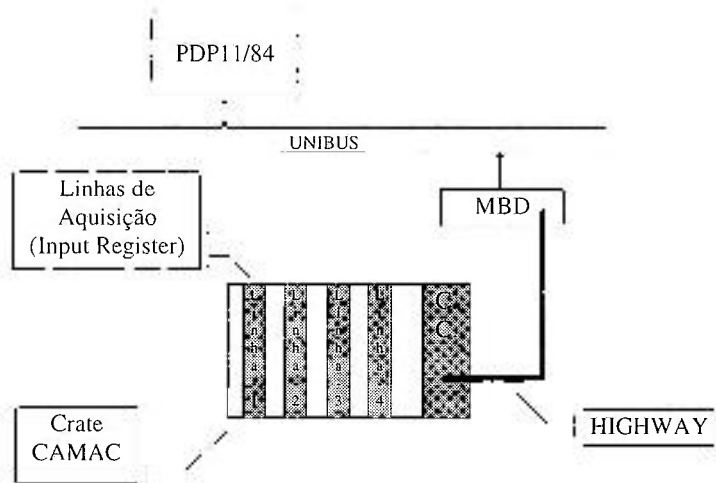


Figura 2 - Esquema simplificado do sistema de aquisição do LAL, baseado no computador PDP-11. O Controlador de bastidor CAMAC (C.C.) era conectado do MBD através do barramento *HIGHWAY*.

Nesta configuração, era possível realizar a aquisição dos dados de até 4 experimentos simultaneamente. Os sinais lidos nos *Input Register* eram recolhidos pelo MBD que os escrevia diretamente na memória do PDP-11, tornando-os disponíveis para visualização e análise em tempo real.

2.2 Por que substituir o sistema de aquisição de dados ?

Embora o sistema baseado no conjunto PDP-11 + MBD atendesse aos requisitos do LAL para um sistema de aquisição de dados, ele tornou-se desatualizado forçando a sua substituição devido à grande evolução ocorrida no mundo da computação neste período. Entre os motivos que forçaram esta atitude podemos destacar os seguintes:

1. A descontinuidade de produção do computador PDP-11, acrescida ao abandono pela indústria da arquitetura *UNIBUS*, fez com que a conexão do MBD com outro computador se tornasse impossível.
2. O alto custo de manutenção do PDP-11.
3. A necessidade de programar o MBD em *assembler* tornava a manutenção dos programas de aquisição muito difícil e dispendiosa em tempo.
4. O vertiginoso aumento da capacidade de processamento das estações de trabalho e dos microcomputadores, a forte queda em seu preço e a sua padronização, tornou-os altamente vantajosos em relação a este sistema particular e dedicado.

Assim, no final de 1994 iniciaram-se os estudos visando a substituição do sistema de aquisição baseado no PDP-11. Este novo sistema deveria não só substituir tecnologicamente o PDP-11, mas também proporcionar um salto no sentido de atingir taxas de aquisição mais elevadas e de tornar o desenvolvimento e manutenção do *software* mais fácil e barato. Logo optou-se por um sistema que não só fosse atual como também obedecesse aos padrões ditados pela indústria. Quanto às taxas de aquisição, desejava-se que o novo sistema que permitisse adquirir mais que os 20.000 eventos por segundo atingidos pelo PDP-11 + MBD em condições especiais. Esta

necessidade vinha do fato de que se planejava realizar experiências envolvendo muitos detetores em coincidência, o que geraria uma quantidade de informações muito maior que os poucos bytes lidos a cada evento, que era o padrão do LAL até então. Finalmente quanto à manutenção do sistema procurava-se algo que permitisse escrever programas em linguagens de nível mais alto que o *assembler*, sem perda da eficiência, proporcionando assim uma economia muito grande de tempo e pessoal necessário para o desenvolvimento e manutenção.

2.3 Alternativas ao sistema de aquisição de dados LAL

Como podemos comprovar em nosso dia-a-dia, os microcomputadores tiveram um imenso aumento em seu poder de processamento e de armazenagem na década de 90, transformando-os em candidatos naturais à base do novo sistema de aquisição de dados. Foram também analisadas outras alternativas, tais como estações de trabalho ou até mesmo o minicomputador μ VAX que na época era o único computador a utilizar o sistema *UNIBUS*. Mas elas, além de caras, não eram padrões amplamente aceitos pelo mercado, o que poderia acarretar o mesmo problema encontrado com o PDP, ou seja, simplesmente desaparecerem.

Nesta disputa, o microcomputador padrão IBM-PC foi o vencedor devido à sua ampla aceitação, utilização, facilidade de manutenção e expansão, além do seu baixo preço, o que possibilita a substituição do computador sem muitas dificuldades técnicas ou financeiras.

No outro extremo do sistema, optou-se por manter o padrão CAMAC, já que havia no laboratório um bom número de módulos e bastidores, e uma grande experiência não só na utilização como também na construção e manutenção de módulos neste padrão. Um fator que também contribuiu para esta decisão, foi o fato deste padrão ainda ser amplamente utilizado em física

nuclear e também porque o novo sistema de coincidência entre detetores desenvolvido no LAL utiliza um *Time to Digital Converter* (TDC) que é facilmente encontrado no padrão CAMAC.

Tomada a decisão sobre o computador a ser utilizado e pela manutenção do CAMAC, havia ainda uma dificuldade: como conectar o microcomputador ao bastidor CAMAC. Na verdade o problema se resume a unir o barramento de dados do microcomputador ao barramento de dados *Dataway* do bastidor CAMAC. Os barramentos de uma maneira geral podem ser divididos em três categorias^[16] :

1. Barramentos de computadores tais como ISA, EISA, NuBus, VME, Sbus, PCI ou *MicroChannel*. Estes barramentos são projetados para interligar subsistemas e periféricos com o computador. Além das linhas de dados, há linhas destinadas à comunicação entre os diferentes processadores para que seja possível arbitrar sua utilização.
2. Barramentos de bastidores de instrumentos como CAMAC, VXI ou *FastBus*. Neste caso, a conexão é efetuada entre módulos de instrumentos, sendo que o bastidor fornece energia para todos os módulos. Geralmente, estes módulos são incapazes de funcionar fora do bastidor e eles devem ser controlados por um gerenciador que normalmente é conectado a um computador, com a função de arbitrar o uso do barramento de dados.
3. Barramentos de módulos independentes. Os mais conhecidos são o IEEE-488 (também conhecido como *General Purpose Instrument Bus*, GPIB), RS-232 e SCSI. O barramento é utilizado apenas para comunicação e troca de dados entre

equipamentos, sendo que estes podem funcionar de forma independente.

De acordo com a definição acima, o nosso problema consiste em interligar um barramento do tipo 1 com um barramento do tipo 2. Para isto, há no mercado várias propostas, normalmente utilizando barramentos do tipo 3. Dois tipos de controladores de bastidor CAMAC diferentes foram testados. Um deles utiliza o barramento GPIB para comunicação com o microcomputador e o outro utiliza o SCSI. Foi desenvolvido *software* básico de comunicação do microcomputador com o bastidor CAMAC e os dois barramentos atingiram resultados semelhantes quanto à taxa máxima de aquisição, da ordem de 6000 eventos por segundo.

O baixo desempenho em ambos os casos é devido à característica de nosso evento típico. Estes barramentos foram projetados para transferências de grandes blocos de dados. Quando uma transferência é solicitada, os equipamentos envolvidos trocam entre si algumas informações através de um protocolo, que são palavras de controle no caso do SCSI ou troca de sinais no GPIB. Nesta etapa da comunicação é gasto um tempo na execução deste protocolo, mas como se transfere uma grande quantidade de dados, este tempo se torna imperceptível, ou seja, a relação entre o tempo despendido no protocolo para iniciar a transferência e o tempo utilizado na transferência dos dados em si é bem pequena. Em nosso caso como o evento possui apenas 2 bytes, esta relação fica muito grande e portanto demora-se mais tempo na realização do protocolo inicial do que na transmissão dos dados.

Diante desta situação só havia duas alternativas para resolver o problema da taxa de aquisição. Uma era encontrar alguma forma de armazenar os dados e transferi-los de uma só vez, diminuindo assim a relação entre tempo gasto com o protocolo e com a transferência do dado. A outra alternativa era otimizar a transferência para o tipo de dados que utilizamos. A

primeira alternativa demandava um controlador de bastidor “inteligente”, ou seja, que soubesse tomar “decisões” e armazenar os dados de tal forma que, ao atingir uma dada quantidade de dados, ele os transferiria em bloco para o computador. Infelizmente, não se encontrou no mercado um tal controlador. Existem vários controladores que possuem até processadores em seu interior, mas eles estão ali apenas para controlar o barramento do tipo 3 que conecta o controlador de bastidor ao computador, e não para otimizar a tomada e armazenamento dos dados adquiridos. Outros controladores podem até executar pequenas listas de procedimentos ao receber um dado gatilho, mas eles são muito restritos na quantidade e qualidade dos comandos possíveis de serem executados.

Como não se encontrou no mercado uma solução para o armazenamento ou empacotamento dos dados no bastidor CAMAC, restou apenas a alternativa de otimizar a transferência de eventos de poucos bytes. Há duas maneiras de solucionar este problema. Uma é desenvolver um barramento do tipo 3 que seja otimizado para transferência de poucos bytes em grande quantidade de forma assíncrona. A outra alternativa é eliminar a necessidade de um terceiro barramento, simplesmente interligando os barramentos do computador diretamente ao barramento do bastidor CAMAC. Em ambos os casos é necessário o desenvolvimento de equipamento, pois no primeiro caso é necessário desenvolver um barramento e a sua interface com o bastidor CAMAC e com o computador. Já no segundo desenvolve-se apenas uma interface entre os dois barramentos existentes conectando-os diretamente, visto que no mercado não há nada semelhante. Optou-se pela segunda alternativa, já que o desenvolvimento de um barramento é algo para toda uma equipe especializada, pois demanda conhecimentos e experiências muito peculiares, tais como: controle, linha de transmissão, interferências e

etc. Já a interligação de dois barramentos necessita que se construa uma interface que realize a conexão e tradução dos protocolos.

Assim definido o computador a ser utilizado, IBM-PC, e o padrão de interface para a instrumentação nuclear, CAMAC, iniciou-se o desenvolvimento da interface para interligar estes dois “mundos”.

2.4 Controlador de bastidor CAMAC

No bastidor CAMAC o módulo responsável pela comunicação com o computador é o controlador. Já no microcomputador a maneira de se conectar qualquer periférico ou dispositivo externo é através dos *slots* de expansão. Logo, a interface que deveria ser desenvolvida precisava estar conectada a um *slot* de expansão no microcomputador e ocupar as posições 24 e 25 no bastidor CAMAC, que são destinadas ao controlador. É completamente impossível fazer e usar uma única placa para conectar ambos, mesmo por motivos mecânicos, logo, um cabo de aproximadamente 1,5 m deveria ser utilizado para dar liberdade ao conjunto. Mas uma dificuldade se põe então: como conectar o barramento do microcomputador ao bastidor CAMAC sem ser diretamente e ainda através de um cabo. Para isto basta estender um dos dois barramentos, o *Dataway* do CAMAC ou o barramento do microcomputador através do cabo. Esta solução parece simples à primeira vista, mas é de difícil implementação, visto que estamos falando de inúmeros sinais trafegando por dezenas de linhas de transmissão em paralelo a uma taxa de alguns MHz, podendo não só emitir ruído eletrônico para o ambiente, como também captar ruído que atrapalharia o funcionamento do sistema. Para implementarmos esta solução é necessário estudarmos um pouco do funcionamento dos barramentos do microcomputador e do CAMAC.

2.5 Barramentos do microcomputador padrão IBM-PC

A família de computadores IBM-PC surgiu no início da década de 80 com o lançamento de um microcomputador baseado no processador *Intel* 8088, com um clock de 4,77MHz, 64kb de memória RAM (*Random Access Memory*) e ainda possuía cerca de 5 conectores para expansão^[17]. Estes conectores eram de encaixe permitindo o contato das placas de expansão com a placa mãe do microcomputador. Ao se encaixar uma placa nestes conectores, ela entrava em contato com o barramento do computador, por onde trafegavam os sinais de dados, de endereçamento e controle. Desta forma a placa poderia utilizar as linhas necessárias ao seu funcionamento e ignorar as demais. O barramento disponibilizado no IBM-PC original era o ISA^[31] (*Industry Standard Architecture*) de apenas 8 bits de linhas de dados. Posteriormente, quando do lançamento do IBM-PC-AT este barramento foi expandido para dados de 16 bits.

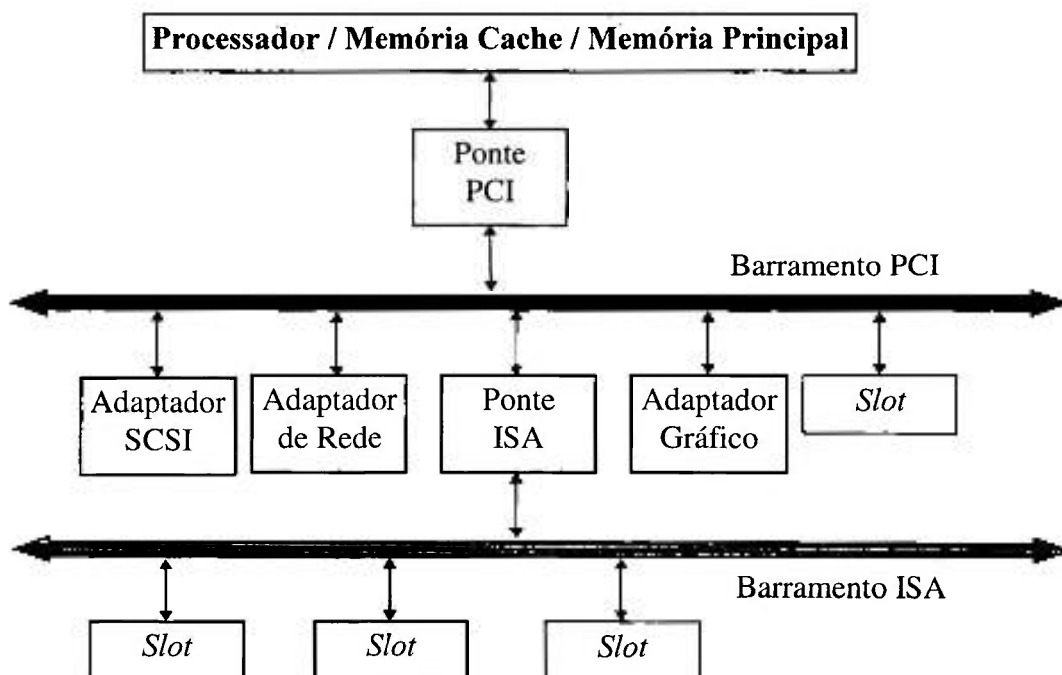


Figura 3 - Esquema simplificado dos barramentos de um microcomputador IBM-PC atual. Alguns dos dispositivos PCI podem estar integrados à placa mãe, mas mesmo assim estão ligados a este barramento.

Vários outros tipos de barramentos foram introduzidos ao IBM-PC nestes anos, tais como MCA, EISA, VLB, mas todos eles foram com o passar dos anos, ou sendo abandonados, ou ocupando pequenos nichos de mercado. Atualmente, os microcomputadores utilizam uma estrutura na qual coexistem dois barramentos: o ISA de 16 bits e o PCI de 32/64 bits como vemos na figura 3.

O processador, a memória cache e a memória principal também possuem um barramento mas, como este é dedicado a apenas estes componentes e não é acessível externamente, não nos interessa para o nosso propósito. O barramento PCI possui como principais características:

1. Taxa máxima de 133 Mbytes/s com a largura de 32 bits para dados.
2. Opera em frequência máxima de 33 MHz.
3. Multiplexa as linhas de dados e endereçamento para a redução do número de pinos no conector.
4. O conector possui 124 pinos.

Em comparação o ISA possui:

1. 16 bits de largura para dados.
2. Frequência de 8,33 MHz podendo operar até 12 MHz, mas com possibilidade de incompatibilidade com determinadas placas de expansão.
3. Dados e endereços trafegam em linhas separadas.
4. O conector possui 98 pinos.

Das características apresentadas acima, vemos que os dois barramentos se complementam no sentido a atender demandas bem distintas. O ISA se destina a dispositivos lentos, baixa taxa de transmissão e de no máximo dois bytes por vez. Já o PCI se destina a altas taxas de transmissão, grandes

volumes de dados e dispositivos rápidos. É mais caro porque exige eletrônica mais sofisticada devido a rapidez, e mais complexo porque multiplexa dados e endereços, além de possuir mais linhas em paralelo.

Desde já podemos perceber algumas vantagens e desvantagens de um em relação ao outro com vistas ao nosso propósito. O menor número de pinos do ISA permite a utilização de um cabo com menor número de sinais. Também a menor frequência de trabalho diminui a probabilidade de emissão ou recepção de ruídos através da linha de transmissão. O barramento PCI é bem recente e portanto há pouca literatura e experiência a seu respeito. Já o ISA é bem difundido e utilizado pela indústria, visto os quase 20 anos de utilização no IBM-PC e portanto há muita literatura a seu respeito, inclusive algumas que demonstram como ele pode ser estendido^[7].

Agora, para podermos decidir qual barramento disponível no microcomputador utilizar, devemos estudar um pouco o barramento do CAMAC para ver qual dos dois será mais indicado.

2.6 *Dataway, o barramento do CAMAC*

A figura 4 nos apresenta um bastidor CAMAC típico, onde vemos 5 módulos conectados ao *Dataway* localizado ao fundo do bastidor, sendo visíveis apenas os conectores verticais. O módulo mais à direita da figura 4 é o controlador do bastidor que sempre ocupa as posições 24 e 25.

Todas as posições do bastidor, menos a 25, são equivalentes entre si, no sentido de possuírem os mesmos sinais. A estação 25 é diferente porque, ao invés de ter acesso às linhas de dados, esta posição possui as linhas de endereçamento de módulos N (Número da estação a ser acessada) e de pedido de atendimento dos módulos individuais LAM (*Look-At-Me*). O barramento de dados é composto por 24 linhas de escrita e 24 linhas de leitura, as quais,

somando-se às linhas de controle, sinalização, tensão e outras, levam a um total de 86 pinos em cada conector. Como os sinais de N e LAM ocupam as linhas de dados na estação 25, então vemos que é possível ter 24 N's e 24 LAM's, correspondentes ao total de estações disponíveis no bastidor. Nenhum módulo pode acessar as linhas de dados sem que o controlador o habilite para isto através das linhas N. Desta forma, o controle do bastidor é totalmente efetuado pelo controlador e quanto mais ele se aproximar do computador, mais “inteligência” e mais rapidez teremos no atendimento aos módulos e no acesso aos mesmos.

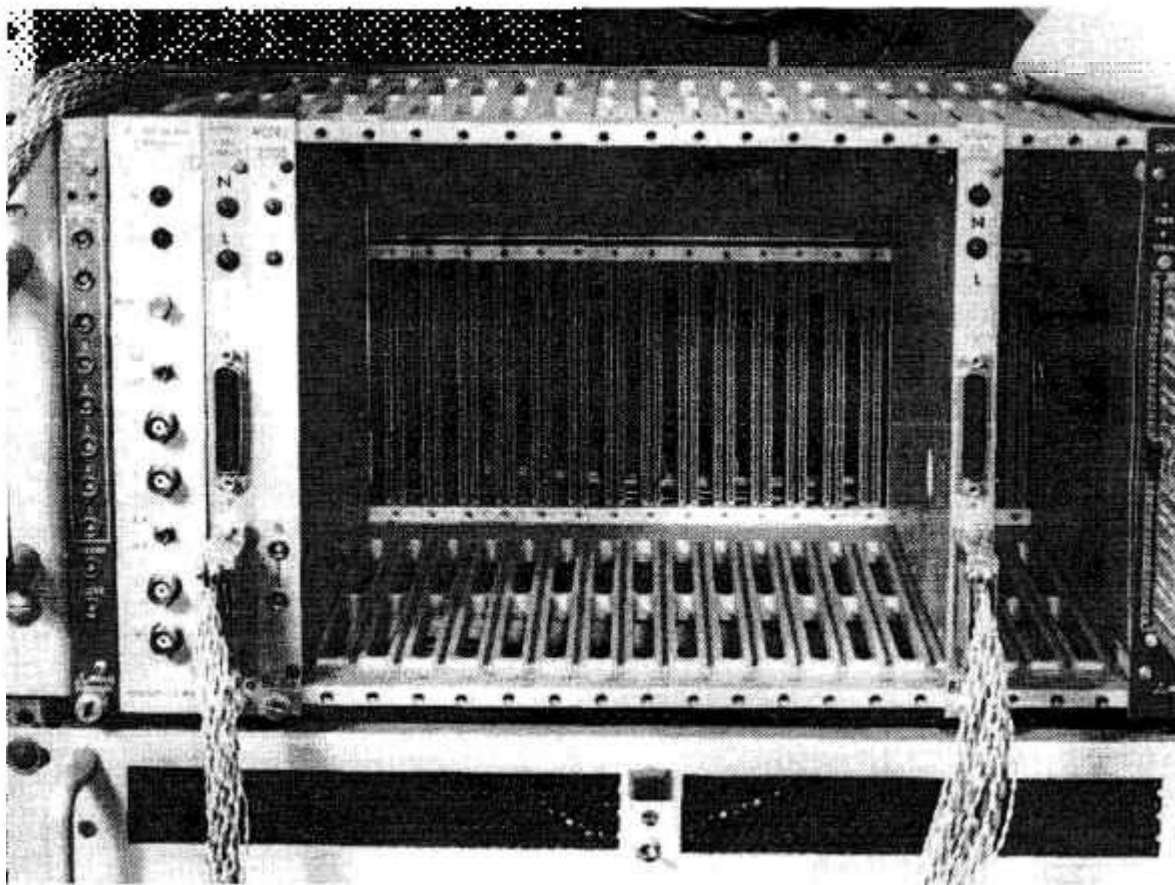


Figura 4 - Foto de um bastidor CAMAC. O módulo mais à direita é o controlador do bastidor e os outros são módulos típicos neste padrão. No fundo do bastidor, vê-se os conectores do *Dataway* e atrás deles está fisicamente o barramento.

Ao fundo do bastidor se encontra o *Dataway* que é o barramento utilizado para comunicação entre os módulos e o controlador. O *Dataway*

trabalha com uma frequência de 1 MHz o que o tornaria mais indicado dentre os três barramentos analisados a ser estendido, no sentido de ruídos na linha de transmissão. Mas ao somarmos as linhas normais às estações de 1 a 24 e às linhas específicas da estação 25, chegamos ao total de 136 sinais a serem transmitidos pelo cabo, o que é um número bastante elevado.

2.7 Qual dos barramentos devemos estender ?

Após esta rápida descrição dos barramentos com os quais estamos lidando, podemos finalmente decidir como elaborar esta conexão. O PCI é o mais rápido de todos e isto traz vários inconvenientes, tais como: 1º se for estendido ele será mais sujeito à emissão e recepção de ruídos, 2º a grande diferença de frequência entre ele e o *Dataway* pode gerar conflitos ou dessincronia entre os dois.

Assim a conexão entre o ISA e o *Dataway* é a mais recomendada do ponto de vista de frequência. Tomando apenas este aspecto em consideração, nós deveríamos estender o *Dataway* ao ISA, visto que ele possui a frequência mais baixa. Por outro lado, ele possui o maior número de sinais tornando o cabo muito volumoso. Também deve-se levar em conta que em algum ponto, seja no bastidor CAMAC ou no gabinete do micro, ou ainda numa caixa intermediária, deverá existir a eletrônica que converta o protocolo de um barramento no do outro. Se estendermos o *Dataway* ao ISA, deverá existir dentro do gabinete do micro o circuito tradutor, mas o gabinete não é projetado, tanto em relação ao espaço quanto a ventilação, para receber um circuito de tal tamanho. Logo o mais natural é estender o ISA ao *Dataway*, pois assim o tradutor ficaria na própria caixa do controlador do bastidor, o número de sinais seria o menor dentre os três barramentos, com a frequência de funcionamento intermediária.

Uma última consideração deveria ser feita. O PCI tem 32 linhas para dados, o *Dataway* 24 e o ISA 16, assim o PCI levaria vantagem sobre o ISA, já que numa única leitura ele poderia ler todos os 24 bits de dados do *Dataway* e ainda poderia receber alguns sinais de controle nos bits restantes. Mas uma limitação de software anula esta vantagem do PCI, pois embora todos os processadores da *Intel* acima do 80386 possuam uma instrução para leitura de até 4 bytes de uma porta de entrada e saída, os compiladores comerciais apenas disponibilizam os comandos para leitura de 1 byte e de 2 bytes ou uma palavra. Logo, seriam necessárias duas operações de leitura para ler os 32 bits, o que do ponto de vista de eficiência o deixa ao mesmo nível do ISA, já que o fator limitante do desempenho, neste caso, será a frequência de 1 MHz do *Dataway*.

2.8 Dominando o barramento ISA

Escolhida a estrutura da conexão, o próximo passo foi o início do aprendizado do funcionamento do barramento ISA. Para isto, iniciou-se o estudo para a construção de uma placa que obedecesse a este padrão e funcionasse como uma placa de expansão qualquer do microcomputador.

A primeira placa desenvolvida era apenas uma entrada e saída de 16 bits e funcionou corretamente, após superar-se algumas dificuldades inerentes a qualquer desenvolvimento. Nesta primeira placa também se testou o funcionamento das interrupções de *hardware* e foi possível gerar interrupção e capturá-la através do *software*.

O próximo passo era o mais importante, pois havia ainda uma dúvida: seria possível estender o ISA? Assim decidiu-se tomar uma placa típica do barramento ISA, uma placa controladora de discos IDE, e projetar um extensor que a fizesse funcionar a 1,5 metros do computador. Para isto, foi

desenvolvida uma placa que era conectada ao barramento ISA e que acondicionava os sinais de tal forma a prepará-los para a transmissão que era efetuada através de dois cabos planos constituídos de 32 pares trançados (cada par continha um sinal e uma terra). A principal preocupação nesta etapa de acondicionamento era separar as linhas de dados de leitura e de escrita, de tal forma que elas seguissem por caminhos independentes e unidirecionais. Na outra extremidade do cabo, um circuito seria responsável pela reconstrução dos sinais e pela disponibilização destes através de um conector padrão ISA. Neste conector seria colocada a placa controladora IDE^[17] e a esta seriam ligados o disco rígido, o disco flexível e o *mouse* do computador. Tudo isto estando a 1,5 metros do computador deveria ser capaz de funcionar. Isso funcionou^[4] (vide figura 5), provando que a idéia básica para a construção de um controlador CAMAC-ISA era totalmente viável.

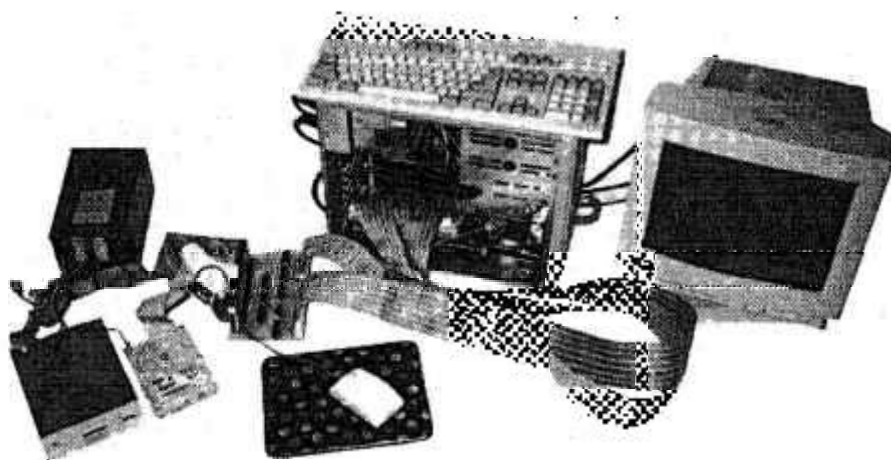


Figura 5 - Extensor do barramento ISA. Nesta foto são visíveis os dois cabos de extensão conectados às placas condicionadoras dos sinais. Uma se encontra no interior do microcomputador, conectada ao barramento ISA, e a outra se encontra ao lado, onde está colocada a placa controladora de discos IDE com o disco rígido, disco flexível e *mouse*, conectados.

O funcionamento do microcomputador nestas condições mostrou que era possível estender o barramento ISA, e o fato do ruído na extremidade do cabo ser desprezível vem comprovar a qualidade do projeto deste extensor, veja a figura 6.

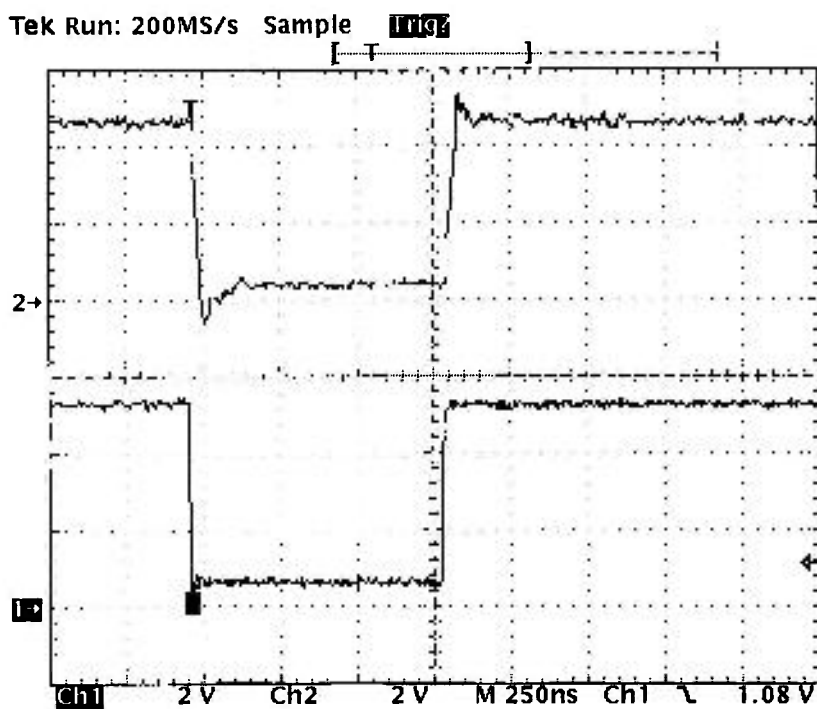


Figura 6 - *Hardcopy* da tela do osciloscópio mostrando na parte inferior (canal 1) um sinal do barramento ISA no conector localizado na placa-mãe do computador e na parte superior (canal 2) está o mesmo sinal após passar pelo extensor. Vemos que o ruído é desprezível e o sinal não se degradou.

2.9 Especificação do controlador CAMAC-ISA

Tendo dominado a técnica de construção de placas no padrão ISA e do extensor, o próximo passo foi definir do controlador ISA. No esquema utilizado para a construção do extensor ISA, o circuito responsável pela reconstrução dos sinais do barramento ISA seria acoplado a um tradutor ISA-

Dataway. Para tanto, deveríamos especificar como este tradutor deveria responder aos comandos ISA e quais as atitudes que ele tomaria no *Dataway*.

O fato de ligarmos estes dois barramentos diretamente trouxe como consequência a facilidade de podermos tratar o *Dataway* como se fosse um dispositivo ISA comum. Portanto, ele deveria possuir alguns endereços de entrada e saída, poderia gerar interrupção de *hardware* no microcomputador e até mesmo efetuar DMA (*Direct Memory Access*). Assim, definiu-se que o controlador seria acessível através de quatro endereços de memória pares e sequenciais a partir de um endereço BASE, selecionado através de um *jumper* no próprio controlador. Além destes endereços o controlador deveria poder gerar interrupções de *hardware* ao receber determinados sinais de LAM (Veja a especificação completa no apêndice 1). A seleção de qual posição do bastidor deveria gerar uma determinada interrupção também seria por *jumper's*.

Finalmente, embora a capacidade de efetuar DMA não estivesse programada para o primeiro controlador, os sinais necessários para a sua execução foram acrescentados ao extensor, para futura utilização. Assim, de acordo com as necessidades do controlador foi projetado um novo extensor do barramento ISA que contivesse apenas os sinais utilizados. Isto tornou desnecessário a extensão dos 98 sinais do ISA, sendo utilizados apenas 46. Como no ISA as linhas de dados são bidirecionais e no extensor elas são unidimensionais, foi necessário acrescentar mais 16 sinais. Assim, foram necessários 62 pares trançados para efetuar esta extensão.

Nesta etapa do desenvolvimento deste módulo ficou patente a necessidade e as vantagens da interação entre o programador de *software* e o projetista do *hardware*. O programador tem como objetivo obter um equipamento que lhe seja o mais amigável na comunicação, mas não está preocupado com a dificuldade em implementar esta facilidade. Já o projetista

do módulo conhece as dificuldades e limitações no projeto, construção e funcionamento elétrico e eletrônico, e portanto, se preocupa com o funcionamento e não com a interface. Se os dois interagirem será possível cada um negociar com o outro e no final poderão obter uma solução de compromisso que permita uma fácil interação entre *software* e *hardware* com um funcionamento correto e seguro.

Um bom exemplo disto é a quantidade de modos que o LAM de um módulo no bastidor pôde ser captado pelo computador (*software*). No total são quatro formas diferentes de informar se um dado LAM foi ativado e assim o programa pode atender da forma que lhe for mais conveniente a cada situação específica, uma necessidade típica para um programador. Já o programador gostaria de ter ainda mais flexibilidade podendo associar uma determinada posição no bastidor à uma interrupção de *hardware*, via programação, mas esta solução traria dificuldades - e riscos - adicionais ao projeto. Logo uma solução de compromisso foi tomada e o uso de *jumper's* se mostrou uma solução adequada.

2.10 Construção do controlador CAMAC-ISA

Ultrapassada a etapa da especificação da interface do módulo como um dispositivo ISA, e dos testes de viabilidade e da construção do extensor do barramento, deu-se início ao projeto e construção do controlador propriamente dito. Para tanto foram utilizadas técnicas modernas de projeto eletrônico e roteamento de circuitos elétricos, como o uso de um CAD (*Computer Aided Designer*)^[11] para montagem do esquema eletrônico e um programa roteador^[36] do circuito elétrico das placas. O resultado^[29] pode ser observado nas figuras 7 e 8 pela alta qualidade do trabalho executado e complexidade deste projeto.

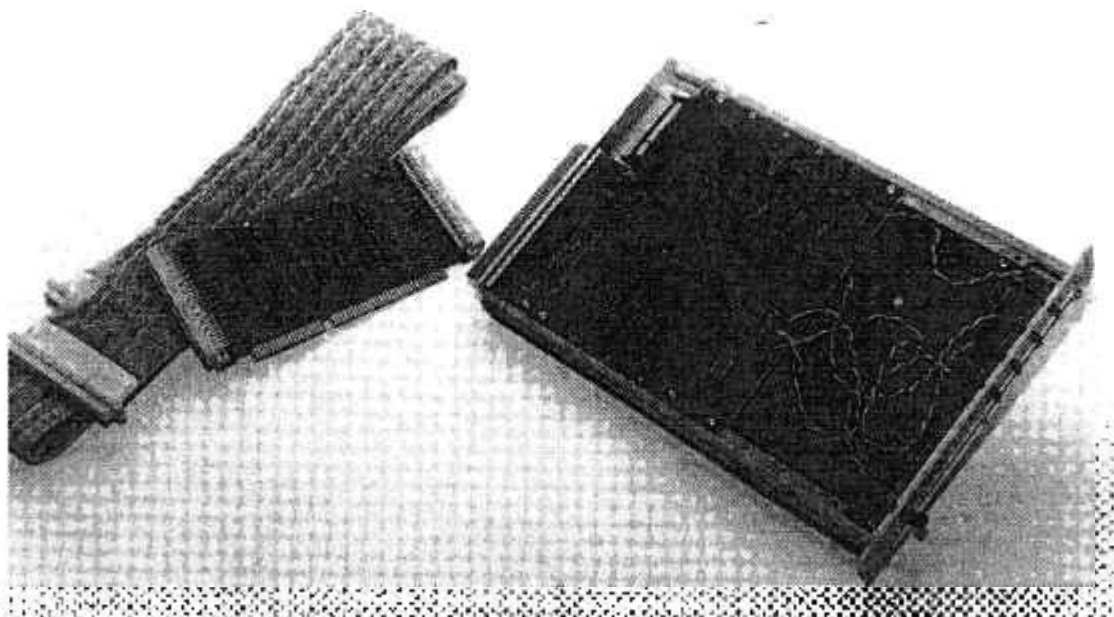


Figura 7 - Da esquerda para a direita vemos, os cabos conexão, a placa ISA que é conectada ao barramento do microcomputador e o controlador CAMAC-ISA.

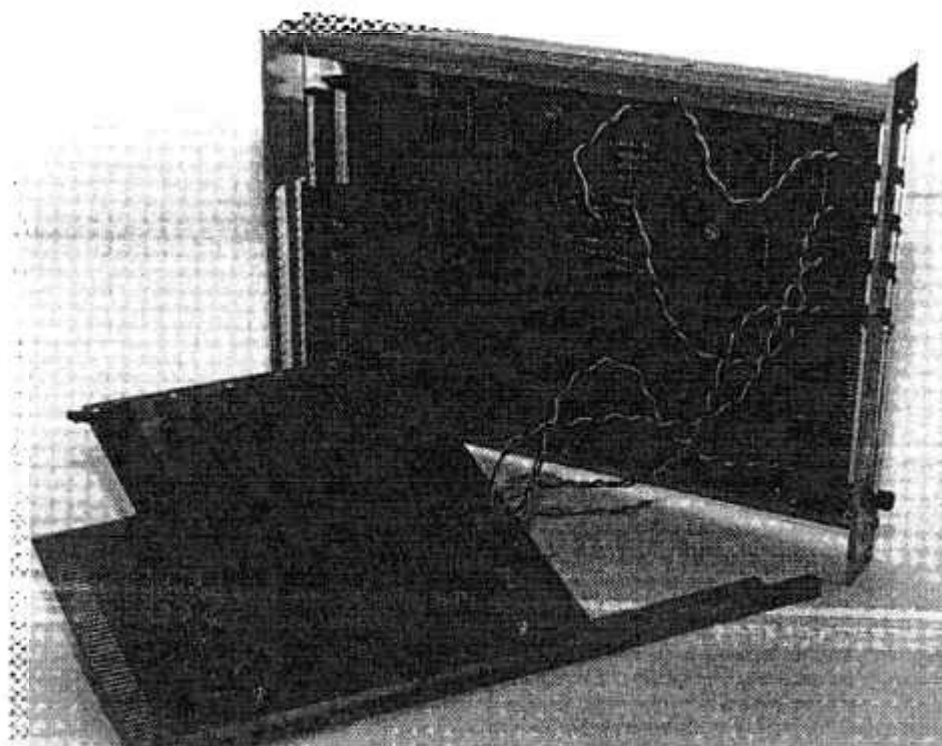


Figura 8 - Vemos as duas placas que formam o controlador CAMAC-ISA, que são conectadas às posições 24 e 25 do bastidor. Estas duas placas são interligadas através de um cabo plano.

Terminada a construção do controlador, iniciaram-se os testes. Como todo módulo novo, ele foi energizado sozinho dentro do bastidor, para verificar qualquer erro nas linhas de tensão. A seguir, foi conectado ao computador para verificar se o mesmo não sofria alguma alteração pela sua presença no barramento ISA. A comunicação entre o computador e o *Dataway* foi o próximo passo e nesta etapa um módulo CAMAC foi de extrema importância: o *Dataway Display* Jorway modelo M203^[11] (DD), pois ele possibilitou a visualização em uma matriz de LED's das informações escritas e lidas pelo computador no *Dataway*, que agora funciona como um dispositivo ISA.

A etapa de testes de controle do *Dataway* e comunicação com o computador durou menos de uma semana, sendo encontrados apenas pequenos problemas, tais como alguns *chips* que o roteador não conectou à tensão de alimentação e outros pequenos ajustes comuns a qualquer equipamento novo. Agora era necessário verificar o desempenho deste módulo nas situações para as quais ele foi desenvolvido, visando atingir taxas maiores que as alcançadas pelos controladores comerciais testados.

O primeiro passo para obter a taxa que o controlador atingiria foi efetuar escritas e leituras no módulo DD. Este módulo nos permite um total controle do que está acontecendo, pois apresenta em seu painel o que é escrito e o que é lido, assim não só poderíamos observar a taxa como também a coerência dos dados. Assim, foi escrito um pequeno programa que escrevia um dado no DD e logo a seguir lia o módulo, comparando o dado escrito com o lido, validando assim a coerência entre a escrita e a leitura. Algo não esperado aconteceu neste teste. O controlador possui um circuito que verifica se ele é endereçado pelo barramento ISA enquanto está sendo executado um ciclo CAMAC. Quando isto acontece um LED é acesso. A princípio, este circuito foi colocado como uma segurança extra e só situações de exceção

poderiam acender este LED. Na execução do programa, este LED se acendeu, indicando que o computador estava efetuando acessos ao controlador durante o ciclo CAMAC. Verificando o circuito e o programa, chegou-se à conclusão que realmente o barramento ISA do computador estava sendo muito rápido e portanto foi acrescentado, no programa, um pequeno atraso logo após cada comando CAMAC executado pelo controlador. Este laço tem como único objetivo atrasar a execução do programa para que o comando seja terminado no *Dataway*. Este acontecimento, ao invés de trazer maus presságios, trouxe muita esperança, pois vimos que agora era o computador que esperava a execução do comando CAMAC e não mais o *Dataway* que deveria ficar ocioso esperando o atendimento do computador. Desta forma taxas muito maiores seriam possíveis, pois o tempo morto do sistema devido ao *Dataway* seria reduzido ao mínimo. Com este módulo e programa, atingiu-se uma taxa de escrita e leitura de dados de 16 bits no *Dataway* de 1,8 Mbytes/s e alcançou-se uma taxa de 730.000 ciclos CAMAC por segundo. A taxa máxima do *Dataway* que seria de 1Mhz não é possível ser atingida por causa da latência do barramento ISA, mas vemos que chegou-se a aproximadamente $\frac{3}{4}$ do total.

O próximo passo foi efetuar uma medida real e para tanto foi utilizado um ADC modelo 114^[25] da Ortec (ADC114). Este ADC, além de ser do padrão CAMAC, utiliza o método de aproximações sucessivas para a conversão, possui 14 bits de resolução e o seu tempo de conversão é fixo em 5 μ s, o que permite uma alta taxa de aquisição. A primeira medida efetuada com alta taxa utilizou um programa que verificava a máscara de LAM do controlador e quando o bit correspondente à posição do ADC no bastidor era ligado, o programa lia o dado convertido. Com este programa obteve-se uma taxa de 60 k eventos por segundo, ou seja, três vezes a maior taxa máxima do sistema PDP-11 + MBD. Posteriormente o programa foi modificado para que

esperasse a interrupção de *hardware* gerada pelo LAM do ADC, que é associada pelo controlador, ao invés de ficar verificando a máscara de LAM continuamente. Este método tem a grande vantagem de deixar o programa e o computador livre para executar outras tarefas entre um dado e outro. Neste caso a taxa máxima obtida foi de 48 k eventos por segundo. Nestas duas medidas foram utilizados detectores de Germânio Hiperpuro (HPGe) com uma fonte de ^{152}Eu de 18,5 MBq.

2.11 Arquitetura do novo sistema de aquisição

Agora que possuíamos um controlador de bastidor CAMAC que atendia aos requisitos do LAL, se tornou necessária a definição da montagem do novo sistema de aquisição. Em princípio, decidiu-se pela limitação de medidas em paralelo, num mesmo bastidor CAMAC, o que diminui o conflito pela utilização do *Dataway* entre as variadas experiências. Assim ficou acertado que apenas duas medidas deveriam ser efetuadas num mesmo bastidor simultaneamente.

A primeira utilizaria o ADC114, conectado a um módulo histogramador HM413^[27] da Ortec que efetua a histogramação automática. Este sistema seria destinado a experiências com altas taxas de aquisição devido ao baixo tempo de conversão deste ADC que é fixo em 5 μ s e ao uso do HM413. Nesta configuração obteve-se uma taxa de 160 k eventos por segundo e esta taxa só é possível porque não há interferência do *Dataway*, do controlador, ou mesmo do computador durante a aquisição. A aquisição e histogramação é feita automaticamente pelos módulos que se comunicam através de um barramento FERABus^[27] que conecta os dois módulos, sendo que o CAMAC só é utilizado quando a medida termina e é necessário ler o histograma armazenado. Este sistema possui controle de tempo vivo, pois o

ADC114 é dotado internamente de um relógio de alta precisão, que efetua este controle usando dados do próprio ADC e do amplificador utilizado.

A segunda medida seria efetuada de forma similar às linhas do antigo sistema de aquisição, com um ADC NIM modelo 8701^[24] da Canberra, transferindo o dado convertido ao *Dataway* através de um módulo *Input Register* modelo 2322^[23] da BiRa. Neste caso o ADC utilizado seria do tipo *Wilkinson*, o que permite uma grande precisão no dado convertido. O controle de tempo vivo também seria disponível, mas utilizando um módulo 2203^[21] da BiRa que efetua o controle do tempo morto através do uso do sinal de ocupado do ADC. A figura 9 mostra as duas linhas de medida montadas no bastidor CAMAC.

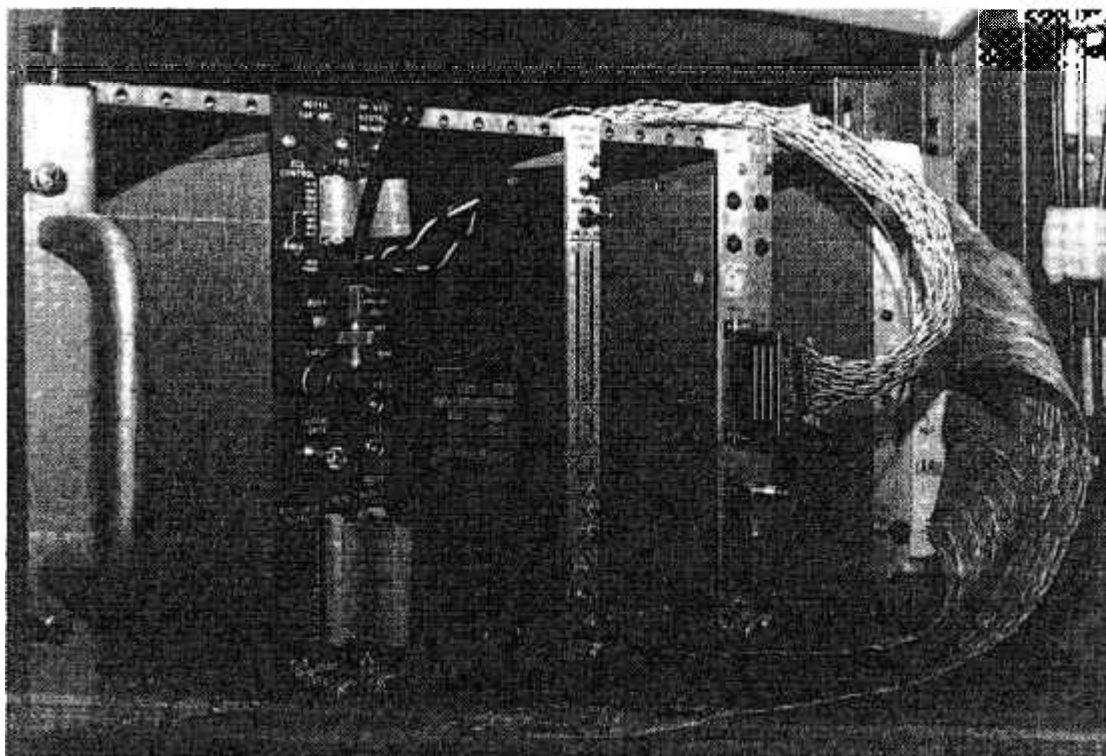


Figura 9 - Novo sistema de aquisição de dados do LAL. Vemos à esquerda o ADC114 conectado ao módulo histogramador HM413, ao centro vemos o *Dataway Display* M203, a seguir o *Input Register* e o relógio de tempo morto 2203. Na extrema direita o controlador de bastidor CAMAC-ISA.

Desta forma vemos que embora o número de entradas num mesmo bastidor tenha diminuído no novo sistema, o desempenho aumentou consideravelmente, visto que temos uma alta taxa de aquisição e um ótimo controle de tempo vivo, que é muito adequada para medidas de intensidade; e uma linha com maior precisão, proporcionada pelo ADC *Wilkinson*, para medidas de energia. Assim as necessidades de medidas em espectroscopia γ do laboratório são satisfeitas, tanto na necessidade de taxa de aquisição quanto na precisão das medidas, seja na intensidade, na energia ou mesmo no controle do tempo vivo.

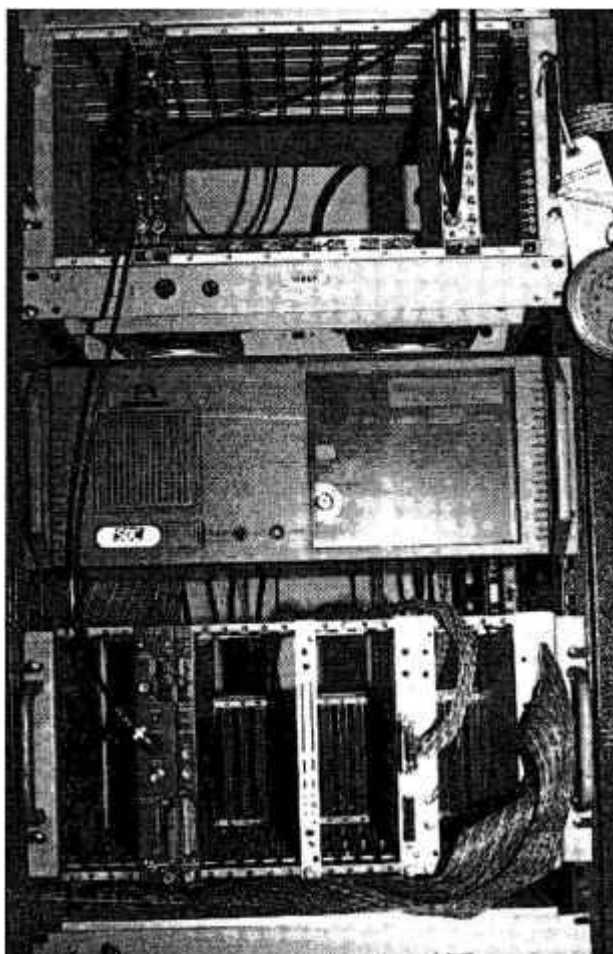


Figura 10 - Sistema de aquisição do LAL. No bastidor superior vemos à esquerda o amplificador utilizado com o ADC114 que se encontra à esquerda no bastidor inferior, CAMAC. Já à direita no bastidor superior está o ADC NIM que é conectado ao *Input Register*, que se encontra no bastidor inferior. Entre os dois bastidores se encontra o gabinete industrial do microcomputador que controla a aquisição de dados.

3 Aquisição Multiparamétrica

3.1 O que é uma aquisição multiparamétrica

No capítulo anterior descrevemos um sistema de aquisição de dados que efetua medidas chamadas de uniparamétricas, pois obtém um único dado ou parâmetro que é a energia do raio γ que atinge o detector. Medidas que envolvam vários detectores, trabalhando em coincidência ou não, são chamadas de multiparamétricas. Quando o LAL teve necessidade de realizar experimentos envolvendo a coincidência entre dois detectores, foi desenvolvido um módulo chamado Sincronizador Biparamétrico (SINC-BIP^[37]) que verificava a existência da coincidência entre os dois detectores e condicionava os sinais dos ADC's, de forma a serem lidos pelo sistema de aquisição. A figura 11 mostra o papel do SINC-BIP no sistema de aquisição.

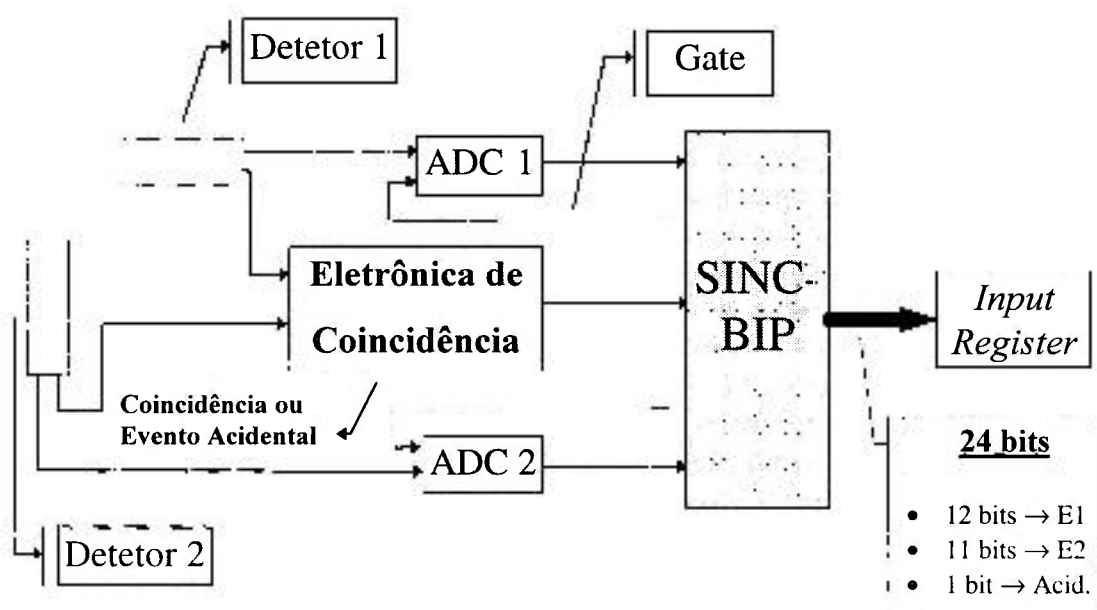


Figura 11 : Esquema da aquisição de dados em experiências com coincidência entre dois detectores utilizando o módulo SINC-BIP.

Como vemos na figura 11, este sistema fornecia dois parâmetros de energia e um *bit* que indicava se o evento era real ou uma coincidência acidental entre os dois detetores. Com o decorrer do tempo tornou-se necessário o desenvolvimento de um módulo que trabalhasse com mais de dois detetores, e que também fornecesse espectros de tempo das coincidências ocorridas e não só um *bit* de sinalização. Este módulo, chamado MULTI, deveria permitir a montagem de um aparato experimental que recebeu o nome de MULTIDETETOR, que consistiria num sistema de oito detetores que mediriam coincidências de fontes de raios X e γ .

3.2 O módulo MULTI

O controlador do sistema multidetetor, MULTI, é um módulo capaz de interfacear o sistema automático de aquisição de dados com até 8 detetores, verificando a existência de coincidência temporal entre pelo menos 2 deles.

Inicialmente pensou-se em construir “... *um módulo inteligente, com microprocessador interno.*”, mas dificuldades de tal empreita fizeram com que entre 1991 e 1992^[32], a decisão se desse por construí-lo no padrão CAMAC, porque desta maneira ele poderia ser facilmente ligado ao sistema de aquisição existente no LAL, o que permitiria o uso de todo o conhecimento de desenvolvimento de *software* existente no laboratório. De 1991 a 1995, efetuou-se a construção do protótipo deste módulo, e finalmente no início de 1996 deu-se início aos testes utilizando um microcomputador padrão IBM-PC conectado ao bastidor CAMAC através de uma interface GPIB.

3.3 Descrição do protótipo

Como já vimos o módulo MULTI foi projetado para funcionar num bastidor CAMAC. O protótipo do MULTI ocupa 8 posições deste bastidor. Embora internamente a eletrônica não utilize todo este espaço, o módulo acabou ficando com esta largura devido à grande quantidade de conectores para entrada e saída de sinais, como mostram as figuras 12 e 13.

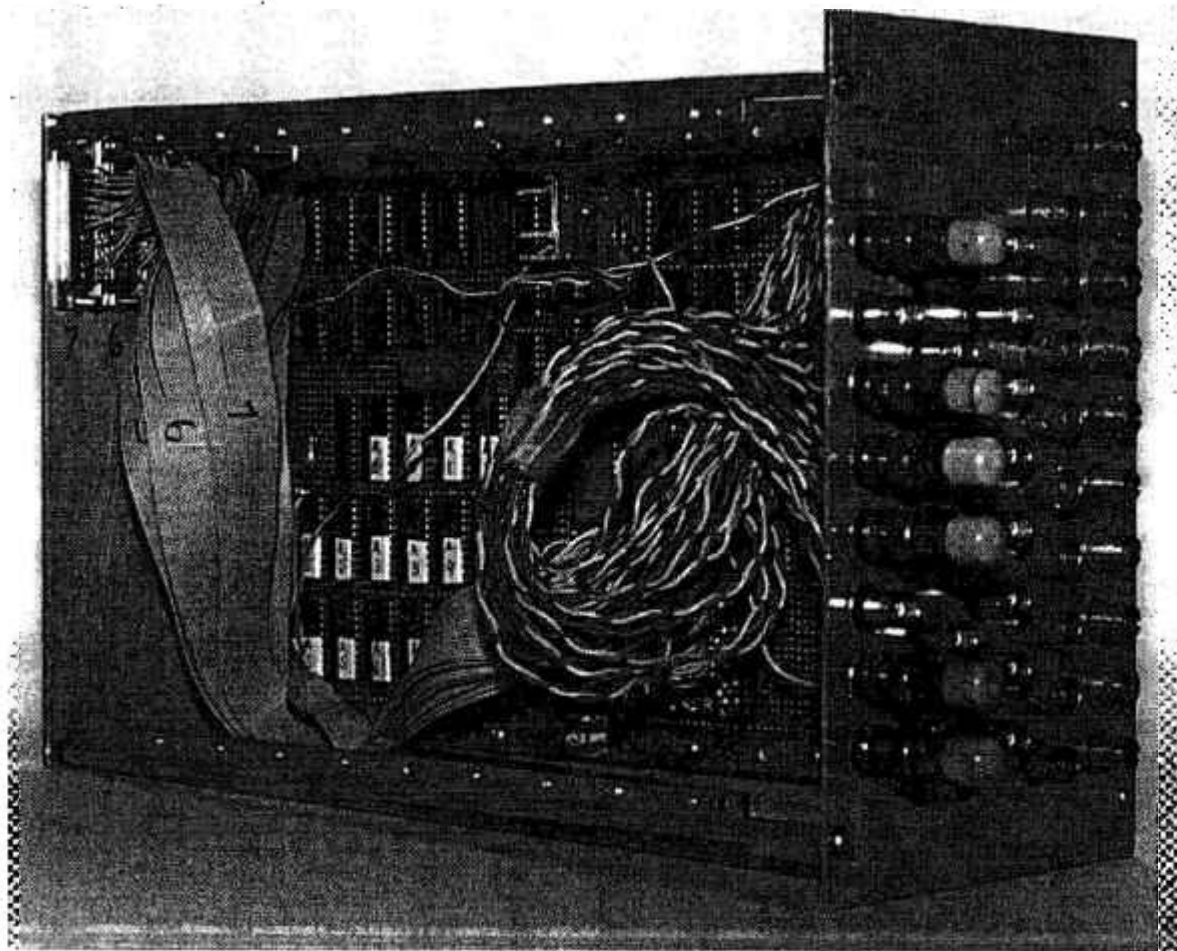


Figura 12 - Foto do protótipo do controlador MULTI. Nesta foto é possível observar a grande quantidade de conectores padrão BNC que se localizam na parte frontal do módulo. São também visíveis os cabos paralelos que trazem os sinais digitais dos ADC's para a placa do MULTI.

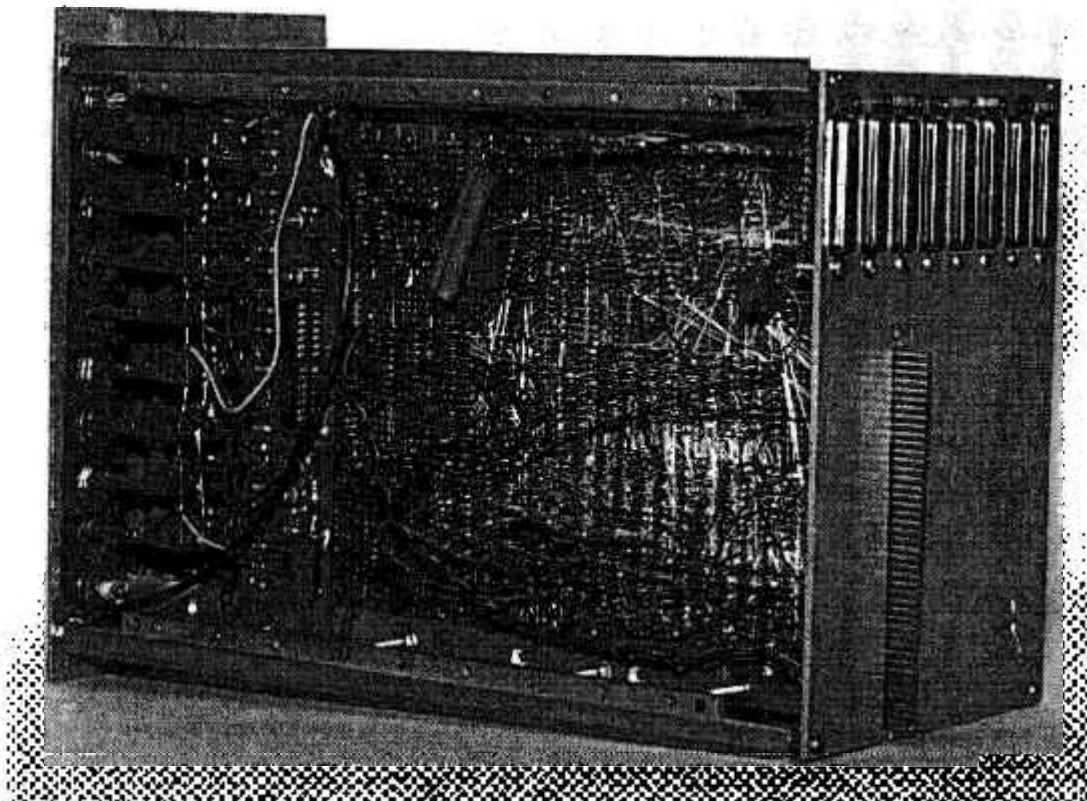


Figura 13 - Nesta foto do MULTI é possível visualizar o conector para ligação ao *Dataway* do bastidor CAMAC e os oito conectores padrão DB25 que interligam os ADC's ao MULTI. Também é possível observar o grande número de ligações elétricas necessárias pela técnica de construção utilizada, *wire-wrap*.

Os conectores padrão BNC frontais do módulo se destinam a: receber os sinais de ocupado (*BUSY*) e empilhamento (*INH*) dos amplificadores; receber os sinais de tempo provenientes dos *Constant Fraction Discriminator* (CFD)^[19]; fornecer os sinais de *start*, *clear* e *stop* para o *Time to Digital Converter* (TDC)^[22] e enviar *gate* para os ADC's. Este protótipo foi desenvolvido visando a utilização do ADC Ortec 800^[20], que é um módulo padrão *Nuclear Instrumentation Module* (NIM)^[13], o que acarretou a necessidade da introdução dos conectores DB25 traseiros para receber o sinal de energia convertido, para posterior leitura via CAMAC.

Como vemos na figura 13, este protótipo foi construído utilizando-se a técnica de *wire-wrap*, que consiste na utilização de uma placa padrão onde são dispostos os chips com as ligações entre eles feitas através de fios. A

vantagem desta técnica é que não é necessário o projeto de uma placa de circuito impresso, o que seria muito difícil se feito manualmente, devido à complexidade do módulo. A grande desvantagem é a fragilidade das ligações e a alta probabilidade de ocorrerem erros nas conexões.

3.4 Descrição do funcionamento do protótipo

Medidas de coincidência envolvem um grande número de sinais, que podem ser de energia, tempo ou mesmo sinais de controle como rejeição de empilhamento, ou sinal de ocupado de amplificadores, ou ADC's. O módulo MULTI tem por função agrupar todos estes sinais e verificar se existe uma coincidência em tempo dentro de uma faixa da ordem de centena de nanosegundos. (Veja na figura 15, o esquema de montagem do sistema multidetector.) Em caso afirmativo, ele deve verificar se existe coincidência em energia, ou seja, se existem ao menos dois amplificadores ocupados e sem rejeição por empilhamento. Após esta segunda etapa, o módulo gera os sinais de *gate* para os ADC's, para que estes iniciem a conversão. Se houver pelo menos dois ADC's convertendo ou com dado pronto, o MULTI considera que houve sucesso na verificação da coincidência e emite o sinal LAM no bastidor CAMAC, indicando a existência de um evento válido. Se não houver a coincidência entre pelo menos dois detetores, em qualquer uma das etapas, um sinal de *reset* é acionado e o módulo volta ao seu estado inicial, esperando uma nova coincidência. A figura 14 nos mostra simplificadamente o diagrama das verificações de coincidência até a geração do sinal de LAM no CAMAC, ou a rejeição do evento e reinicialização do módulo.

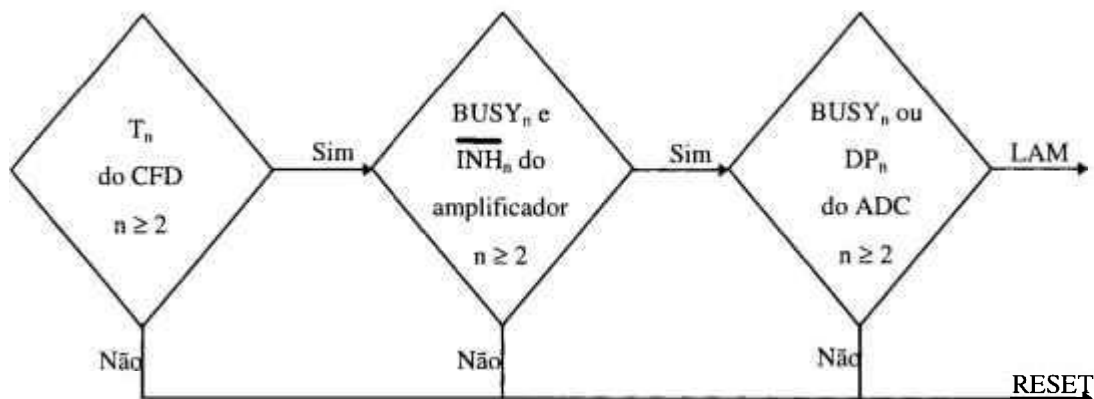


Figura 14 - Esquema de verificação de coincidência do MULTI, onde T_n é o sinal de tempo e DP_n é o sinal de dado pronto do n -ésimo detetor.

A principal dificuldade no projeto e construção deste módulo foi resolver o problema da definição do sinal que dispara todo o processo de verificação de coincidência em tempo. Nos sistemas em linha, como num acelerador, o sinal de disparo ou gatilho é dado pelo próprio pulso do acelerador, que inicializa todo processo de aquisição. Mas quando temos um sistema com muitos detetores semelhantes e não temos um gatilho externo, como o pulso de um acelerador, gostaríamos que qualquer detetor pudesse disparar o processo.

Este impasse foi resolvido de uma maneira engenhosa possibilitando a todos os detetores serem equivalentes. Basicamente o processo consiste em estender a duração do sinal de tempo que possui o formato NIM rápido através de mono-estáveis por aproximadamente 200 ns. Desta forma é possível efetuar operações lógicas sobre estes sinais e, assim, não só determinar qual sinal foi o primeiro a chegar como também saber se houve coincidência entre dois ou mais pulsos dentro da janela de tempo especificada.

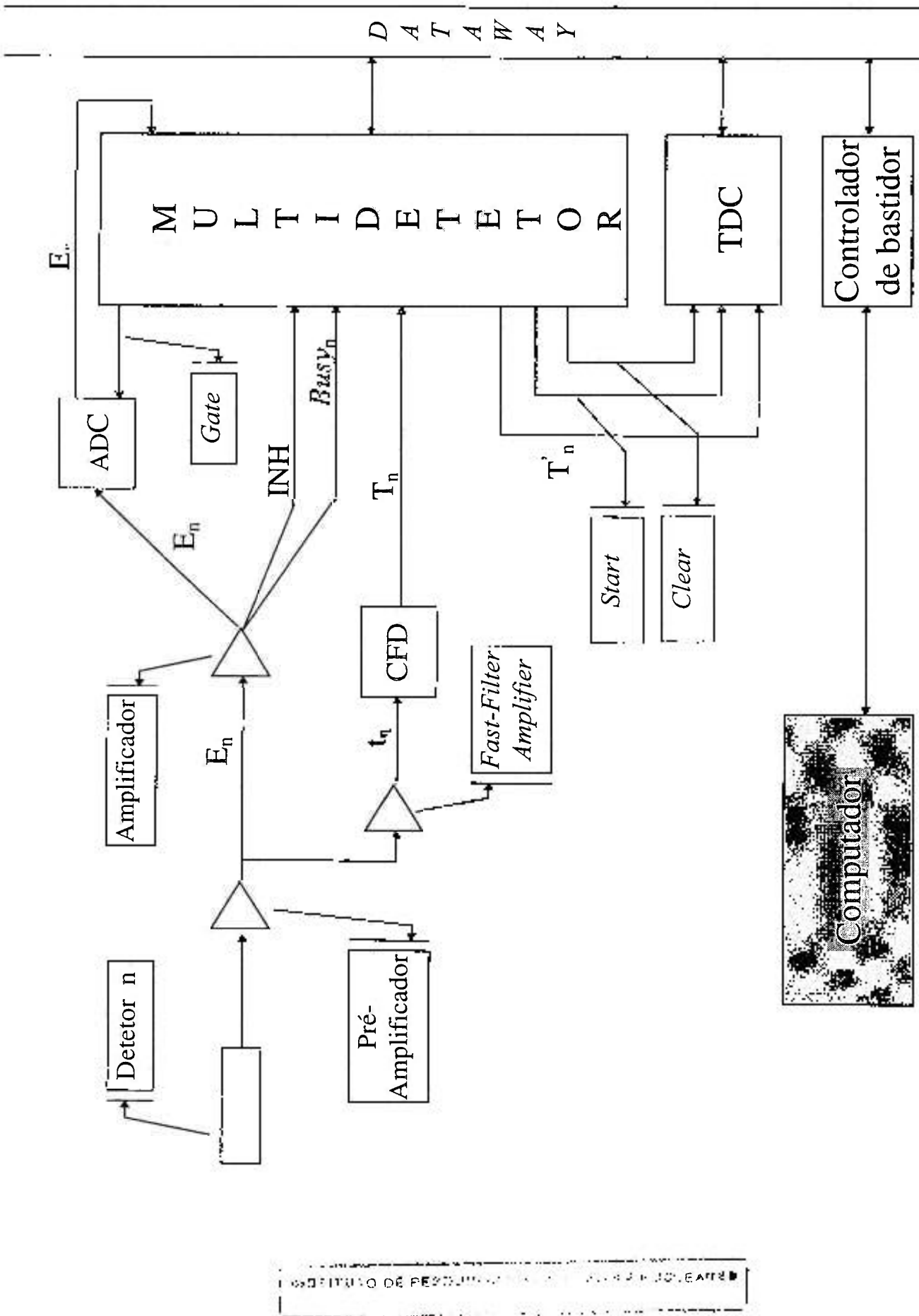


Figura 15 : Esquema da montagem do sistema

A figura 16 é um diagrama temporal que mostra como este processo se desenvolve. Desta figura vemos que a grande vantagem deste processo é que ele é genérico, pois não importa o número de detetores envolvidos. Além do mais, este método permite ainda que eventos de coincidência entre mais de dois detetores sejam obtidos.

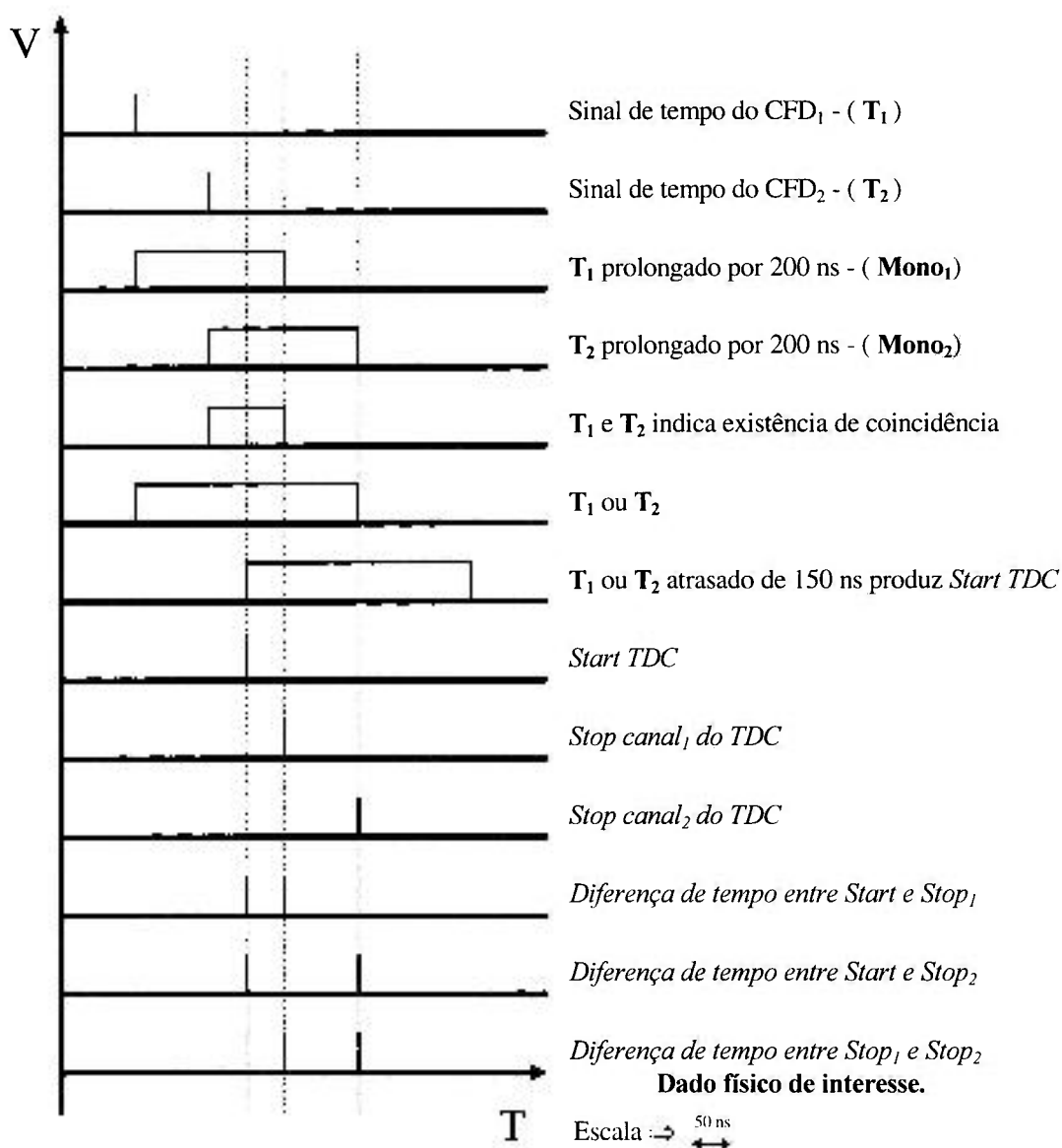


Figura 16 : Processo de coincidência temporal num intervalo de 200 ns, utilizado pelo protótipo do MULTI. (Os sinais NIM rápidos foram representados por um traço vertical, devido à sua curta duração.) Desta figura vemos que a existência do sinal (T_1 e T_2) sinaliza a existência de coincidência, permitindo que qualquer par ou conjunto de detetores valide o evento de uma forma completamente genérica.

Desta forma é eliminada a rigidez apresentada por outros sistemas de coincidência, onde o número de detetores e o tipo de coincidência é determinado de forma fixa pela montagem eletrônica do sistema^[2].

Finalmente, a diferença de tempo entre o sinal de *start* comum e de *stop* de cada detetor é convertido em digital utilizando-se o TDC, neste caso, um modelo 2228A^[22] da LeCroy.

3.5 Testes e Medida preliminar

Ao término da construção do protótipo do MULTI, inicialmente verificou-se o seu funcionamento como um módulo padrão CAMAC legítimo, pois a comunicação entre o módulo, o bastidor CAMAC, o controlador do bastidor e o computador, é essencial para qualquer teste posterior. Nesta etapa encontrou-se apenas dificuldades normais num projeto de desenvolvimento, como o adequação da largura e amplitude de sinais e a sua temporização, o que foi rapidamente resolvido com ajustes eletrônicos. Estabelecida a comunicação com o módulo, a lógica de coincidência foi o próximo passo.

Como vemos na figura 14, um evento de coincidência válido consiste em três etapas de verificação de coincidência: a coincidência rápida entre os sinais de tempo NIM rápidos, a coincidência lenta entre os amplificadores e a coincidência lenta entre os ADC's. A metodologia utilizada, no teste das várias etapas de verificação, consistiu em forçar coincidências em dois dos três circuitos enquanto se testava o terceiro. Isto era conseguido, para a coincidência rápida, colocando-se o mesmo sinal em duas entradas do MULTI. Na coincidência dos amplificadores deixou-se aberto os sinais de *BUSY* e foram colocados terminadores nos sinais de *INH*. Finalmente no caso dos ADC's gerou-se externamente o sinal de dado pronto.

Dos três níveis de coincidência, a rápida é a mais importante, pois dispara todo o processo. Portanto uma atenção toda especial foi dada a ela. Para isto, efetuou-se uma medida da linearidade da diferença de tempo entre o sinal de *start* que é comum a todos os canais do MULTI e o de *stop* que é individual (veja a figura 16). Para tanto, dividiu-se o sinal de um pulsador, sendo que um dos sinais foi atrasado através de caixas de atraso, possibilitando assim variar a diferença de tempo entre o *start* (sinal sem atraso) e o *stop* (sinal atrasado), em unidade de nanosegundo. Este levantamento é apresentado na figura 17 e vemos claramente que o MULTI apresentava uma acentuada não-linearidade nos sinais de *start* e *stop* quando a diferença de tempo entre os sinais era menor que 20 ns. A solução encontrada foi utilizar transistores de maior velocidade no circuito responsável pela geração destes sinais, e o resultado é apresentado na figura 18.

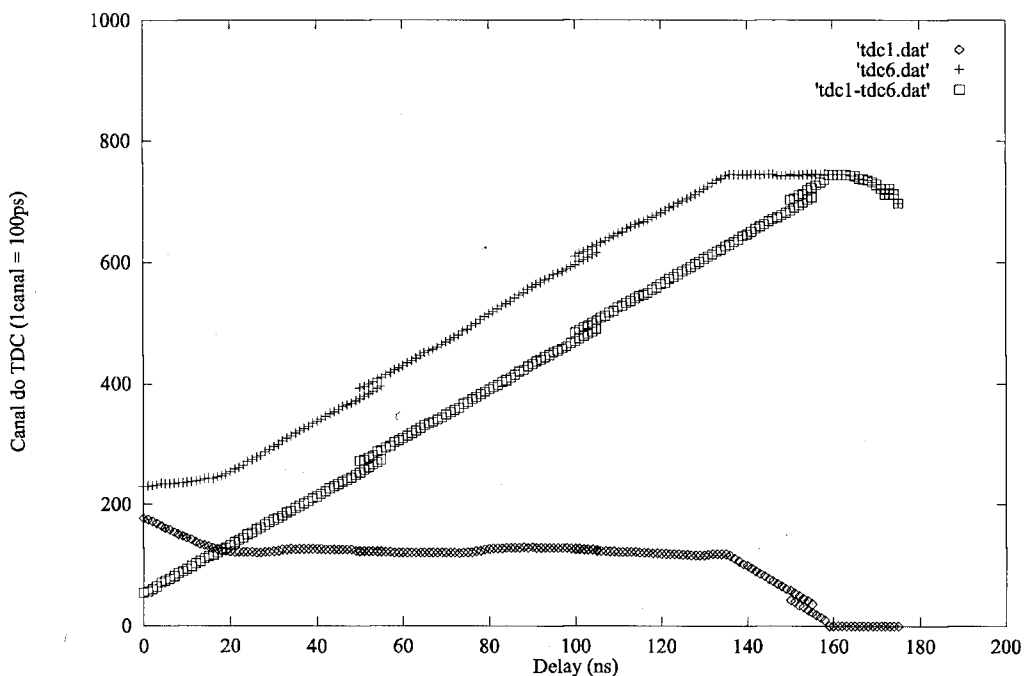


Figura 17 - Neste gráfico, onde "tdc1.dat" corresponde ao valor lido no canal 1 do TDC (*start*), "tdc6.dat" (*stop*) ao canal 6 do TDC e "tdc1-tdc6.dat" é a diferença entre os dois valores anteriores. Vemos claramente a não-linearidade nos sinais até 20ns, embora a diferença se mantenha linear em toda a região. Os dois degraus localizados em 50ns e 110ns se devem à mudança de faixa de atraso de sinal nas caixas de atraso.

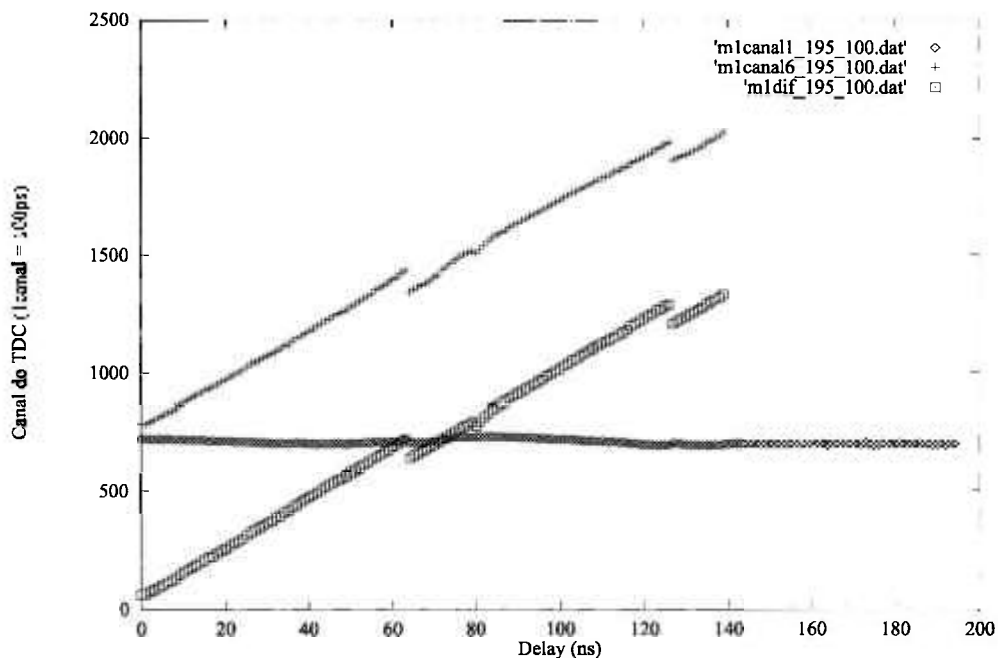


Figura 18 - Neste gráfico, onde “m1canal1_195_100.dat” corresponde ao valor lido no canal 1 do TDC, “m1canal6_195_100.dat” ao canal 6 do TDC e “m1dif_195_100.dat” é a diferença entre os dois valores anteriores. Vemos claramente que a não-linearidade nos sinais desapareceu e a diferença se manteve linear em toda a região. Os dois degraus localizados em 60ns e 125ns se devem à mudança da faixa de atraso de sinal nas caixas de atraso.

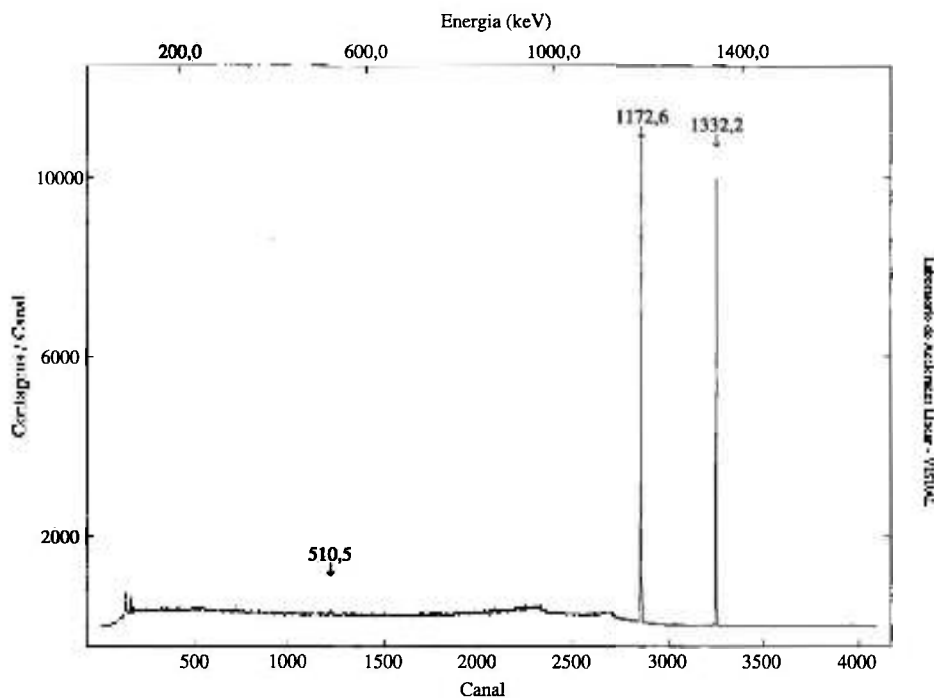


Figura 19 - Espectro de energia do detetor 1. Fonte de ^{60}Co .

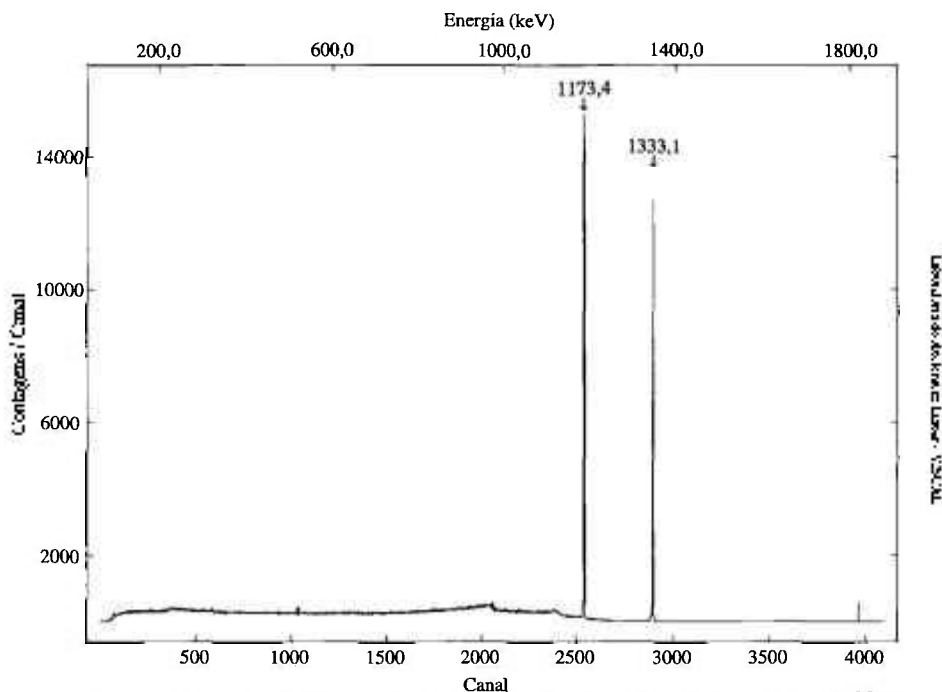


Figura 20 - Espectro de energia do detetor 2. Fonte de ^{60}Co .

Finalmente o último teste foi a execução de uma medida com uma fonte de calibração de ^{60}Co , utilizando-se 2 detetores de HPGe. As figuras 19 e 20 mostram os espectros de energia obtidos e as figuras 21 e 22 apresentam os espectros de tempo correspondentes.

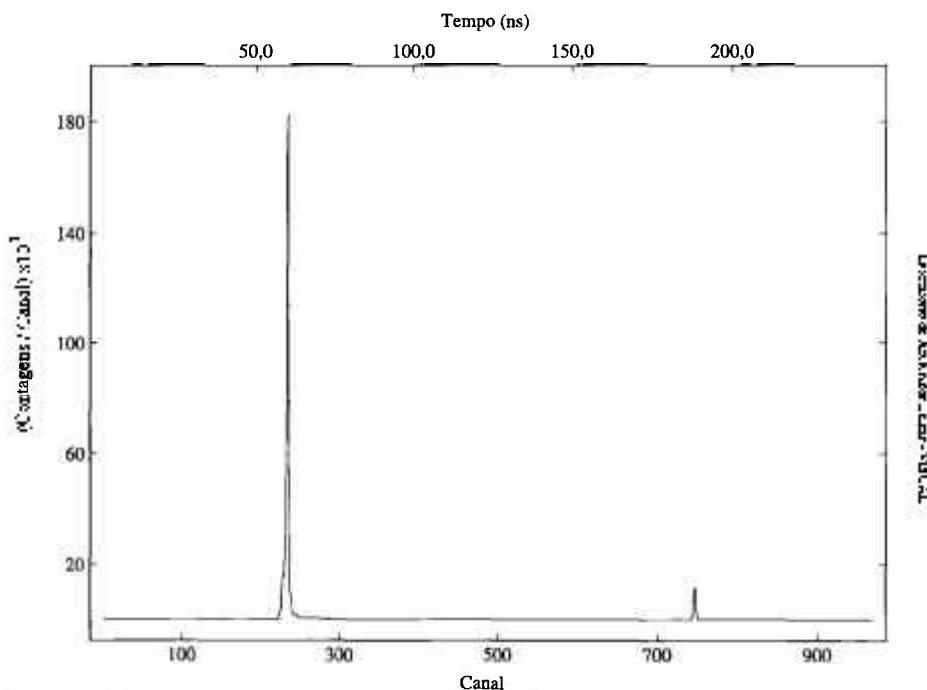


Figura 21 - Espectro de tempo do detetor 1. O pico à esquerda é denominado de auto-disparo e é formado pelo atraso introduzido pelo MULTI no sinal de tempo que disparou a verificação de coincidência (Veja a figura 16). Já o pequeno pico à direita é um artefato eletrônico que surge no final da janela de coincidência.

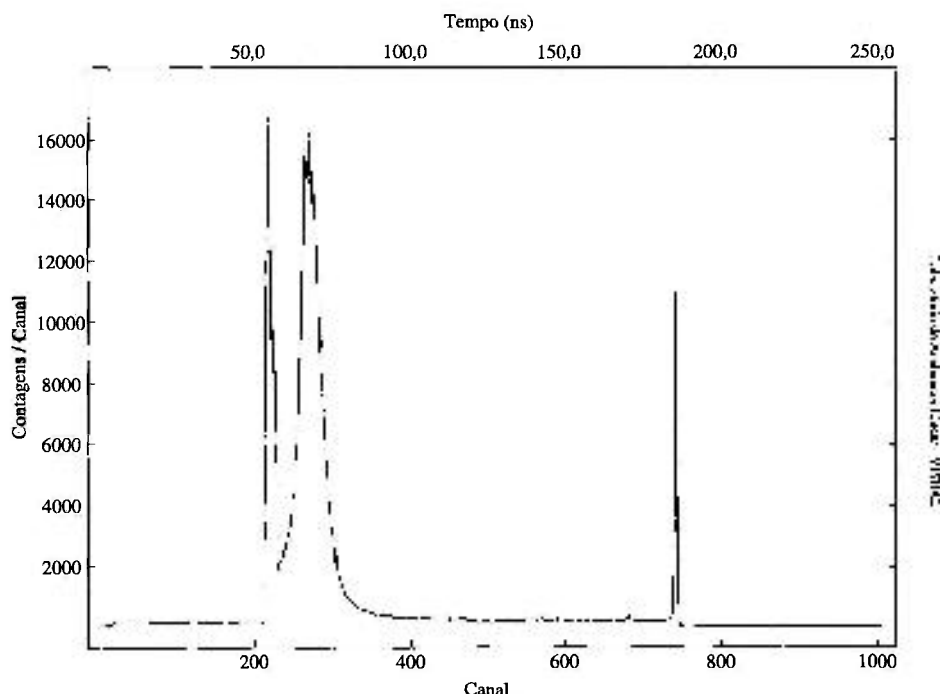


Figura 22 - Espectro de tempo do detetor 2. O pico largo entre o pico de auto-disparo à esquerda e o artefato eletrônico à direita é o pico de coincidência entre os dois detetores. A diferença de tempo de resposta dos dois detetores faz com que um deles sempre esteja atrasado em relação ao outro, e assim vemos o pico de coincidência apenas neste espectro.

Analisando os espectros de tempo apresentados nas figuras 21 e 22 observamos que há duas estruturas comuns a ambos. Uma é o pico bem estreito que se encontra em torno do canal 200 que é consequência do auto-disparo, ou seja, dos eventos em que o detetor em questão disparou o processo de verificação de coincidência. A outra estrutura é um pico à direita, perto do canal 700, que é um artefato eletrônico que surge quando a diferença de tempo entre os dois γ 's é comparável ao tempo de abertura da janela de coincidência. Finalmente o que diferencia os dois espectros é a estrutura larga que aparece apenas na figura 22. Esta estrutura só aparece neste espectro, pois o sinal deste detetor está atrasado em relação ao outro, devido à diferença do tempo de coleta de cargas entre os dois detetores.

Estes espectros foram montados simplesmente histogramando os dados de energia e tempo obtidos. Mas a aquisição de dados foi efetuada em modo

fila, ou seja, todos os eventos adquiridos foram armazenados seqüencialmente em um arquivo de computador. Isto permite a análise posterior dos dados pelo usuário, já que estes se encontram no seu estado bruto. Lendo o arquivo no modo fila, podemos calcular a diferença de tempo em cada evento e assim obter o espectro de diferença de tempo, que possui a informação física esperada. A figura 23 apresenta este espectro para a totalidade dos eventos obtidos, já na figura 24 fez-se uma seleção em energia, ou seja, apenas foram histogramados os eventos que possuíam as energia de 1173 e 1332 keV do ^{60}Co .

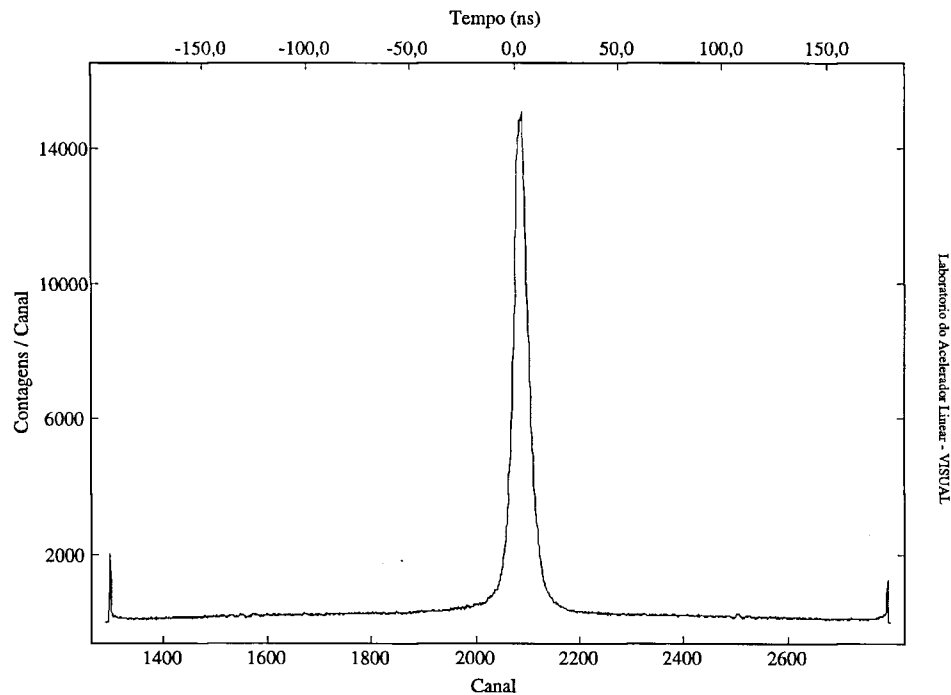


Figura 23 - Espectro construído com a diferença de tempo entre os sinais dos dois detetores.

Das figuras 21 e 22, vemos que os picos de auto-disparo simplesmente desaparecem restando apenas a informação física buscada que é o pico de tempo. Também é possível observar que ao efetuar a janela em energia obtemos uma resolução em tempo da ordem de 6 ns o que é um resultado muito bom.

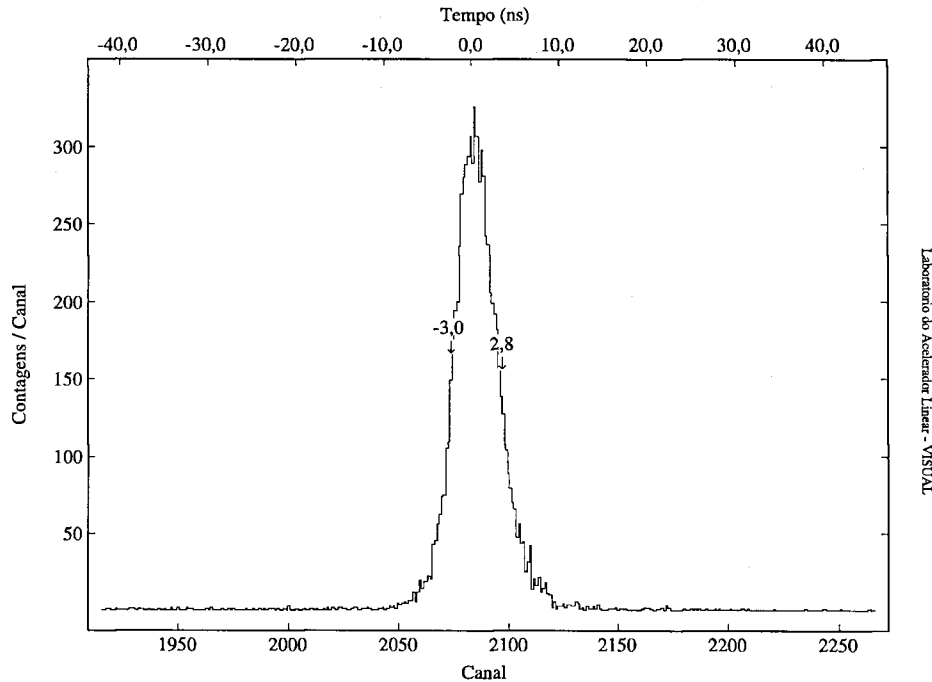


Figura 24 - Espectro construído com a diferença de tempo entre os sinais dos dois detetores, mas com seleção das energias dos gamas. Num detetor foi escolhida a região do pico de 1173 keV e no outro a região de 1332 keV. Vemos que foi possível obter uma resolução em tempo de apenas 6 ns.

3.6 Um defeito sutil

Próximo ao término dos testes do protótipo do MULTI, o sistema de aquisição de dados do LAL baseado no PDP-11 teve de ser desativado e junto o antigo sistema baseado no módulo SINC-BIP. Como havia algumas medidas de coincidência programadas, o protótipo do MULTI teve de ser disponibilizado para uso. Assim alguns pequenos programas foram desenvolvidos de forma que os usuários pudessem adquirir e visualizar os dados brutos como espectros de energia, tempo e diferença de tempo. Até então só haviam sido efetuadas medidas para teste do módulo e nenhuma análise física dos dados fora levada até o fim.

Uma das medidas que estavam programadas era uma medida dos núclídeos ^{232}U e ^{233}U , que se destinava ao trabalho de dissertação de

GUIMARÃES^[8]. Para analisar os seus dados GUIMARÃES desenvolveu o programa *Bidim*^[8], pois todos os programas de análise do laboratório se tornaram obsoletos diante da mudança do sistema de aquisição de dados e ao analisar os dados de uma medida de calibração com ^{133}Ba surgiu um resultado inesperado: havia uma coincidência entre um mesmo γ desta fonte com intensidade várias vezes maior que as coincidências reais nas quais este mesmo γ participava. A figura 25 apresenta uma visão da matriz formada pela coincidência entre dois detetores e vemos na escala de contagem que há uma estrutura bem grande quando o canal corresponde ao pico maior nos dois detetores. A figura correta seria apresentar apenas as estruturas correspondentes ao cruzamento do pico menor num detetor, com o pico maior no outro detetor, e vice-versa.

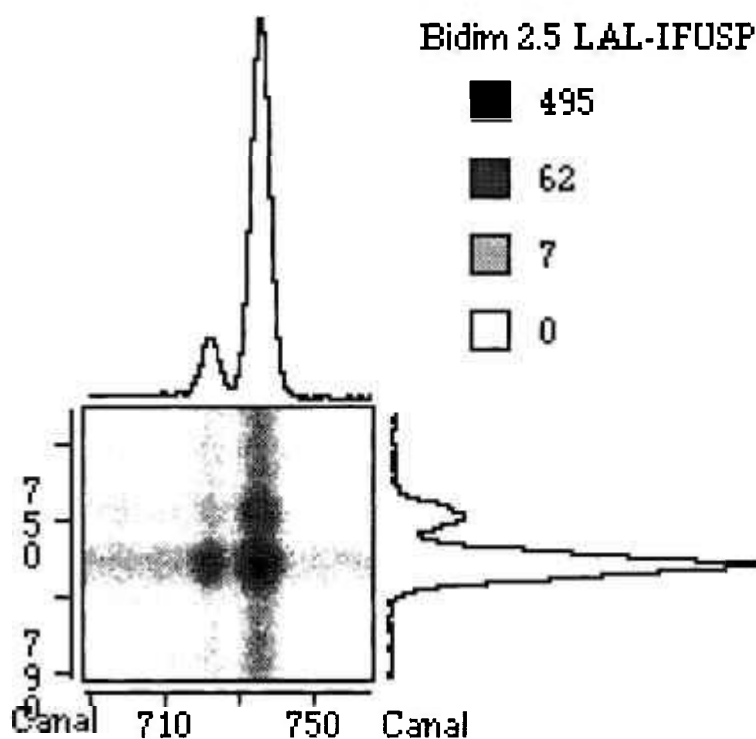


Figura 25 - Vista da matriz de coincidência entre dois detetores. A estrutura mais acentuada é formada pela coincidência entre os dois γ de maior intensidade, o que é fisicamente incoerente, pois esta coincidência é apenas acidental. As duas estruturas que relacionam os picos de menor intensidade de um detetor, com o de maior intensidade do outro, são as coincidências verdadeiras. Ou seja, o acidental está muito mais intenso que o verdadeiro.

Efetuada algumas medidas de controle, GUIMARÃES chegou a conclusão de que este fenômeno era devido ao fato de que o MULTI não estava reinicializando os ADC's após uma coincidência abortada. Assim se um dos ADC's havia conseguido converter o dado, ao ocorrer uma coincidência verdadeira o novo *gate* fornecido pelo MULTI era ignorado, pois já havia um dado pronto e não lido. Como a probabilidade do dado armazenado no evento anterior e no novo é maior para um sinal do pico de maior intensidade, então surge o pico de falsa coincidência.

O módulo foi modificado de tal forma a forçar a reinicialização dos ADC's após cada evento abortado, eliminando assim o dado convertido, e como consequência esta coincidência desapareceu como demonstra a figura 26.

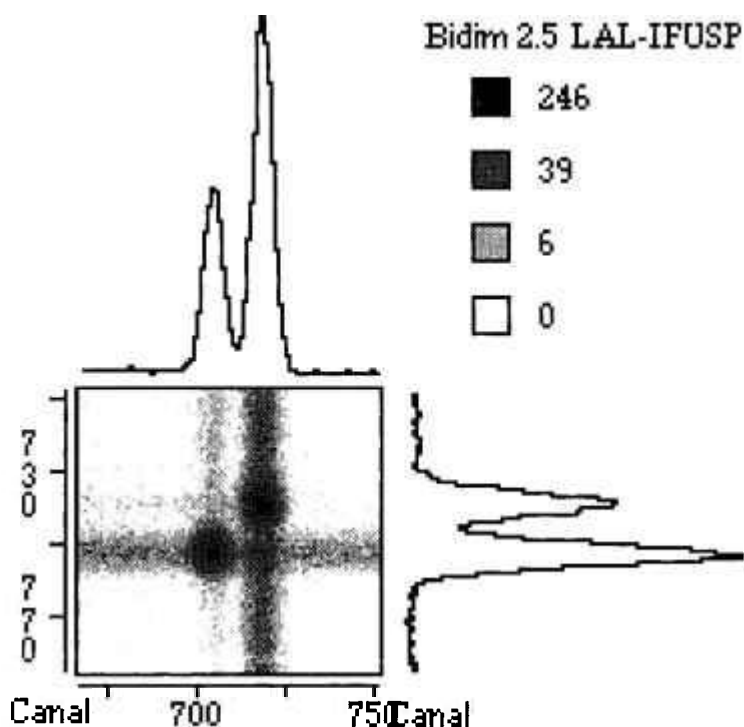


Figura 26 - Vista da matriz de coincidência após a alteração do módulo. Comparando com a figura 25 vemos como a estrutura que relacionava os dois γ de maior intensidade desapareceu e as estruturas verdadeiras se tornaram relativamente mais acentuadas.

Neste episódio ficou claro que os testes de um novo módulo não terminam quando o seu funcionamento eletrônico ou computacional está concluído. Vemos que a análise da coerência dos dados obtidos é muito importante, pois é ela que vai comprovar a eficácia e veracidade dos dados obtidos com o novo equipamento.

3.7 Estudos para a construção do módulo definitivo

O módulo até aqui descrito é o protótipo do controlador do MULTIDETETOR, que serviu para demonstrar a viabilidade do projeto e indicar possíveis modificações e correções para o projeto do módulo definitivo. Para tanto, além das medidas já efetuadas e analisadas, ainda era necessária a execução de uma medida visando o esclarecimento de uma dúvida sobre os espectros de tempo e a definição das principais mudanças que o módulo deveria receber.

3.7.1 Não-linearidade no sinal de tempo

Após a alteração efetuada no protótipo do MULTI, como descrito no item anterior, havia uma sutil estrutura no espectro de tempo apresentado pelo TDC. Nas figuras 27 e 28, vemos a região entre o pico de auto-start e o artefato no final do espectro, onde há várias oscilações no espectro que não possuem nenhum significado físico. Como as estruturas se repetiam em todas as entradas do TDC, a primeira suspeita foi de que ele era o responsável pelo aparecimento destas estruturas devido a algum tipo de não-linearidade no seu conversor analógico-digital.

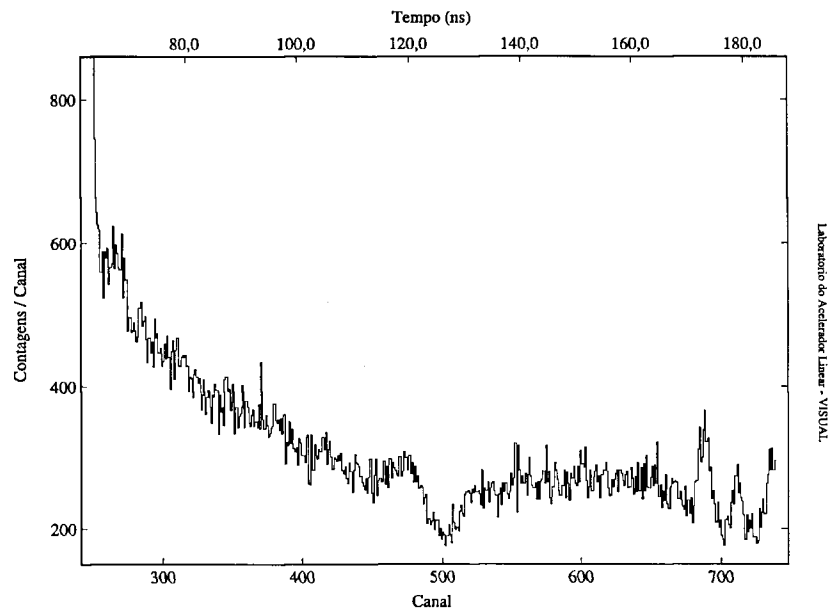


Figura 27 - Região compreendida entre o pico de *auto-start* e o artefato ao final do espectro correspondente ao canal 1 do TDC. (Vide figura 21) Observe o "vale" em torno do canal 500 e os pequenos picos perto do canal 700.

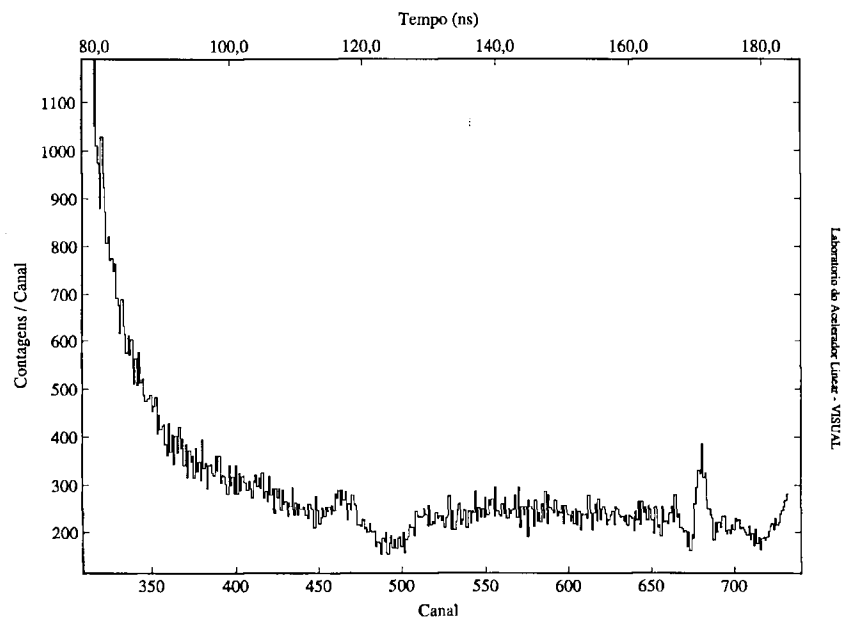


Figura 28 - Região compreendida entre o pico de *stop* e o artefato ao final do espectro correspondente ao canal 6 do TDC. (Vide figura 22) Observe como o "vale" em torno do canal 500 e os pequenos picos perto do canal 700 são muito parecidos aos apresentados na figura 27.

Para verificar esta hipótese uma medida foi proposta utilizando-se o módulo *Time-Calibrator* (TC) Ortec 462^[18]. Este equipamento tem a característica de gerar pulsos de *start* e *stop* uniformemente equidistantes dentro de uma faixa de tempo. Assim é possível varrer toda a faixa de tempo

que o TDC consegue analisar. A figura 29 mostra o esquema eletrônico da montagem para efetuar esta medida.

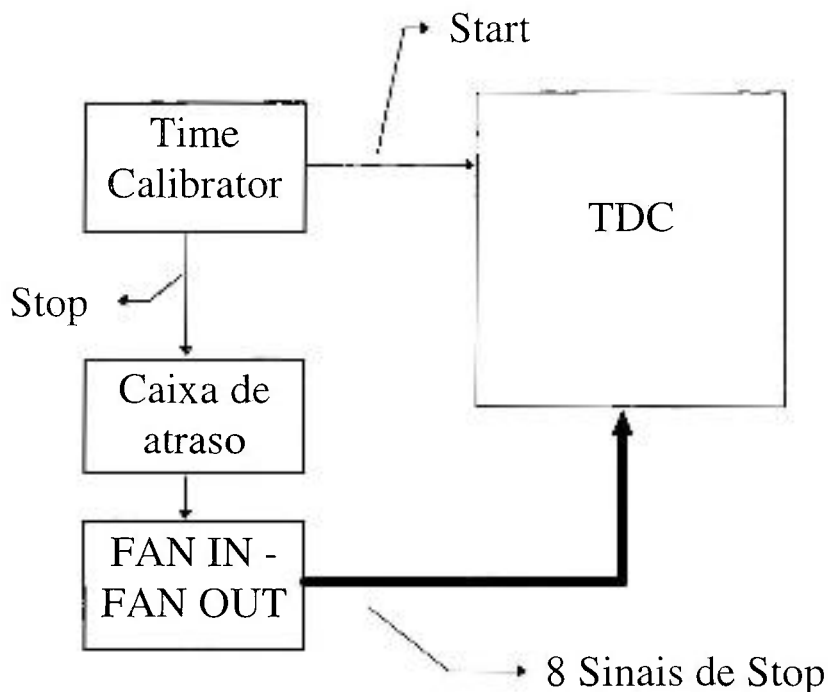


Figura 29 - Esquema da montagem eletrônica utilizada para a verificação da não-linearidade do TDC. A caixa de atraso serviu para fornecer uma variação mais fina, 1ns, da região varrida, já que o menor passo do TC é de 10ns. O *Fan In - Fan Out* foi utilizado apenas para reproduzir oito vezes o sinal e desta forma efetuar a medida em todas as oito entradas do TDC de uma única vez, com sinais idênticos.

Efetuada esta medida verificou-se que, na escala utilizada (250ps por canal) a não-linearidade máxima em todas as entradas do TDC não ultrapassava 0,3 de canal, ou seja aproximadamente 80ps. Desta forma não poderíamos atribuir as estruturas apresentadas nas figuras 27 e 28 à não-linearidade do TDC e sim ao próprio módulo MULTI. Assim foi montado um arranjo eletrônico, figura 30, muito parecido ao da figura 29, mas que possibilitaria efetuar a medida da não-linearidade introduzida pelo MULTI nos sinais de tempo.

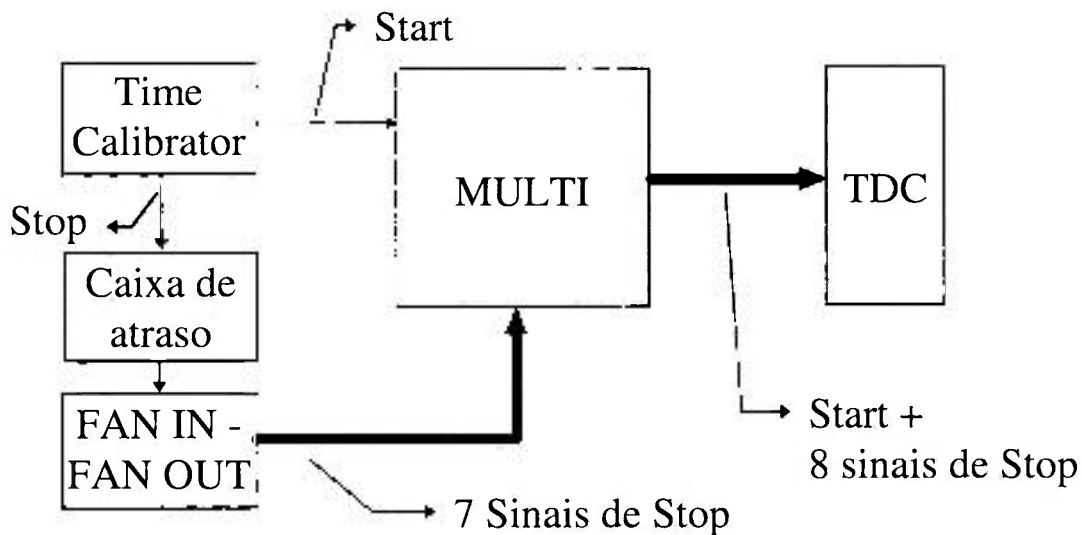


Figura 30 - Esquema da montagem eletrônica efetuada para a verificação da não-linearidade do MULTI. A única diferença desta montagem em relação à apresentada na figura 29 é a presença do MULTI.

A análise da medida efetuada com a montagem eletrônica descrita na figura 30, demonstrou que a não-linearidade introduzida pelo MULTI realmente é muito grande. A distorção do sinal, mostrada na figura 31, chegava a 10 canais do TDC, ou seja, 2,5 ns o que com certeza justificava as estruturas encontradas nos espectros de tempo.

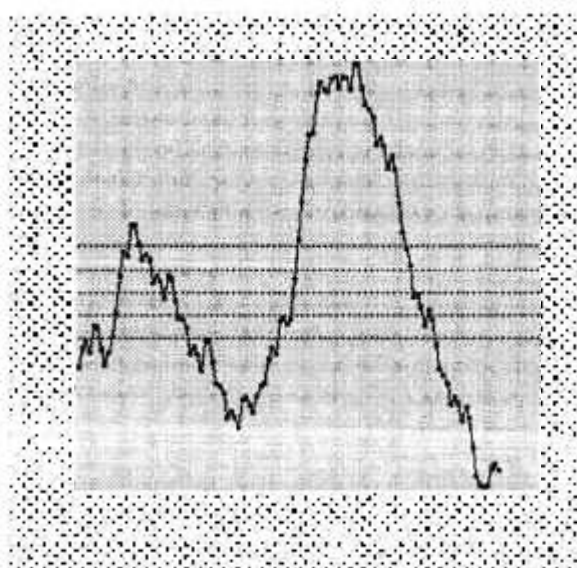


Figura 31 - Resíduo do ajuste de uma reta sobre os dados obtidos com a montagem mostrada na figura 30, para um canal do TDC. O desvio padrão utilizado é de um canal (250 ps) e a estrutura encontrada é a mesma em todos os canais do TDC.

O fato de todos os canais do MULTI apresentarem a mesma não-linearidade e estrutura nesta medida, deixou claro que o circuito responsável pela introdução deste efeito nos sinais deveria ser comum a todos. Verificando o esquema do MULTI, vê-se que o circuito que gera o sinal de *start* para o TDC é comum, sendo que os sinais de *stop* eram independentes entre si. Desta forma identificou-se algo que poderia influenciar todos os sinais ao mesmo tempo e da mesma forma, mas como cada um destes circuitos poderiam introduzir não-linearidades nos sinais, decidiu-se efetuar uma medida visando quantificar a influência dos circuitos de *start* e do *stop* em cada sinal do MULTI. Para tanto, montou-se o arranjo eletrônico apresentado na figura 32, onde, utilizando dois TDC's é possível verificar independentemente a não-linearidade introduzida pelo *start* e pelos *stop's* do MULTI.

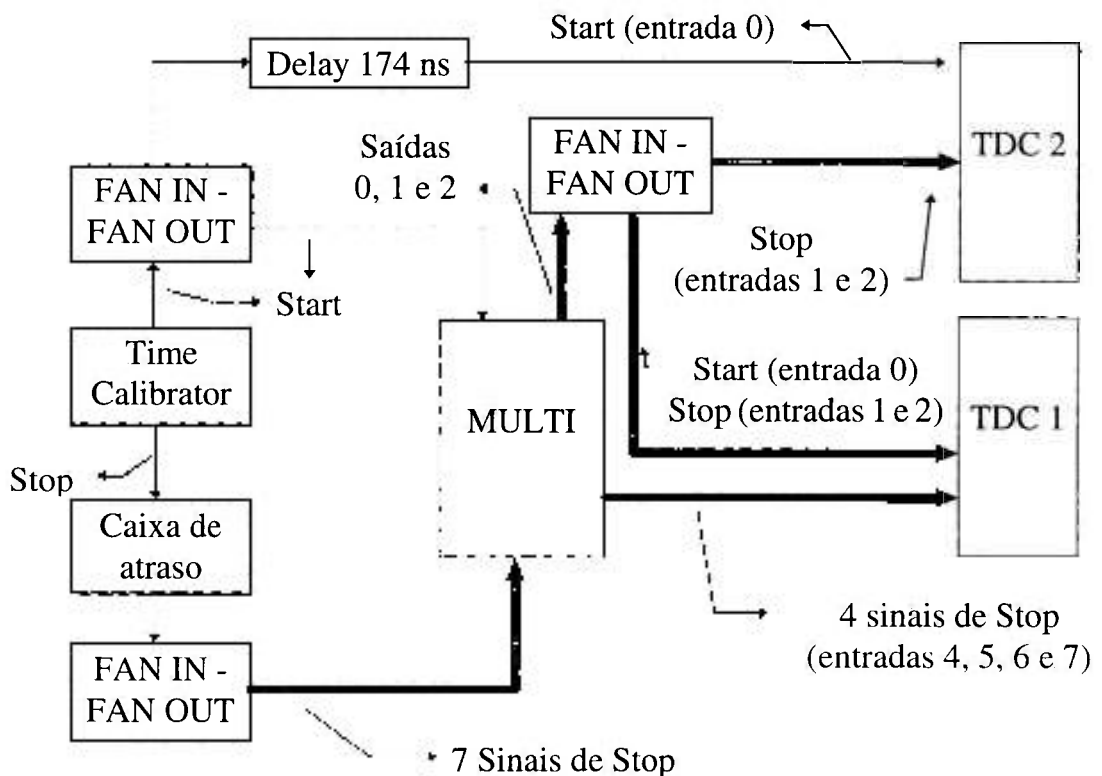


Figura 32 - Montagem eletrônica para a verificação da não-linearidade introduzida pelo *start* e pelos *stop's* do MULTI. A comparação entre as entradas 1 e 2 dos TDC's fornece a informação esperada.

O sinal de *start* do TDC 2 é independente do MULTI, e os sinais de *stop* dos canais 1 e 2 são iguais em ambos os TDC's, então é possível verificar a contribuição de cada circuito para a não-linearidade. Se no TDC 2 a estrutura desaparecesse, então o sinal de *start* gerado pelo MULTI seria o responsável pela não-linearidade e poderíamos quantificar a sua contribuição comparando com o resultado do TDC 1.

9

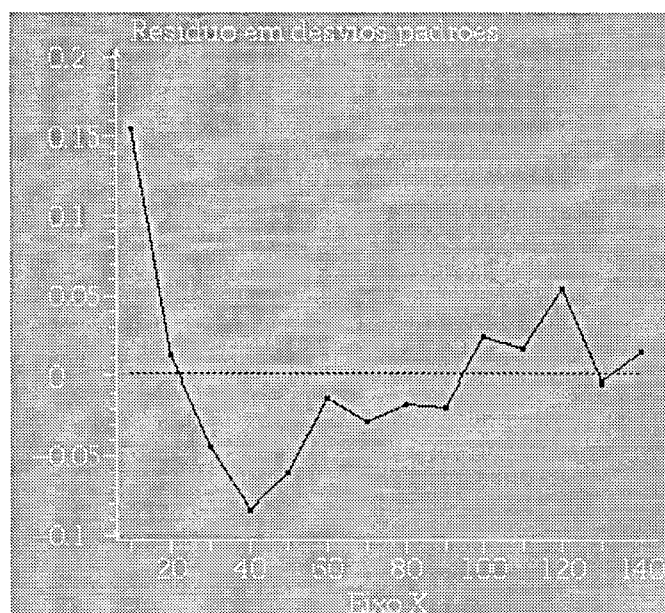


Figura 33 - Resíduo do ajuste de uma reta efetuado sobre os dados obtidos com a montagem mostrada na figura 31, para um canal do TDC 2. O desvio padrão utilizado é de um canal (250 ps) e a estrutura encontrada é a mesma em todos os canais do TDC 2. (O eixo X representa os sinais gerados pelo *Time-Calibrator* em ns)

Vemos na figura 33 que a não-linearidade apresentada pelo canal 1 do TDC 2 é da mesma ordem do próprio TDC, ou seja, toda a estrutura que vimos na figura 31 e que se repetiu no canal 1 do TDC 1 é introduzida pelo circuito do MULTI que gera o sinal de *start* para o TDC. Desta forma ficou claro que este circuito deveria sofrer uma modificação radical no projeto do módulo MULTI definitivo.

3.7.2 Outras modificações necessárias

Além de corrigir os problemas encontrados no protótipo do MULTI, a construção de um módulo definitivo deveria introduzir características advindas da experiência de uso do protótipo.

O tempo despendido pelo computador na leitura dos dados acrescenta um tempo morto ao sistema. Este tempo deve ser minimizado, já que ler um dado para descartá-lo logo em seguida, devido a alguma inconsistência, é um tempo morto desnecessário. Desta forma, pensou-se em acrescentar ao módulo algumas verificações suplementares à existência de coincidência, com vista a diminuir o tempo gasto com a rejeição de eventos indesejáveis.

O primeiro circuito proposto para ser introduzido, foi a verificação da coerência entre quais detetores participavam em cada etapa da verificação de coincidência. No protótipo, se os detetores 2 e 4 validassem a verificação de coincidência em tempo, os detetores 2 e 5 validassem os amplificadores e os detetores 3 e 5 os ADC's, o dado seria aceito, embora fosse uma coincidência falsa. É claro que esta situação possui uma probabilidade baixa de ocorrer e ela poderá ser descartada numa análise simples, pois o valor de tempo na entrada 5 do TDC seria um *overflow*, mas esta análise seria feita pelo programa de análise e um tempo precioso teria sido perdido na aquisição deste dado. Assim, o módulo definitivo deveria verificar se os mesmos detetores validam todas as etapas de coincidência, descartando eventos desta natureza.

Visando aumentar a versatilidade do MULTI e também eliminar tempo morto com dados indesejados, duas máscaras deveriam ser introduzidas. Cada máscara seria composta por um byte onde cada bit representaria um certo detetor do sistema. A primeira máscara seria a de coincidência obrigatória, ou seja, se um bit desta máscara estiver ligado o detetor correspondente deve

participar obrigatoriamente da coincidência, se não o evento é descartado. A outra máscara seria a de rejeição obrigatória, onde o sinal de um detetor que tivesse o seu bit correspondente ligado provocaria a rejeição do evento. Desta forma é possível aumentar muito a versatilidade do módulo, pois com a rejeição obrigatória ele poderia ser usado com supressores Compton, por exemplo. Já a máscara de coincidência obrigatória poderia aumentar muito o número de eventos de coincidência γ -X, pois poderíamos aceitar apenas os eventos onde os detetores de Raio-X participam, e descartar as coincidências entre os detetores γ .

Outra modificação que se tornou obrigatória foi a troca dos ADC's utilizados pelo sistema. O protótipo foi projetado para trabalhar com o ADC800 NIM da Ortec, pois já haviam alguns destes ADC's no laboratório. Mas, com o fim da produção deste ADC, optou-se por trocar os ADC's 800 por dois ADC's quádruplos CAMAC modelo 413A^[26] da Ortec. Com esta troca, o novo módulo não necessitaria dos conectores traseiros, e portanto ele ocuparia menos espaço no bastidor CAMAC. Esta troca leva, porém, a uma perda que não é pequena. Como não seria mais possível obter dos ADC's os sinais de *BUSY* e Dado Pronto (vide figura 14), o último nível de verificação de coincidência se tornaria impraticável. Assim o MULTI passaria a ter apenas dois níveis de verificação de coincidência, sendo cada nível muito mais complexo devido às verificações adicionais acrescentadas.

Finalmente, chegou-se à conclusão que a inserção de um número maior de LED's de estado no módulo, possibilitaria obter com maior rapidez uma avaliação sobre o estado do próprio. No protótipo, como podemos ver na figura 12, só haviam três LED's que indicavam LAM, existência de coincidência rápida e acesso ao módulo pelo CAMAC. O módulo definitivo deveria apresentar os mesmos sinais do protótipo acrescidos dos sinais de alimentação de tensão, *Clear* e mostrar em forma de LED's acesos quais

detetores participaram da última coincidência validada. Desta forma seria possível obter visualmente uma noção do estado de funcionamento do módulo.

3.8 Construção do MULTI definitivo

Para a construção do módulo MULTI definitivo, foram utilizadas as mesmas técnicas avançadas de projeto e roteamento de circuitos usados no projeto do controlador CAMAC. Assim foi possível projetar uma placa de circuito impresso com a complexidade deste módulo, podendo descartar o uso da técnica de *wire-wrap* utilizada na construção do protótipo.

Visando diminuir o número de estações ocupado pelo MULTI no bastidor CAMAC, optou-se pela utilização de conectores padrão LEMO ao invés do BNC do protótipo. Nas figuras 34 e 35 vemos como ficou o módulo MULTI definitivo.

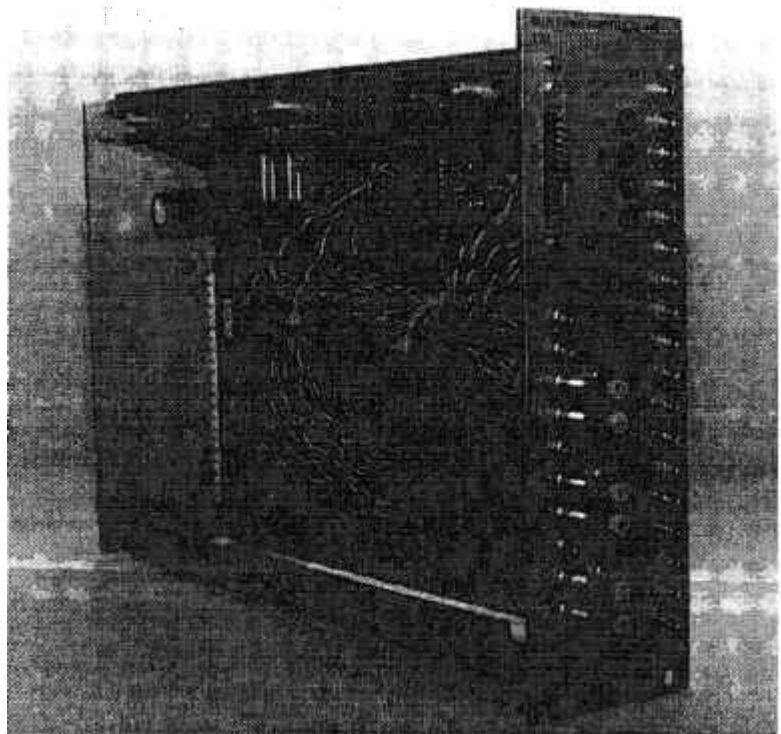


Figura 34 - Foto em ângulo do módulo MULTI definitivo. São visíveis os conectores padrão LEMO frontais e os LED's indicativos de estado do módulo. Internamente vêem-se os cabos dos conectores e a placa de circuito impresso.

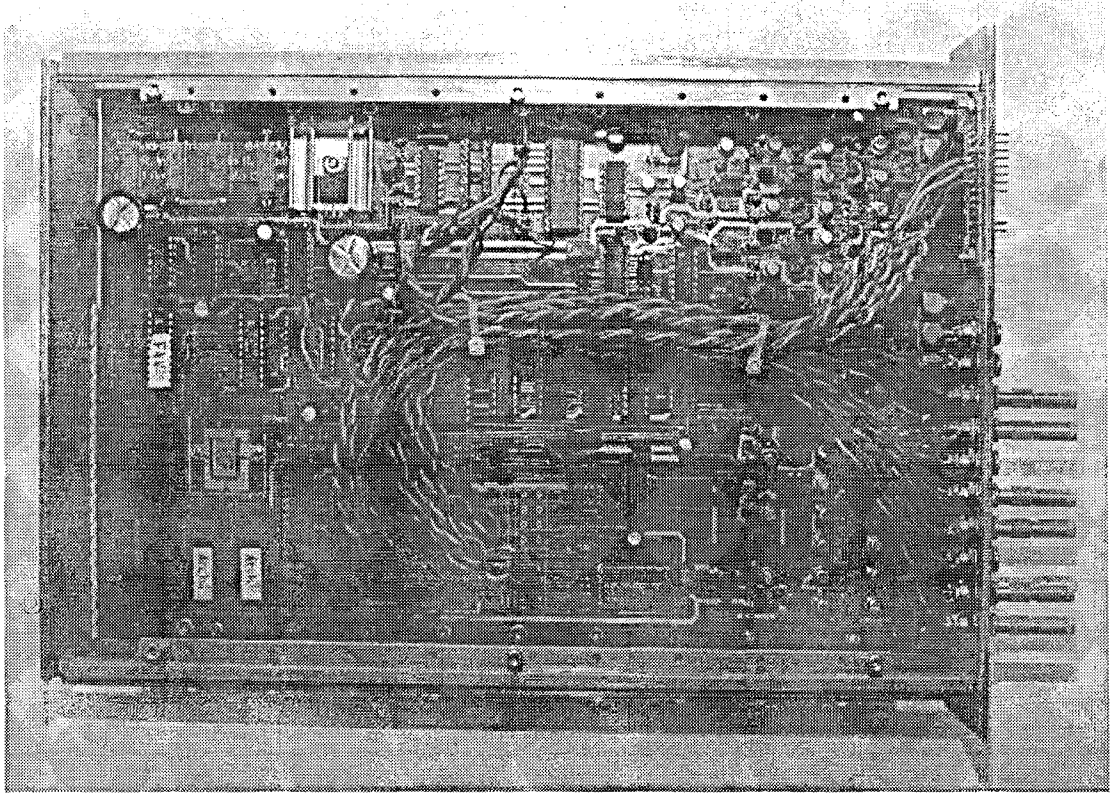


Figura 35 - Nesta foto lateral do MULTI definitivo é visível a placa de circuito impresso gerada pelo programa roteador. Na parte direita da placa vemos os transistores utilizados no tratamento dos sinais analógicos de tempo, e no resto da placa estão os chips responsáveis pela parte digital do módulo.

Na construção deste módulo, além de modernas técnicas de projeto e roteamento, foram utilizados componentes que nunca haviam sido utilizados em projetos do LAL como matrizes lógicas programáveis e transistores de alta frequência. Como se vê na figura 36, além da grande diferença de tamanho entre o módulo definitivo e o protótipo, alguns conectores dianteiros do protótipo desapareceram. O ADC413A pode trabalhar com o barramento FERABus para comunicação com outros módulos, então utilizou-se esta facilidade para conectar o MULTI a este ADC e enviar-lhe os sinais de *gate* e *clear*. Desta forma substituiu-se oito conectores frontais por um cabo. O uso de conectores LEMO também contribuiu para a diminuição da largura do módulo possibilitando colocar em uma única coluna todos os sinais de tempo. Assim, ao invés de ocupar oito posições no bastidor CAMAC, o MULTI definitivo ocupa apenas quatro.

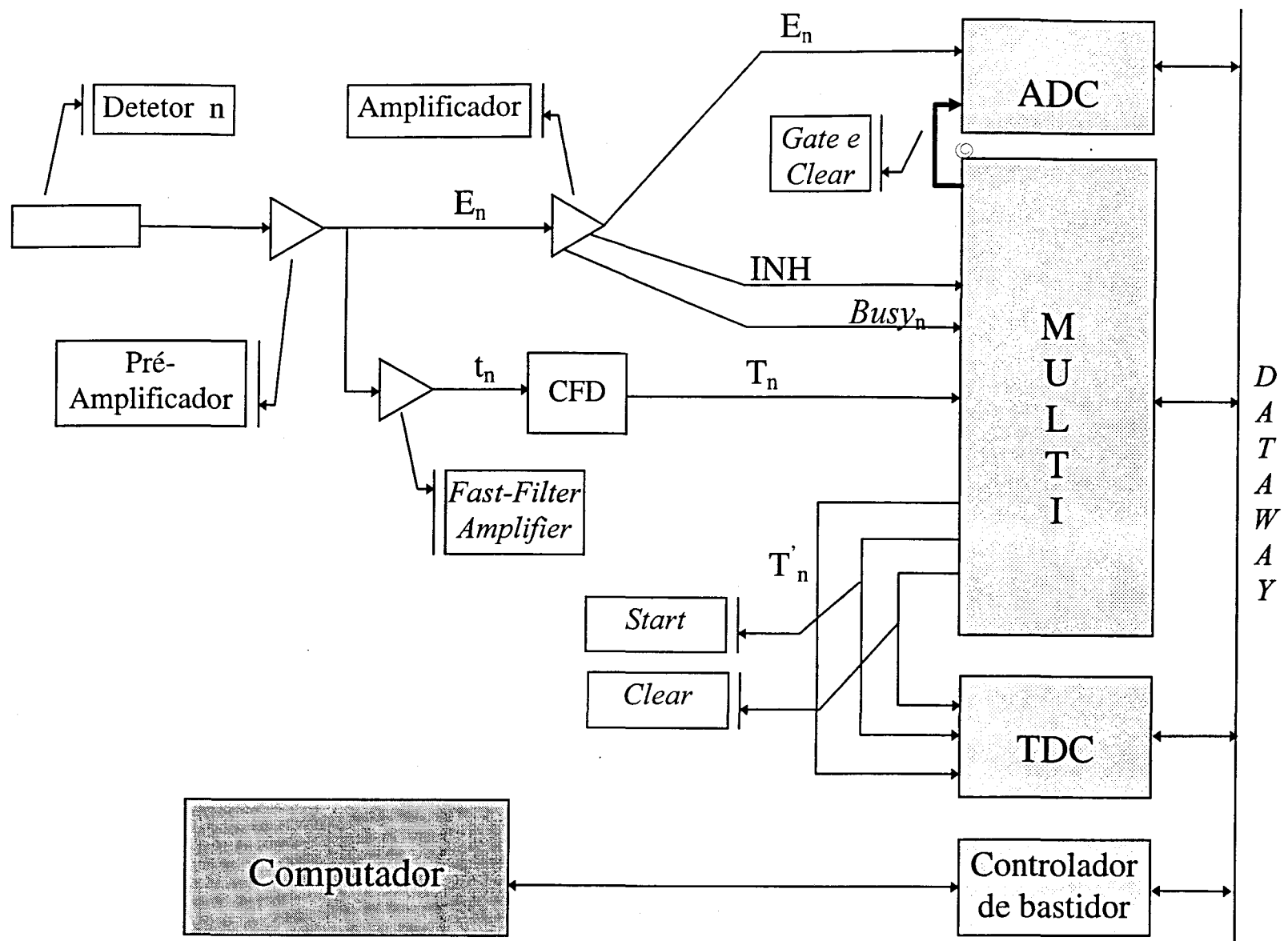


Figura 37 : Esquema da montagem do sistema Multidetetor

Outra importante modificação foi efetuada no circuito de verificação de coincidência em tempo. No protótipo esta verificação era efetuada inteiramente de forma analógica e agora, foi introduzida uma matriz lógica que verifica a existência da coincidência através de uma equação matemática. Esta modificação eliminou vários componentes e uniformizou o caminho seguido pelos sinais, procurando promover não só uma melhora na não-linearidade do sinal de *start* como também diminuir a largura do pico de auto-disparo nos espectros de tempo.

3.9 Testes preliminares

Como no protótipo os primeiros testes efetuados no módulo definitivo foram a energização e verificação das tensões nos vários circuitos que integram o módulo, foi necessário ajustar a largura e amplitude de alguns sinais analógicos que estavam fora do padrão especificado.

Utilizando pequenos programas de computador, o módulo foi endereçado e respondeu a contento a comandos simples como *clear*. O próximo passo seria introduzir no módulo sinais de forma a gerar coincidências e assim testar os circuitos de coincidência rápida (tempo) e lenta (energia), como também dos vários circuitos de verificação de consistência das coincidências ocorridas. A figura 38 apresenta a montagem experimental utilizada para o teste dos circuitos de coincidência.

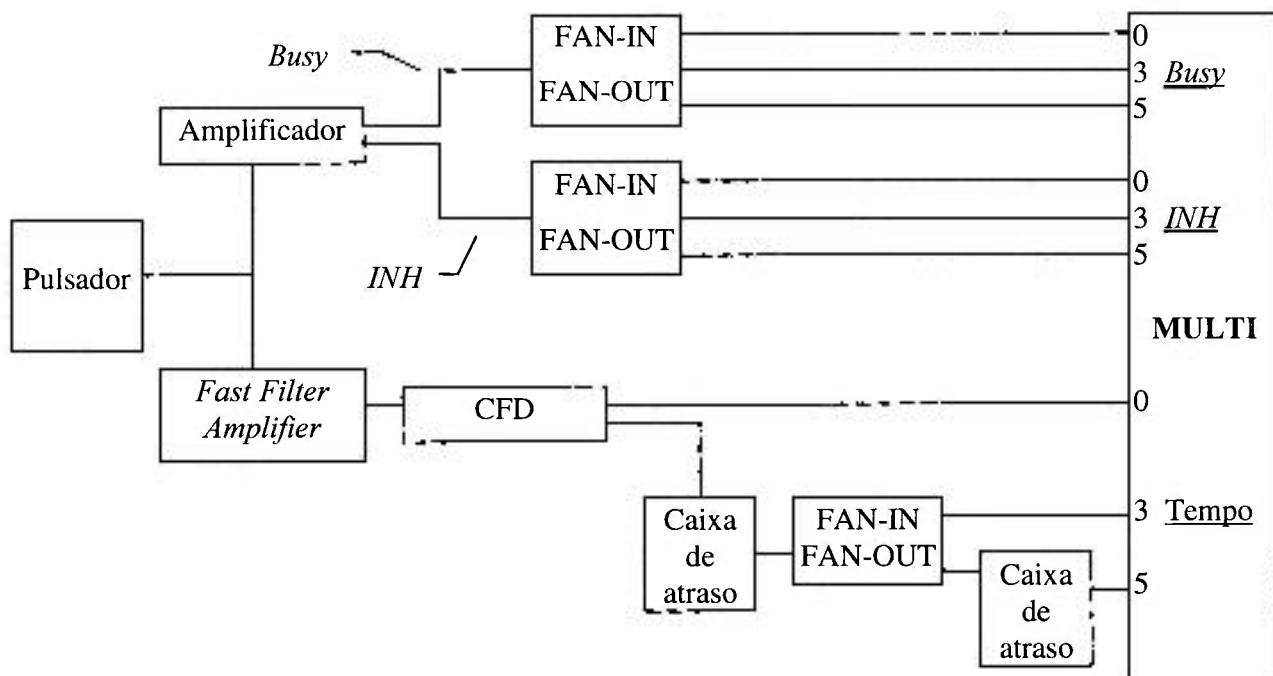


Figura 38 - Esquema da montagem experimental para o teste dos circuitos de coincidência do MULTI. Vemos claramente que esta montagem permite o teste simultâneo de até três canais com um único sinal de entrada.

Ao se tentar verificar coincidência com o aparato apresentado na figura 38 não se obteve nenhuma coincidência. Após uma longa análise, chegou-se à conclusão que as impedâncias dos circuitos de entrada do sinal analógico de tempo do MULTI e a saída dos CFD's eram muito diferentes, e isto não permitia que o circuito de coincidência funcionasse. Um longo tempo foi gasto na modificação dos circuitos de entrada dos sinais de tempo do MULTI, mas finalmente foi possível verificar a existência de coincidência. Nesta etapa dos testes ficou claro que o uso de transistores de 2 GHz na montagem dos circuitos provocava a emissão de radiofrequência que influenciava os outros circuitos. Assim, foi necessário introduzir uma placa de blindagem sobre os circuitos analógicos de tempo, o que resolveu este inconveniente. A figura 39 apresenta a blindagem instalada.

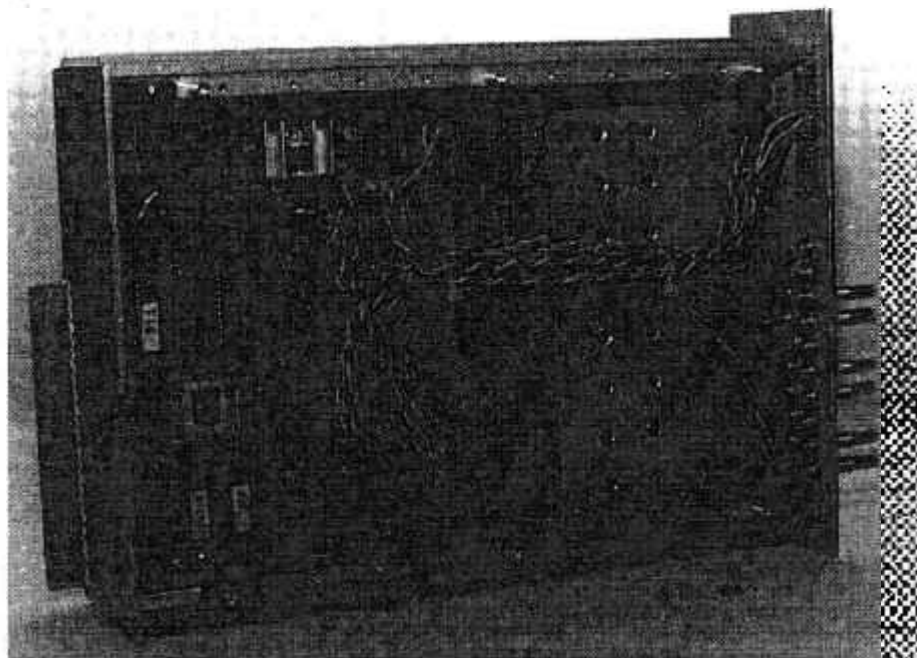


Figura 39 - Foto do lado direito do módulo MULTI, mostrando claramente a blindagem introduzida sobre o circuito analógico de tempo para evitar a emissão e recepção de radiofrequência. Os furos na placa metálica permitem acessar os ajustes dos monoestáveis de cada canal do MULTI.

Outras pequenas dificuldades foram encontradas nesta fase do processo de teste do módulo, como a comunicação através do barramento FERABus com o ADC413A, mas todas foram superadas com pequenos ajustes eletrônicos.

3.10 Medidas com fontes radioativas

Os testes até aqui executados indicaram que, apesar das dificuldades encontradas o novo MULTI funcionava como esperado, mas apenas uma medida de uma fonte radioativa mostraria a coerência dos dados obtidos. Utilizando uma fonte radioativa de ^{60}Co , três detetores (dois HPGe e um Ge(Li)), efetuou-se uma medida que apresentou muitos artefatos nos

espectros de tempo. A princípio pensou-se que seriam relacionados ao ajuste da eletrônica rápida utilizada na formação dos sinais de tempo. Após uma longa análise destes sinais ficou claro que eles se mostraram corretos, e portanto o problema deveria se encontrar no circuito de geração do sinal de *start* ou dos *stop's* individuais de cada entrada do TDC. Para tanto, ao invés de usarmos o sinal de *stop* gerado pelo MULTI, utilizamos a segunda saída do CFD acrescido de um atraso que simulava o tempo de processamento dos sinais pelo MULTI e introduzimos este sinal diretamente no TDC, apenas o sinal de *start* provinha do MULTI. Desta forma, se os artefatos desaparecessem eram os circuitos de *stop* que geravam os artefatos, e se continuasse era o circuito de *start*. A figura 40 mostra o esquema da montagem experimental utilizada para esta verificação.

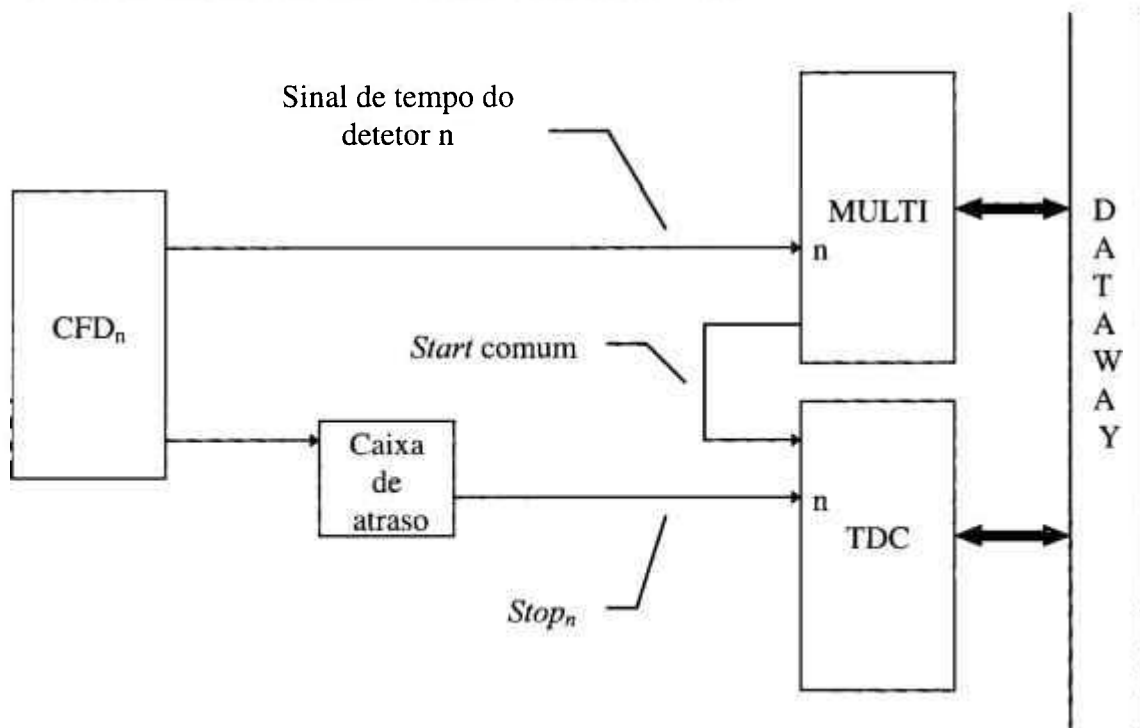


Figura 40 - Esquema da montagem experimental para verificação da procedência dos artefatos apresentados nos espectros de tempo gerados pelo MULTI. (n indica o e -nésimo detetor)

O resultado da medida mostrou que os artefatos eram introduzidos pelos circuitos de *stop* com o desaparecimento dos artefatos. Estes sinais

deveriam ser uma cópia dos sinais enviados pelo CFD ao MULTI, apenas atrasados de tal forma a serem enviados ao TDC após o sinal de *start* gerado pelo circuito de coincidência. Assim o uso de cabos ou caixas de atraso devidamente dimensionadas permitiu substituir completamente o uso dos circuitos de *stop* do MULTI. Resolvida esta dificuldade, efetuou-se uma medida de 60 horas de uma fonte de ^{60}Co com atividade de 4×10^5 Bq que é apresentada a seguir.

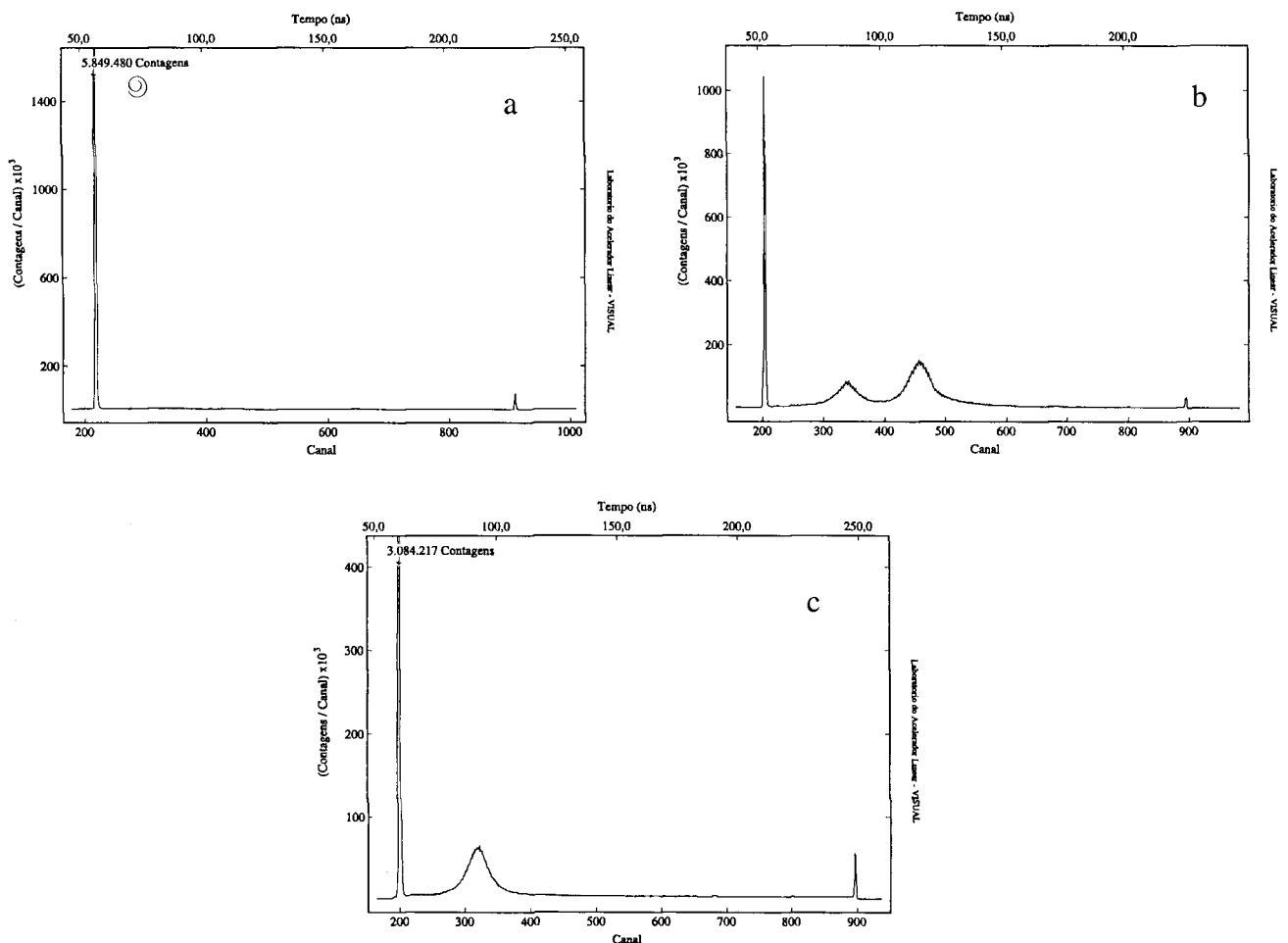


Figura 41 - Espectros de tempo absolutos lidos diretamente do TDC. O espectro *a* possui apenas o pico de auto-disparo, porque o seu sinal chega antes dos sinais dos outros detetores. Já o espectro *b* possui dois picos de coincidência porque o seu sinal é atrasado em relação aos sinais dos detetores *a* e *c*. Finalmente, o espectro *c* possui apenas um pico de coincidência pois é atrasado apenas em relação ao detetor *a*.

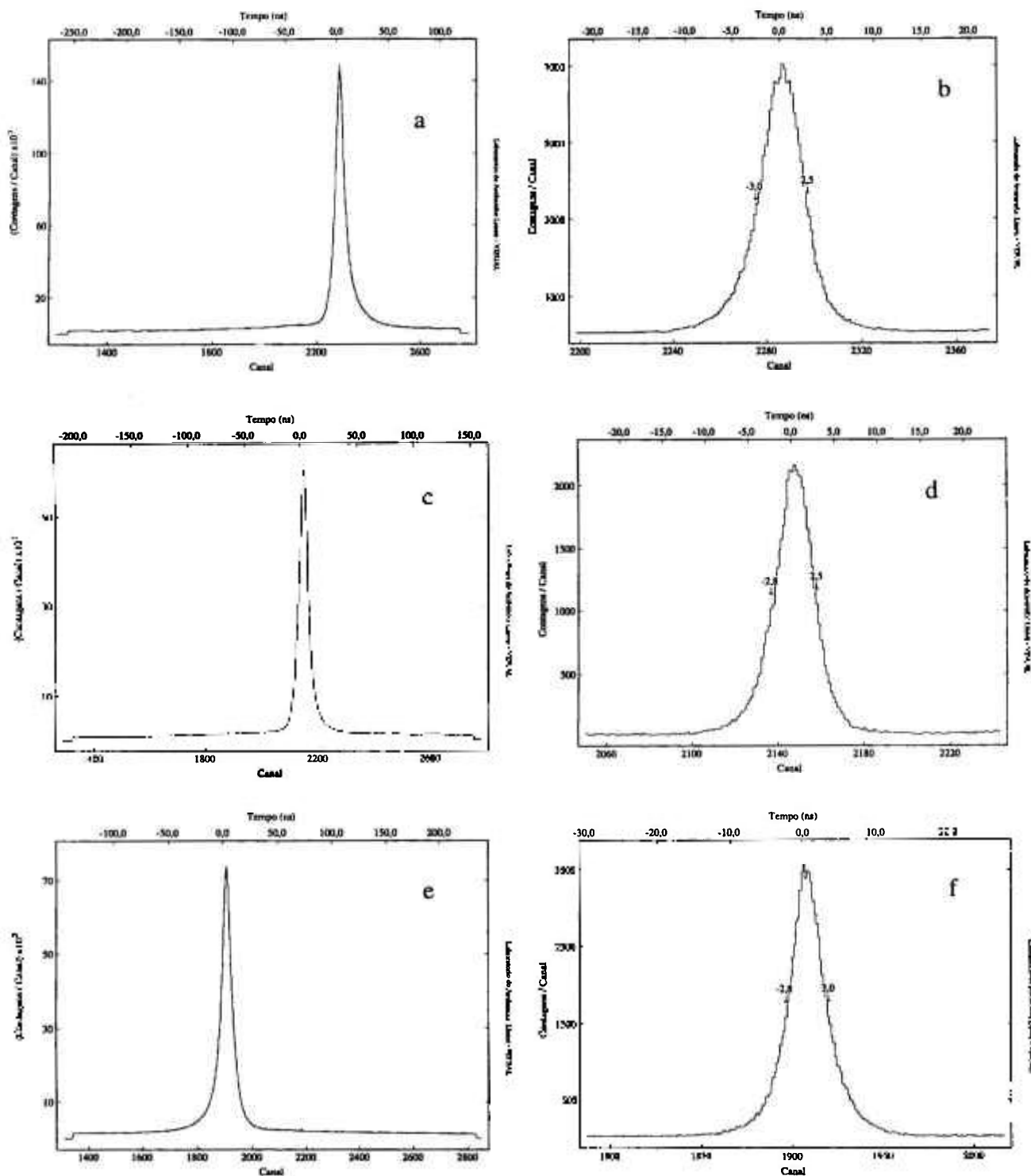


Figura 42 - Os espectros *a*, *c* e *e* são formados pela diferença de tempo entre os sinais dos três detetores utilizados. O espectro *a* é a diferença entre os sinais que formaram as figuras 40a e 40b, o espectro *c* é a diferença entre os sinais das figuras 40a e 40c e o espectro *e* é a diferença dos sinais das figuras 40b e 40c. Já os espectros *b*, *d* e *f* são os espectros de diferença de tempo, mas com janelas em energia. Vemos que a largura dos picos de coincidência variam de 5,3 a 5,8 ns o que é uma resolução muito boa para detetores de Germânio.

Comparando a figura 41 com as figuras 21 e 22, vemos que a relação entre o pico de auto-disparo e o artefato à extrema direita do espectro é muito maior no módulo definitivo, isto indica que houve uma melhora substancial no sentido de diminuir estes eventos indesejados. Mas se compararmos os espectros de diferença de tempo da figura 42 com o da figura 23, vemos que o artefato desaparece por completo nas extremidades do espectro. A largura dos espectros de diferença de tempo com janela em energia se manteve a mesma devido ao fato de que esta largura é característica do tipo de detetores utilizados e não do MULTI. Outra grande diferença encontrada na comparação entre os resultados do protótipo e do módulo definitivo é a largura do pico de auto-disparo (veja a figura 43).

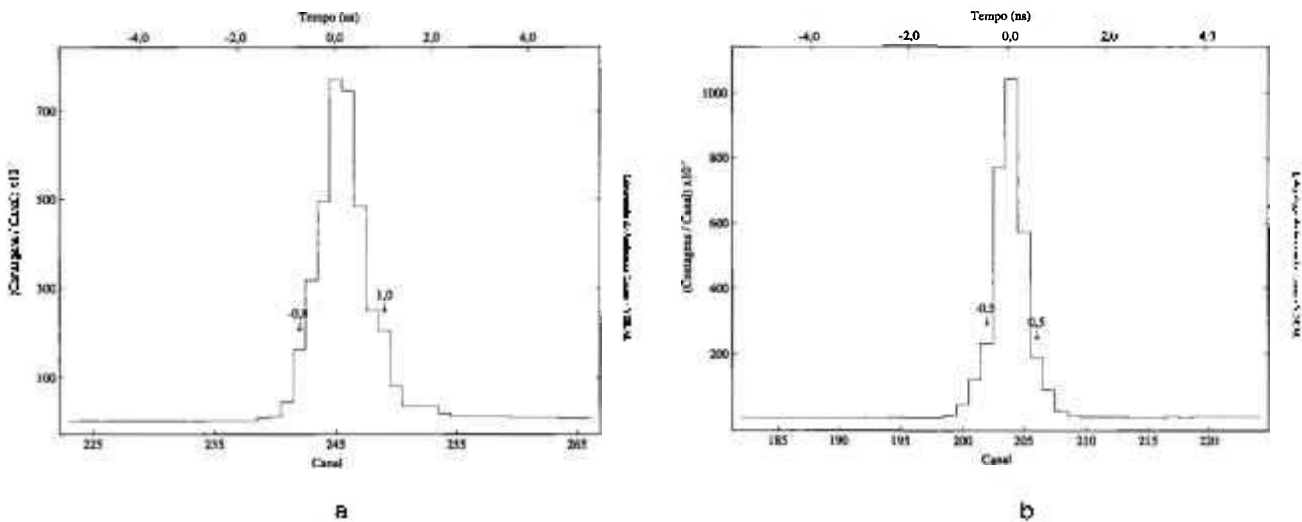


Figura 43 - O espectro *a* apresenta o melhor pico de auto-disparo conseguido com o protótipo do MULTI. Normalmente este pico apresenta vários nanosegundos de largura e uma estrutura. Já o espectro *b* apresenta um pico típico do módulo definitivo, sendo que a pior situação encontrada foi de 4 canais à meia-altura do pico, o que representa 1 ns.

O novo circuito de *start* do MULTI, que utiliza uma matriz lógica programável, possibilitou a redução da largura do pico de auto-disparo para 0,75 ns o que é uma melhora muito grande comparado aos vários

nanossegundos do protótipo. A medida da não-linearidade introduzida pelos circuitos de *start / stop* não foi efetuada, pois como o primeiro foi reduzido a apenas 3 canais de largura, a sua variação foi claramente reduzida e o segundo foi substituído pelos atrasos introduzidos nos sinais dos CFD's, não havia mais necessidade de tal medida.

Finalmente, faltava uma análise para verificação da existência das coincidências irreais devido à não reinicialização do ADC ao abortar um evento. Uma análise efetuada utilizando-se o programa BIDIM^[8], mostrou que este módulo não apresenta tal defeito, o que só veio mostrar que a introdução de um comando de reinicialização do ADC antes de fornecer o *gate*, a este módulo, eliminou a possibilidade de ocorrer tal problema.

3.11 Montagem do sistema MULTIDETETOR

O MULTIDETETOR é um sistema para aquisição de eventos de coincidência entre dois ou mais detetores de um total de oito. O protótipo do controlador do MULTIDETETOR, MULTI, por utilizar o ADC800 para conversão dos sinais de energia era limitado a no máximo três detetores, já que o LAL só dispunha de três módulos deste tipo. Com a substituição dos ADC's utilizados por dois ADC413 quádruplos, esta limitação foi eliminada. A figura 43 apresenta a montagem dos módulos ADC413, MULTI e TDC no bastidor CAMAC em sua disposição final.

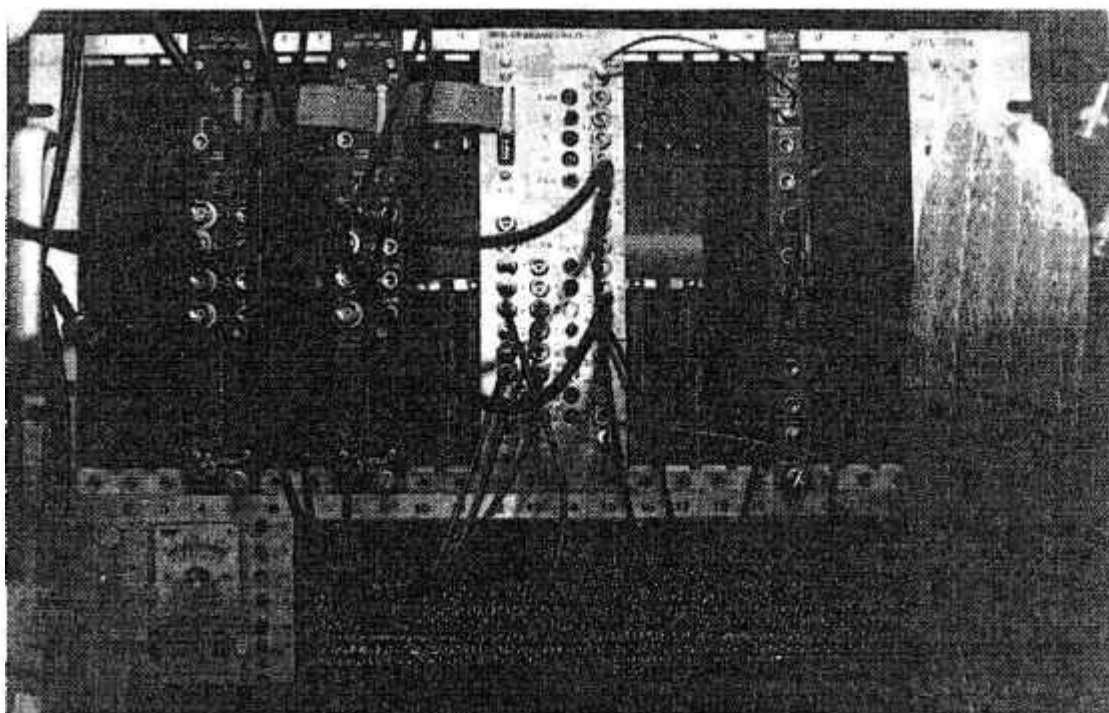


Figura 44 - Montagem final dos módulos padrão CAMAC que compõem o sistema MULTIDETECTOR. À esquerda, no bastidor vemos os dois ADC's quádruplos modelo ADC413A. No centro está o módulo MULTI definitivo e à sua direita o TDC. Na extrema direita se encontra o controlador de bastidor CAMAC-ISA também desenvolvido no LAL. Ligando o MULTI aos dois ADC's 413A vemos o cabo plano do barramento FERAbus .

4 Programas de Aquisição de Dados

4.1 Introdução

Nos capítulos anteriores foram apresentados os estudos, testes, resultados e indicadores de desempenho dos equipamentos tais como ADC, TDC, Controlador do bastidor CAMAC e do MULTIDETECTOR que compõem o novo sistema de aquisição de dados do LAL. Vimos que o sistema responsável pela aquisição de dados de duas linhas uniparamétricas seria instalado em um bastidor CAMAC e o MULTIDETECTOR, instalado noutro bastidor. Toda esta eletrônica precisa ser controlada pelo *software* que deve ser executado no computador ao qual é conectado o controlador do bastidor CAMAC.

Antes do início do desenvolvimento de um *software* de controle e aquisição de dados devemos estabelecer claramente os anseios dos futuros usuários e definir desta forma a abordagem que será tomada no sentido de atender estas expectativas dentro das possibilidades que o *hardware* nos oferece. Da experiência adquirida pelos integrantes do LAL^[6] durante os anos de utilização do sistema de aquisição baseado no PDP-11 e também nos sistemas MAESTRO^[14] e REGULUS^[15] baseados em microcomputadores padrão IBM-PC, foi possível verificar as virtudes e fraquezas destes sistemas e assim estabelecer o que se espera do novo sistema de aquisição de dados uniparamétrico. O desenvolvimento do *software* de controle do MULTIDETECTOR foi adiado até a conclusão da construção do sistema uniparamétrico, pois estas deveriam demonstrar a viabilidade da montagem de um sistema tão complexo como este.

4.2 *Requisitos do novo sistema de aquisição de dados*

Existem três escolhas básicas a serem feitas quando se decide desenvolver um programa de computador, o tipo de computador em que o programa funcionará, o sistema operacional e a linguagem na qual o programa será escrito. Como o IBM-PC seria o computador utilizado para a montagem do novo sistema de aquisição, então devemos observar o espectro de sistemas operacionais e linguagem disponíveis para este computador.

Um dos requisitos apresentados foi que o sistema operacional deveria ser multi-tarefa, pois dessa forma seria possível executar vários programas simultaneamente e por exemplo efetuar a transferência ou análise dos dados ao mesmo tempo em que a aquisição é efetuada. Desta forma o MS-DOS e outros sistemas que não possuem esta característica foram descartados. Como haveriam duas linhas uniparamétricas sendo controladas por um bastidor CAMAC e conseqüentemente por um único computador, ficou claro que o sistema operacional deveria ser multi-usuário. Desta forma o WINDOWS foi descartado, pois não permite que dois usuários possam utilizar completamente um microcomputador ao mesmo tempo. Assim estes dois únicos requisitos indicaram que apenas os sistemas *UNIX-like*, permitiriam obter o resultado almejado. Mas havia ainda um terceiro item, o sistema deveria ser *preemptivo* que significa que ele permite o total controle de qual programa está sendo executado a cada momento. Este último item definiu, dentre os vários “sabores” de UNIX disponíveis no mercado, o sistema operacional QNX, que é especialmente construído para aquisição e controle em tempo real.

Outro requisito muito importante era que o sistema deveria ser escrito numa linguagem de programação de fácil manutenção, o mais portátil possível, ou seja, deveria ser uma linguagem que permite que os programas escritos sejam facilmente adaptados a diferentes tipos de computadores e sistemas operacionais, mas que também fosse de alto desempenho. Dentre as linguagens disponíveis a escolha recaiu sobre linguagem C^[12], que é a linguagem em que foi escrito o Unix, é de uso genérico e gera códigos binários menores e mais rápidos que a maioria das linguagens de programação. Pelo fato de ser uma linguagem de médio nível, ela possibilita trabalhar facilmente com informação codificada em dígitos binários, como também permite escrever um código que é bastante genérico e de fácil portabilidade. Para o presente trabalho foi adquirido junto com o sistema operacional QNX, o compilador C/C++ da WATCOM^[39] que é muito utilizado neste tipo de aplicação, além de estar perfeitamente integrado ao QNX e à arquitetura do IBM-PC.

Um item muito importante era que o usuário dispusesse de uma interface gráfica, para visualizar os espectros obtidos, e que fosse de fácil utilização. Esta interface deveria possibilitar o controle da aquisição de dados a partir de qualquer computador conectado à rede do laboratório, facilitando o acompanhamento da aquisição efetuada. Para tanto esta interface foi desenvolvida seguindo o padrão X-WINDOWS / MOTIF, que é o padrão utilizado nos sistemas Unix. Programas desenvolvidos neste padrão permitem que sejam executados em computadores remotos independentemente do sistema operacional utilizado, desde que disponibilize a arquitetura cliente-servidor. Também contribuiu para esta escolha o conhecimento já acumulado no LAL sobre o desenvolvimento de interfaces neste padrão, na construção de vários programas já em uso pelo laboratório.

4.3 Descrição geral do sistema de aquisição uniparamétrico

Como vimos no capítulo 2, o sistema de aquisição de dados do LAL permitiria efetuar duas tomadas de dados uniparamétricas simultaneamente num mesmo bastidor CAMAC. Uma medida usaria o ADC114 da Ortec conectado a um módulo histogramador HM413, também da Ortec, e a outra utilizaria um ADC NIM modelo 8701 da Canberra conectado ao bastidor CAMAC através de um *Input Register* 2322 da BiRa. O ADC114 efetuaria a própria contagem de tempo vivo e, no caso do ADC8701, o relógio de tempo vivo 2203 da BiRa contabilizaria este controle. Para efeito de facilidade de referência, chamaremos o conjunto do ADC114 e HM413 de linha de aquisição do ADC114 (Linha do ADC114) e ao conjunto do ADC114, *Input Register* 2322 e contador de tempo vivo 2203 de linha de aquisição do ADC NIM (Linha do ADC NIM), pois qualquer modelo de ADC padrão NIM pode ser conectado ao *Input Register*.

Do ponto de vista do usuário do sistema, ele apenas deve saber que há duas linhas de aquisição uniparamétricas, às quais podem ser conectados os sinais provenientes da eletrônica associada ao seu experimento, respeitando as características inerentes a cada linha como já foi discutido no capítulo 2.

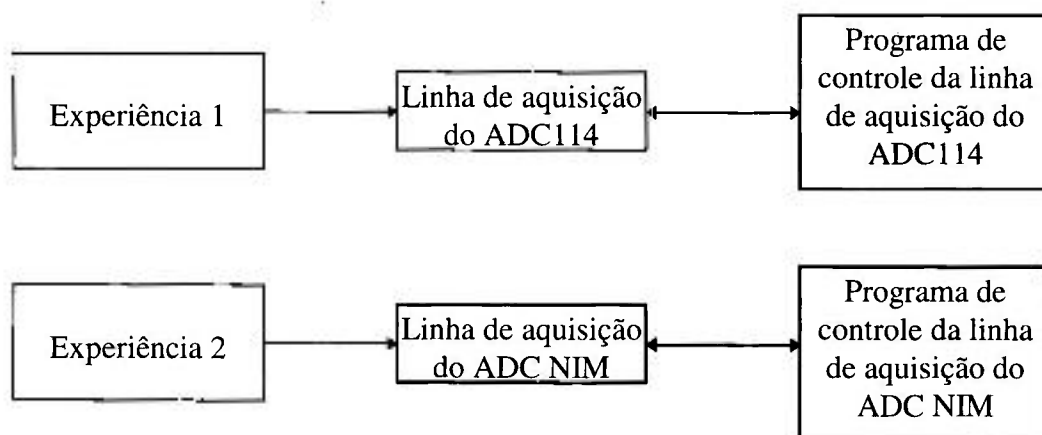


Figura 45 - Esquema do sistema de aquisição de dados do ponto de vista de um usuário.

Para controlar cada linha de aquisição, o usuário tem à sua disposição dois programas de computador que fazem a interface entre o usuário e o sistema de aquisição de dados. Portanto, se a sua experiência necessita das características da linha de aquisição do ADC114, ele deve utilizar o programa correspondente, o mesmo acontecendo com a linha do ADC NIM. A figura 45 mostra o esquema do sistema de aquisição do ponto de vista do usuário, mas uma análise um pouco mais acurada da mesma figura nos mostra que as duas linhas de aquisição não são completamente independentes. Uma relação existente entre os dois sistemas é que os programas são executados num mesmo computador. Outra é que as linhas de aquisição são montadas num mesmo bastidor CAMAC, e portanto compartilham o mesmo barramento, sendo comandadas por um controlador de bastidor, que também é a única linha de comunicação com o computador.

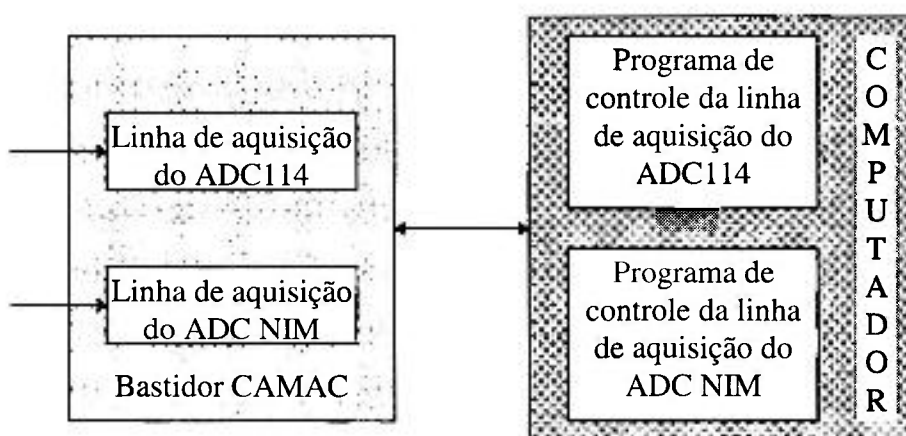


Figura 46 - Esquema do sistema de aquisição, mostrando que as linhas de aquisição e os programas não são totalmente independentes. Estas inter-relações são facilmente observáveis através da montagem física do sistema. A comunicação entre o bastidor CAMAC e o computador é efetuada pelo controlador de bastidor e isto explica a seta única entre os dois.

Ao se efetuar um exame mais detalhado vemos que, ao invés de independentes, os dois são altamente concorrentes no uso dos recursos do computador, bem como do bastidor CAMAC.

Como vemos na figura 46, temos dois programas que se destinam a controlar e efetuar a aquisição em dois ADC's distintos que são controlados pelo mesmo controlador de bastidor. O fato do acesso e controle dentro do bastidor ser efetuado unicamente pelo controlador, soluciona o conflito entre as duas linhas no mesmo bastidor CAMAC. Mas no computador, o fato de que qualquer um dos programas pode acessar o controlador a qualquer momento, pode gerar conflitos que tornariam impossível a obtenção dos dados experimentais. Assim se torna imprescindível a existência de um "árbitro" que gerencie o acesso ao bastidor CAMAC pelos programas de aquisição. Dentre as várias configurações possíveis a apresentada na figura 47 foi considerada a mais adequada.

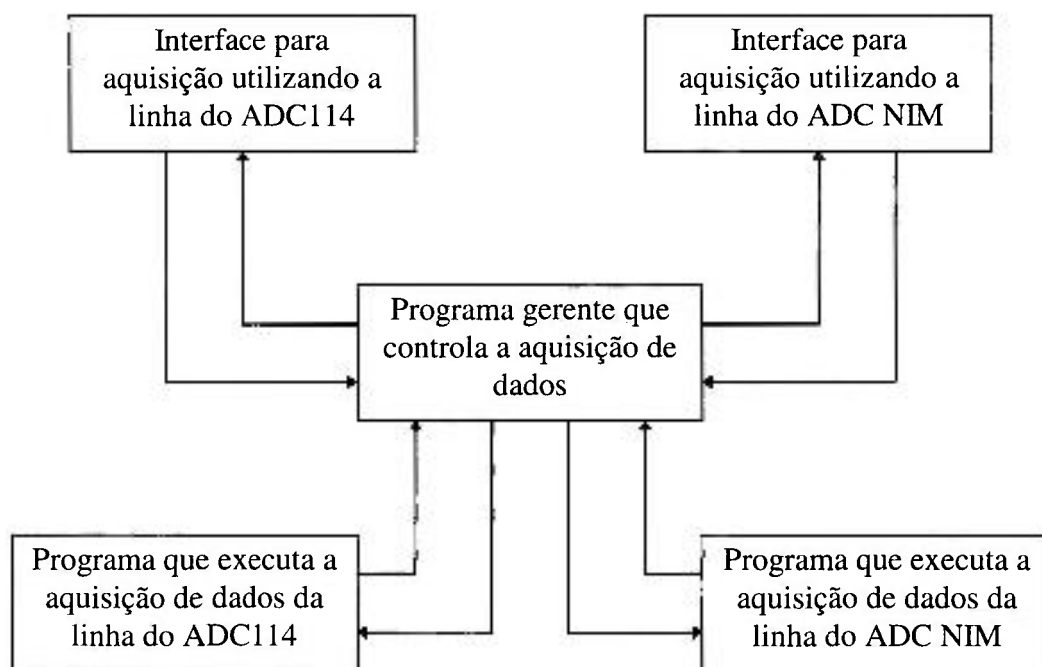


Figura 47 - Configuração dos programas de aquisição das linhas do ADC114 e ADC NIM. Vemos que o programa gerente ou "árbitro" executa toda comunicação entre a interface com o usuário e os programas de aquisição. Como o gerente tem acesso ao estado de cada programa, ele controla o acesso ao bastidor CAMAC sinalizando quando este está ocupado ou não.

As vantagens da estrutura apresentada na figura 47 são as seguintes:

1. Separação entre a interface com o usuário e o programa que executa o acesso ao bastidor CAMAC ou a aquisição. Desta forma o usuário pode iniciar uma medida e a seguir fechar a interface, liberando o terminal para outros usos, enquanto a aquisição se processa automaticamente.
2. A existência de dois programas de aquisição simplifica a construção destes, pois a linha do ADC114 tem características diversas da linha do ADC NIM. Desta forma, ao invés de escrever um intrincado programa que efetue o controle das duas linhas, criam-se dois programas pequenos e dedicados, sendo a arbitração do acesso ao controlador CAMAC efetuada pelo programa gerente.
3. Possibilidade de que as interfaces possam ser executadas em computadores remotos de forma a aumentar a disponibilidade da CPU (*Central Process Unit*) para a tarefa primordial que é a aquisição de dados. A comunicação entre as interfaces e o programa gerente se daria através do protocolo TCPIP^[9] utilizando RPC^[5] (*Remote Process Control*). A princípio todos os programas devem ser executados no mesmo computador para testar a estrutura e funcionamento do sistema. Mesmo assim as interfaces podem ser visualizadas remotamente, sendo exportadas pelo computador que efetua a aquisição.

Uma desvantagem desta estrutura é que a existência de cinco programas que se comunicam entre si tornando-os interdependentes, dificulta a depuração do sistema como um todo, pois em determinados casos é impossível executar apenas um único programa.

4.4 A comunicação entre os programas

O sistema operacional QNX^[31] disponibiliza três maneiras de realizar IPC (*Interprocess Communication*) num mesmo computador, são eles: Mensagens, Procuradores e Sinais.

As Mensagens enviam pacotes de alguns bytes de um programa para outro. Quando um programa envia uma mensagem a sua execução é interrompida e ele entra num estado chamado de *SEND-blocked* até que o outro processo receba a informação, neste momento o seu estado muda para *REPLY-blocked*, ou seja, ele continua parado mas está à espera de um retorno do programa destino. Mas para que um programa receba uma mensagem ele deve estar esperando por ela, senão a mensagem é completamente ignorada. Este estado de espera se chama *RECEIVE-blocked*, sendo que a sua execução também é interrompida até a chegada de uma mensagem. Para que o processo termine, o programa destino deve sinalizar o recebimento da mensagem, permitindo que o programa origem possa voltar à sua execução normal. Desta breve explanação podemos perceber o quão importante é a perfeita sincronia entre os dois programas, pois ambos podem ficar indefinidamente esperando por uma mensagem ou por um retorno de uma mensagem enviada.

O Procurador é um tipo especial de mensagem no qual nenhuma informação é transmitida a não ser a própria intenção de sinalizar algo ao outro programa. Neste caso a sincronia também é muito importante.

Finalmente, os Sinais são interrupções de *software*, e portanto não necessitam que o programa destino esteja esperando, basta que ele esteja preparado para receber o sinal. Esta preparação consiste em indicar uma determinada rotina a ser executada quando um sinal específico for recebido.

Assim, a execução normal do programa é interrompida e redirecionada para a rotina específica de tratamento daquele sinal e, ao término desta rotina, o programa volta à sua execução normal.

Lembrando a figura 47 que nos mostra a divisão dos programas do sistema de aquisição, observando as setas indicativas das comunicações entre eles, e analisando os processos de IPC disponíveis no QNX, vemos que o programa gerente é o único programa que se comunica com todos os outros processos, seja recebendo ou mandando informações. Já as interfaces e os programas de aquisição apenas se comunicam com o gerente, e em hipótese alguma o seu funcionamento pode ser interrompido à espera de uma comunicação, pois eles devem atender às solicitações do usuário no caso das interfaces, e efetuar a aquisição de dados no caso dos programas de aquisição.

Então a forma mais indicada para se efetuar as comunicações entre os processos é a seguinte: o gerente se mantém num estado de *RECEIVE-blocked*, pronto a receber qualquer mensagem mandada pelos outros programas e, assim que ele receber uma mensagem, deve respondê-la o mais rápido possível para não bloquear o funcionamento do programa emissor da mensagem. Mas quando o gerente necessitar se comunicar com um outro programa, ele deve usar um sinal, de tal forma que provocará no programa destino apenas um desvio da sua execução e não a sua interrupção.

As mensagens e sinais trocados pelos programas apenas sinalizam que o programa destino deve executar alguma tarefa. Embora as mensagens possam transmitir informações, optou-se pela transferência de apenas um código que indica ao programa gerente qual o programa que originou a mensagem e qual tarefa ele deve executar para a troca de dados. Como parâmetros de aquisição e espectros obtidos, optou-se pela utilização de arquivos em disco. A troca de arquivos é a única ligação direta entre as

interfaces e os programas de aquisição, mas a sinalização da disponibilidade de um determinado arquivo se dá através do programa gerente. A figura 48, onde as linhas pontilhadas e orientadas para baixo representam a seqüência de execução dos programas e as setas na horizontal indicam a troca de mensagens entre os programas, mostra um exemplo simples de troca de informações entre um programa de aquisição e sua interface através das mensagens ao gerente e do arquivo disponibilizado à interface.

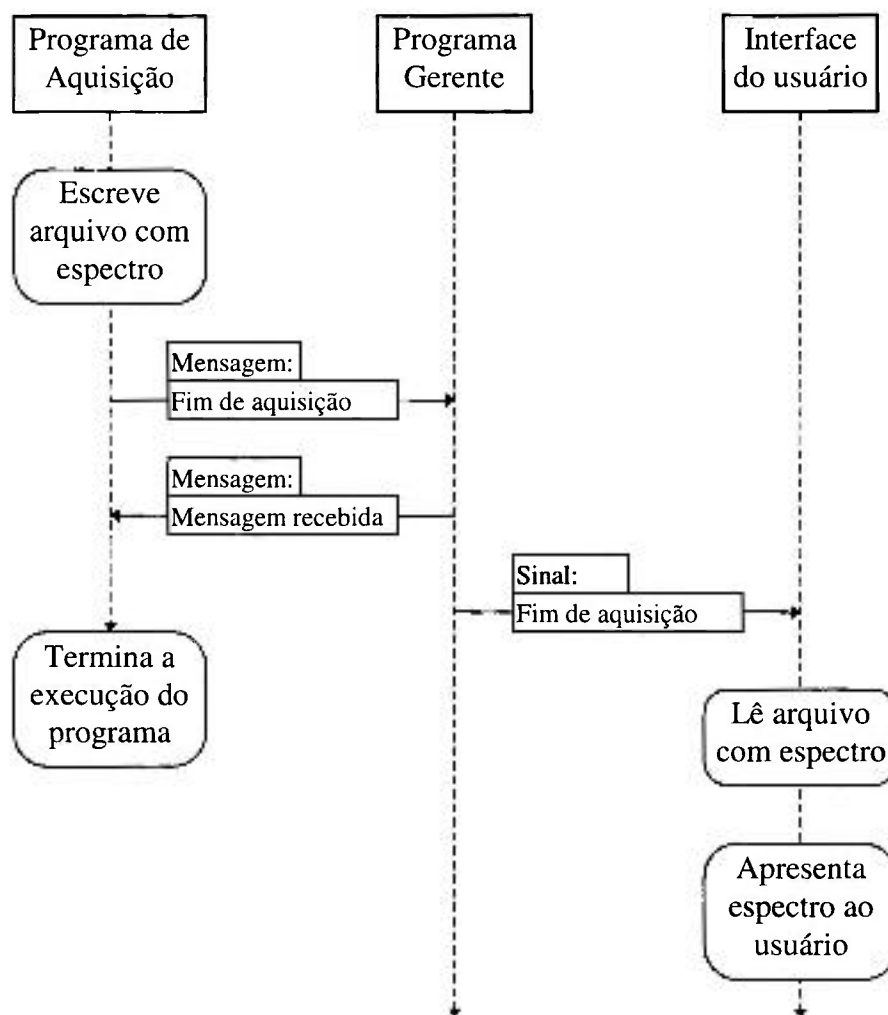


Figura 48 - Esquema de troca de mensagens e sinais entre os programas do sistema de aquisição de dados. Neste exemplo fica claro a necessidade da ida e vinda da mensagem entre o programa de aquisição e o gerente, se não houver o sinal de mensagem recebida o programa de aquisição não termina e fica esperando a mensagem. Também mostra que a troca de informações através de arquivos é utilizada apenas e diretamente entre o programa de aquisição e a interface.

4.5 As interfaces do sistema de aquisição

A interface de um programa de aquisição deve permitir ao usuário o controle total dos parâmetros disponíveis no *hardware* utilizado, bem como permitir fácil acesso às informações pertinentes ao estado do sistema e principalmente a visualização e manipulação básica dos dados adquiridos. Com o advento das interfaces gráficas para os sistemas operacionais dos computadores, foi possível atender a todos estes requisitos de uma forma muito mais organizada e de fácil aprendizado e utilização. Mas, o desenvolvimento de uma interface gráfica para o usuário GUI (*Graphical User Interface*) tem uma complexidade muito maior que uma simples interface de perguntas e respostas, montadas de uma forma linear e previsível pelo programador.

Numa GUI o usuário pode efetuar qualquer ação em qualquer seqüência, o que torna difícil para o programador a estruturação do programa. Assim, um programa com GUI tem necessariamente uma estrutura que consiste na construção da interface propriamente dita, ou seja, a montagem da janela vista pelo usuário onde estão os botões, menus, etc... Após a construção e exibição desta janela o programa deve entrar num estado de espera, aguardando a primeira ação do usuário. Se o usuário, por exemplo, escolher um determinado item de um menu, o programa automaticamente verifica qual escolha foi feita e executa a tarefa pedida se possível, se não, deve avisar ao usuário a falta de condições para execução do pedido deste, e em ambos os casos o programa volta ao estado de espera do próximo comando. Daqui vemos que qualquer comando requisitado pelo usuário sempre será executado numa forma circular, pois no final o programa sempre

deve voltar ao seu estado de espera, independentemente do que faça. Esta estrutura às vezes gera situações de difícil compreensão, portanto, para exemplificá-la a figura 49 mostra a estrutura do tratamento da ação de apertar e soltar o botão do *mouse*.

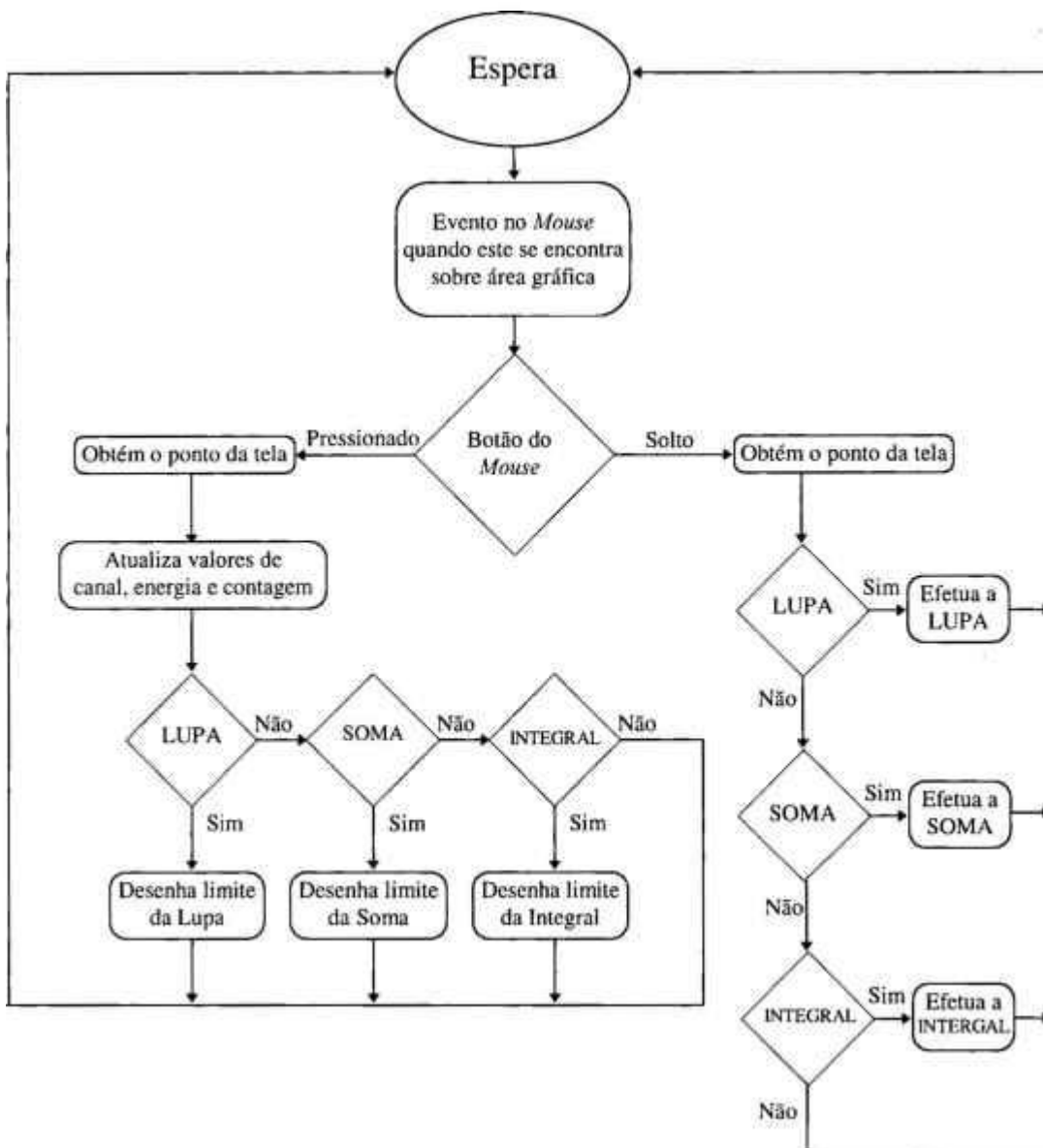


Figura 49 - Esta figura apresenta o esquema típico de uma ação produzida pelo usuário numa GUI. Quando o botão do *mouse* é pressionado ou solto, o programa sai do seu estado de espera e executa a ação apropriada de acordo com parâmetros previamente estabelecidos. Ao final da execução das tarefas requisitadas o programa volta ao estado de espera.

Em nosso caso o padrão GUI escolhido foi o OSF-MOTIF^[30] e o compilador foi o WATCOM-C^[39], sendo que ambos não possuem nenhum ambiente para desenvolvimento de programas com interfaces gráficas. Portanto todas as interfaces tiveram de ser construídas da forma mais elementar possível, ou seja, construindo cada item da janela, como menus, botões, gráficos de tal forma a assumirem a posição e formato desejado.

O sistema de aquisição de dados proposto consistia em duas linhas de aquisição independentes, e portanto duas interfaces deveriam ser desenvolvidas. A filosofia de construção de GUI's propõe que as interfaces sejam o mais padronizadas possível para permitir o fácil aprendizado pelo usuário pela simples similaridade com outras GUI's já utilizadas. Então em vista disto e do fato de que a construção de uma interface não é uma tarefa simples, como já exposto acima, as interfaces das duas linhas de aquisição foram construídas o mais parecidas possível, de tal forma a permitir o rápido aprendizado dos usuários e o reaproveitamento de código já desenvolvido e testado.

A interface deveria possuir dois estados, sendo eles: teste e aquisição. O primeiro deles, o modo teste, deveria consistir num estado onde o usuário teria acesso aos controles do ADC e poderia efetuar curtas medidas automáticas no sentido de repetição, ou manuais com intuito de visualizar e ajustar os parâmetros da medida, tais como ganho do amplificador, limites de discriminadores, etc. Já o modo aquisição indica que o sistema está efetuando uma medida dos dados para posterior análise. A implementação deste conceito foi conseguido com a divisão da interface em duas janelas. Numa se encontra todos os comandos para manipulação do espectro obtido e para a configuração dos parâmetros para a execução da aquisição. Já a segunda janela contém os comandos referentes ao controle do modo teste, sendo que a

existência desta segunda janela indica de forma clara ao usuário o modo em que o sistema de aquisição se encontra no momento. Se a janela de controle do modo teste estiver visível o sistema se encontra neste modo, senão, está no modo de aquisição.

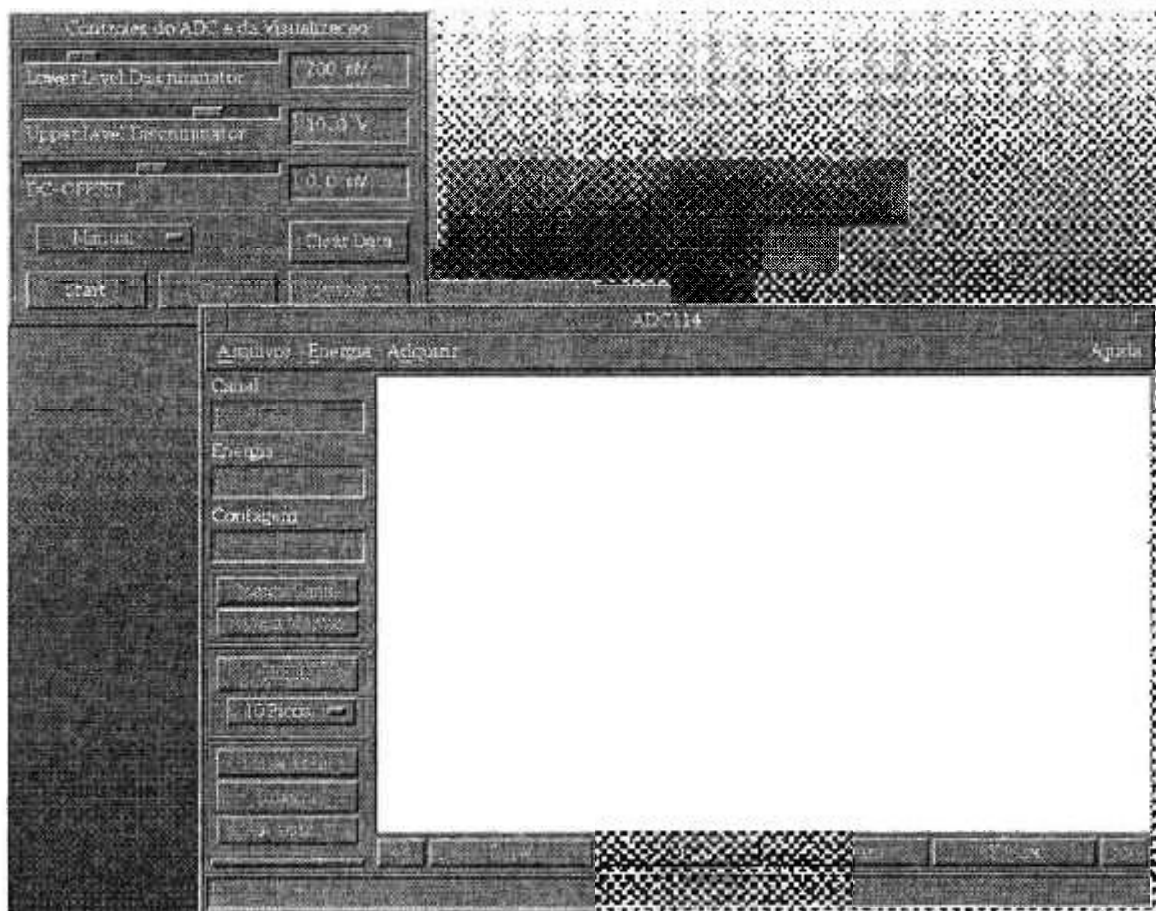


Figura 50 - Interface da linha de aquisição de dados do ADC114. A janela pequena no alto e à esquerda é a janela de controle do modo teste, e a maior é a janela responsável pela visualização do espectro adquirido e pelo controle da aquisição.

A figura 50 mostra o *layout* da interface de controle e aquisição para a linha do ADC114, ficando bem visível as duas janelas descritas acima. A janela grande é completamente similar para as duas linhas de aquisição, já o controle do modo teste é otimizada para cada linha.

4.5.1 A janela de controle de aquisição

A figura 51 apresenta em maior detalhe a janela de visualização de espectros e controle de aquisição.

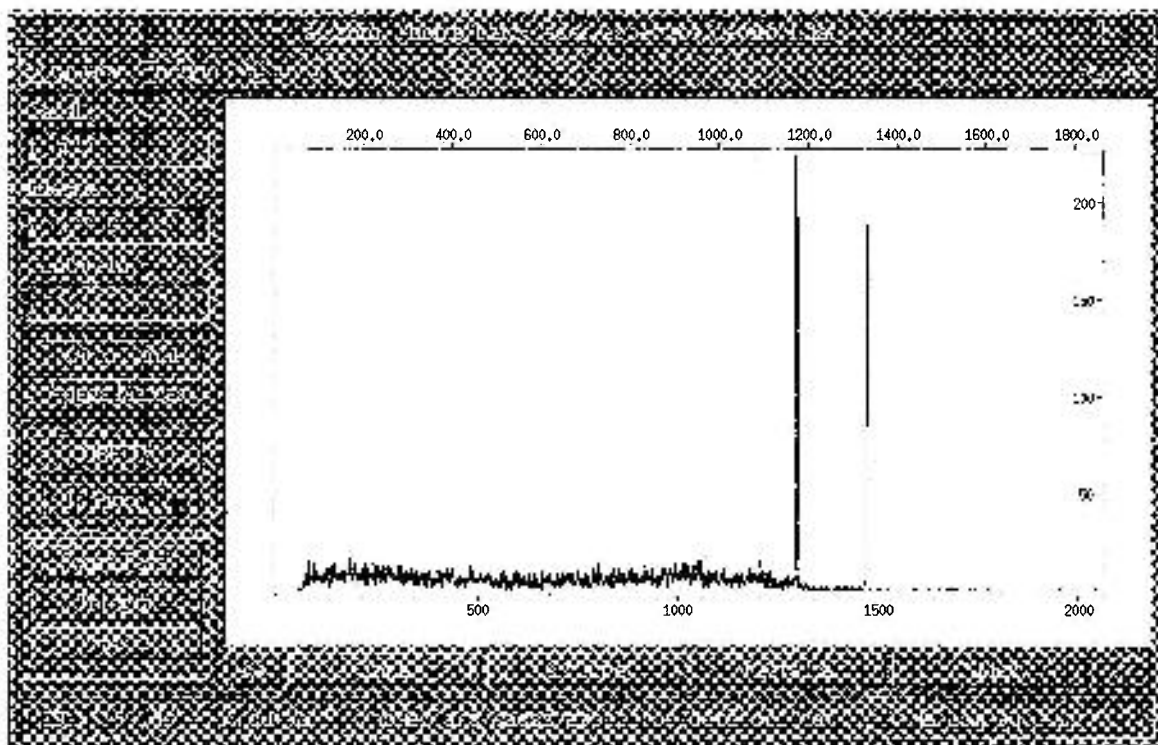


Figura 51 - Janela de exibição do espectro.

Na parte inferior da janela, logo abaixo do espectro, se encontram os botões que controlam a apresentação do espectro, tais como ampliação da região observada, deslocamentos para esquerda ou direita da região apresentada. Desta forma é possível enfocar pequenos detalhes de interesse do espectro utilizando os comandos que se encontram na parte inferior esquerda da janela sendo possível marcar canais, efetuar a procura automática de picos na região apresentada, somar ou integrar as contagens de uma determinada região. Estes controles permitem ao usuário obter informações, tanto visuais como quantitativas, que o auxiliam na análise primária necessária para a determinação da condição ótima para execução da aquisição de dados.

No alto da janela se encontram três menus: Arquivos, Energia e Adquirir. O primeiro menu permite ao usuário leitura e gravação em disco de um espectro, possibilitando a criação de um arquivo contendo a lista de todos os picos encontrados pelo processo automático de procura, e de um arquivo gráfico na linguagem PostScript^[34] que permite a impressão do espectro visualizado. Já o segundo menu permite a entrada dos valores de uma calibração de energia por um polinômio de grau menor ou igual a três, para a construção da escala de energia que se encontra na superior do gráfico do espectro. Se esta calibração de energia estiver ativa, toda informação referente a canais do ADC será fornecida em keV.

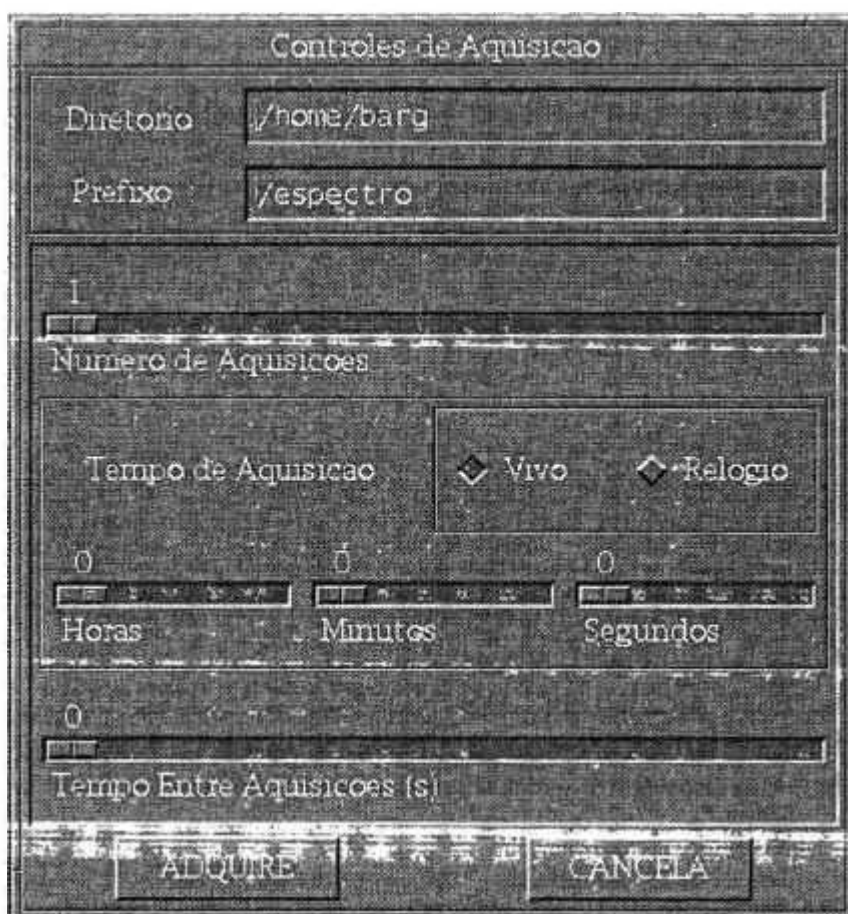


Figura 52 - Janela de entrada dos parâmetros de aquisição. O botão ADQUIRE (parte inferior esquerda da janela) inicia o processo e muda o estado da interface fechando a janela de controle do modo teste.

Finalmente, o menu mais importante é o Adquirir, pois este menu ativa uma janela, veja figura 52, que permite a entrada dos parâmetros de aquisição, tais como diretório de escrita dos arquivos de espectros, prefixos dos arquivos, número total de medidas a serem efetuadas, tempo de cada medida, tipo de relógio a ser utilizado e tempo de espera entre aquisições consecutivas. Além do mais, esta janela ativa a aquisição de dados fazendo com que a interface mude o seu estado para o modo de aquisição, fechando a janela de controle de visualização. Este último menu ainda possibilita a interrupção de uma aquisição em andamento fazendo com que a interface volte ao estado de visualização, e permite o acompanhamento da aquisição corrente mostrando um registro dos acontecimentos.

4.5.2 As janelas de controle de visualização

A existência da janela de controle do modo teste indica que nenhuma aquisição de dados está sendo efetuada no momento. Esta janela permite efetuar pequenas aquisições que permitem visualizar o efeito dos parâmetros do ADC e do amplificador no espectro gerado. A figura 53 apresenta o controle do modo teste da interface da linha de aquisição do ADC114. Nesta figura podemos observar três barras de rolagem que controlam os valores do *Lower Level Discriminator* (LLD), *Upper Level Discriminator* (ULD), *DCOffset* do ADC114. Este controle só é possível devido ao fato de que este ADC é padrão CAMAC, e portanto é possível controlar digitalmente estes parâmetros.

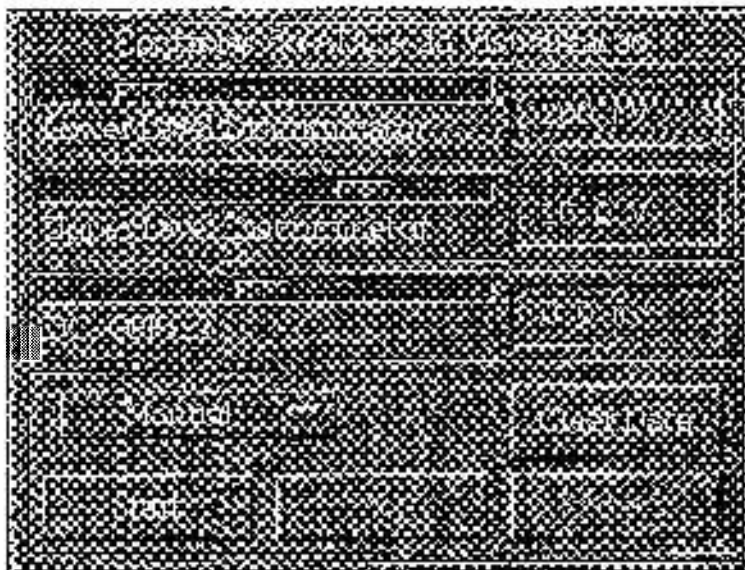


Figura 53 - Janela de controle do modo teste da interface do programa de aquisição da linha do ADC114. Na parte superior vê-se os controles dos parâmetros do ADC e na parte inferior está o controle da aquisição para efetuar os testes necessários.

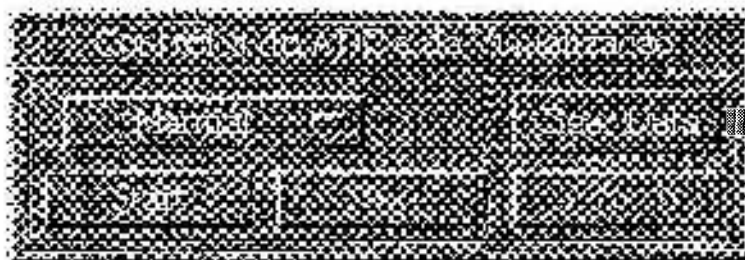


Figura 54 - Janela de controle do modo teste da interface do programa de aquisição da linha do ADC NIM. Apenas os controles da aquisição para os testes estão disponíveis.

Se compararmos com a janela da interface da linha do ADC NIM (veja a figura 54), vemos que a única diferença é que nesta segunda não existem os controles dos parâmetros do ADC, visto que no padrão NIM não há controle digital, sendo possível apenas o controle manual através de chaves e botões no próprio módulo.

Como a parte inferior da janela do ADC114 é similar à do ADC NIM, basta descrevermos esta segunda. O botão que se encontra na parte superior esquerda permite ao usuário efetuar testes manuais ou automáticos. O teste automático consiste em efetuar aquisições rápidas e seguidas com tempos pré-fixados pelo usuário (1, 2, 5, 10 e 15 segundos), sendo que não há acúmulo de dados entre uma aquisição e outra. Esta facilidade é muito útil quando o usuário está, por exemplo, ajustando o ganho do amplificador, obtendo na tela a situação do espectro a cada período determinado por ele. Já no teste manual, o controle do início, fim e limpeza do espectro é efetuado pelo usuário, utilizando os outros três botões disponíveis nesta janela. Um quarto botão é necessário para atualizar o espectro apresentado na janela principal, pois como o sistema é composto por vários programas separados, é necessário que um comando específico seja mandado ao programa de aquisição para que este disponibilize o espectro para a interface.

A facilidade de uso de um programa construído utilizando interfaces gráficas esconde a complexidade da construção do programa. A figura 57 apresenta o esquema das ações efetuadas ao ser pressionado o botão de *Start* da janela de controle de visualização. Vemos nesta figura que este botão inicia uma troca de mensagens entre a interface e o programa gerente, que por sua vez inicia a execução de um terceiro programa que efetuará a aquisição. Já a figura 55 mostra o esquema de ações disparados pelo botão *Stop* e finalmente o esquema de ações do botão *Refresh* é apresentado na figura 56. Muitos outros esquemas poderiam ser apresentados aqui, mas estes dois demonstram o quão complexo um sistema de aquisição de dados em tempo real pode ser.

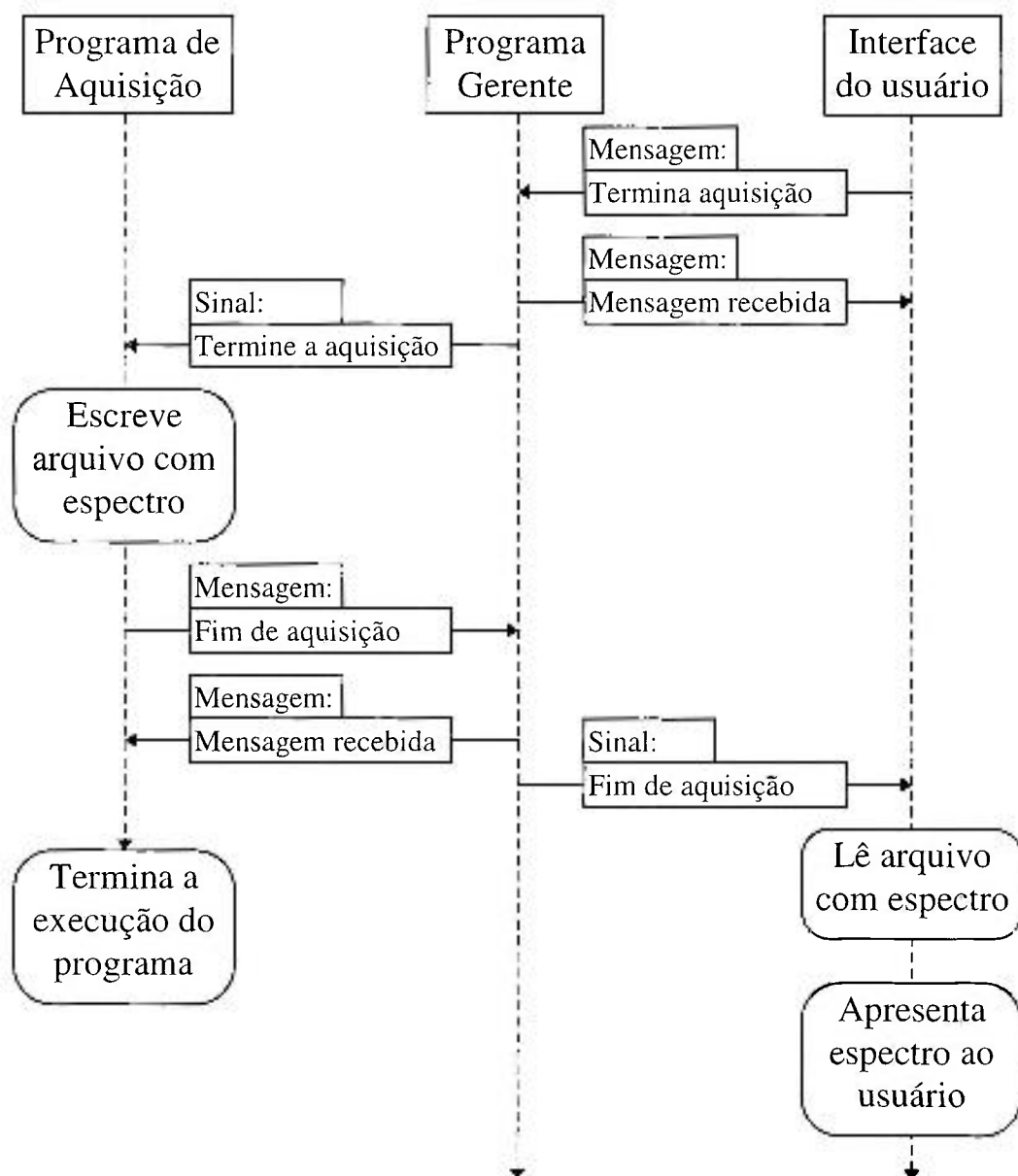


Figura 55 - Esquema de mensagens, sinais e ações iniciados pelo pressionar do botão *Stop* da janela de controle do modo teste quando é executado um teste manual. Comparando com a figura 48 que mostra o final de uma aquisição normal, vemos que a única diferença é a mensagem mandada ao gerente e repassada ao programa de aquisição, indicando que este deve terminar a aquisição em curso.

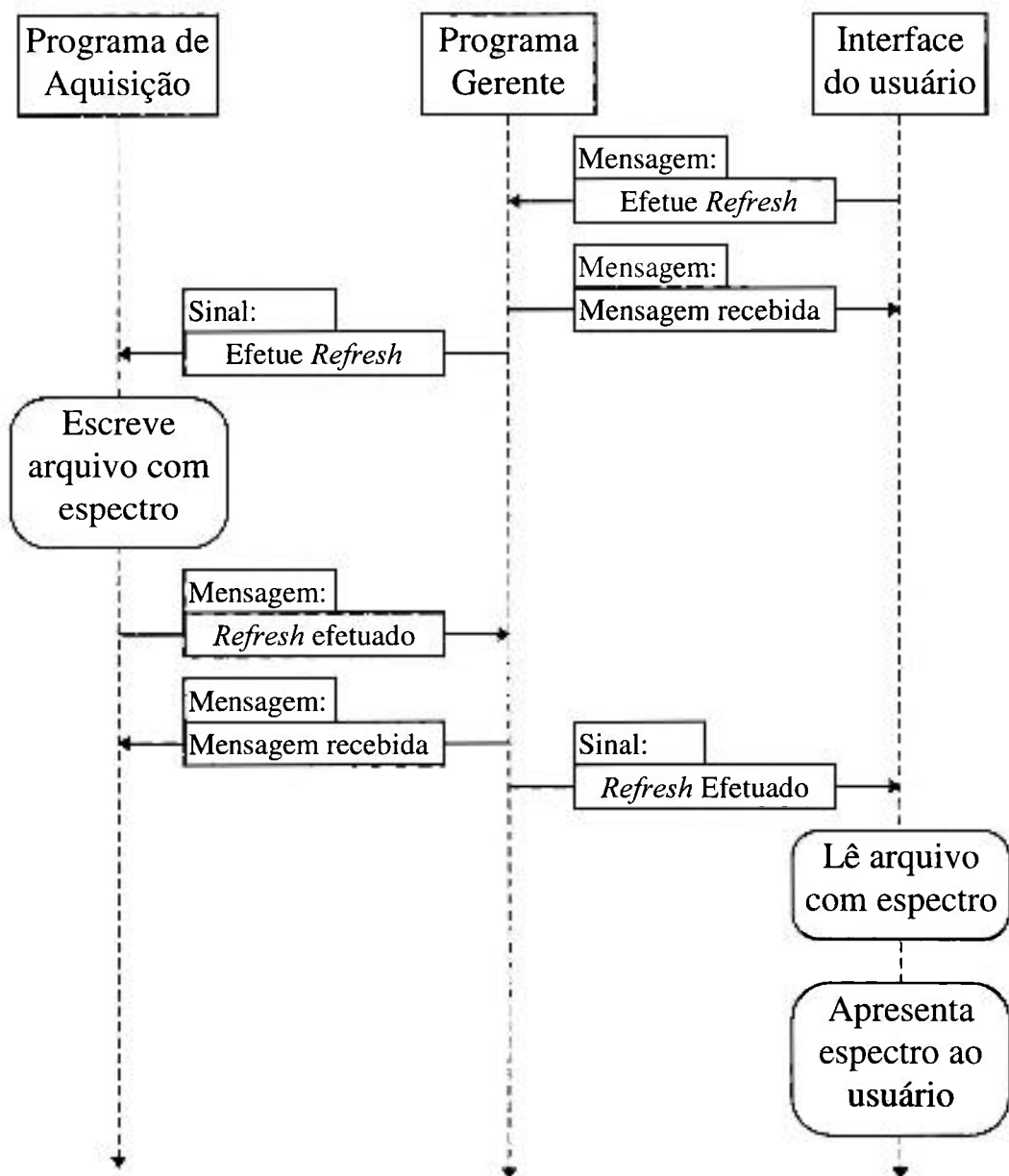


Figura 56 - Esquema de mensagens, sinais e ações iniciados pelo pressionar do botão *Refresh* da janela de controle do modo teste quando é executado um teste manual. Neste caso a execução do programa de aquisição não é abortada, sendo apenas interrompida no caso do ADC114, para a escrita do arquivo de espectro.

4.6 O programa gerente

O funcionamento do sistema de aquisição depende fundamentalmente do programa gerente, cuja responsabilidade é a sincronia e a comunicação entre todos os programas que compõem o sistema. Este programa é inicializado quando o sistema operacional é colocado em funcionamento e fica ativo durante todo o tempo que o computador estiver ligado. Como ele não possui interface e apenas recebe e emite mensagens e ativa os programas de aquisição, pode ser considerado como um *daemon*^[35], que são pequenos programas que intermediam a execução de tarefas entre processos no Unix.

Como discutido anteriormente, a comunicação das interfaces e dos programas de aquisição com o programa gerente é feita utilizando mensagens. Assim, o programa gerente deve estar constantemente à espera de mensagens e, quando uma é recebida, ela é analisada e a tarefa requisitada é imediatamente executada de forma que o programa possa voltar ao estado de espera para a próxima mensagem. Basicamente, guardando as proporções, corresponde ao estado de espera em que a GUI se encontra entre uma ação e outra do usuário.

As mensagens utilizadas na comunicação entre os programas, foram definidas procurando-se o menor conjunto possível que atendesse as tarefas requeridas pelo sistema, dentro da estrutura do sistema. Refletindo a similaridade que existe entre as duas linhas de medidas, as mensagens são exatamente iguais para ambas. A diferenciação entre uma linha e outra foi conseguida utilizando-se faixas de números para os conjuntos de mensagens, como mostra a tabela 1. A única exceção refere-se às mensagens referentes ao bloqueio de acesso ao CAMAC.

Emissor	ADC114	ADC NIM	Descrição
Interface	10	20	Requer a execução do programa de aquisição.
Programa de aquisição	11	21	Indica o final de uma aquisição de dados.
Interface	12	22	Requer o aborto da aquisição de dados em execução.
Interface	13	23	Sinaliza pedido de atualização do espectro a ser apresentado ao usuário <i>Refresh</i> .
Programa de aquisição	14	24	Indica que o espectro para <i>Refresh</i> foi atualizado.
Programa de aquisição	15		Requer o bloqueio do acesso ao CAMAC pelo programa de aquisição do ADC NIM.
Programa de aquisição		25	Indica que o acesso ao CAMAC foi bloqueado.

Tabela 1 - Mensagens recebidas pelo programa gerente.

O uso de faixas de números para os códigos das mensagens facilita a identificação da linha de aquisição que está requisitando a tarefa, bem como torna bem flexível a inclusão de novas mensagens ou mesmo outras linhas de aquisição, utilizando um mesmo programa gerente.

Como o gerente é o responsável pela execução dos programas de aquisição, ele tem o controle total sobre a existência ou não de uma aquisição em andamento. Mas como a interface é comandada pelo usuário, o gerente precisa saber se a interface está ativa quando ele necessitar se comunicar com ela. Para tanto, cada interface e o gerente ao serem executados criam um pequeno arquivo que possui como único conteúdo o seu PID (*Process Identifier*). Desta forma, a simples existência do arquivo correspondente à determinado programa indica que ele está sendo executado e se for necessária a comunicação entre os processos, o PID está disponível em seu interior.

4.7 Os programas de aquisição

Dos cinco processos que compõem o sistema de aquisição de dados (vide figura 46), apenas dois efetuam o acesso ao *hardware*. Como o funcionamento do ADC114 + Histogramador é muito diferente do conjunto *Input Register* + Relógio de Tempo Vivo, estes processos possuem metodologias distintas para efetuar a aquisição dos dados. Mas, embora completamente diferentes no acesso, controle e leitura dos dados, eles possuem uma estrutura padronizada para efetuar a comunicação com os outros programas do sistema de aquisição. Desta forma é possível simplificar a escrita e depuração dos códigos referentes à comunicação entre os programas de aquisição e os outros do conjunto. A seguir, analisaremos as similaridades entre os dois programas de aquisição e depois nos aprofundaremos em suas diferenças.

4.7.1 Similaridades entre os programas de aquisição de dados

Quando o usuário inicia uma aquisição de dados através de sua interface, uma mensagem é mandada para o programa gerente e este inicia a execução do programa de aquisição. Mas vários parâmetros são necessários e devem ser fornecidos pelo usuário através da interface, portanto, antes de mandar a mensagem ao gerente, a interface cria um arquivo contendo todos os parâmetros a serem utilizados na aquisição de dados a ser efetuada. Este arquivo é lido pelo programa de aquisição assim que é executado, e a existência deste arquivo em disco indica uma aquisição em andamento, pois ao final da aquisição dos dados este arquivo é apagado pelo próprio programa de aquisição.

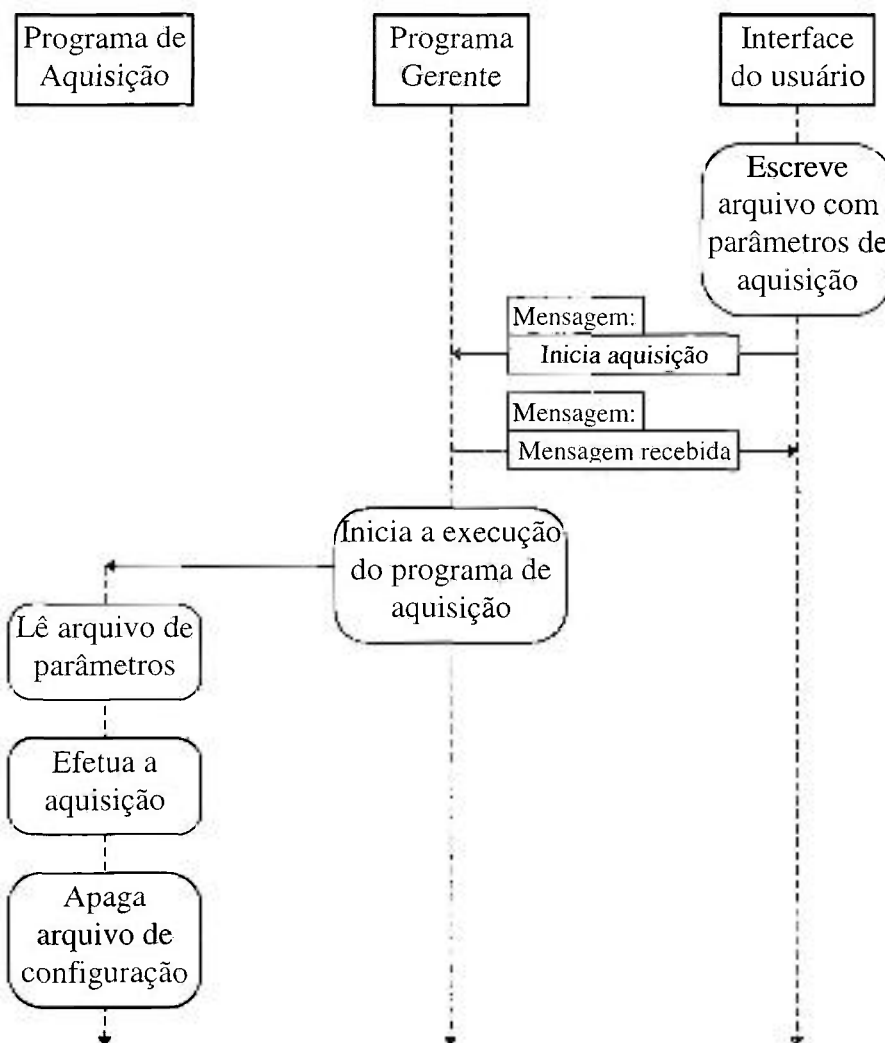


Figura 57 - Esta figura complementa a figura 48 pois mostra o início de uma aquisição de dados. Aqui podemos ver como se efetua a passagem dos parâmetros de aquisição entre a interface do usuário e o programa de aquisição. Quando o usuário indica na interface que a aquisição deve ser iniciada, o arquivo de parâmetros é escrito e uma mensagem é mandada ao gerente que executa o programa de aquisição que por sua vez lê o arquivo de parâmetros e ao final da aquisição o apaga. Os passos seguintes podem ser acompanhados na figura 48.

O arquivo de parâmetros de aquisição é muito similar tanto para a linha de aquisição do ADC114 quanto do ADC NIM, diferindo apenas nos parâmetros de controle de LLD (*Lower Level Discriminator*), ULD (*Upper Level Discriminator*) e DCOFF (*DC Offset*) que são disponíveis apenas para

o ADC114. A tabela 2 mostra a composição dos arquivos de parâmetros de ambas as linhas de aquisição e indica o que cada item representa.

Item	ADC114	ADC NIM	Descrição
LLD	sim	não	<i>Lower Level Discriminator</i>
ULD	sim	não	<i>Upper Level Discriminator</i>
DCOFF	sim	não	<i>DC Offset</i>
TEMP_AQUI	sim	sim	Tempo de aquisição (segundos)
LIMPEZA	sim	sim	Limpa espectro em memória
AQUISICAO	sim	sim	Aquisição ou Visualização
UID	sim	sim	<i>User Identifier</i>
ESPECTRO.path	sim	sim	Diretório dos arquivos
ESPECTRO.nome	sim	sim	Nome dos arquivos de espectros
N_AQUI	sim	sim	Número de aquisições
ESPERA	sim	sim	Tempo de espera entre aquisições (segundos)
RELOGIO	sim	sim	Tempo de relógio ou tempo vivo

Tabela 2 - Esta tabela mostra o conteúdo dos arquivos de parâmetros fornecidos pelas interfaces aos programas de aquisição de dados. O item AQUISICAO indica se os últimos seis itens estarão presentes no arquivo ou não, pois numa visualização estes parâmetros não são utilizados.

Embora a interface permita ao usuário determinar o estado do sistema de aquisição, se é uma visualização do espectro ou uma aquisição de dados, para os programas de aquisição a diferença é muito sutil, pois ela se resume à não necessidade dos seis últimos parâmetros da tabela 2 e internamente apenas não serão criados os arquivos de resultados para os usuários.

Finalmente observando a figura 58, vemos que embora a implementação dos dois programas seja muito diferente devido às diferenças de *hardware*, internamente o algoritmo de ambos é muito similar.

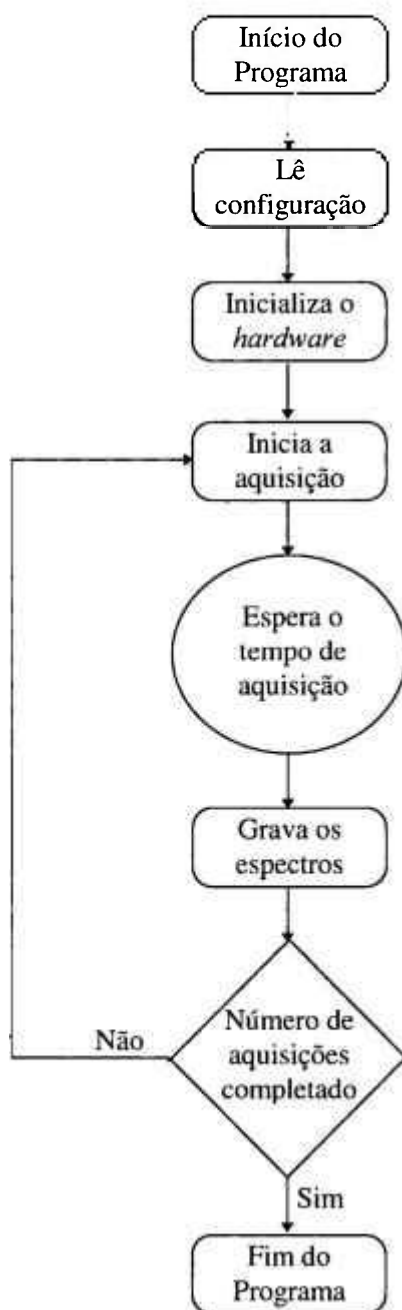


Figura 58 - Diagrama de blocos mostrando a estrutura dos programas de aquisição de dados. Nesta figura, não ficam evidentes as diferenças de implementação específica a cada *hardware*, mas mostra que os dois programas possuem a mesma estrutura apesar das diferenças do *hardware* utilizado.

4.7.2 Particularidades da linha de aquisição do ADC114

A aquisição de dados utilizando o ADC114 conectado ao módulo histogramador HM413 é muito fácil devido ao fato de que não é necessário atender o ADC a cada conversão, já que os dados são automaticamente histogramados e armazenados pelo HM413. O programa de aquisição deve basicamente configurar o ADC114 e o HM413 e habilitá-los para a aquisição, podendo a seguir ficar esperando o tempo determinado pelo usuário. Ao final deste tempo, basta parar o ADC114, ler o HM413, escrever o arquivo contendo o espectro e a aquisição está pronta.

Agora, se o usuário definiu que o tempo de aquisição deve ser o tempo vivo do sistema, então o programa se complica um pouco, visto que devemos contar este tempo e não simplesmente esperar o tempo de relógio passar. O ADC114 possui um relógio de tempo vivo que não só usa os dados do próprio ADC como também os sinais de ocupado e empilhamento do amplificador utilizado para efetuar a correção de tempo vivo do sistema. Infelizmente este relógio não pode ser programado para contar um determinado tempo e ao final deste avisar o controlador do bastidor CAMAC que o tempo se passou. Ele apenas pode ser zerado e inicializado, sendo necessário parar o funcionamento do ADC114 através do comando *Inhibit* do CAMAC para efetuar a leitura do tempo que se passou, desde o início do processo.

Assim, quando é definido que o tempo de aquisição deve ser o tempo vivo do sistema, o programa espera em tempo de relógio metade do tempo total de medida e verifica no relógio do ADC114 quanto tempo vivo se passou. Subtraindo do tempo total de medida o tempo vivo que já se passou, o programa espera novamente, em tempo de relógio, metade do tempo total restante e novamente efetua a leitura do relógio do ADC114, subtrai este

tempo do tempo total. Este processo é repetido até que a diferença seja menor que 20 milissegundos que é a precisão do relógio do ADC114, então ele faz uma projeção do tempo restante, espera este tempo e termina a aquisição.

4.7.3 Particularidades da linha de aquisição do ADC NIM

O fato de utilizarmos o controlador CAMAC ISA, que permite associar o LAM de uma determinada posição do bastidor à uma interrupção de *hardware* do microcomputador, tornou a construção do programa de aquisição da linha do ADC NIM muito fácil. Como esta linha não dispõe de um módulo histogramador, a aquisição deve ser efetuada evento a evento e o histograma deve ser montado na memória do computador. O programa de aquisição deve apenas esperar o fim do tempo de medida, pois a cada conversão do ADC o módulo *Input Register* gera um *LAM* que por sua vez provoca uma interrupção de *hardware* no computador. Neste momento o processador interrompe a sua atividade e executa a rotina que deve tratar da interrupção gerada, lendo e histogramando o dado e reinicializando o *Input Register*. Ao final do tempo de medida o programa de aquisição deve apenas guardar em disco o histograma adquirido.

Se o usuário indicou que o tempo de aquisição deveria ser o tempo vivo do sistema, então mais uma interrupção de *hardware* do computador é associada ao *LAM* do módulo de controle de tempo vivo. O módulo 2203 da BiRa possui um relógio interno que pode ser programado de tal forma que ao final do tempo pedido ele gera um *LAM* no bastidor CAMAC. Desta forma basta programá-lo para contar 1 segundo de tempo vivo do ADC, e assim a cada segundo de tempo vivo será gerada uma interrupção de *hardware* no computador que contará o número de segundos “vivos” pedidos pelo usuário.

4.7.4 Algumas pequenas diferenças entre os programas

No arquivo de parâmetros de aquisição o item LIMPEZA tem um significado bem diferente nos dois programas de aquisição devido às diferenças de *hardware* das duas linhas de aquisição. Quando o usuário está em modo de visualização manual, ele pode parar a aquisição e ao retomá-la ele pode ter pedido para limpar o espectro em memória ou não, querendo que a nova visualização se sobreponha à antiga. No caso do ADC114 isto indica que, ao inicializar a nova aquisição, o programa não deve efetuar a limpeza do módulo histogramador, pois o espectro que o usuário está vendo ainda está armazenado ali. Já no caso do ADC NIM, como é o programa que efetua a histogramação, ao término da aquisição o espectro é mandado para a interface e o que se encontrava na memória deste programa é perdido. Assim este parâmetro indica ao programa de aquisição que ele deve efetuar a leitura de um espectro que é fornecido pela interface, e efetuar a nova aquisição de dados sobre este espectro fornecido.

Uma outra diferença entre as duas linhas de aquisição é o número de canais utilizados na conversão do sinal analógico em digital. Como o ADC114 possui um número fixo de canais de conversão, a saber 16384 canais, a interface não disponibiliza nenhum controle a esse respeito, visto que é desnecessário. Mas a maioria dos ADC's padrão NIM permite a variação do número de canais para conversão e desta forma a interface deveria conter um controle deste valor. Neste caso, infelizmente este controle não atuaria sobre o ADC, visto que ele é padrão NIM, e portanto teria apenas um caráter informativo dando margem a possíveis enganos do usuário. Assim, para que as duas interfaces fossem o mais parecidas possível, criou-se no programa de aquisição da linha do ADC NIM um método automático de

determinação do número de canais utilizados pelo ADC em uso. O programa guarda o maior valor de canal convertido e efetua o seguinte cálculo:

$$N = 2^{\lceil \log_2(n+1) \rceil}$$

onde N = Número máximo de canais utilizado pelo ADC
 n = maior canal convertido pelo ADC

desta forma obtêm-se a faixa de conversão do ADC de uma forma automática sem a necessidade de um controle específico na interface. Uma análise um pouco mais criteriosa nos mostra que este método não é infalível, pois se o ADC for ajustado para trabalhar com 8192 canais e o ganho do amplificador for muito baixo de forma que o maior pulso que ele forneça seja convertido no canal 3000, então o uso da fórmula 1 nos dará um número máximo de canais de 4096, que é um valor falso. Mas um erro da mesma magnitude seria cometido se o usuário fornecesse ao programa a informação de que o ADC estaria configurado para converter até 4096 canais.

Por outro lado, se houvesse a possibilidade de o usuário informar o número de canais e fosse introduzido um valor menor que o ajustado no ADC, então o espectro seria cortado obedecendo a informação fornecida pelo usuário. Neste caso o método automático não comete este erro, pois ao receber um valor maior que N durante a aquisição, o valor de N é automaticamente modificado.

Portanto, este método automático substitui a informação do usuário com igual chance de cometer erros com a vantagem de eliminar um controle a mais na interface com o usuário e eliminar por completo um erro que o usuário poderia cometer.

4.7.5 Acesso simultâneo ao bastidor CAMAC

O principal problema em montar duas linhas de aquisição independentes num mesmo bastidor CAMAC é a possibilidade de conflito no acesso ao controlador. Como temos dois programas independentes que controlam módulos no bastidor através do controlador, devemos tomar cuidado para que eles não efetuem acessos simultâneos, o que provocaria não só a perda de informações como também o total descontrole do bastidor e travamento do sistema devido a perda de sincronia do sistema como um todo. Desta forma é necessário um sistema de “semáforos” de forma a indicar se o controlador e conseqüentemente o bastidor CAMAC está livre para uso.

Devido às características peculiares das duas linhas de aquisição, já discutidas anteriormente, sabemos que a linha do ADC114 necessita muito menos acesso ao controlador do bastidor que a linha do ADC NIM que necessita acessá-lo a cada evento convertido pelo ADC. Desta forma o programa de aquisição do ADC114 comandaria o “semáforo”, pois como o seu acesso é menos freqüente então o acionamento do “semáforo” não sobrecarregaria o sistema.

Pela estrutura do sistema de aquisição apresentado na figura 46, vemos que a comunicação entre os programas de aquisição não é permitida, logo o programa gerente deveria ser acionado para conduzir a comunicação entre os dois programas. Para tanto utilizou-se novamente das mensagens e dos sinais para efetuar a construção deste “semáforo” de acesso ao bastidor CAMAC.

Quando o programa de controle do ADC114 necessita efetuar um acesso ao bastidor CAMAC, ele manda uma mensagem ao gerente indicando esta necessidade. Se o programa do ADC NIM não estiver ativo o gerente responde à mensagem indicando assim que o bastidor está livre para uso. Agora se houver uma aquisição sendo executado no ADC NIM, então um

sinal é mandado ao programa indicando que este deve bloquear qualquer acesso ao bastidor CAMAC. Neste momento o programa do ADC NIM desabilita o relógio de tempo vivo, pois ele ficará “cego” por alguns momentos, e manda uma mensagem ao gerente indicando que está pronto para esperar. Ao receber esta mensagem o gerente retorna a mensagem do programa do ADC114 e este finalmente pode efetuar o seu acesso ao bastidor CAMAC.

Ao término do acesso ao bastidor pelo programa do ADC114, ele manda uma nova mensagem ao gerente indicando que o bastidor está livre para acesso. Neste momento o gerente manda um novo sinal ao programa do ADC NIM que reativa o relógio de tempo vivo e manda um procurador para si mesmo indicando que o bastidor está novamente livre e a aquisição pode ocorrer normalmente.

O atendimento de um sinal pelo programa do ADC NIM interrompe o seu funcionamento, mas este continua normalmente ao término da rotina de tratamento do sinal recebido. Para que não seja recebido nenhum sinal durante um acesso ao bastidor por este programa, nestes momentos a sua prioridade é elevada ao máximo permitido a este tipo de programa, de tal forma a inibir o gerente a mandar o sinal. Desta forma ao término do seu acesso ao bastidor, quando a prioridade deste programa volta ao normal, o gerente conseguirá enviar o sinal e o processo de bloqueio de acesso ao bastidor ocorrerá. A figura 59 mostra esquematicamente o processo de bloqueio do acesso ao bastidor do programa do ADC NIM. A figura 60 mostra a carta de tempo deste processo de onde vemos que o tempo máximo, utilizado pelo sistema, é da ordem de $35\mu\text{s}$.

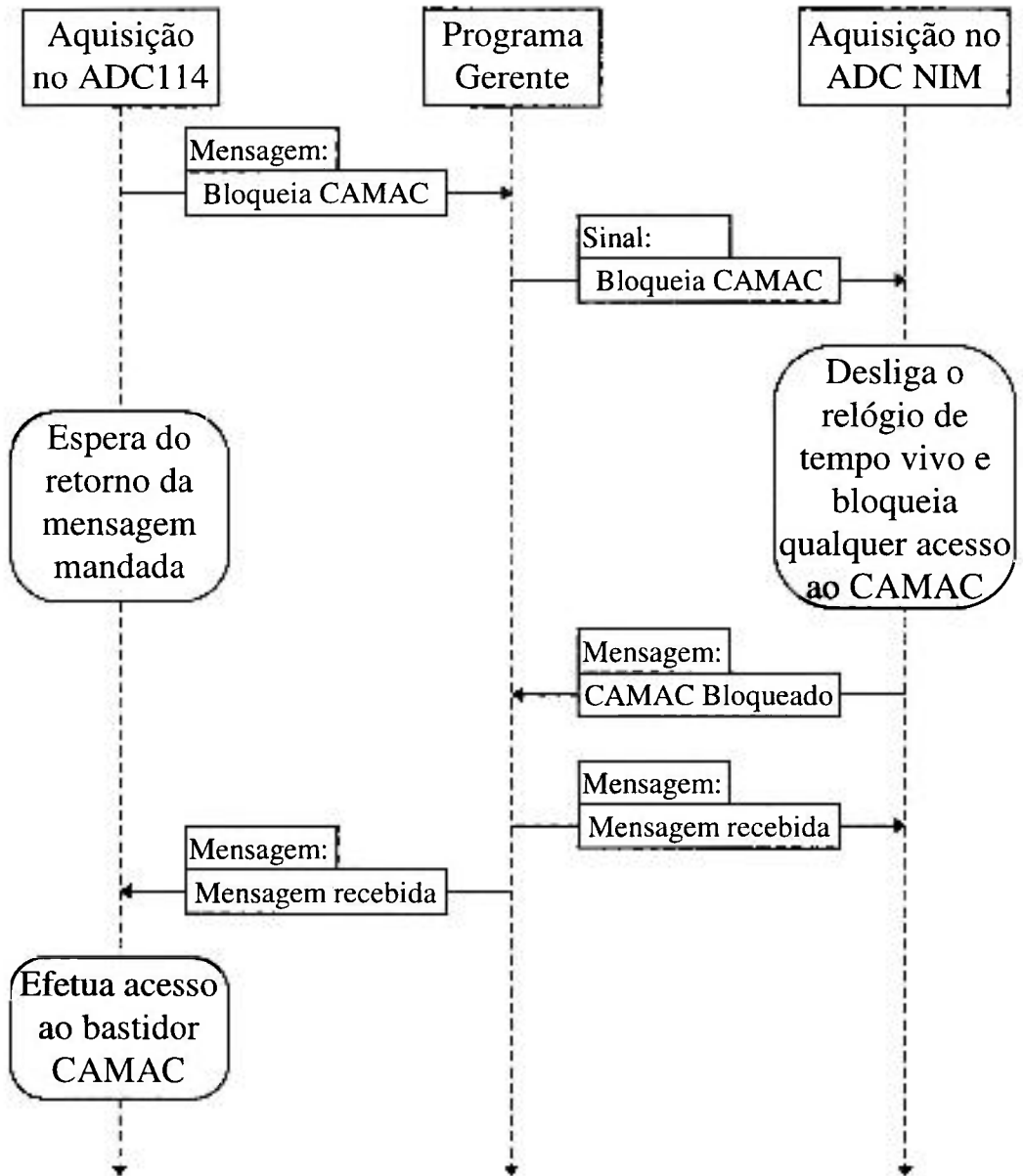


Figura 59 - Esquema das mensagens e sinais que compõem o “semáforo” de acesso ao CAMAC. Nesta figura vemos o processo de bloqueio requerido pelo programa do ADC114 e efetuado pelo programa do ADC NIM.

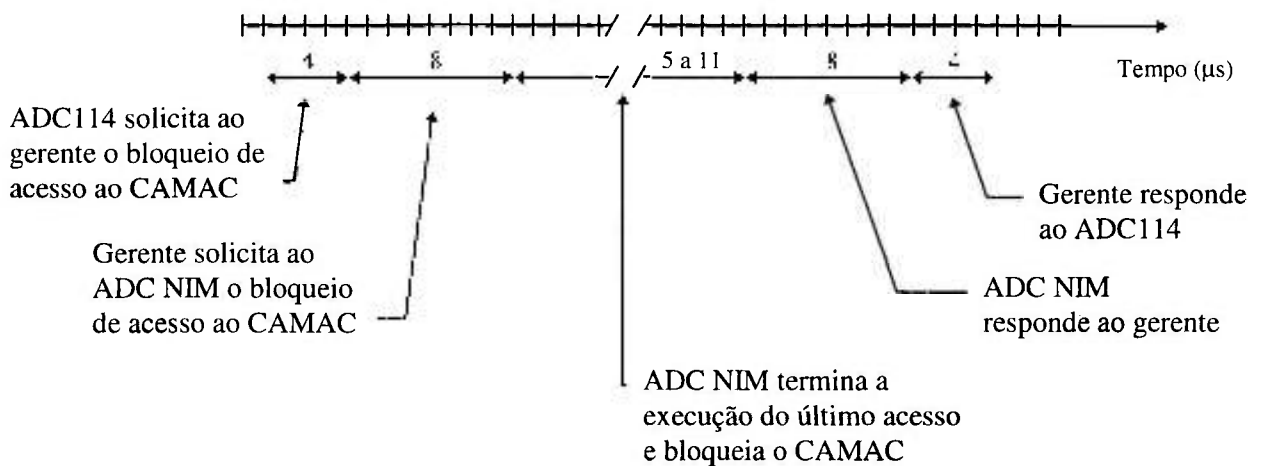


Figura 60 - Carta de tempo do bloqueio de acesso ao CAMAC pelo programa de aquisição do ADC NIM. Se no momento em que recebe a solicitação o ADC NIM estiver efetuando a inicialização do sistema o tempo será máximo, 11 μs , senão o tempo típico é da ordem de 5 μs . O tempo para o desbloqueio é igual ao mínimo, pois neste caso o programa do ADC NIM está no estado de espera e a sua resposta será imediata.

5 Conclusão

O presente trabalho apresenta o desenvolvimento de um sistema de aquisição de dados completamente novo em vários aspectos. O primeiro é o desenvolvimento do controlador de bastidor CAMAC padrão ISA que introduziu um novo patamar em termos de desempenho no acesso ao *Dataway*. O segundo aspecto é a introdução do módulo MULTI, que permite a montagem do sistema MULTIDETECTOR que permitirá a ampliação da gama de experimentos de coincidência possíveis de serem realizados no LAL. O desenvolvimento de um equipamento tão complexo como este trouxe uma bagagem de conhecimentos que vão do projeto e construção eletrônica, ao desenvolvimento de programas de aquisição que permitem o acesso e visualização de um número de parâmetros, passando pelos fenômenos novos físicos que se tornaram acessíveis pela utilização deste equipamento.

Em terceiro lugar está o desenvolvimento dos programas de visualização e controle do novo sistema de aquisição de dados uniparamétricos que permitiu a substituição dos antigos sistemas bem como introduziu uma melhora significativa; tanto no sentido de facilidade de utilização como também no desempenho, aumento da taxa de aquisição e na precisão do controle do tempo vivo, das medidas efetuadas.

Cabe ressaltar que o módulo MULTI, mesmo em seu protótipo, e os programas desenvolvidos para a sua utilização serviram para efetuar as medidas de trabalhos de mestrados^{[GU98], [NA99]} e doutorado^[AS98] já concluídos e a concluir. O novo sistema de aquisição está sendo utilizado nos testes de montagem do MULTIDETECTOR com quatro detetores e será utilizado muito em breve para medidas.

Para o futuro ainda há muitas propostas de desenvolvimento, como o aprimoramento do controlador CAMAC, acrescentando novas funções e técnicas de projeto e construção. No módulo MULTI a solução para os circuitos que geram o sinal de *Stop* para o TDC deve ser encontrada, para evitar o uso de cabos de atraso que, além de incômodos, diminuem a amplitude dos sinais. Quanto aos programas de aquisição uniparamétrica, a implementação do RPC (*Remote Process Control*) permitindo que a interface seja executada em outro computador que não o de aquisição, possibilitará um maior desempenho dos programas de aquisição, bem como facilitará a execução da interface de qualquer ponto da rede Internet sem o pesado tráfego necessário para a execução da interface remotamente.

Finalmente, o desenvolvimento de um programa de aquisição para o sistema MULTIDETECTOR nos moldes do programa de aquisição uniparamétrica é imprescindível, visto que o programa atual só permite o uso de até quatro detetores devido às limitações do sistema operacional utilizado, o DOS. Assim, um novo programa, escrito sobre o QNX, com as interfaces gráficas para controle e visualização dos parâmetros adquiridos se faz necessário para a montagem completa do sistema.

Portanto, vemos que o objetivo do presente trabalho: o desenvolvimento de um sistema de aquisição de dados que substituísse o antigo sistema do LAL, foi alcançado e em muitos aspectos ultrapassado.

REFERÊNCIAS BIBLIOGRÁFICAS.

1. **ACCEL Schematic, User's Guide and Reference** - Accel Technologies, Inc., 1996
2. ASAI, M.; KAWADE, K.; YAMAMOTO, H.; OSA, A.; Koizumi, M.; Sekine, T. - **A five-HPGe detector system for γ - γ angular correlation measurements for mass-separated short-lived nuclei** - Nuclear Instruments and Methods in Physics Research A 398, p. 265-275, 1997
3. ASSUNÇÃO, M. P. M. - **Espectroscopia gama e correlação angular gama-gama em ^{188}Os** - Tese de Doutorado, Instituto de Física da USP, São Paulo, 1998, USP/IF/SBI - 083/98
4. BÄRG, D. B.; VANIN, V.R.; NEVES, R. C. - **ISA-to-CAMAC interface** - Contribuição à XX Reunião de Trabalho sobre Física Nuclear no Brasil, Guaratinguetá, 1997
5. BLOOMER, J. - **Power Programming with RPC** - O'Reilly Associates, Inc., Cambridge, 1992
6. COHENCA, J. M. - **Manual do Programa de Aquisição Bidi** - Universidade de São Paulo, Instituto de Física, Laboratório do Acelerador Linear, São Paulo, 1983

7. EGGBRECHT, L. C. - **Interfacing the IBM personal Computer** - 2^a Edição, Howard W. Sans & Co., 1990

8. GUIMARÃES, Z. O. - **Medidas precisas de energias de transições gama em coincidência: espectroscopia das séries do ^{232}U e ^{233}U** - Dissertação de Mestrado, Instituto de Física da USP, São Paulo, 1998, USP/IF/SBI - 081/98

9. HUNT, C. - **TCP/IP Network Administration** - O'Reilly & Associates, Inc., Cambridge, 1992

10. IEEE - **Standard Modular Instrumentation and Digital Interface System (CAMAC) (ANSI/IEEE std 583-1975)** - The Institute of Electrical and Electronics Engineers, Inc., 1975

11. **Jorway Model 203 Dataway Display** - Jorway Corporation, New York, 1997

12. KERNIGHAN, B. W.; RITCHIE, D. M. - **C linguagem de programação** - Editora Campus, Rio de Janeiro, 1990

13. KNOLL, G. F. - **Radiation Detection and Measurement** - 2^a Edição - John Wiley & Sons. Inc., Singapura, 1989

14. **Maestro II Operator's Manual**, EG&G ORTEC, 1990

15. MALAFRONTTE, A. A.; PASSARO, A.; RADOVICH, A.; BONINI, A. L.
- **Low Cost Gamma-Ray Acquisition / Analysis System** - Revista de Física Aplicada e Instrumentação, 11 (2), p. 71, 1996

16. MATEY, J. R.; STEIN, P. - **Catching the right bus I : An overview of architectures for instrument systems** - Computer in Physics, 7 (2), p. 130-135, 1993

17. MENDONÇA, A.; ZELENOVSKY, R. - **PC e Periféricos, um guia completo de programação** - Editora Ciência Moderna Ltda, Rio de Janeiro, 1996

18. **Model 462 Time Calibrator, Operating and Service Manual**, EG&G ORTEC, 1972

19. **Model 463 Constant Fraction Discriminator, Operating and Service Manual**, EG&G ORTEC, 1971

20. **Model 800 Analog to Digital Converter, Operating and Service Manual**, EG&G ORTEC, 1971

21. **Model 2203, Death Time Counter** - BiRa Systems, Inc., 1973

22. **Model 2228A Octal Time-to-Digital Converter, Operator's Manual**, LeCroy, 1994
23. **Model 2322, Dual 24-bit Parallel Input Register - BiRa Systems, Inc.**, 1973
24. **Model 8701 ADC, User's Manual - Canberra Industries**, 1997
25. **Model AD114 CAMAC 16K ADC, Operating and Service Manual**, EG&G ORTEC, 1996
26. **Model AD413A CAMAC Quad 8K ADC, Operating and Service Manual**, EG&G ORTEC, 1990
27. **Model HM413 CAMAC FERAbus Histogramming Memory Hardware Manual**, EG&G ORTEC, 1996
28. **NASCIMENTO, E. do - Estudo do alargamento doppler dos gamas da aniquilação elétron-pósitron - Dissertação de Mestrado**, Instituto de Física da USP, São Paulo, 1999, USP/IF/SBI - 046/99
29. **NEVES, R. C.; BÄRG, D. B.; PASCHOLATI, P. R.; VANIN, V.R. - CAMAC controlado por um dispositivo ISA - Contribuição à XXI Reunião de Trabalho sobre Física Nuclear no Brasil**, Itatiaia, 1998

30. Open Software Foundation - **OSF/Motif Style Guide** - Prentice Hall, Inc., New Jersey, 1993
31. **QNX Operating System - System Architecture** - QNX Software Systems Ltd., 1996
32. ROSA, M. C. S.; PINTO, R. D.; VANIN, V. R.; PASCHOLATI, P. R. - **Sistema de Aquisição de Dados Multidetector** - Contribuição à XV Reunião de Trabalho sobre Física Nuclear no Brasil, Caxambú, “Programas e Resumos”, P-21, p. 42, 1992
33. SHANLEY, T.; ANDERSON, D. - **ISA System Architecture** - Mindshare Inc., 3^a Edição, Reading, Massachusetts, 1995
34. SMITH, R. - **Aprendendo PostScript Guia Visual**, Editora Berkeley, Rio de Janeiro, 1993
35. SOBELL, M. G. - **A practical guide to the UNIX system**, Addison-Wesley Publishing Company, 3^a. Edição, Menlo Park, Califórnia, 1994
36. **SPECTRA, User Guide** - Cadence Design Systems, Inc., 1997
37. VANIN, V.R. - **Estudo dos Decaimentos do ^{101m}Rh e ^{101f}Rh e Estrutura Nuclear do ^{101}Ru** , PhD Thesis, Instituto de Física da USP, São Paulo, 1984, USP/IF-B35-84

38. VANIN, V. R.; AICHE, M. - Nuclear Instruments and Methods in Physics Research, A284, p. 452-458, 1989

39. **Watcom, Compiler & Tolls** - QNX Software Systems Ltd., 1996

Apêndice 1

O controlador CAMAC-ISA possui endereços de leitura e escrita e interrupções de *hardware* para comunicação com o microcomputador. Aqui apresentamos de forma mais detalhada todos os endereços e interrupções disponíveis.

Escolhido o endereço BASE entre os disponíveis (300, 320, 340, 360 hexadecimal) a comunicação pode se efetuar através dos quatro endereços seguintes:

- BASE + 0 → FNA → 16bits → Escrita e Leitura

Escrita: Neste endereço será escrita a função FNA (F = Função, N = estação do bastidor, A = sub-endereço no módulo em questão) que será mandada ao controlador para execução.



F_n → Função a ser executada

N_n → Número da estação a ser endereçada

A_n → Sub-endereço da estação endereçada

Leitura: Quando efetuada leitura neste endereço, o controlador retornará o seu STATUS.



- Q → *Response Q*
 X → *Command Accepted*
 B → *Busy*
 I → *Inhibit*
 C → *Clear*
 Z → *Initialize*
 ΣL → Existência de um LAM (“ou” de todos os LAMs do bastidor)
 F → Falha - ciclo ISA não aguardou ciclo CAMAC terminar

- BASE + 2 → DADOS → 16bits → Escrita e Leitura

Escrita: Dados a serem escritos no *Dataway*. Dois bytes de menor ordem.

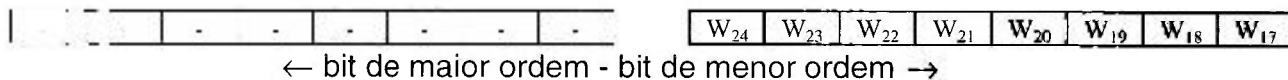


Leitura: Dados lidos do *Dataway*. Dois bytes de menor ordem



- BASE + 4 → LAM & DADOS → 16bits → Escrita e Leitura

Escrita: *Byte* de maior ordem a ser escrito no *Dataway*



Leitura: *Byte* de maior ordem lido do *Dataway* e máscara de LAM



- BASE + 6 → LAM → 16bits → Leitura

Leitura: Status de LAM



Além destes endereços o controlador deveria poder gerar interrupções de *hardware* ao receber determinados sinais de LAM. Desta forma definiu-se que as seguintes linhas de interrupção do microcomputador são acessíveis ao controlador.

- IRQ 5 :
Ativada quando há um pedido de LAM por qualquer estação do bastidor.
- IRQ 9 :
Associada ao pedido de LAM, de uma estação (1 a 6).
- IRQ 10 :
Associada ao pedido de LAM, de uma estação (7 a 12).
- IRQ 11 :
Associada ao pedido de LAM, de uma estação (13 a 18).
- IRQ 15 :
Associada ao pedido de LAM, de uma estação (19 a 23).

A seleção de qual estação pode gerar interrupção, em cada faixa, deve ser através de *jumper* no próprio controlador.