

UNIVERSIDADE FEDERAL DA PARAIBA

CENTRO DE CIENCIAS E TECNOLOGIA

CURSO DE MESTRADO EM ENGENHARIA ELETRICA

Executivo em Tempo Real
Para Controle de Processos

ROBERT RICE BRANDT

CAMPINA GRANDE, PB

OUTUBRO - 1985

Executivo em Tempo Real
Para Controle de Processos



ROBERT RICE BRANDT

Executivo em Tempo Real
Para Controle de Processos

Dissertação apresentada ao
Curso de MESTRADO EM ENGENHARIA
ELETRICA da Universidade
Federal da Paraíba, em
cumprimento às exigências para
obtenção do Grau de Mestre.

AREA DE CONCENTRAÇÃO: PROCESSAMENTO DA INFORMAÇÃO

WILSON GUERREIRO PINHEIRO

Orientador

JOSE HOMERO FEITOSA CAVALCANTI

Co-orientador

CAMPINA GRANDE, PB

OUTUBRO - 1985



B821e Brandt, Robert Rice.
Executivo em tempo real para controle de processos /
Robert Rice Brandt. - Campina Grande, 1985.
46 f.

Dissertação (Mestrado em Engenharia Elétrica) -
Universidade Federal da Paraíba, Centro de Ciências e
Tecnologia, 1985.
"Orientação : Prof. Wilson Guerreiro Pinheiro, Prof.
José Homero Feitosa Cavalcanti".
Referências.

1. Sistema Distribuído em Tempo Real. 2. Controle de
Processos. 3. Processamento da Informação. 4. Dissertação -
Engenharia Elétrica. I. Pinheiro, Wilson Guerreiro. II.
Cavalcanti, José Homero Feitosa. III. Universidade Federal
da Paraíba - Campina Grande (PB). IV. Título

CDU 621.3(043)

Executivo em Tempo Real
Para Controle de Processos

ROBERT RICE BRANDT

DISSERTAÇÃO APROVADA EM 10/12/85


WILSON GUERREIRO PINHEIRO

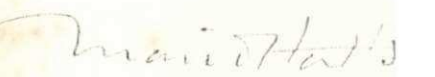
Orientador


JOSE HOMERO FEITOSA CAVALCANTI

Co-orientador


JOÃO MARQUES DE CARVALHO


Componente da Banca


MARIO TOYOTARO HATTON

Componente da Banca

CAMPINA GRANDE, PB

DEZEMBRO - 1985



AGRADECIMENTOS

Agradeço primeiramente a Deus que me tem dado saúde e força para o trabalho e, acima de tudo, me tem dado VIDA.

Agradeço também aos seguintes professores:

WILSON GUERREIRO PINHEIRO, pela orientação e revisão final do manuscrito;

JOSE HOMERO FEITOSA CAVALCANTI, pela orientação;

ANTONIO MARCUS NOGUEIRA LIMA, pela cooperação no desenvolvimento do Conversor A/D e pelas valiosas sugestões;

ALEX VIEIRA PINTO, pelas sugestões e comentários.

Agradeço também ao técnico:

LUCIO FERNANDO DIAS DA SILVA, pela cooperação no traçado dos desenhos.

DEDICATORIA

Ao meu filho DANIEL, que não
compreendeu a ausência do pai
durante o desenvolvimento
deste trabalho.

CONTEUDO

CAPITULO 1

INTRODUÇÃO	58.100 - Computação Gerencial - Capitulo 1	1
1.1 - SURGIMENTO DE CONTROLE DE PROCESSOS		1
1.2 - COMPUTADORES EM CONTROLE DE PROCESSOS		3
1.3 - CONTROLE DE PROCESSOS EM TEMPO REAL		4
1.4 - PROGRAMAS GERENCIADORES		5
1.4.1 - Classificações		5
1.4.2 - Características Essenciais		6
1.5 - OBJETIVOS DESTE TRABALHO		7

CAPITULO 2

DESENVOLVIMENTO DE SOFTWARE		8
2.1 - INTRODUÇÃO		8
2.2 - EXECUTIVO EM TEMPO REAL		9
2.2.1 - Descritores		9
2.2.2 - Atualizador de Descritores		11
2.2.3 - Acionador de Tarefas		14
2.3 - SUBROTINAS DE APOIO GERAL		15
2.3.1 - Comunicações com o Operador		15
2.3.2 - Subrotinas de Posicionamento do Cursor		17
2.4 - TAREFAS DO SISTEMA		18
2.4.1 - Tarefas de Menu		19
2.4.2 - Tarefas de Relógio		20
2.4.3 - Tarefa de Display de "Buffer"		21
2.4.4 - Tarefas de Display dos Descritores		21
2.4.5 - Tarefas que Modificam Descritores		22
2.4.6 - Tarefas do Conversor Analógico-Digital		23
2.5 - LISTAGEM DOS PROGRAMAS		23

CAPITULO 3

DESENVOLVIMENTO DE HARDWARE		24
3.1 - INTRODUÇÃO		24
3.2 - CONVERSOR A/D		25
3.3 - INTERFACE		25
3.4 - FONTE DE TENSÃO DE REFERENCIA		28
3.5 - CONDICIONADOR DE SINAL		28
3.6 - CIRCUITO DE AMOSTRAGEM E RETENÇÃO		29

CAPITULO 4

CONCLUSÃO		33
---------------------	--	----

BIBLIOGRAFIA	37
------------------------	----

APENDICE A

RECURSOS DO SISTEMA UTILIZADO	39
A.1 - RECURSOS DE HARDWARE	39
A.1.1 - Placa da Unidade Central de Processamento	39
A.1.2 - Placa de Memória Volátil (RAM)	40
A.1.3 - Placa de Interface para Unidade de Disco Flexivel	40
A.1.4 - Placa da Fonte de Alimentação	40
A.1.5 - "Backplane"	41
A.2 - RECURSOS DE SOFTWARE	41
A.2.1 - Editor de Textos (WordStar)	41
A.2.2 - Macro Assembler (M80)	43
A.2.3 - Linker (L80)	43
A.2.4 - Depurador de Programas (ZSID)	44

APENDICE B

VALORES DOS CAMPOS DOS DESCRITORES	45
--	----

LISTA DE FIGURAS

Fig. 1.1 - Diagrama de blocos simplificado de um controle de processo típico	2
Fig. 2.1 - Diagrama de estados dos processos	11
Fig. 2.2 - Diagrama de blocos do atendimento ao RTC	13
Fig. 3.1 - Diagrama de blocos do conversor A/D	26
Fig. 3.2 - Conversor A/D	29
Fig. 3.3 - Circuito da fonte de tensão de referência	30
Fig. 3.4 - Tensão de referência versus temperatura	30
Fig. 3.5 - Circuito do condicionador de sinal	31
Fig. 3.6 - Circuito do amostrador	32
Fig. A.1 - Barramento do CAMAÇARI	42



LISTA DE TABELAS

Tabela 2.1 - Conteúdo dos descritores das tarefas do sistema	10
Tabela 4.1 - Tempos máximos e mínimos de execução do programa executivo	34
Tabela B.1 - Valores normais dos campos das tarefas .	46

RESUMO

Este trabalho apresenta o desenvolvimento de um programa executivo, preemptivo e com prioridade mista, para controle de processos em tempo real à base do microprocessador Z80. Diversas rotinas básicas, entre as quais relógio, display de descritores e de aquisição de dados, são descritas. O desenvolvimento de um conversor analógico-digital para aquisição de dados de processos externos é apresentado em detalhe.

ABSTRACT

This work presents the development of a real time, preemptive executive, with mixed priority, for use in process control applications, using the Z80 microprocessor. Various routines are described, such as: clock routine, data acquisition and descriptor display. The development of an analog-to-digital converter for data acquisition of external processes is presented in detail.

CAPITULO 1

I N T R O D U Ç Ã O

No suor do teu rosto comerás
o teu pão... (Genesis 3:19)

1.1 - SURGIMENTO DE CONTROLE DE PROCESSOS

Desde os tempos mais remotos, o homem tem procurado meios de simplificar ou aliviar o seu trabalho. Ao longo dos tempos, tem usado escravos, animais de carga e, mais recentemente, produtos de sua própria engenharia (embarcações a vela, roda d'água, máquinas a vapor, veículos motorizados, etc.) para trabalhar em seu lugar, ou seja, tem procurado canalizar energias para alcançar objetivos ou obter produtos de seu interesse (processos externos) e controlar os parâmetros para que o resultado final seja o mais próximo possível do desejado.

Em suma, o homem tem, ao longo do tempo, usado sua engenhosidade para controlar processos (externos) utilizando tipicamente o conceito descrito na figura 1.1. O resultado da comparação feita entre o valor obtido e o valor desejado deve atuar junto ao processo (atuador) para fazer um ajuste corretivo no processo (controle de processo) para que o valor obtido se aproxime do valor desejado.

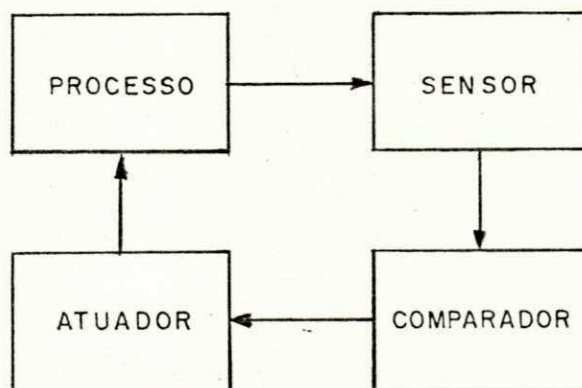


Fig. 1.1 Diagrama de blocos simplificado de um controle de processo típico

A primeira evidência de controle automático de uma máquina foi o regulador centrífugo, inventado por James Watt em 1788 [1], dispositivo estritamente mecânico, que permitia controlar automaticamente a velocidade de uma máquina a vapor.

Neste século, já no fim da década de 40, iniciou-se o uso de controladores pneumáticos nas indústrias de processos [2] e, a partir da década de 60, foram introduzidos dispositivos semicondutores nos controladores de processos, dispositivos estes que foram ficando cada vez mais complexos com o aparecimento de circuitos integrados, de amplificadores operacionais e, finalmente, de computadores.

1.2 - COMPUTADORES EM CONTROLE DE PROCESSOS

Com o desenvolvimento de circuitos eletrônicos a semicondutores mais complexos, tem-se conseguido colocar sistemas inteiros, como é o caso dos microprocessadores, em uma única "pastilha" de silício. O baixo custo dos microprocessadores tem facilitado a sua atuação na área de controle de processos. Pelo fato de os computadores serem bastante semelhantes entre si na comunicação com o mundo externo e de já existirem muitos computadores versáteis no mercado, atualmente a pesquisa se está voltando para a parte de programação (software) dos microcomputadores para uso em controle de processos, apesar de ainda haver lugar para pesquisa na área de hardware.

Classicamente, existem três modos de incluir um computador na malha de controle de um processo [2]:

- "off line";
- "on line" em malha aberta;
- "on line" em malha fechada.

No modo "off line", o computador recebe os dados do operador, faz os cálculos necessários e devolve os resultados ao operador. Em seguida, o operador faz os ajustes necessários no processo para que os resultados se aproximem do valor desejado.

No modo "on line" em malha aberta, o computador recebe os dados diretamente do processo, mas os resultados ainda são entregues ao operador para fazer os ajustes.

Finalmente, no modo "on line" em malha fechada, o computador assume todas as tarefas de aquisição dos dados do processo, bem como o processamento destes dados e o controle das variáveis do processo para obter os valores desejados.

1.3 - CONTROLE DE PROCESSOS EM TEMPO REAL

O termo "controle de processos em tempo real" caracteriza normalmente um sistema de controle que utiliza um computador "on line" em malha fechada. O processo é controlado pelo computador sem necessidade de intervenção do operador, a não ser para mudança eventual do ponto de operação. Geralmente, o computador pode emitir mensagens ou relatórios ao operador enquanto controla o processo. Normalmente, a comunicação com o mundo externo (entrada e saída) é feita através de interrupção do processador. Em alguns sistemas, o computador também verifica se algum valor está fora dos limites e avisa ao operador a condição de alarme ou de perigo [2].

Em sistemas de computador para atuação em tempo real, é comum a utilização de um programa principal gerenciador e de vários sub-programas de execução ou tarefas [3].

Nestes sistemas também é necessário um Relógio de Tempo Real (RTC - Real Time Clock) que gere interrupções periódicas [2]. A escolha da frequência deste RTC influencia na eficiência do sistema como um todo e também determina o tempo máximo de execução de uma tarefa sem que

esta seja interrompida (e possivelmente suspensa) pelo sistema.

1.4 - PROGRAMAS GERENCIADORES

1.4.1 - Classificações

Os programas gerenciadores de processos em tempo real, também chamados de executivos em tempo real [4], podem ser classificados em termos de:

- prioridade de execução das tarefas;
- suspensão temporária de execução das tarefas.

Quanto à prioridade de execução das tarefas, os programas gerenciadores se classificam em:

1) Sem prioridade, quando não existe prioridade diferenciada entre as tarefas [5].

2) Com prioridade estática, quando as prioridades são diferentes, mas fixas pelo programa original, ou seja, não mudam durante a execução do programa.

3) Com prioridade dinâmica, quando a prioridade pode mudar ao longo da execução do programa.

4) Com prioridade mista, quando coexistem prioridades estáticas e dinâmicas no mesmo sistema de computação.

Em relação à execução ou suspensão de tarefas, os programas gerenciadores se classificam em:

- Não preemptivos, quando executam cada tarefa completamente sem aceitar interrupções durante sua execução;

- Preemptivos, quando as tarefas podem ser interrompidas, podendo não manter a sua execução.

1.4.2 - Características Essenciais

Os programas gerenciadores precisam ser executados várias vezes por segundo, devendo, portanto, ser compactos e rápidos. Normalmente, são escritos em linguagem "assembly", que permite maior velocidade de execução em comparação com outras linguagens. As tarefas (sub-programas de execução) podem ser escritas em "assembly" ou em linguagem de alto nível (FORTRAN, PASCAL, C, etc.), dependendo da complexidade dos algoritmos e da rapidez de execução necessária.

Em sistemas de controle de processos em tempo real, em que há variáveis compartilhadas nas comunicações entre tarefas, o programa gerenciador necessita de algum dispositivo para garantir sincronização nessas comunicações [5],[6]. Dois tipos de sincronização são usualmente utilizados: exclusão mútua e sincronização condicional [5].

Exclusão mútua garante que uma seqüência de programa é tratada como operação indivisível. Em sincronização condicional, a execução de um programa é "retardada" até serem satisfeitas certas condições exigidas pelo programa.

1.5 - OBJETIVOS DESTE TRABALHO

A idéia original deste trabalho surgiu da necessidade de se controlar alguns parâmetros (como temperatura, pressão, iluminamento, depósito de reagentes) de uma câmara ambiental em desenvolvimento no Laboratório de Pesquisas Multidisciplinares do Departamento de Engenharia Elétrica da Universidade Federal da Paraíba. Como o projeto da câmara propriamente dita e dos vários atuadores não se completaria em tempo hábil, optou-se, neste trabalho, pelo desenvolvimento de um sistema de controle de processos em tempo real mais geral que poderia ser aplicado posteriormente a esse caso específico com um mínimo de modificações.

Especificamente, este trabalho objetiva:

- O desenvolvimento de um executivo em tempo real para controle de processos;
- o desenvolvimento de algumas tarefas básicas;
- projeto e implementação de um conversor analógico-digital para aquisição de dados.

CAPITULO 2

DESENVOLVIMENTO DE SOFTWARE

Porque na muita sabedoria há muito enfado; e o que aumenta em ciência, aumenta em trabalho. (Eclesiastes 1:18).

2.1 - INTRODUÇÃO

O executivo em tempo real e algumas tarefas básicas foram desenvolvidos na linguagem "assembly" do microprocessador Z80 da Zilog [7], cujas principais características são baixo custo e rapidez no atendimento às interrupções. Esta opção deveu-se principalmente ao fato de o Departamento de Engenharia Elétrica dispor de um sistema CAMAÇARI, da CALCON TECNOLOGIA S. A. (Vide Apêndice A), baseado no Z80 e dispor de facilidades como editor de textos, macro assembler, depurador de programas, terminal de vídeo, unidade de disco flexível e impressora, além de informações técnicas detalhadas sobre o hardware.

O software desenvolvido é composto de três partes distintas: executivo em tempo real, subrotinas de apoio geral e tarefas.

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso, 882 - Tel (083) 321-7222-R 355
58.100 - Campina Grande - Paraíba

2.2 - EXECUTIVO EM TEMPO REAL

O executivo em tempo real, também chamado de gerenciador ou escalonador de tarefas, é o programa principal, que tem por função o gerenciamento, em tempo real, de todos os processos envolvidos. Optou-se, neste trabalho, pelo desenvolvimento de um executivo preemptivo, com prioridade mista, aplicável em sistemas multi-tarefas, e que utiliza mecanismos de sincronização para garantir exclusão mútua no acesso às operações sobre variáveis compartilhadas (regiões críticas).

O programa desenvolvido age sobre os descritores de cada tarefa, ativando e desativando as tarefas. O executivo está dividido em duas partes distintas: atualizador de descritores e acionador de tarefas.

2.2.1 - Descritores

Os programas necessários a cada processo externo são agrupados em tarefas do sistema, sendo que cada tarefa tem um descritor contendo informações indispensáveis ao executivo. Os descritores são tabelas, ligadas em lista, de modo que cada descritor contém o endereço do descritor seguinte, à exceção do último descritor da lista, que contém o endereço zero. No sistema desenvolvido, cada descritor contém 11 (onze) bytes, divididos nos campos descritos a seguir e apresentados sucintamente na Tabela 2.1.

O LINK é um apontador de dois bytes que aponta para o endereço do início do próximo descritor.

O registrador de frequência (FREQ) contém um valor que, multiplicado pelo período do RTC, dá o tempo entre acionamentos da tarefa. Quando uma tarefa não depende do relógio do sistema, ou seja, não é periódica no tempo, o registrador de frequência (FREQ) deve ser zero.

O registrador de contagem (CONT) contém o número de interrupções do relógio desde o último acionamento da tarefa.

Tabela 2.1 - Conteúdo dos descritores das tarefas do sistema.

CAMPO	DESCRIÇÃO	NUMERO DE BYTES
LINK	Apontador para o próximo descritor	2
FREQ	Registrador de frequência	2
CONT	Registrador de contagem	2
STAK	Apontador de pilha da tarefa	2
STAT	Status da tarefa	1
PRIO	Prioridade inicial da tarefa	1
DECR	Decremento do status da tarefa	1

O apontador de pilha da tarefa (STAK) aponta para a locação de RAM onde são armazenados todos os valores dos registradores na hora da suspensão da tarefa, bem como os valores do contador de programa nas chamadas de subrotinas.

O status da tarefa (STAT) é um byte que indica o

estado da tarefa (se inativa ou pronta) e sua prioridade (no caso de a tarefa estar pronta para execução). Observe-se que, neste sistema, a tarefa em execução é considerada pronta (Vide figura 2.1).

O byte da prioridade (PRIO) indica o valor inicial da prioridade de uma determinada tarefa. Este valor será transferido para o campo de status da tarefa (STAT) quando a tarefa passar do estado inativo para pronto.

O decremento do status da tarefa (DECR) é um byte usado para modificar a prioridade de uma tarefa a cada interrupção do relógio (RTC). Para tarefas periódicas que devam terminar antes de um determinado tempo ("deadline"), deve ser calculado um decremento que garanta que a tarefa seja executada antes do "deadline".

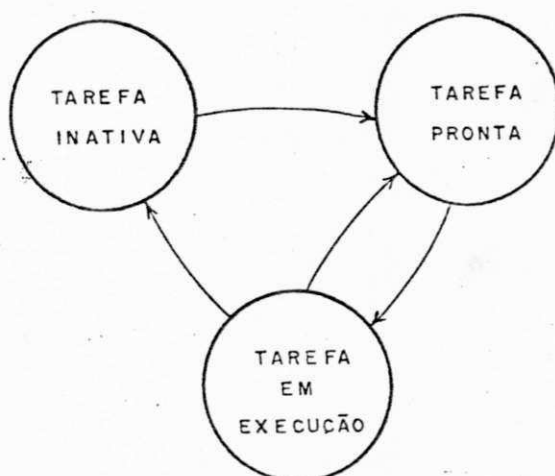


Fig. 2.1 Diagrama de estados dos processos

2.2.2 - O Atualizador de Descritores

O atualizador de descritores de tarefas é inicializado por interrupções periódicas do relógio (RTC).

Logo após a ocorrência da interrupção do RTC, o atualizador guarda todas as informações pertinentes à tarefa interrompida na pilha atual de dados. Em seguida, faz um teste para saber se interrompeu o acionador de tarefas ou alguma outra tarefa específica. Neste sistema, não é necessário gerenciar as ocorrências de interrupções do tipo INT (Interrupt Request) do microprocessador Z80, visto que o RTC está ligado ao pino INT desse microprocessador e não ao pino NMI (Non-Maskable Interrupt). Não se optou pelo uso de interrupção do tipo NMI neste sistema por causa da complexidade adicional que acarretaria no desenvolvimento do executivo.

No caso de o atualizador ter interrompido uma tarefa (e não o acionador de tarefas), o valor do apontador de pilha é armazenado no descritor da tarefa interrompida.

Após ter armazenado todas as informações pertinentes à execução anterior, o programa atualizador passa a fazer a modificação nos descritores. O atualizador age sobre os campos CONT e STAT dos descritores.

Para saber se o valor contido no registrador de contagem (CONT) deve ser alterado, o atualizador examina o valor contido no registrador de frequência (FREQ) para saber se é igual a zero ou não (Vide figura 2.2). Se esse valor for diferente de zero, o atualizador incrementa o valor contido em CONT e compara o resultado com o valor contido em FREQ. Se o valor contido em CONT for maior ou igual ao contido em FREQ, o atualizador faz uma comparação

entre os valores contidos em STAT e PRIO. Se o valor contido em STAT for maior que o contido em PRIO, são realizados dois passos: 1) o conteúdo de CONT é decrementado do valor contido em FREQ; 2) o valor contido em PRIO é transferido para STAT.

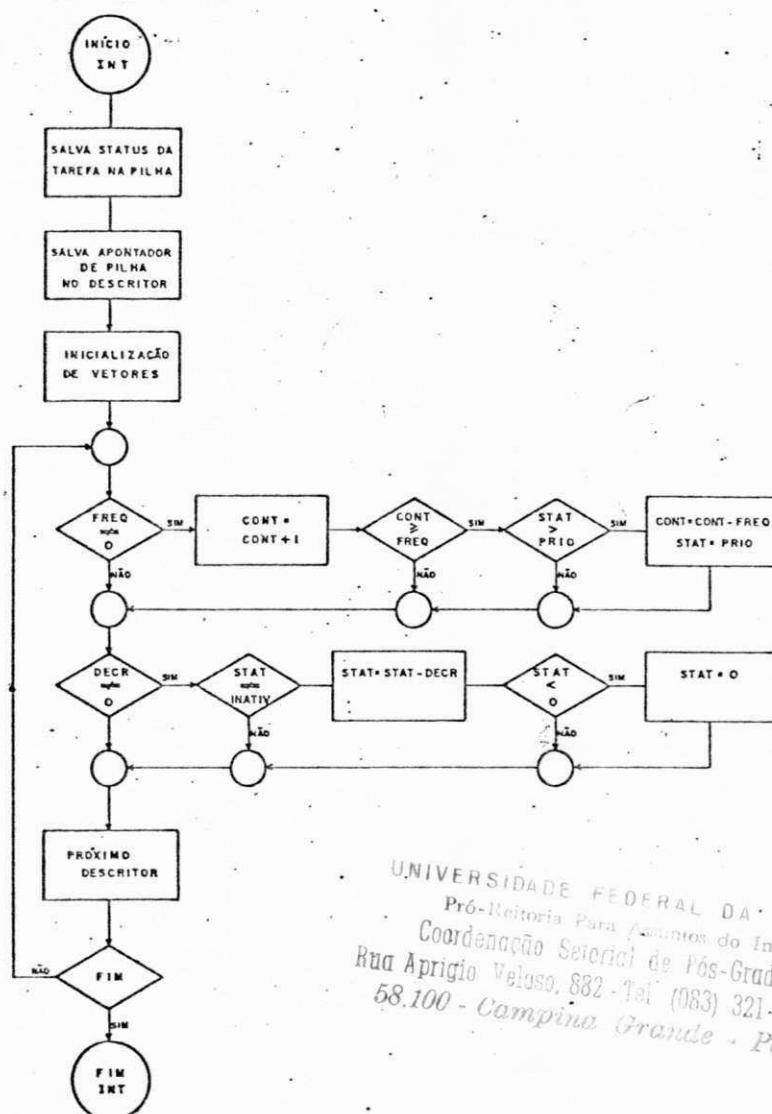


Fig. 2.2 - Diagrama de blocos do atendimento ao RTC.

O segundo parâmetro verificado pelo atualizador é o conteúdo do decremento do status da tarefa (DECR). Se o conteúdo de DECR for zero, nada é modificado. Por outro

UNIVERSIDADE FEDERAL DA PARAÍBA
 Pró-Reitoria Para Assuntos do Interior
 Coordenação Setorial de Pós-Graduação
 Rua Aprígio Veloso, 882 - Tel (083) 321-7222-R 355
 58.100 - Campina Grande - Paraíba.

lado, se o conteúdo de DECR for diferente de zero, o valor do status da tarefa (STAT) é verificado para saber se a tarefa está pronta para execução. Sendo atendidas estas exigências, o valor do status é decrementado do valor contido no DECR. Se o valor do status resultante for menor que zero, ou seja, se houver um transporte ("carry") gerado na subtração, o valor contido em STAT é feito igual a zero, que é a prioridade máxima do sistema.

A utilização do campo DECR no sistema permite a variação dinâmica do status das tarefas, aumentando a prioridade efetiva da tarefa pronta para execução a cada interrupção do RTC.

2.2.3 - Acionador de Tarefas

O acionador de tarefas, programa que decide qual a próxima tarefa a ser executada, é executado ao término de cada tarefa e após a execução do atualizador de descritores de tarefas. Este programa foi desenvolvido de tal modo a fazer uma varredura em todos os descritores, em busca da tarefa de maior prioridade, a qual, uma vez localizada, é executada logo a seguir. Se forem encontradas mais de uma tarefa com essa mesma prioridade, será executada a que for encontrada primeiro. Se não for encontrada nenhuma tarefa que esteja pronta para execução, o acionador será novamente executado.

2.3 - SUBROTINAS DE APOIO GERAL

Foram desenvolvidas algumas subrotinas de uso geral para tornar mais eficiente o desenvolvimento das tarefas em si. Estas subrotinas facilitam a comunicação com o operador, o posicionamento do cursor no vídeo e a formatação dos dados.

2.3.1 - Comunicações com o Operador.

Neste sistema, usa-se um terminal de vídeo, ligado através de linhas seriais no padrão RS-232-C, para todas as comunicações com o operador. As subrotinas de comunicação estão divididas em dois grupos: 1) subrotinas de teclado; 2) subrotinas de vídeo.

A recepção de caracteres na linha serial vinda do teclado é feita pela DART (Dual Asynchronous Receiver/Transmitter) [7], periférico que transforma os dados seriais em palavras paralelas e gera interrupção da tarefa em execução. Dentro da subrotina de atendimento à interrupção, o carácter recebido é colocado em um "buffer" na memória e os vetores que apontam para o "buffer" são atualizados. A seguir, a interrupção é finalizada.

Os caracteres armazenados podem, então, ser retirados do "buffer" pela subrotina "CONIN" (Console IN). Qualquer tarefa que necessite de caracteres, poderá chamar esta subrotina para retornar com um carácter. Apenas uma tarefa que chama esta subrotina deverá estar habilitada,

isto é, pronta para execução, a cada instante, para não haver interferência entre as tarefas.

A transmissão de caracteres ao video é feita de modo análogo. A tarefa que tem algum caracter a transmitir chama a subrotina "CONOUT" (Console OUT). Esta subrotina coloca o caracter no "buffer" de transmissão e habilita a DART a interromper o processador se a DART estiver pronta para transmitir outro caracter. Como podem existir várias tarefas de video prontas para execução a qualquer instante, e é necessário manter as mensagens separadas umas das outras (por causa da posição especifica no video), foi utilizado um mecanismo de sincronização para garantir exclusão mútua [6] no acesso à subrotina "CONOUT" enquanto uma tarefa estiver transmitindo uma mensagem.

O controle de cada "buffer" é feito por dois vetores: um que aponta para o local onde podem ser acrescentados caracteres e outro que aponta para o lugar de onde devem ser retirados caracteres do "buffer". Quando os dois vetores estão apontando para o mesmo endereço, significa que o "buffer" está vazio.

Observe-se que as regiões destas subrotinas que alteram os apontadores de "buffer" são protegidas por desabilitação de interrupções do tipo INT, impedindo, assim, a alteração dos apontadores por uma subrotina de atendimento à interrupção.

Esta técnica de "buffers" foi escolhida porque aumenta a eficiência de tarefas que enviam mensagens ao video ("buffer" do video), evita a perda de um caracter

digitado ("buffer" do teclado), e também por poder ser adaptada à maioria dos sistemas operacionais existentes.

2.3.2 - Subrotinas de Posicionamento do Cursor

Este trabalho foi desenvolvido utilizando áreas diferentes da tela (do video) para mensagens diferentes. Cada mensagem deve começar com o posicionamento do cursor no local desejado da tela antes de enviar a mensagem. Assim, pode-se colocar o relógio na linha superior do lado direito, o menu principal em outra linha, etc.

Ao longo do desenvolvimento do trabalho, por motivos superiores, foi necessário utilizar terminais de video de diversos fabricantes, tais como: VT100 e VT52 da Digital (U.S.A), Sagita-150 da Scopus (Brasil), VCM-12 da Volker-Craig (Canadá). Como o tamanho das telas e os comandos de posicionamento do cursor são diferentes para os diversos terminais, optou-se, neste trabalho, pelo desenvolvimento de subrotinas de apoio dedicadas ao posicionamento do cursor. Foram aproveitados alguns conceitos do editor de textos (WordStar) que estava sendo usado na escrita dos textos.

Foi desenvolvido um conjunto de subrotinas para auxiliar no posicionamento do cursor e saída de caracteres. As subrotinas constantes deste conjunto são as seguintes:

POSCUR - posicionamento do cursor;

SLETRA - posicionamento do cursor e saída da letra encontrada no acumulador "A";

SAICHR - saída de uma letra sem posicionamento do cursor, mas com atualização dos registradores de posicionamento;

SAIDE - saída da letra apontada pelo registrador "DE" sem posicionamento do cursor, mas com atualização dos registradores de posicionamento;

SAISTR - posicionamento do cursor e saída do "string" apontado pelo registrador "DE" onde o primeiro caracter é o número de bytes no "string";

BINBCD - conversão de binário para "BCD" até 199, onde unidades e dezenas estão contidas em "A" e centenas, no bit do "carry";

HXASCI - transformação dos 4 bits menos significativos do acumulador "A" (formato HEX) em um byte ASCII;

OUTAHX - transformação do valor HEX contido em "A" para 2 bytes ASCII;

OUT2HX - transformação do valor HEX apontado pelo registrador "DE" para 2 bytes ASCII;

OUT4HX - transformação de dois bytes HEX consecutivos, apontados pelo registrador "DE", para 4 bytes ASCII.

2.4 - TAREFAS DO SISTEMA

Neste trabalho, entende-se como tarefas do sistema as rotinas de atendimento específico aos processos externos.

Foram desenvolvidas várias tarefas básicas para servir de modelo para as tarefas específicas de um sistema

de controle dedicado em malha fechada. Descrições sucintas das tarefas básicas desenvolvidas são apresentadas nas subseções seguintes.

2.4.1 - Tarefas de Menu

As funções possíveis do sistema são apresentadas na forma de menus, tendo em vista suas características interativas. Desta forma, é solicitada apenas uma letra na escolha de cada função. Durante o funcionamento normal do programa, o menu principal é apresentado na tela. Quando o operador solicita alguma alteração, pode ou não apresentar um menu secundário ou outra instrução, dependendo da alteração solicitada. Estas mensagens auxiliares aparecem apenas enquanto o operador fornece os dados solicitados para completar a função desejada. Depois disto, reaparece o menu principal.

As apresentações de menu consistem basicamente da chamada da subrotina SAISTR (que posiciona o cursor e envia o "string" apontado por "DE" ao console) e da chamada de CONIN (que espera um caracter do console). Uma vez especificada a função pelo operador, através da digitação de uma letra, esta é comparada com as letras contidas em uma tabela das funções possíveis. Se for encontrada a função especificada nesta tabela, a tarefa correspondente é ativada. Se não, permanece o menu principal e o sistema espera outro comando.

2.4.2 - Tarefas de Relógio

Um programa básico desenvolvido diz respeito à colocação de um relógio (em tempo real) no video. Este programa está dividido em quatro tarefas distintas: 1) atualização periódica do relógio; 2) apresentação do relógio no video; 3) acerto do relógio; 4) liga/desliga a tarefa de apresentação do relógio no video.

A tarefa de atualização periódica do relógio usa locações em RAM (memória volátil) para armazenar os valores do relógio. No programa desenvolvido, esta tarefa é executada a cada 40 interrupções do RTC, ou seja, a cada segundo, visto que o período do RTC é de 25 ms. Após a atualização das locações em RAM, esta tarefa ativa a tarefa de apresentação no video.

A tarefa de apresentação do relógio no video envia ao console (video) o "string" de posicionamento do cursor e, em seguida, o conteúdo das locações do relógio em RAM. Ao final da tarefa, outro "string" de posicionamento do cursor é enviado, para colocar o cursor na posição de "default".

A tarefa de acerto do relógio aceita seis dígitos para a atualização do relógio. A cada par de números digitados, o registrador correspondente é atualizado. A ordem de digitação deve seguir o seguinte formato: dois dígitos para hora, dois dígitos para minuto e dois dígitos para segundo. Digitando um retorno (CR) a qualquer instante, finaliza o comando imediato.

A tarefa liga/desliga apresentação do relógio atua no descritor da tarefa apresentação do relógio no vídeo, alterando-o de "baixa prioridade" para "inativa". Acionado pela segunda vez, altera de "inativa" para "baixa prioridade", e assim sucessivamente.

2.4.3 - Tarefa de Display de "Buffer"

Outro programa abordado é o que trata da impressão do "buffer" de recepção de dados. Este programa pode ser dividido em duas partes distintas, que não necessitam de prioridades diferentes. Portanto, optou-se por usar apenas uma tarefa para executar todo o programa. A primeira parte da tarefa diz respeito ao teste se realmente há caracteres no "buffer" ou não ("buffer" vazio). A segunda parte da tarefa faz o display propriamente dito (obviamente, se houver caracteres no "buffer"). Esta segunda parte copia o conteúdo do "buffer" de entrada no vídeo, sem alterar o conteúdo desse "buffer".

2.4.4 - Tarefas de Display dos Descritores

Outro conjunto de programas elaborados é o de display dos descritores. Em um sistema dedicado de controle de processos, este programa provavelmente só irá constar durante os testes das tarefas. Uma vez colocado o sistema em funcionamento, estas tarefas provavelmente serão desativadas. Este programa foi dividido em duas

tarefas independentes, com prioridades diferentes. A primeira delas, com alta prioridade, colhe os dados e coloca-os em um "buffer" em RAM. A segunda, com prioridade baixa, envia estes dados colhidos para o video.

Da mesma forma que foi incluída no conjunto de tarefas de relógio uma tarefa para ligar e desligar o display do relógio, foi incluído no menu principal um comando para ligar e desligar o display dos descritores. Após a ativação do comando, qualquer tecla digitada desliga o display dos descritores, volta ao menu principal e tenta interpretar novo comando. Nestas tarefas, foram incluídos apenas os parâmetros julgados mais importantes para o operador (o LINK, por exemplo, não foi incluído).

2.4.5 - Tarefas que Modificam Descritores

Outro programa desenvolvido, que está ligado aos descritores de tarefas, é o que altera o conteúdo dos descritores. Este programa foi dividido em cinco tarefas distintas, mas poderia ser apenas uma continuação da tarefa do menu principal.

A primeira destas tarefas aceita um par de dígitos para especificar qual o descritor a ser modificado e, em seguida, apresenta um menu secundário que indica as alterações possíveis em cada descritor. A resposta à solicitação deste menu ativa uma das outras tarefas que irá aceitar o número de dígitos necessários e colocá-los no registrador certo dentro do descritor especificado.

Deve-se tomar muito cuidado ao modificar os

valores nos descritores porque uma modificação errada poderá desativar o sistema. Apenas poucos valores poderão ser alterados sem causar danos ao programa (Vide Apêndice B).

2.4.6 - Tarefas do Conversor Analógico-Digital

Foram desenvolvidas duas tarefas e uma subrotina para atendimento ao conversor A/D. A primeira tarefa, cuja finalidade é iniciar a conversão do canal especificado, é acionada no menu principal e aceita um dígito hexadecimal para a especificação do canal a ser convertido. No final da conversão, o conversor interrompe o microprocessador. A subrotina de atendimento à interrupção do conversor A/D permite ao microprocessador executar os seguintes passos: 1) leitura do valor convertido e da polaridade do canal; 2) armazenamento na RAM dos valores lidos em registradores; 3) acionamento da tarefa de display dos valores convertidos.

A outra tarefa desenvolvida envia ao vídeo o valor convertido. Se o valor analógico for negativo, esta tarefa envia o sinal de polaridade negativa antes do valor convertido.

2.5 - LISTAGEM DOS PROGRAMAS

A listagem dos programas desenvolvidos encontra-se à disposição dos interessados na Coordenação dos Cursos de Pós-Graduação em Engenharia Elétrica da UFPb, Campus II.

CAPITULO 3

DESENVOLVIMENTO DO HARDWARE

Melhor é uma mão cheia com descanso do que ambas cheias com trabalho e aflição de espirito. (Eclesiastes 4:6).

3.1 - INTRODUÇÃO

Visando tornar possível a aquisição de dados de processos externos, foi desenvolvido um cartão para o microcomputador CAMAÇARI contendo um conversor analógico-digital e circuitos auxiliares (fonte de tensão de referência, amostragem e retenção, condicionador de sinal, interface).

Para que um controlador digital possa receber os dados sobre o estado atual do processo sob seu controle, é essencial dispor de um circuito de interface que permita a aquisição destas informações, ou seja, de um circuito que transporte para o controlador os valores das variáveis que caracterizam a operação do processo. Tal circuito pode ser genericamente classificado como um conversor. No caso em que os dados dos processos externos são analógicos, faz-se necessário utilizar um conversor que transforme as grandezas da forma analógica para uma representação digital, ou seja, um conversor analógico-digital (conversor A/D).

3.2 - CONVERSOR A/D

Uma vez que atualmente já existem conversores A/D sob a forma integrada, foi adotada a solução de implementar uma placa de conversor A/D a partir de um conversor integrado disponível no comércio (ADC0817). Esse circuito integrado utiliza a técnica de aproximação sucessiva para efetuar a conversão do sinal analógico numa palavra binária de 8 bits. No ADC0817, há 16 canais multiplexados e um único conversor A/D com tempo típico de conversão de 100 μ s [8]. O tempo de conversão medido experimentalmente foi de 64 μ s, para um sinal de relógio de 1 MHz.

3.3 - INTERFACE

Na placa desenvolvida, cujo diagrama funcional se encontra na Fig. 3.1, além do conversor integrado, foram utilizados: uma PIO - Parallel Input/Output [7] para transferência dos dados para o microprocessador; uma fonte de tensão de referência; um "latch" de 16 canais para armazenar o bit de polaridade de cada canal no instante da amostragem; um multiplexador de 16 canais para informar a polaridade do sinal convertido; os circuitos lógicos de seleção do cartão; um circuito divisor binário; e dois multivibradores monoestáveis (um para o sinal de amostragem e outro para o sinal de fim de conversão).

Os circuitos lógicos de seleção do cartão, os

multivibradores monoestáveis e o divisor binário programável foram implementados com circuitos integrados TTL do tipo LS para evitar sobrecargas ao barramento.

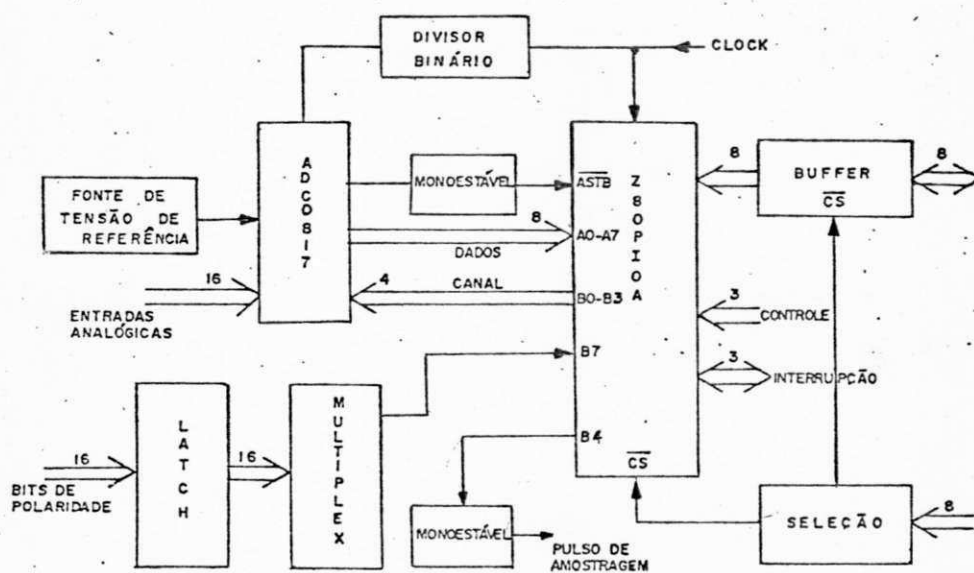


Fig. 3.1 Diagrama de blocos do conversor A/D

A informação de polaridade, selecionada por multiplex adicional na placa do conversor, é gerada pelo circuito de condicionamento de sinal e armazenada no "latch". Este multiplexador de polaridade permite a utilização de todos os oito bits do ADC0817 na representação da magnitude (256 níveis), garantindo uma resolução de $1/256$ da tensão de referência.

A transferência dos dados do conversor para o microprocessador é feita através do canal A da PIO - Parallel Input/Output (AO - A7), enquanto que o bit de polaridade é transferido por um bit do canal B (B7). Deste modo, a leitura de um valor convertido é feita em

duas etapas: leitura do bit de polaridade e leitura do módulo. A ordem destas duas leituras não tem importância na operação do conversor A/D. A especificação do canal a ser convertido é feita no canal B (B0 - B3).

O pulso de comando que aciona os circuitos de amostragem é obtido a partir de um multivibrador monoestável ligado ao pino B4 da PIO. A duração do pulso fornecido pelo multivibrador monoestável é definida pelo tempo de amostragem do circuito de amostragem e retenção. No presente caso, este pulso tem a duração de 13 microssegundos. Para dar início a uma operação de amostragem, é necessário efetuar uma operação de escrita na PIO, produzindo uma transição de 0 para 1 no pino B4.

O pulso de início de conversão é gerado por software. O programa desenvolvido envia, através do pino B5 da PIO, uma transição de 0 para 1 para preparar o conversor para converter e uma transição de 1 para 0 para inicializar a conversão propriamente dita. O tempo em nível 1 tem de ser, no mínimo, 200 ns para o conversor inicializar a conversão [8].

Ao fim de cada conversão, há uma transição de 0 para 1 no sinal EOC (End of Conversion) do ADC0817. Esta transição é processada em um multivibrador monoestável para gerar o pulso que informa à PIO o fim da conversão (pulso ASTB). A PIO, se estiver programada adequadamente, ao receber este sinal, interrompe o microprocessador, que deve, então, processar o valor binário convertido.

Deste modo, para a operação correta do cartão do conversor A/D, o programa de aquisição de dados deve

programar o canal A da PIO como entrada (Modo 1), preparar a PIO para interromper o processador no recebimento do pulso ASTB e programar o canal B da PIO como bit entrada/saída (Modo 3). Além destas tarefas de inicialização da PIO, deve ser providenciado um programa de atendimento da interrupção que será gerada ao fim de cada conversão. O circuito completo do conversor A/D desenvolvido é mostrado na Fig. 3.2.

3.4 - FONTE DE TENSÃO DE REFERENCIA

A fonte de tensão de referência foi implementada com um regulador de tensão (Circuito Integrado uA7808) [9] e um amplificador operacional de alto desempenho (CA3140). O circuito utilizado encontra-se na Fig. 3.3. Na Fig. 3.4, é ilustrada a variação com a temperatura da tensão de referência, ajustada na temperatura ambiente para 4,9 V. O ensaio experimental revelou uma variação de 4 mV na tensão de referência para uma variação de 55°C no circuito. A tensão de referência foi ajustada, posteriormente, para 5,12 V, de modo a se obter um valor múltiplo de 256 (degraus de 20 mV).

3.5 - CONDICIONADOR DE SINAL

O circuito do condicionador de sinal utilizado é mostrado na Fig. 3.5, consistindo basicamente de um retificador de precisão, implementado com dois amplificadores operacionais de alto desempenho, e um

circuito de detecção de polaridade.

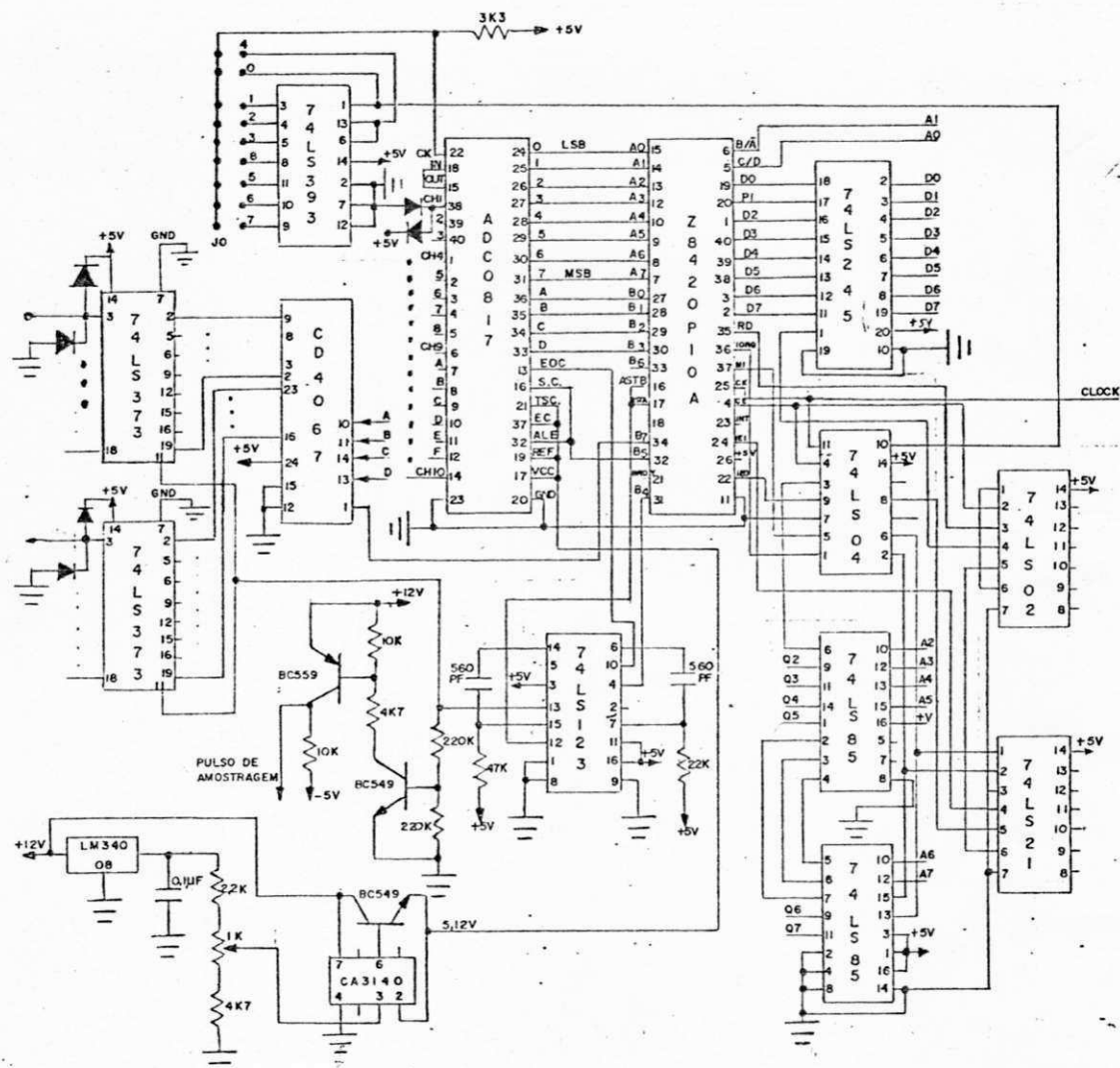


Fig. 3.2 Conversor A/D

O circuito do retificador de precisão fornece um sinal de saída na faixa de 0 a 5 volts para o circuito de amostragem e retenção.

3.6 - CIRCUITO DE AMOSTRAGEM E RETENÇÃO

Um circuito de amostragem e retenção ("sample & hold") é um circuito que amostra e armazena o valor de um sinal analógico, em um dado instante. Quando no modo

RETENÇÃO ("hold"), o valor amostrado é mantido constante na saída do circuito e pode, então, ser convertido em uma palavra digital por meio de um conversor analógico-digital [10].

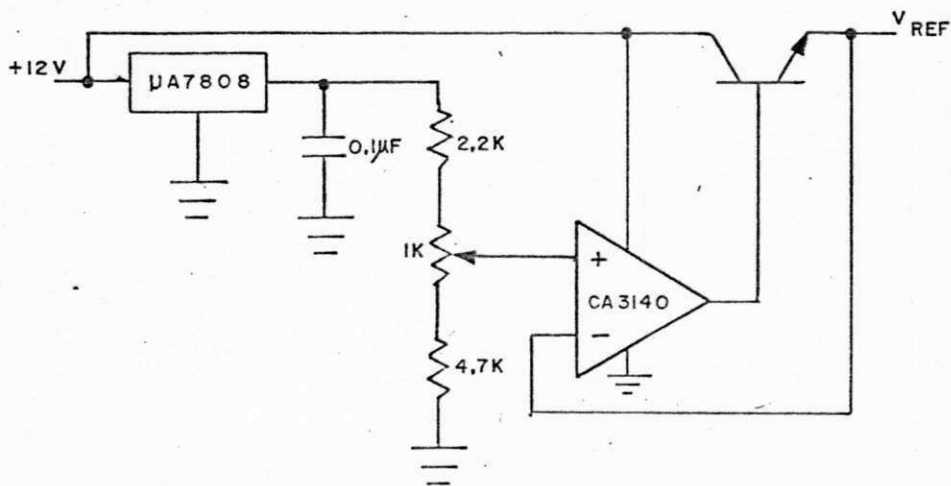


Fig. 3.3 Circuito da fonte de tensão de referência

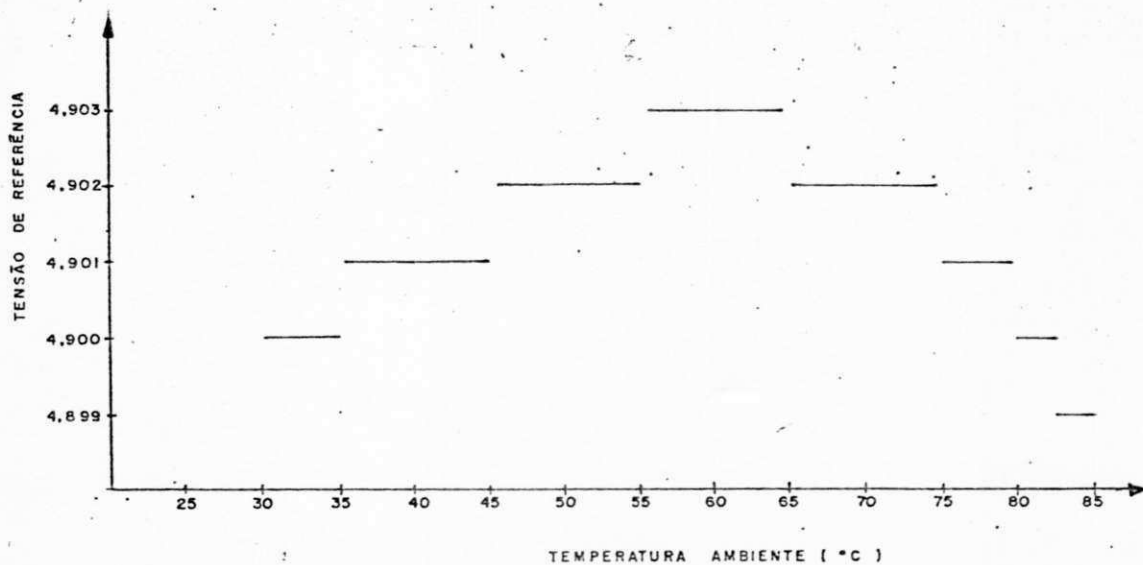


Fig. 3.4 Tensão de referência versus temperatura

O circuito de amostragem e retenção desenvolvido é

mostrado na Fig. 3.6. Note-se que este circuito é híbrido, utilizando transistores discretos e circuitos integrados. Os transistores bipolares foram utilizados para reduzir o tempo de aquisição, que é definido como o tempo necessário para que o circuito, no modo AMOSTRAGEM ("sample"), obtenha o valor correto do sinal. Este tempo de aquisição, quando excessivo, conduz a erros de amostragem que dependem da frequência do sinal de entrada, da taxa de amostragem escolhida e dos parâmetros do circuito.

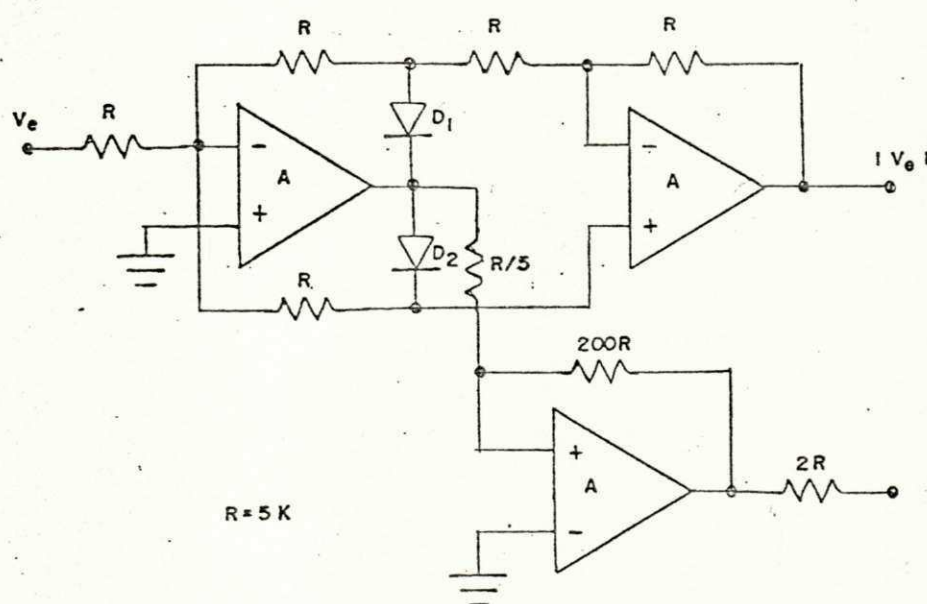


Fig. 3.5 Circuito do condicionador de sinal

Com o presente circuito, foi conseguido um tempo de aquisição da ordem de 5 microssegundos, tempo esse que garante que o erro introduzido pelo circuito de amostragem e retenção pode ser desprezado [10]. Os amplificadores operacionais empregados são do tipo CA3140, que possuem corrente de polarização típica de 20 pA e "slew rate" de 9 V/microssegundo.

Para reduzir o efeito da corrente de fuga das

chaves analógicas empregadas, foi utilizado o resistor de 100 quiloohms (Vide Fig. 3.6). Este resistor garante que, durante o tempo de retenção, a tensão sobre a chave Ch1 é limitada a um valor abaixo de 100 milivolts. Com tensão reduzida na chave analógica, a corrente de fuga sofre um decréscimo substancial.

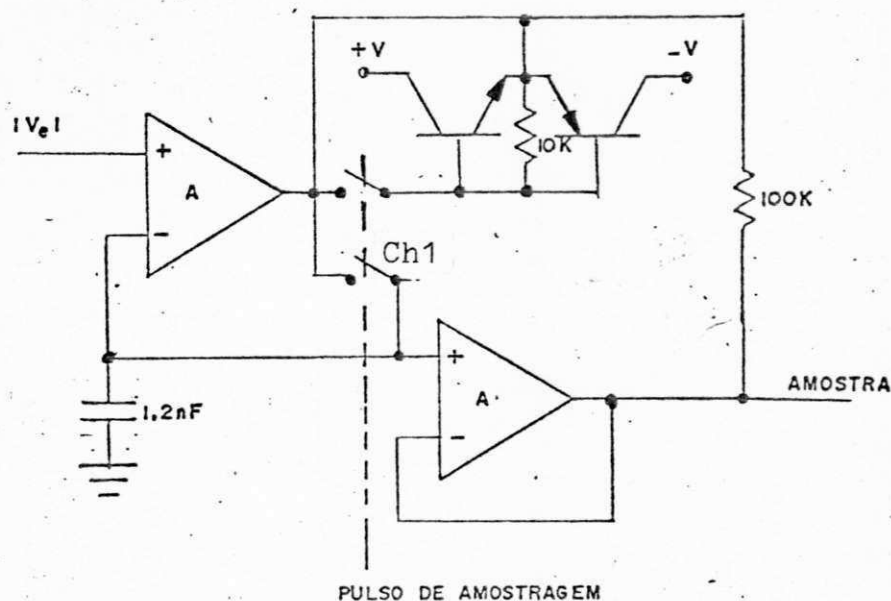


Fig. 3.6 Circuito do amostrador

O tempo de retenção obtido com o presente circuito é equivalente a uma constante de tempo da ordem de 60 segundos.

CAPITULO 4

CONCLUSÃO

Portanto, meus amados irmãos, sede firmes e constantes, sempre abundantes na obra do Senhor, sabendo que o vosso trabalho não é vão no Senhor. (I Corintios 15:58).

Foi desenvolvido um programa executivo, preemptivo e com prioridade mista, para controle de processos em tempo real à base do microprocessador Z80. Este programa foi escrito em "assembly", sendo bastante compacto e rápido, ocupando menos de 512 bytes de memória e necessitando de cerca de 12% do tempo do processador, tempo este medido considerando-se 18 tarefas no sistema.

Para manter pequeno o tempo de execução deste programa, ou seja, para não se ter um "overhead" muito grande, é aconselhável manter o número de tarefas relativamente pequeno. Por exemplo, para se manter o "overhead" máximo abaixo de 20% do tempo de execução, é necessário manter o número de tarefas abaixo de 30, e para mantê-lo abaixo de 50%, deve-se ter menos que 78 tarefas.

Os tempos máximos e mínimos de execução do programa executivo são apresentados na Tabela 4.1.

Tabela 4.1 - Tempos máximos e mínimos de execução do programa executivo (T = 1 ciclo do clock do sistema).

	MAXIMOS	MINIMOS
Interrupção do sistema (1 tarefa)	1418T	1057T
-----	-----	-----
Adicional para cada tarefa	630T	397T
Chaveamento entre tarefas (1 tarefa)	915T	863T
-----	-----	-----
Adicional para cada tarefa	252T	207T

Portanto, pode-se concluir, por exemplo, que, para um clock de 4 MHz e 15 tarefas distintas, o tempo máximo para atender a interrupção do sistema será de 2,56 ms e o mínimo, de 1,65 ms, e que o tempo máximo para realizar o chaveamento de uma tarefa para outra será de 1,11 ms e o mínimo, de 0,94 ms.

Neste trabalho, foram desenvolvidas 18 tarefas distintas para servir de modelo para o desenvolvimento de tarefas específicas de um sistema de controle dedicado em malha fechada. Estas tarefas foram agrupadas em seis conjuntos, a saber: tarefas de menu, tarefas de relógio, tarefa de display de "buffer", tarefas de display de descritores, tarefas que modificam descritores e tarefas do conversor A/D.

As tarefas de menu apresentam, no vídeo, as funções possíveis do sistema na forma de menus, sendo necessária a digitação de uma única letra para a escolha de cada função.

As tarefas de relógio permitem a colocação, ou não, de um relógio em tempo real no vídeo, bem como o acerto desse relógio e sua atualização periódica.

A tarefa de display de "buffer" verifica a existência, ou não, de caracteres no "buffer" de recepção de dados e, em caso afirmativo, apresenta-os no vídeo.

As tarefas de display de descritores armazenam, em RAM, os valores contidos nos descritores das tarefas, podendo, se desejado, mostrá-los no vídeo.

As tarefas que modificam descritores permitem a alteração do conteúdo dos campos `FREQ`, `PRIO`, `STAT` e `DECR` dos descritores.

As tarefas do conversor A/D especificam o canal do conversor a ser convertido e, após a conversão, armazenam na RAM os valores convertidos e apresentam os resultados no vídeo. Além disso, permitem a indicação da polaridade do valor do canal convertido.

No desenvolvimento das tarefas citadas acima, preocupou-se fundamentalmente com a simplicidade das rotinas necessárias ao sistema, e não com a minimização do número total de tarefas.

Objetivando tornar eficiente o desenvolvimento das tarefas propriamente ditas, foram desenvolvidas algumas subrotinas de uso geral, que facilitam a comunicação com o operador (via teclado e vídeo), o posicionamento do cursor no vídeo e a formatação dos dados.

Para tornar possível a aquisição de dados de processos externos, foi projetado, construído e testado um cartão para o microcomputador CAMAÇARI, contendo um

conversor A/D de 16 canais multiplexados e circuitos auxiliares (fonte de tensão de referência, amostragem e retenção, condicionador de sinal, interface). O conversor A/D desenvolvido aceita sinais positivos e negativos até 5,12 V de amplitude, convertendo-os com resolução de 20 mV. Cada conversão pode ser feita em menos de 70 μ s, ou seja, os 16 canais podem ser convertidos em menos de 1,2 ms.

Saliente-se que o circuito de amostragem e retenção, que foi implementado com transistores bipolares discretos e circuitos integrados, apresenta um tempo de aquisição da ordem de 5 μ s e tempo de retenção equivalente a uma constante de tempo da ordem de 60 s.

Finalmente, como sugestões para aperfeiçoamento do software desenvolvido, podem-se citar:

- 1) Desenvolvimento de rotinas dentro do executivo com a finalidade de proteger o sistema contra tarefas com erros que poderiam provocar uma pane no sistema;
- 2) modificação das tarefas de display, visando dar solução às mensagens de alta prioridade;
- 3) aumento do número de listas de descritores para diminuir o tempo de chaveamento entre tarefas.

B I B L I O G R A F I A

- [1] Jacob H. Ruiter, Jr e R. Gordon Murphy, Basic Industrial Electronic Controls, Holt, Rinehart and Winston, Inc., New York, 1962.
- [2] Paulo Roberto Campos de Araújo, "PROSAD, Um Sistema de Aquisição de Dados e Controle de Processos a Microprocessador", Tese de Mestrado, janeiro de 1981.
- [3] Kenneth Burgett e Edward F. O'Neil, "An Integral Real-Time Executive For Microcomputers," Computer Design, July 1977.
- [4] Kevin C. Kahn, "A Small-Scale Operating System Foundation for Microprocessor Applications", Proceedings of the IEEE, Vol. 66, No. 2, February 1978.
- [5] Gregory R. Andrews e Fred B. Schneider, "Concepts and Notations for Concurrent Programming", Computing Surveys, Vol. 15, No. 1, March 1983.
- [6] Alex Vieira Pinto, Gurdip Singh Deep e José Homero Feitosa Cavalcanti, "Exclusão Mútua de Processos Concorrentes em Sistemas de Controle Por Microprocessador", 1º Congresso Latino-Americano de Automática/5º Congresso Brasileiro de Automática, Universidade Federal da Paraíba, Campus II, Campina Grande, Paraíba, 3 a 6 de setembro de 1984.
- [7] Zilog Components Data Book, 1983

- [8] National Semiconductor Corporation, Linear Databook, 1982.
- [9] National Semiconductor Corporation, Voltage Regulator Handbook, 1980.
- [10] J. J. Hill, "On Aquisition Error in the Sample and Hold Circuit", Transactions on Industrial Electronics, Control Instrumentation, Vol. IECI-25, No. 2, pp. 181-183, May 1978.
- [11] Hewlett-Packard Company, "Analog to Digital Conversion", Bench Briefs, pp. 1-4, March/May 1985.
- [12] E. W. Dijkstra, "Solution of a Problem in Concurrent Programming Control", Communications of the ACM, Vol. 8, No. 9, September 1965.

A P E N D I C E A

RECURSOS DO SISTEMA UTILIZADO

A.1 - RECURSOS DE HARDWARE

O microcomputador CAMAÇARI, no qual foi desenvolvido este trabalho, é fabricado pela CALCON TECNOLOGIA S. A. O sistema está configurado para um barramento padronizado segundo o padrão ECB (EuroCard Bus) e está montado num rack de 19 polegadas, que dispõe de vários conectores para adicionar cartões de interface. Os cartões básicos que constituem o sistema utilizado estão descritos abaixo:

A.1.1 - Placa da Unidade Central de Processamento

A placa da Unidade Central de Processamento (UCP), montada no cartão tipo "Eurocard" simples, com conector traseiro do tipo Euroconector (2 fileiras de 32 pinos), e configurada para o barramento "ECB", contém: Um microprocessador Z80-CPU/A, um temporizador quádruplo programável (Z80-CTC/A), uma porta paralela dual de oito bits (Z80-PIO/A), e uma porta serial dual (Z80-SIO/O/A ou Z80-DART/A) com saídas, no padrão RS-232-C, para terminal de vídeo e para impressora serial. O Sistema Operacional é o CP/M 2.2

A.1.2 - Placa de Memória Volátil (RAM)

A placa de memória volátil, contendo 65.536 locações de 8 bits (64K/8bits) com tempo de acesso da ordem de 150 nanosegundos, está montada num cartão tipo "Eurocard" simples, com conector traseiro do tipo Euroconector (2 fileiras de 32 pinos), configurado para o barramento "ECB".

A.1.3 - Placa de Interface para Unidade de Disco Flexível

A placa de interface para unidade de disco flexível, montada no cartão tipo "Eurocard" simples, com conector traseiro do tipo Euroconector (2 fileiras de 32 pinos), configurada para o barramento "ECB", contém um controlador de disco flexível (uPD765) e um controlador de acesso direto à memória (Z80-DMA/A) para fazer a transferência de dados entre a memória e o disco flexível ou vice-versa. Esta placa é compatível com a placa da UCP Z80-CPU/A e tem a possibilidade de controlar de uma a quatro unidades de disco flexível de 5,25 ou 8 polegadas, face dupla e densidade dupla.

A.1.4 - Placa da Fonte de Alimentação

A placa da fonte de alimentação, montada no cartão tipo "Eurocard" simples, com conector traseiro do tipo Euroconector (2 fileiras de 32 pinos), configurada para o barramento "ECB", tem capacidade para fornecer +5 volts

com 10 ampères, +12 volts com 1 ampère e -12 volts com 1 ampère.

A.1.5 - "Backplane"

O "backplane" está montado no cartão tipo "Eurocard" simples, com conector traseiro do tipo Euroconector (2 fileiras de 32 pinos), configurado para o barramento "ECB".

O sistema de interconexão entre os cartões é feito por meio do "backplane", configurado com duas fileiras de 32 pinos, conectados segundo o padrão ECB conforme ilustra o diagrama da figura A.1

A.2 - RECURSOS DE SOFTWARE

Os programas abaixo relacionados foram fornecidos pela CALCON TECNOLOGIA S. A. para serem executados no ambiente CP/M do CAMAÇARI. Foi necessário instalar o programa de edição de textos (WordStar) para o terminal de vídeo e impressora em uso. Uma versão diferente foi criada para cada tipo de periférico (terminal ou impressora).

A.2.1 - Editor de Textos (WordStar)

Para escrever os programas e os textos foi usado um editor de textos para uso em processamento de palavras e impressão de textos. Tanto o texto já digitado como o texto em digitação são mostrados no vídeo. A maioria das

UNIVERSIDADE FEDERAL DA PARAÍBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso, 882 - Tel (083) 321-7222-R 355
58.100 - Campina Grande - Paraíba

funções de formatação é apresentadas imediatamente na tela (as demais só aparecem no texto impresso).

	a	b	c
+5V	0	01	0 +5V
D5	0	02	0 D0
D6	0	03	0 D7
D3	0	04	0 D2
D4	0	05	0 A0
A2	0	06	0 A3
A4	0	07	0 A1
A5	0	08	0 A8
A6	0	09	0 A7
WAIT	0	10	0 A16
BUSRQ	0	11	0 IEI
BAI	0	12	0 A17
+12V	0	13	0 A18
A19	0	14	0 DI
-5V	0	15	0 -15V
20	0	16	0 IEO
BA0	0	17	0 A11
A14	0	18	0 A10
-	0	19	0 -
MI	0	20	0 NMI
-	0	21	0 INT
DMARDY	0	22	0 WR
-	0	23	0 -
-	0	24	0 RD
-	0	25	0 HALT
deselect	0	26	0 PWCLR
IORQ	0	27	0 A12
RFRSH	0	28	0 A15
A13	0	29	0 0
A9	0	30	0 MRQ
BUSAK	0	31	0 RESET
GND	0	32	0 GND

Fig. A.1 Barramento do CAMAÇARI

Durante a edição, são apresentados menus na tela mostrando os diversos comandos possíveis. Durante a execução de funções especiais, são apresentados menus secundários.

Saliente-se que todos os programas e todo o texto desta Dissertação foram escritos usando o editor WordStar.

A.2.2 - Macro Assembler (M80)

O Macro Assembler utilizado aceita todos os mneumônicos do Z80 e do 8080 e gera o código objeto correspondente. Aceita quatro campos distintos por linha de programa. Estes campos são: 1) símbolo ou "label"; 2) operação ou mneumônico; 3) argumento; 4) comentário.

O Macro Assembler permite ainda o uso de variáveis públicas (externas) e de trechos condicionais (só gerará o código objeto mediante a condição verdadeira). Permite, como o próprio nome sugere, o uso de macros, ou seja, trechos de programas que são inseridos no corpo do programa principal quando o nome de uma macro se encontra no campo do mneumônico.

Este Assembler usa um arquivo de entrada e gera dois arquivos de saída: o do programa objeto relocável e o da listagem para imprimir.

A.2.3 - Linker (L80)

Este programa usa arquivos relocáveis, gerados pelo M80 e cria um arquivo comando. Este pode ser armazenado no disco em duas formas diferentes: um comando executável e um arquivo no formato HEXA. Pode ainda criar, no disco, um arquivo dos símbolos públicos usados.

A.2.4 - Depurador de Programas (ZSID)

O depurador de programas pode executar programas passo-a-passo, mostrando o conteúdo de cada registrador, ou pode executar o programa sem interrupção até alcançar um endereço determinado. Tem também comandos para mostrar e alterar o conteúdo da memória, e gerar código objeto a partir dos mneumônicos.

Este programa auxilia na depuração de programas ou de trechos de programas complicados.

UNIVERSIDADE FEDERAL DA PARAIBA
Pró-Reitoria Para Assuntos do Interior
Coordenação Setorial de Pós-Graduação
Rua Aprígio Veloso, 882 - Tel (83) 321-7222-R 355
58.100 - Campina Grande - Paraíba

A P E N D I C E B

VALORES DOS CAMPOS DOS DESCRITORES

Na tabela B.1 são apresentados os valores normais dos campos dos descritores das tarefas. Apesar de terem sido desenvolvidas tarefas para alterar os valores contidos nos campos de `FREQ`, `STAT`, `PRIO` e `DECR`, apenas alguns valores de algumas tarefas podem ser modificados sem causar prejuízo ao sistema.

No presente caso, somente poderão ser mudados os campos `PRIO` e `DECR` da tarefa de número 01, o campo `FREQ` da tarefa de número 03 e os campos `FREQ`, `PRIO` e `DECR` da tarefa de número 04.

Observe-se que a tarefa de número 01 admite também mudança no campo `FREQ` sem necessariamente causar pane no sistema, alterando, porém, o incremento normal do relógio.

Tabela B.1 - Valores normais dos campos das tarefas

Nº	NOME DA TAREFA	FREQ	STAT	PRIO	DECR
00	Sistema CP/M	0000	FF	FF	00
01	Incremente relógio	0028	FF	7F	07
02	Display relógio	0000	FF	FE	00
03	Display buffer	002D	FF	FE	00
04	Armazene descritores	004F	FF	7F	07
05	Display descritores	0000	FF	FE	00
06	Recebe caracter	0000	FF	FE	00
07	Menu principal	0000	FF	FE	00
08	Liga/desl. display do relógio	0000	FF	FE	00
09	Liga display dos descritores	0000	FF	FE	09
0A	Acerto do relógio	0000	FF	FE	00
0B	Ativa display dos descritores	0000	FF	FE	00
0C	Modifica FREQ	0000	FF	FE	00
0D	Display conversor A/D	0000	FF	FE	00
0E	Modifica STAT	0000	FF	FE	00
0F	Modifica PRIO	0000	FF	FE	00
10	Modifica DECR	0000	FF	FE	00
11	Especifica canal do conversor	0000	FF	FE	00

UNIVERSIDADE FEDERAL DA PARAÍBA
 Pró-Reitoria Para Assuntos do Interior
 Coordenação Setorial de Pós-Graduação
 Rua Aprígio Veloso, 882 - Tel (083) 321-7222-R 355
 58.100 - Campina Grande - Paraíba