TÉCNICAS DE PROJETO UTILIZADAS NA CONSTRUÇÃO DO SUPERMICROCOMPUTADOR PEGASUS-32X E DO SISTEMA OPERACIONAL PLURIX

Newton Faller, Manuel Lois Anido, Pedro Salenbauch
Núcleo de Computação Eletrônica
Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brasil

ABSTRACT

O sistema PEGASUS-32X é uma família de supermicrocomputadores de 32 bits com múltiplos processadores baseado na linha 68000 da Motorola. O PLURIX é um sistema operacional compatível com o UNIX da AT&T, especialmente desenvolvido para aproveitar o multiprocessamento do PEGASUS-32X.

Este trabalho apresenta as características de ambos os sistemas, dando-se ênfase especial às técnicas de projeto utilizadas na sua construção.

O PEGASUS-32X e o PLURIX estão em desenvolvimento no Núcleo de Computação Eletrônica da UFRJ e mostram a capacitação tecnológica brasileira para desenvolver um supermicrocomputador e um sistema operacional orientado para multiprocessamento.

INTRODUÇÃO E MOTIVAÇÃO

O sistema PEGASUS-32X é uma família de supermicrocomputadores de 32 bits com múltiplos processadores baseado na linha 68000 da Motorola. O PLURIX é um sistema operacional compatível com o UNIX da AT&T, especialmente projetado para o máximo aproveitamento dos recursos computacionais do PEGASUS-32X (múltiplos processadores, espaço de endereçamento de 16Mbytes, processadores de E/S, etc.).

Este projeto foi motivado pela necessidade de se dar continuidade ao desenvolvimento tecnológico da área de Informática no Brasil através da construção de um sistema computacional com porte equiparável e até superior aos superminicomputadores importados existentes no país. A reserva de mercado amparada pelo governo já propiciou a independência tecnológica na área dos microcomputadores de 8 e 16 bits.

Trabalho publicado no VI Congresso Chileno de Engenharia Elétrica, Santiago,Chile,nov.85

Com o surgimento dos microprocessadores de 32 bits e dos "chips" de memória de alta densidade viabilizou-se a construção de computadores de baixo custo e alto desempenho, sem ter que recorrer à importação de tecnologia.

O PLURIX está em desenvolvimento, segundo as especificações do UNIX da AT&T, cujo núcleo possui uma interface muito bem definida e conhecida. Uma versão inicial para o controle de múltiplos processadores está prevista para o final de 1985.

O PEGASUS-32X

Descrição Geral

O PEGASUS-32X representa uma família de supermicrocomputadores, homogêneos, simétricos, de 32 bits, construídos com diversas unidades de processamento (UCP's) da família MC68000 operando em paralelo (Multiprocessamento), Unidades de Processamento Periférico (UPP's) inteligentes para entrada e saída (E/S), memória global e barramento de intercomunicação VME bus.

Na fase de concepção levou-se em consideração que a construção de sistemas de multiprocessamento homogêneos (aqueles em que os módulos básicos como UCP's, Unidades de Memória (UM) e Unidades de Processamento Periférico (UPP's) podem ser compartilhados indistintamente, apresentam uma série de vantagens sobre outros, tais como:

- o aumento do desempenho (porte) do sistema é imediato, já que ele é composto de módulos homogêneos e, portanto, basta replicá-los;
- quanto mais homogêneo, mais robusto (menos sujeito a falhas), pois ainda restam recursos semelhantes, quando um falha;
- a utilização de um módulo básico em cada parte do sistema multiprocessador diminui os custos de projeto, fabricação e programação;
- a homogeneidade propicia uma só versão do "software", independente do porte do sistema;
- o tempo de desenvolvimento é muito menor, pois o desempenho não está relacionado à complexidade dos módulos, mas à sua quantidade.

A simetria dos módulos do sistema permite uma melhor utilização dos recursos computacionais, tanto em operações de cálculo, como em E/S.

Todas as UCP's, Unidades de Processamento Periférico (UPP's) e Unidades de Memória (UM's) são interconectados pela barra global do sistema (VME bus).

Todas as UPP's e UM's são endereçadas de forma idêntica por todas as UCP's. Isto significa que um programa pode ser executado em qualquer um dos processadores, sem qualquer modificação no modo de endereçar a memória ou as Unidades de Processamento Periférico. Da mesma forma, todas as UPP's possuem uma visão simétrica das UCP's e da memória.

Esta simetria é também explorada para melhor a confiabilidade do sistema. Quando o mesmo é ligado, cada módulo realiza um auto-teste e reporta o resultado a um dos processadores, que é designado como meste. Esta UCP mestre inicia a execução de código do Sistema Operacional, que determina o número de componentes do sistema que passaram nos auto-testes e configura o Sistema, baseado nos componentes operacionais.

Desde o início do projeto, deu-se uma atenção especial à capacidade e facilidade de auto-teste de cada um dos módulos do sistema. Isto provou ser de enorme valia, pois os testes dos módulos foram feitos em paralelo e independentemente, concorrendo para o rápido desenvolvimento do sistema.

Técnicas Adotadas na Construção do "Hardware" do PEGASUS-32X:

O Objetivo.

Construir uma máquina de 32 bits, utilizando os atuais microprocessadores de 16/32 bits, que pudesse competir em desempenho com o VAX-780 ou o IBM 4341-MG2, a um custo inferior e que utilizasse o sistema operacional PLURIX. Esta máquina deveria possuir também uma elevada capacidade de auto-teste e replicabilidade.

Características Gerais do PEGASUS-32X:

- suporte para memória virtual;
- gerência de memória para relocação e proteção de programas em ambientes multi-usuários e multi-tarefas;
- UCP's da família MC68000, com "cache", para controle do Sistema Operacional e programas de usuário;
- instruções de 8,16 ou 32 bits;
- barramento "VME bus", com caminhos de 32 bits para dados e 24 bits para endereço;

- taxa de transferência de dados na barra de 6,7 MBps (1);
- "cache" de 4Kbytes por UCP;
- velocidade de 1,2 Mips (2) (1 UCP) a 3 Mips (4 UCP's);
- unidades de processamento periférico , (UPP's) inteligentes, utilizando o micro processador Z80-A;
- as placas são auto-testáveis;
- rápida reconfiguração em caso de falhas, devido à existência de diversos processadores e à modularidade do projeto.

A Figura 1 compara o PEGASUS-32X com outros superminis.

A Família de Microprocessadores Adotada.

Após estudar diversos microprocessadores, concluímos que a linha MC680XX da Motorola era a mais adequada, pois:

- apresentava arquitetura de 32 bits;
- apresentava instruções, modos de endereçamento, tratamento de exceções compatíveis com os atuais superminis;
- possuía alta velocidade de operação;
- permitia a evolução para microprocessadores mais potentes;
- apresentava dispositivos auxiliares disponíveis (MMU, FPP, etc.).

(1) MBps - milhões de bytes por segundo;
(2) Mips - milhões de instruções por segundo.

O Barramento do Sistema.

O barramento escolhido foi o VME bus, pois além de ser de 32 bits e não proprietário, é suportado pela MOTOROLA e por uma série de companhias européias. Além disso apresenta protocolo assíncrono (permite maior velocidade) e diversos níveis de interrupção e de controle de acesso à barra.

As Unidades de Memória.

Por questões de modularidade, confiabilidade e expandibilidade, não existe um controle único para o sistema de memória, mas um controle distribuído, onde cada placa pode evoluir de 256Kbytes até 1Mbyte quando pastilhas de 64K bits são utilizadas. Com as novas pastilhas de 256K bits uma unidade de memória poderá conter 4Mbytes.

	IBM 4341-9	IBM 4341-12	DIGITAL VAX-780	HP3000 68	CHARLES RIVER UNIVERSE 66/05	PEGASUS-32X 321
DESEMPENHO RELATIVO IBM-370/158-3 = 45	24	76	62	64	77	77
MIPS	0,4	1,2	1,06	1,1	1,25	1,25
TAMANHO DE MEMÓRIA (Mb)	1-4	2-16	1-32	3-8	0,5-3	1-15
Nº DE CANAIS E/S	3-6	6	1-8	3-48	1-3	1-6
CACHE	[illegible]	16K	8K	8K	4K	4K
ESTRUTURA DE BUS	NÃO	NÃO	SIM	SIM	SIM	SIM
FONTE: COMPUTERWORLD, SET. 84						

FIGURA 1: CARACTERÍSTICAS DO PEGASUS-32X E SISTEMAS SIMILARES.

As Unidades de Processamento Periférico.

Tendo como um dos objetivos mais importantes o desempenho, procurou-se aliviar a carga dos processadores centrais em tarefas de E/S, tais como discos, impressoras e terminais. Isto foi feito colocando-se um processador especializado em cada um dos módulos. Isto contribuiu também para facilitar o auto-teste.

As Unidades Centrais de Processamento.

Para serem compatíveis em desempenho e potencialidade com as UCP's dos superminis, elas deveriam ter gerência de memória e "cache".

A gerência de memória do PEGASUS-32X é construída com integrados da linha TTL-SHOTTKY, pois caso utilizássemos a MMU-68451 da MOTOROLA, o sistema ficaria mais lento. Também foi construído um sistema de memória virtual para poder rodar programas maiores que a memória física.

O "CACHE" faz-se necessário, por dois motivos:

1) Os microprocessadores são muito mais rápidos que o sistema de memória, logo foi necessário ter uma memória muito rápida, interna às UCP's;

ii) Não seria possível um sistema multiprocessador, sem "cache" nas UCP's, devido ao congestionamento do barramento do sistema.

"CACHE" de Blocos do Disco.

Num sistema com diversas UCP's, executando muitos programas, a solicitação de disco é muito grande e normalmente torna-se o gargalo do sistema. Para resolver este problema, as unidades de interface de disco possuem um "cache" de blocos do disco, com capacidade de até 256Kbytes.

"WATCH-DOGS" nas UCP's.

Para fazer com que os dados dos "caches" e da memória principal fossem sempre consistentes, construiu-se um mecanismo de vigilância das transferências na barra. Se alguma unidade do sistema alterar o conteúdo da memória principal e este conteúdo residir em algum "cache", tal(is) posição(ões) do(s) "cache(s)" será(ão) invalidada(s). Isto possibilita implementar multiprocessamento e torna desnecessária a invalidação de todo o "cache" em transferências de acesso direto à memória.

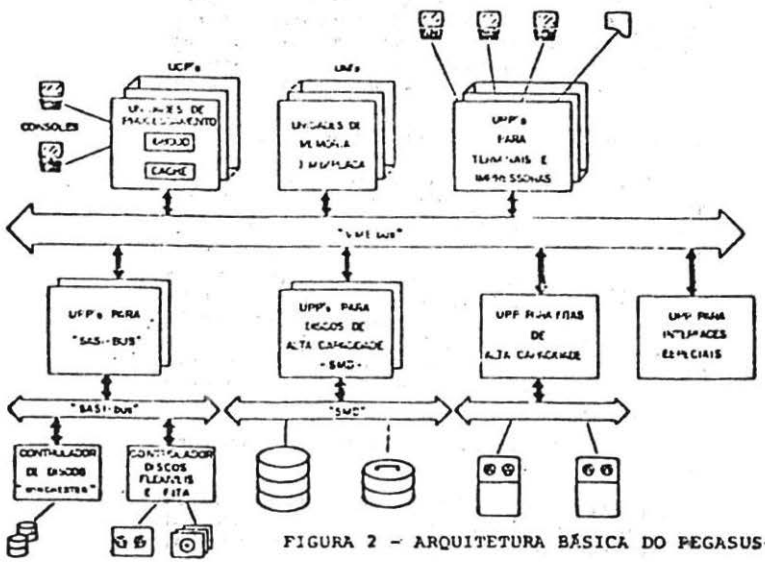

FIGURA 2 - ARQUITETURA BÁSICA DO PEGASUS-32X

Arquitetura do Sistema

Os sistemas da família PEGASUS-32X são formados a partir de múltiplos módulos escolhidos de cada um dos 7 tipos básicos, que são:

a) Módulo UCP com saída para duas consoles;

b) Módulo Unidade de Memória (UM) de até 1Mbytes;

c) Módulo UPP para 16 terminais e 2 impressoras (UPPTI);

d) Módulo UPP para "SASI bus" - permite ligar discos "Winchester", discos flexíveis e fitas do tipo "Streaming";

e) Módulo UPP para 4 discos SMD de alta capacidade;

f) Módulo UPP para fitas magnéticas;

g) Módulo UPP para interfaces especiais, como: "timer", calendário, Processador de Ponto Flutuante, etc.

Deve-se observar que uma configuração simples possuiria apenas 4 módulos, ou seja, um módulo UCP com console, um módulo de memória com até 1Mbytes, um módulo UPP para até 16 terminais e duas impressoras e um módulo UPP para "SASI bus".

Devido à alta homogeneidade do sistema, a evolução da família para a faixa de desempenho dos superminis se dá pela replicação dos módulos básicos.

O sistema utiliza-se como meio de comunicação entre os diversos módulos o barramento "VME bus" que é um padrão internacional.

A mesma filosofia de utilizar padrões tipo "VME bus" e "SASI bus" foi seguida nos controladores de disco de alta capacidade, onde se utiliza o barramento SMD.

A Figura 2 ilustra a arquitetura básica do sistema.

O SISTEMA OPERACIONAL PLURIX

Descrição Geral

O sistema PLURIX baseia-se nas especificações da versão 7 do sistema UNIX, possuindo diversas melhorias encontradas em outras versões, assim como características próprias. Está em andamento a compatibilização do PLURIX com o System V.

Entre as principais características do PLURIX podemos citar:

- suporte para o controle de múltiplos processadores;
- sistema de arquivos hierárquico, incluindo volumes montáveis;
- entrada/saída em arquivos, em dispositivos, e entre processos compatíveis entre si;
- ativação de processos assíncronos;
- seleção do interpretador de comandos do sistema a nível de usuário/aplicação;
- alto grau de portabilidade, sendo a grande maioria do sistema escrito numa linguagem de alto nível (C);
- chamadas às funções do sistema ("System Calls") padronizadas.

O Sistema de Arquivos

Do ponto de vista do usuário, são implementados quatro tipos de arquivos: arquivos regulares, diretórios, arquivos especiais e arquivos fifo.

Um arquivo regular contém qualquer informação colocada nele pelo usuário. Para o sistema estes arquivos não assumem nenhuma estruturação. São simplesmente uma seqüência contínua de bytes. O acesso a estes arquivos pode ser sequencial ou direto a qualquer byte do arquivo.

Os diretórios implementam o mapeamento entre o nome do arquivo e o arquivo propriamente dito, e assim descrevem a estrutura do sistema de arquivos como um todo. Um diretório pode ter um ponteiro ("Link") para qualquer tipo de arquivo. Qualquer arquivo que não seja um diretório pode ser apontado por mais de um diretório. Existe um diretório especial, reconhecido pelo sistema, o diretório "root", a partir do qual pode ser encontrado qualquer arquivo do sistema de arquivos, bastando para tal especificar a seqüência de diretórios a ser procurada.

A cada dispositivo físico de entrada/saída (disco, fita, linhas de comunicação, impressora, memória física, etc.) está associado um ou mais arquivos especiais. O acesso a estes arquivos para a execução de uma operação de E/S, causa, na realidade, a ativação física do dispositivo associado. No caso de dispositivos tipo disco, é possível dividí-los em várias unidades lógicas diferentes, e ter um ou mais arquivos especiais associados a cada uma delas.

Os arquivos fifo são arquivos que não possuem nenhuma informação associada. São utilizados para a comunicação entre quaisquer dois processos. Para tal, o processo consumidor deve abrir um arquivo fifo para leitura, enquanto que o processo produtor deve abrir o mesmo arquivo para escrita. Assim qualquer informação escrita no arquivo pelo processo produtor será passada ao processo consumidor.

Processos

Um processo é a execução de uma imagem, o ambiente de execução do computador. A imagem consiste basicamente de duas partes: a primeira contém várias informações necessárias para a execução do processo, tais como os valores dos registros do processador, registros de gerência, os arquivos abertos, a identificação do usuário, etc. A segunda contém a área de texto do programa sendo executado (o código), a área de dados e a "stack" do processo.

Um processo pode criar um outro processo assíncrono (System Call "Fork"). Este novo processo será uma cópia idêntica do processo antigo, isto é, imagem dos dois processos será a mesma. A diferença entre os dois será o valor retornado pelo System Call. Para o processo original (PAI) será retornado um número único identificando o processo criado (processo FILHO). Para o processo filho será retornado o valor zero.

Um processo pode substituir sua área de texto, dados e "stack" através do System Call "Exec". Este System Call recebe como parâmetros o nome do programa a ser executado e os seus parâmetros. Assim todo o texto e dados do processo são substituídos pelo conteúdo do arquivo, mas a primeira parte da imagem não é alterada. É bom observar que o processo nunca volta a executar o código que deu o "Exec", a não ser em caso de erro na chamada ao System Call.

Um processo pode esperar o fim da execução de um dos seus processos filhos através do System Call "Wait". Este System Call retorna a identificação do processo filho que terminou, assim como o status da terminação. Este status é passado pelo processo filho ao processo pai como parâmetro do System Call "Exit" que tem como função principal terminar um processo.

Para a comunicação entre dois processos "irmãos" existe também o mecanismo chamado "pipe". Um pipe é um canal de comunicação, que, como um arquivo fifo, permite a um processo escrever informação de um lado para ser lida por um outro processo. Para utilizar um pipe o processo deve criá-lo através do System Call "pipe", que será passado ao pro

cesso filho através do System Call Fork, já que um "pipe" é tratado como um arquivo comum.

Controle de Entrada/Saída

Os pedidos de E/S dos usuários passam por uma interface usuário-sistema operacional, sendo analisados e distribuídos a rotinas específicas de E/S (drivers) para os diversos periféricos do sistema.

Existem dois tipos básicos de rotinas de E/S: as que tratam periféricos tipo bloco, como discos, e as que tratam periféricos tipo caracter, como terminais.

As primeiras manuseiam um conjunto comum de buffers de tamanho fixo, que conterão os dados. Através de "cabeças de buffers" encadeadas, são mantidas listas de buffers associados a cada periférico e uma de buffers disponíveis.

As que tratam periféricos tipo caracter também manuseiam filas formadas pelo encadeamento de pequenos blocos, alocados e desalocados quando necessário, que conterão os caracteres. Devido a existência no PEGASUS-32X de Unidades de Processamento Periférico o manuseio de caracteres é feito de forma descentralizada liberando a UCP para funções mais importantes.

Os periféricos são identificados por dois códigos: um seleciona seu driver, através da entrada correspondente numa tabela de configuração do sistema. O outro é passado ao driver, para selecionar a unidade de um controlador, por exemplo.

Controle dos Processos

Os processos no PLURIX podem estar basicamente em três estados: RUN, READY e SLEEP. Um processo no estado RUN está rodando em algum dos processadores do PEGASUS-32X. Os processos no estado READY estão prontos para rodar e aguardam um processador livre. Em SLEEP, um processo aguarda um recurso que não é processador.

Os processos no estado READY, são organizados em filas ordenadas por prioridade. Garante-se que em qualquer instante de tempo os n processos prontos para rodar de maior prioridade estarão rodando nos n processadores disponíveis. Isto só pode não acontecer no caso de especificar-se que um determinado processo somente poderá ser executado por um processador específico. Em geral, um processo pode rodar indistintamente em qualquer dos processadores disponíveis.

O balanceamento da carga de trabalho é feita através da mudança dinâmica das prioridades a medida que os processos vão sendo executados. Um esquema de realimentação negativa garante a estabilidade e eficiência do sistema.

O esquema adotado para o multiprocessamento é o de total simetria em relação as UCP's. Desta forma qualquer trecho de código seja do usuário ou do supervisor pode rodar em qualquer das UCP's disponíveis. Por esta razão especial atenção foi dada aos recursos que exigem exclusão mútua.

Um esquema de semáforos que em seu nível mais baixo está baseado nas instruções de Test-and-Set dos processadores garantem a correta utilização de recursos mutuamente exclusivos. Especial atenção tem sido dada à instrumentação do sistema de forma a localizar e corrigir eventuais "dead locks".

Interface entre o Sistema e o Usuário

O PIURIX apresenta os mesmos comandos e utilitários básicos do UNIX, além de suas principais opções. O principal interpretador de comandos ("Shell"), possui as mesmas características básicas, que são: execução de programas com parâmetros; redirecionamento de entrada e saída; execução de arquivo de comandos.

Inicialização do Sistema

Após o término do auto-teste, a UCP0 entra na rotina de carga, que se encontra em ROM, enquanto que as outras UCP's entram numa rotina de espera de um comando da UCP0.

A rotina de carga é bastante inteligente, no sentido de poder efetuar a carga de qualquer dispositivo do sistema (Diskette, Winchester, etc.). Essa rotina também reconhece o sistema de arquivos do S.O., podendo carregar qualquer programa que se encontre nos dispositivos do sistema. Para tal, a rotina espera que o operador tecle na console o nome do arquivo e o periférico aonde este se encontra.

Carregado o S.O., este inicializa suas tabelas, força a criação do "Scheduler" (processo 0) e dos diversos "Dispatchers" (processos 1 até o número de UCP's). O processo "Init" é criado e o seu código executado criando-se um processo "Login" para cada terminal. O processo "Login" associado a um terminal é transformado num processo "Shell" quando um usuário é admitido no sistema através deste terminal.

Ferramentas de Depuração/Teste

Na fase inicial do desenvolvimento de um sistema baseado em microprocessador é de extrema importância a existência de um conjunto de ferramentas de diagnósticos e de depuração baseados em software à disposição da equipe dos projetistas do hardware do sistema.

O esquema de diagnosticar e depurar o hardware do sistema adotando o software como o seu instrumento principal apresenta uma série de vantagens, sendo a mais importante o fator tempo de depuração ser reduzido.

Tendo em vista não tornar o desenvolvimento das ferramentas de diagnósticos e de depuração parte do caminho crítico do projeto, definiu-se um programa de diagnósticos e um depurador contendo os elementos essenciais para assistir à fase inicial da depuração da UCP do PEGASUS-32X.

INTEGRAÇÃO "HARDWARE-SOFTWARE"

Desde o início do projeto, foi de importância fundamental a interação entre as equipes de "hardware" e de "software" para realizar uma tarefa desta envergadura.

Desta interação podemos citar alguns ítens, tais como:

- escolha das famílias de microprocessadores a utilizar;
- arquitetura do sistema;
- análise do conjunto de instruções para poder suportar as características do PLURIX;
- tratamento de erros no sistema;
- definição do sistema de gerenciamento da memória;
- definição dos "drivers" de entrada/saída.

DIFICULDADES E PERSPECTIVAS

As principais dificuldades foram, e são, como geralmente ocorre nas universidades, de ordem material e financeira. No entanto, foi possível implementar, em tempo muito curto, um primeiro protótipo e provar que é possível construir no Brasil supermicrocomputadores muito potentes, dispensando qualquer importação de tecnologia.

As perspectivas para o PEGASUS-32X são no sentido de se obter um desempenho maior ainda, desenvolvendo módulos que permitam, entre outras coisas:

- interligação de diversos PEGASUS-32X numa rede de alta velocidade (sistemas "loosely-coupled");
- interligação de diversos PEGASUS-32X pela expansão do barramento paralelo de cada um (sistemas "tightly-coupled").

O "cluster" de PEGASUS-32X assim obtido é um sistema distribuído e tem desempenho similar a máquinas de grande porte. O sistema operacional distribuído (PLURIX-D) para suportar a nova arquitetura já está sendo especificado.

CONCLUSÕES

O sucesso na construção do supermicro computador PEGASUS-32X e na implementação do sistema operacional PLURIX através das técnicas descritas neste trabalho provam a viabilidade da obtenção de sistemas computacionais de grande porte em países do Terceiro Mundo sem a necessidade de importação da tecnologia do projeto. Obviamente a dependência tecnológica na área de componentes ainda é apreciável. Isto, entretanto, parece ser um fator menos crítico já que existem inúmeros fabricantes internacionais que produzem circuitos similares não havendo, portanto, necessariamente, a dependência a um fabricante ou a um país específico.

O aproveitamento pela sociedade como um todo da tecnologia desenvolvida nas Universidades se faz através das indústrias nacionais. Seguindo a filosofia do NCE/UFRJ de passar para a indústria os protótipos construídos na Universidade espera-se para breve no mercado brasileiro a disponibilidade de sistemas PEGASUS-32X/PLURIX.

i) UNIX é marca registrada da AT&T Bell Laboratories - USA;

ii) PLURIX e PEGASUS-32X são marcas registradas do NCE/UFRJ - Brasil.

NEWTON FALLER - Ph.D. em Engenharia Elétrica e Ciência da Computação pela Universidade da Califórnia, Berkeley, U.S.A., Professor Adjunto da Universidade Federal do Rio de Janeiro (UFRJ) e Diretor da Divisão de Projetos e Equipamentos Digitais do NCE/UFRJ;
MANUEL LOIS ANIDO - M.Sc. em Engenharia Elétrica pela COPPE/UFRJ, Projetista de Sistemas Digitais Senior do NCE/UFRJ;
PEDRO SALENBAUCH - M.Sc. em Engenharia de Sistemas e Computação pela COPPE/UFRJ, Professor Adjunto da Universidade Federal do Rio de Janeiro (UFRJ), Analista de Sistemas Senior do NCE/UFRJ.