

UM PROCESSADOR DE COMUNICAÇÃO DE REDE LOCAL PARA CONTROLE DE PROCESSOS

Leonardo Appezato
Instituto de Automação - CTI

SUMÁRIO

Este trabalho apresenta o resumo de um projeto atualmente em desenvolvimento no Instituto de Automação do Centro Tecnológico para Informática - CTI, visando a implementação de um processador de comunicação de rede local para aplicações em controle de processos, tendo como referência a norma MAP ("Manufacturing Automation Protocol").

1. INTRODUÇÃO

O módulo REDE/WD constitui-se em um cartão inteligente, o qual permite que um sistema, utilizando um barramento padrão ECB, seja integrado a uma rede local do tipo "token-pass". A taxa de transmissão alcançada pela REDE/WD é de 1Mbit/seg e seu campo de endereçamento permite a presença de até 254 estações na rede local. A REDE/WD se liga a um cartão denominado "Interface Elétrica de Rede", através de sinais no padrão RS-422. Esses mesmos sinais podem ser usados na implementação de uma rede local simples utilizando-se, como meio físico, um fio tipo par trançado ou cabo coaxial. Porém, na hipótese de ser necessário um meio de comunicação mais confiável para certos tipos de aplicação, a Interface Elétrica de Rede permitirá a implementação do meio físico adequado. Inicialmente, será implementada uma Interface Elétrica de Rede com um "modem" FSK, para ser utilizado em um meio físico tipo barramento ("token bus"), com cabo coaxial.

2. DESCRIÇÃO DOS SUBMÓDULOS DE HARDWARE

O módulo REDE/WD é composto pelos submódulos descritos abaixo.

2.1. CPU.

Baseia-se em um microprocessador Z80-A, operando a 4MHz, o qual é responsável pela inicialização e operação dos recursos internos do módulo REDE/WD, na implementação do protocolo de comunicação da rede local.

2.2. Controlador de Rede

Constitui-se, basicamente de um "chip" controlador de rede local (WD2840), de um codificador/decodificador Manchester (HD6409) e de uma interface RS-422 para comunicação entre o módulo REDE/WD e a Interface Elétrica de Rede.

O WD2840 é responsável pela execução do protocolo de comunicação "token-passing" da rede local, possuindo funções que permitem a ele gerar o anel lógico (inicialização da rede local), adicionar ou remover estações da rede, detectar e corrigir erros de transmissão, transmitir em "broadcasting", solicitar que a estação destino de uma mensagem confirme o recebimento desta, etc.

O HD6409 codifica os pulsos NRZ transmitidos pelo WD2840 para o padrão Manchester, antes de enviá-los ao meio físico da rede. Na recepção, os sinais recebidos da rede são decodificados de Manchester para NRZ e enviados para o WD2840.

2.3. Memória RAM/EPROM

A memória EPROM contém o "firmware" do módulo REDE/WD, ou seja, as rotinas de inicialização do módulo e o protocolo implementado. A memória RAM é utilizada com área de trabalho da CPU, armazenando tabelas e dados intermediários do protocolo de comunicação.

2.4. Memória "Dual-Port" CPU/WD

Esta memória é meio pelo qual a CPU comunica-se com a rede local, e vice-versa, utilizando-se dos serviços do controlador de rede WD 2840. Ela permite o acesso compartilhado dos dois dispositivos (Z80 e WD2840), tanto para escrita como para leitura, sendo que acessos simultâneos são resolvidos pelo árbitro de controle de acesso associado à esta memória.

2.5. Memória "Dual-Port" CPU/ECB

Esta memória permite a comunicação entre o módulo REDE/WD e o sistema "host", no qual este módulo estará inserido. Ela é utilizada, basicamente, como "buffer" de transmissão e recepção de mensagens entre esses dois dispositivos. Acessos simultâneos são resolvidos por um árbitro de controle, de acesso à memória.

2.6. Interface com o Barramento ECB

Ela é composta, basicamente, por circuitos "buffers" e "transceivers", ligados aos barramentos de dados, endereço e controle do barramento ECB, e por uma lógica de decodificação de endereços de memória e E/S. Essa lógica permite a seleção de endereços de acesso à memória "dual-port" CPU/ECB e de acesso

ã porta de E/S utilizada para gerar uma interrupção ã CPU do m3dulo REDE/WD, que irã indicar a chegada de uma mensagem naquela mem3ria.

2.7. L3gica de Interrupç3o

Esta l3gica destina-se a gerar uma interrupç3o ã CPU do m3dulo REDE/WD, a partir de eventos, tais como: estouro do "watch-dog" que supervisiona o "host"; pedido de interrupç3o feito pelo WD2840; interrupç3es peri3dicas geradas por um temporizador; e, pedido de interrupç3o gerado pelo "host".

2.8. Temporizadores

Implementado a partir de um Z80-CTC, este subm3dulo tem a funç3o dupla de ser o gerador de uma base de tempo, e a de servir como um perif3rico da fam3lia Z80 no sentido de gerar uma interrupç3o para o ECB. A funç3o desta interrupç3o ẽ a de permitir ao m3dulo REDE/WD avisar o "host" do ECB da exist3ncia de uma mensagem para ele, a qual se encontra colocada na mem3ria "dual-port" CPU/ECB.

2.9. Decodificador de Endereços de E/S e Mem3ria

Possui a l3gica responsãvel pelo mapeamento da 3rea de mem3ria e dos perif3ricos de E/S presentes no m3dulo REDE/WD. Esse mapeamento ẽ feito atrav3s da programaç3o das PROMs usadas na decodificaç3o de endereços.

3. DESCRIÇ3O DO SOFTWARE

Apesar do controlador de rede WD2840 n3o ser conforme ã norma IEEE 802.4, o software a ser implementado no m3dulo REDE/WD deverã ser compat3vel com a norma MAP (Manufacturing Automation Protocol). Assim, ser3o implementadas as subcamadas MAC (controle de acesso ao meio) e LLC (controle de enlace l3gico), que formam a camada 2 (enlace). O software deverã fornecer uma interface MAC/LLC na forma especificada pelo 802.4 (sem a utilizaç3o de prioridade), apesar do MAC n3o implementar todas as funç3es especificadas nessa norma. A partir dessa interface, a subcamada LLC deverã ser implementada seguindo-se a definiç3o da norma IEEE 802.2, com serviços do tipo 1, assim como sua interface com a camada de rede.

O controlador de rede deverã ser utilizado dentro de uma arquitetura do tipo MiniMAP (n3veis OSI 1,2 e 7) onde o n3vel 7 implementarã serviços do protocolo EIA RS-511.

DESCRIÇÃO DO DESENVOLVIMENTO DE UM ANALISADOR/SIMULADOR DE PROTOCOLOS PARA REDES DE COMPUTADORES

Antonio Carlos T. Porto - Engenheiro de desenvolvimento da DIGITEL
João Carlos Gluz - Projetista de software da DIGITEL
André Chignatti - Engenheiro de desenvolvimento da DIGITEL
Ricardo Scop - Engenheiro de desenvolvimento da DIGITEL

RESUMO

Este trabalho descreve o desenvolvimento de um analisador/simulador de protocolos. São apresentadas as fases de desenvolvimento do equipamento, bem como sua engenharia de produto e arquitetura de hardware e software. Além disso são descritas também as aplicações do equipamento em redes e suas especificações técnicas. Com relação a simulação é apresentado o desenvolvimento da linguagem "STEP-BASIC" além de uma descrição sucinta da sintaxe da mesma.

1. INTRODUÇÃO

Nos últimos anos verificou-se um aumento considerável do número de redes de computadores instalados no país, bem como a ampliação das redes já existentes. Além disso inúmeros fabricantes tornaram-se fornecedores de equipamentos p/ redes como CPUs, terminais, multiplexadores, controladoras, modems, etc.

A complexidade das redes, a grande variedade de equipamentos e a necessidade de um instrumento de apoio ao desenvolvimento e instalação de novas máquinas, aumentaram muito a demanda por um analisador/simulador de protocolos. A DIGITEL frente a este contexto optou por desenvolver um equipamento que atendesse a este mercado, aproveitando sua tecnologia em comunicação de dados.

2. FASES DE DESENVOLVIMENTO

2.1. Monitor

A 1ª fase do projeto foi o desenvolvimento de um equipamento para monitoração de protocolos. Nesta fase identificou-se as necessidades do mercado brasileiro de redes.

A partir disto elaborou-se a especificação de um equipamento que atendesse as necessidades mais imediatas dos usuários. Esta especificação cobria principalmente as características de um monitor de protocolos, prevendo, no entanto, a futura expansão deste produto para um simulador, com capacidade de armazenamento em disquete.

Estabelecida a especificação do produto, partiu-se para o desenvolvimento do mesmo. O tempo total de desenvolvimento foi de 12 meses, divididos em dois meses de projeto, 8 meses de implementação e 2 meses de teste. Ao final do desenvolvimento em novembro de 1985, iniciou-se a comercialização do equipamento, com o nome de STEP 2 modelo M.

O pessoal envolvido exclusivamente com o projeto de hardware e software, nesta fase, foi de 4 engenheiros e 1 técnico, além do setor de engenharia de produto.

2.2. Simulador para Protocolos Orientados a Caracter

Esta fase envolveu primeiramente um estudo das linguagens de simulação de protocolos existentes no mercado internacional.

A seguir foi definida a linguagem de simulação de protocolos orientados a caracter para o STEP 2, denominada STEP BASIC. Iniciou-se, então a implantação da simulação no equipamento existente. Para tanto, foi desenvolvido um compilador para a linguagem, um editor de textos, um software de gerenciamento e manipulação de arquivos em disco, além da adaptação do software já existente para suportar as novas características do equipamento.

A duração desta fase foi de 10 meses, dos quais um mês foi gasto no estudo de linguagens de simulação, um mês na definição do STEP-BASIC, 6 meses no desenvolvimento do software e 2 meses de testes e depuração. O lançamento do novo produto foi efetuado em outubro de 1986.

O número de pessoas envolvidas nesta fase em projeto e desenvolvimento e software foi de 3 engenheiros, um programador e dois técnicos.

2.3. Simulador para Protocolos Orientados a Bit

A terceira fase segue os moldes da segunda fase, contando também de um estudo de outras linguagens de simulação, definição da expansão do STEP-BASIC para suportar simulação de protocolos orientados a bit e modificação do software existente para suportar essa nova característica.

O estudo de outras linguagens e a definição de expansão do STEP-BASIC, durou aproximadamente um mês, adaptação do software 2 meses e o teste do equipamento dois meses.

Atualmente estamos na fase de testes do equipamento.

As pessoas envolvidas na 3ª fase são as mesmas envolvidas na segunda.

3. DESCRIÇÃO DO EQUIPAMENTO

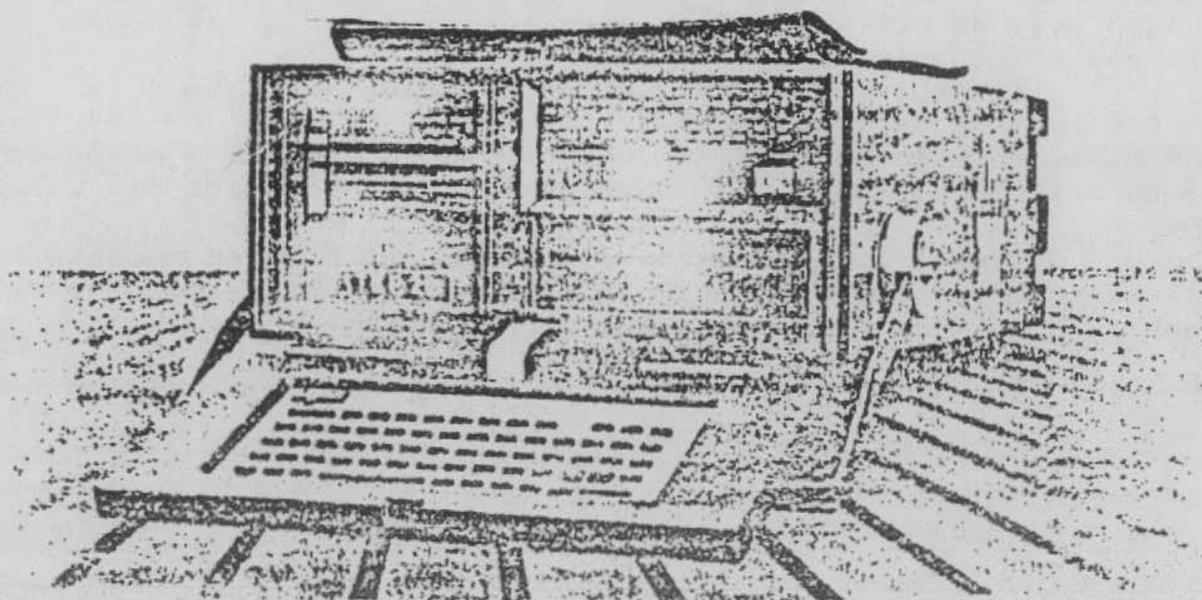
3.1. Engenharia de Produto

O STEP 2 possui no painel frontal um vídeo de 5 polegadas com 16x32 caracteres, um teclado ASCII mais teclas de funções e um drive dupla face, dupla densidade e led's de monitoração dos sinais de interface.

No painel traseiro encontram-se os conectores para ligações ao ETD e ao ECD e a saída para impressora.

Internamente as placas encontram-se dispostas verticalmente sobre uma "MOTHER BOARD" localizada na base do equipamento.

Além disso, o equipamento dispõe de uma alça para transporte e apoio do teclado.



Vista frontal do STEP2

3.2. Arquitetura de Hardware

O STEP 2 possui uma arquitetura multiprocessadora voltada à tarefa de monitorar e simular protocolos.

O Processador Geral realiza as funções de monitoração e simulação de protocolos e executa as rotinas de configuração do equipamento. O Processador de Disquete realiza o gerenciamento dos arquivos em disco. Esta situação onde o Processador Geral é Mestre e o Processador de Disquete é Escravo se inverte no USER MODE. Neste modo é o Processador Geral que presta serviços para o Processador de Disquete.

O hardware está dividido em módulos onde cada módulo executa uma determinada função e está localizado em uma placa.

A placa Processador Geral é a placa da CPU principal do sistema.

As funções da CPU principal são:

- Monitorar a interface ETD ECD
- Analisar os dados monitorados
- Apresentar no vídeo os dados monitorados
- Atender o teclado
- Enviar dados para a impressora
- Rodar o programa de configuração do equipamento
- Emular um terminal para o PD

A função da Placa Controladores é fazer o controle de vídeo e teclado do STEP 2.

Apenas o Processador Geral tem acesso aos circuitos desta placa, isto é, apenas ele pode receber dados do teclado e colocar informações no vídeo.

A placa interfaces faz interface com a linha de monitoração/simulação e com a impressora.

Todos os circuitos desta placa (SIOS e timers) estão sob o controle da placa Processador Geral.

As funções da placa interfaces são as seguintes:

- Receber os dados seriais que trafegam entre o DTE e o DCE.
- Enviar dados para uma impressora serial.
- Implementar a interface de simulação de DTE e de DCE.

A placa Processador de Disquete é uma placa de circuito impresso onde se encontram a CPU auxiliar do sistema STEP 2, seus circuitos associados e os circuitos de controle de diskette.

A função da placa Processador de Disquete é gerenciar o disquete do STEP 2.

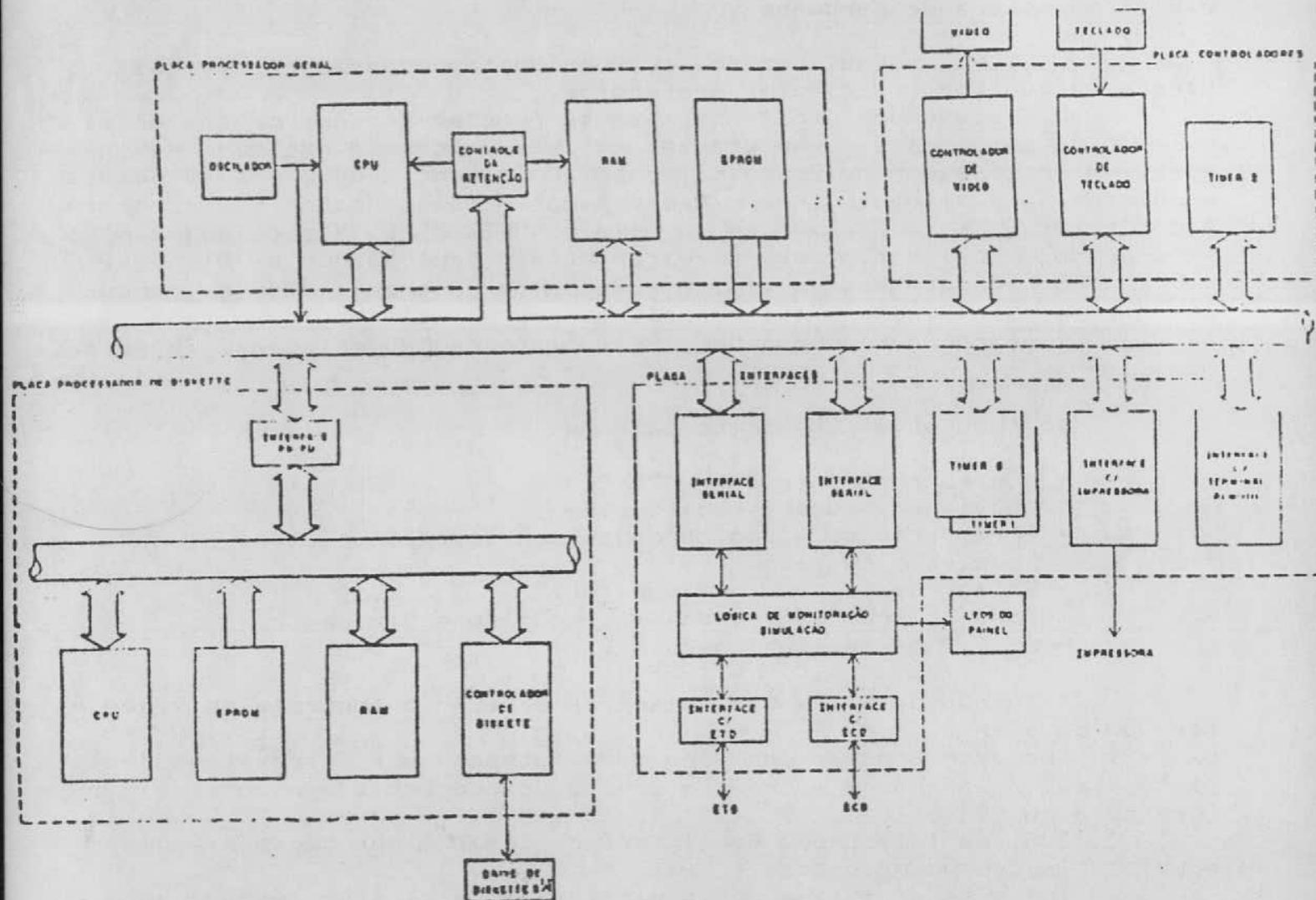


Diagrama de blocos do Hardware do STEP2

3.3. Arquitetura de Software do STEP 2

O software do STEP 2 é organizado em 3 grandes estados de operação. Para cada um destes estados existe uma série de processos responsáveis pela execução de tarefa associada ao mesmo. A arquitetura de software será descrita, então, através de diagramas de fluxo de dados (DFD) que mostram o inter-relacionamento destes processos mais uma curta descrição funcional dos principais processos.

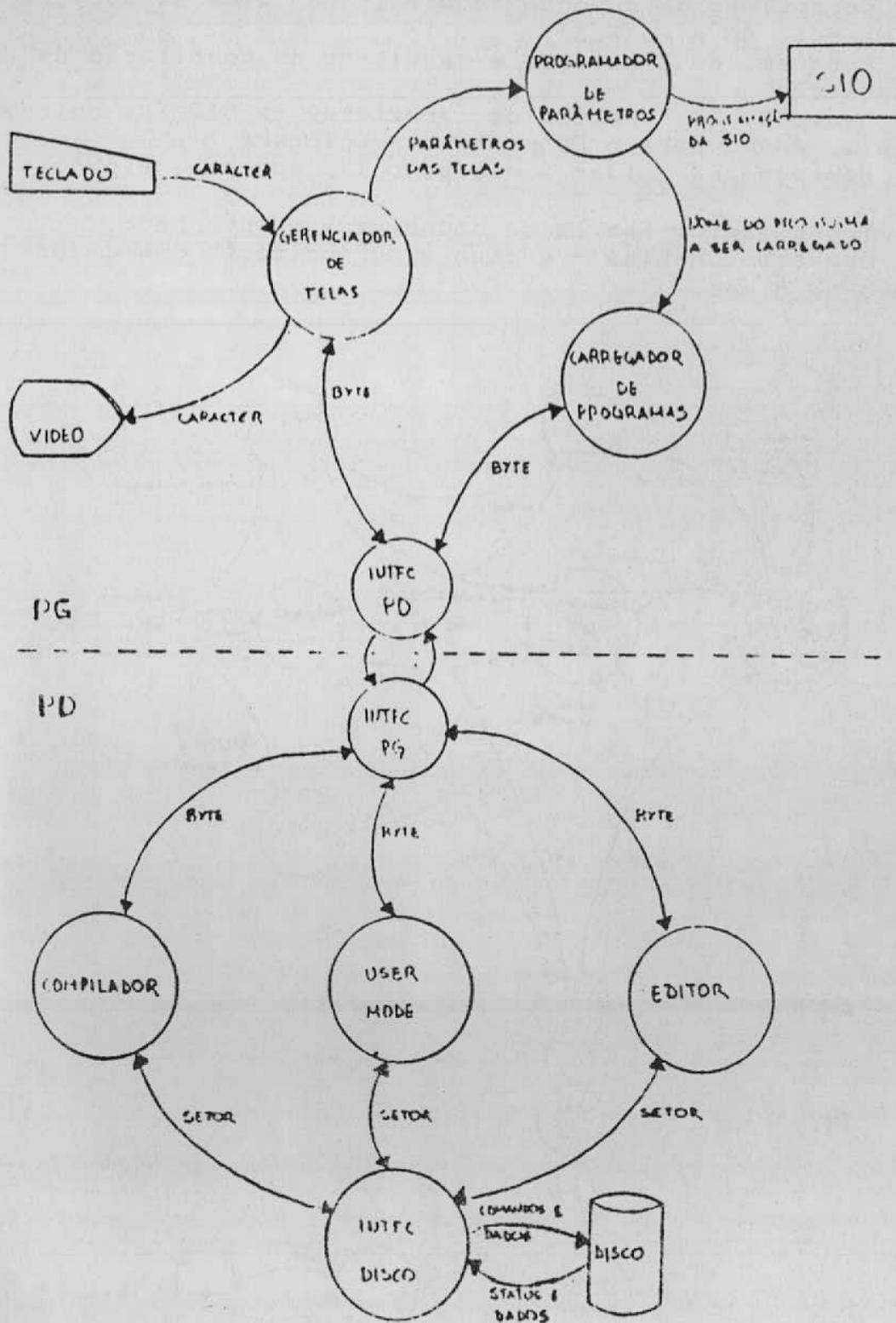
3.3.1. Arquitetura do Estado de Programação

Gerenciador de Telas - gerencia a apresentação das telas de programação e a obtenção dos parâmetros de operação.

Programador de Parâmetros - executa a programação dos SIOS e da lógica de monitoração e seleção, seleciona qual o programa que deve ser carregado do disco para execução.

Carregador de Programas - efetua a carga do programa a ser utilizado no estado de execução.

USER MODE - gerencia e manipula os arquivos em disco (monit-
 ração)



DFD do estado de programação

3.3.2. Arquitetura do Estado de Execução de Simulação

Sistema de Suporte da simulação - Sistema que fornece a biblioteca de rotinas de suporte de simulação, além de detectar a tecla de encerramento de simulação.

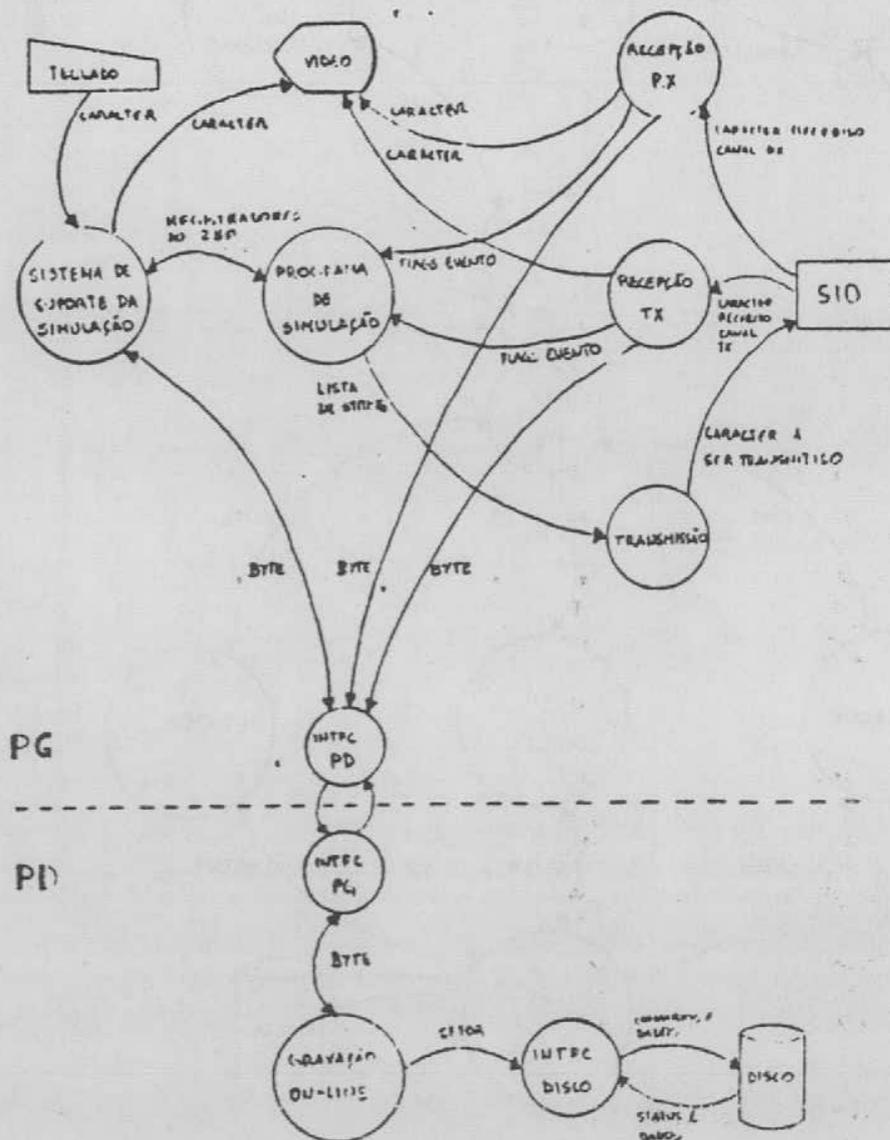
Programa de Simulação - resultado da compilação de um programa de simulação.

Recepção TX - recebe os caracteres da SIO TX, coloca os mesmos na tela, envia para o PD e detecta triggers.

Recepção RX - idem a recepção TX, apenas que para o canal RX da SIO.

Transmissão - envia os dados para a interface.

Gravação On-Line - efetua a bufferização dos dados recebidos e os grava no disco.

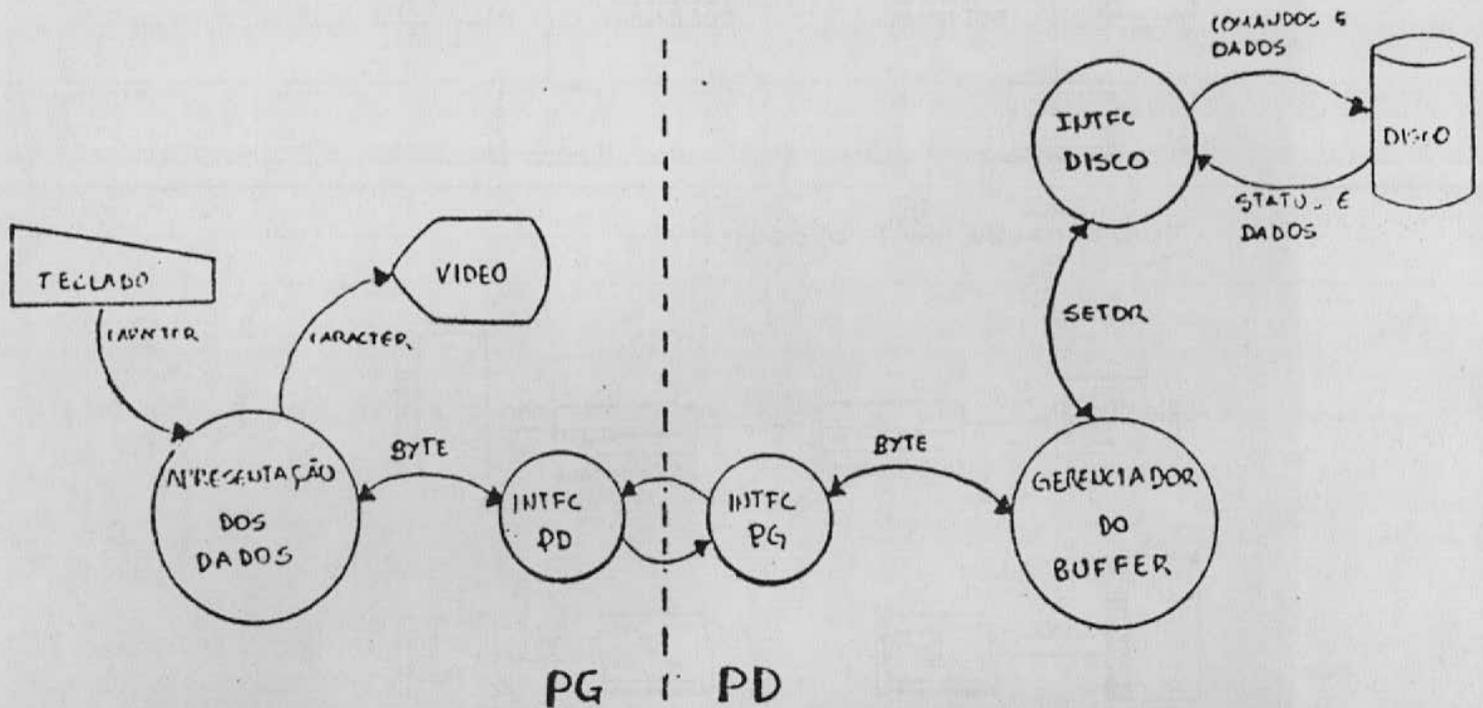


DFD do estado de execução da simulação

3.3.3. Arquitetura do Estado de Apresentação

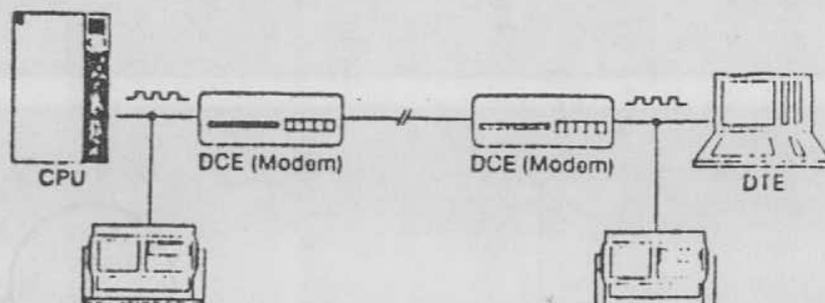
Apresentação dos Dados - recebe os dados armazenados em disco e os apresenta na tela. Atende aos comandos do usuário de movimentação do cursor.

Gerenciador do Buffer - gerencia o buffer de disco que contém os dados recebidos da linha.

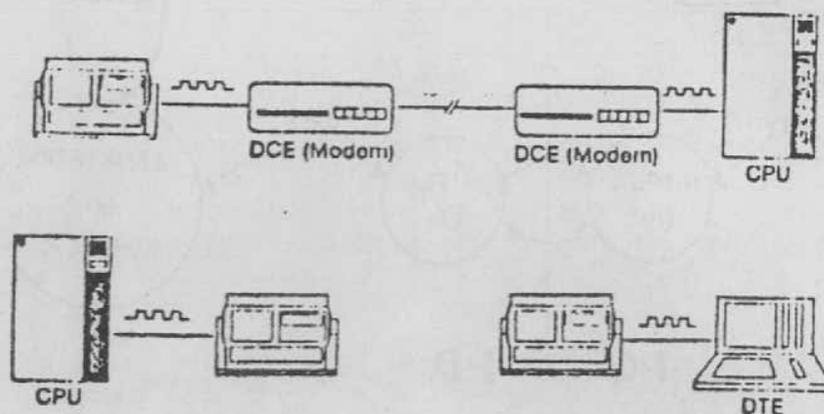


DFD do estado de apresentação

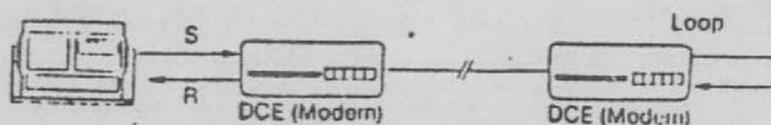
4. APLICAÇÕES DO EQUIPAMENTO



Monitoração de Protocolos



Simulação de Protocolos



Bit/Block Error Rate

5. STEP 2 - ESPECIFICAÇÕES TÉCNICAS

5.1. Modos de Operação

a) Modo MONITOR

- Assíncrono
- Síncrono, orientado a caracter (BSC 1/3)
- Síncrono, orientado a bit (HDLC/SDLC)
- Modo pacote, X.25 nível 3

b) Modo SIMULADOR

- Modo de operação onde pode se executar os programas de simulação (ou monitoração) elaborados na linguagem STEP-BASIC.

Protocolos Simulados

- Assíncrono
- Síncrono, orientado a caracter
- Síncrono, orientado a bit

c) USER - MODE

- Modo de operação onde pode-se editar e compilar programas de simulação. Este modo apresenta todas as facilidades necessárias para a manipulação de arquivos em disco; como copiar, deletar, renomear, imprimir, analisar, etc.

Neste modo dispõe-se de um sistema operacional compatível com o CP/M (STEP-DOS).

d) Modo BERT/BLERT

- Modo de operação no qual o STEP 2 pode enviar para a linha um padrão pseudo-aleatório e receber de linha este padrão verificando o número de bits e blocos errados.

- Padrão pseudo-aleatório 511 bits
- Inserção de erros por tecla
- Modo síncrono até 19200 bps com relógio interno e externo
- Modo assíncrono AS0, AS1, AS2 até 19200bps

e) Modo TIME MEASUREMENT

- Modo de operação no qual o STEP 2 pode realizar medidas de tempo entre os sinais de interface.

- Sinais disponíveis para medida: RTS, CTS, DSR, DCD, DTR, RI.
- Transições positivas e negativas
- Precisão: +/- 1ms

5.2. Interfaces

a) Interface DTE/DCE

Conforme recomendação V24/V28 do CCITT

Monitoração da interface:

- O STEP 2 é provido de led para a monitoração dos seguintes sinais de interface:

TD(103), RD(104), RTS(105), CTS(106), DSR(107), DCD(109), TC(114), RC(115), DTR(108), SQ(110), RI(125), XTC(113).

Led's vermelhos = condição "ON" ou +V

Led's verdes = condição "OFF" ou -V

b) Interface com a Impressora

Serial, assíncrona, totalmente programável:

velocidade, paridade, número de colunas, caracteres de X-ON/X-OFF e tempo de flyback buffering. Controle de fluxo por DTR ou X-ON/X-OFF.

c) Interface com Operador

Display: Vídeo de 5" com tela de 16x32 caracteres, anti-reflexivo.

Teclado: Teclado "QWERT" com 64 teclas e 4 níveis com todos os caracteres ASCII mais teclas de função e caracteres hexa.

5.3. Velocidade de Operação

a) Clock Interno

Velocidade de: 50, 75, 110, 134.5, 150, 300, 600, 1200, 2400, 4800, 9600, 19200 bps. Full-Duplex.

A seleção da velocidade é independente para as linhas de TX e RX em operações assíncronas.

b) Clock Externo

CT114/Pino 15, CT113/Pino 24, qualquer velocidade até 19200 bps. Full-Duplex.

5.4. Códigos

- EBCDIC
- ASCII
- HEX 5, 6, 7, 8, bits

5.5. APRESENTAÇÃO DOS DADOS

a) Tipos de Displays

- Half-Duplex (HDX):
Dados recebidos e transmitidos são apresentados na sequência de captura.

- Full-Duplex (FDX):
Dados transmitidos e recebidos são apresentados em duas linhas diferentes.

b) Identificação dos Dados

- TD: Dados Transmitidos do DTE para o DCE em vídeo normal.
- RD: Dados Transmitidos do DTE para o DCE em vídeo reverso.

Obs.: No modo simulador, a identificação se dá da mesma forma, podendo o STEP 2 ser DTE ou DCE.

c) Apresentação Hexadecimal

- Os caracteres em hexadecimal são mostrados ocupando espaço de apenas um caracter.

d) Apresentação gráfica e estatística nos protocolos bit-oriented.

- Apresenta de forma estatística e gráfica a porcentagem de frames errados e o tráfego de TX e RX.

e) Interpretação do Nível 3 do X.25

f) Apresentação do Nível 2 nos protocolos orientados a bit

- Um frame por linha
- Octeto por octeto nos modos Full-Duplex comprimido e expandido (com correlação de tempo).

5.6. APRESENTAÇÃO DE ERROS

- Erro de Paridade: Caracter em dupla intensidade.
- Erro de Frame: Caracter piscando
- Erro de CRC/LRC: LRC - piscando
CRC - um caracter piscando após o PAD
- Falta de DCD e CTS:

a) Caracter transmitido sem CTS: o mesmo é apresentado seguido por um "raio" (⚡).

b) Caracter recebido sem DCD: o mesmo é apresentado seguido por um raio (⚡).

MODO STATUS - Apresentação gráfica do nível 1
(Protocolos Orientados a Caracter)

- Após a monitoração, todos os caracteres capturados podem ser visualizados com o Status da Interface Digital, existente no instante de sua captura.

- Os sinais apresentados são: RTS, CTS, DSR, DCD, DTR

- Os sinais são apresentados sob forma de onda em relação aos dados capturados.

5.7. PESQUISA DE DADOS

a) Pesquisa de dados no Buffer

- Para frente e para trás
- Página por página
- Linha por linha

b) No modo Status

- Linha por linha
- Caracter a caracter

c) Campo Auxiliar

- Constituído de duas linhas na parte superior do vídeo, indicando a posição do buffer em pesquisa e eventos de trigger.

- Utilizado em conjunto com a tecla "HOME" para operações do tipo "PRINT" e "FIND".

5.8. FUNÇÕES

a) Congelamento (freezing)

Durante a monitoração, o conteúdo da tela pode ser congelado a qualquer momento, sem alteração na captura dos dados.

b) Função FIND

Os dados capturados podem ser encontrados através de uma sequência programável de até 8 caracteres (incluindo o caracter NOT e DON'T CARE).

c) Função PRINT

A tecla Print, utilizada em conjunto com uma impressora serial, visa a impressão de qualquer tela ou de uma área de dados armazenada no disco.

d) Função VERIFY

A função verify apresenta no vídeo um resumo da programação do STEP 2.

e) Função AL/HEX

A função AL/HEX faz uma conversão de dados apresentados na tela do HEXA para alfanumérico ou vice-versa.

5.9. SUPRESSÃO DE DADOS

a) Supressão de Programação

- Podem ser suprimidos de 1 a 3 caracteres, programáveis pelo operador.

b) Supressão por Chave

- Todos os caracteres do campo de informações dos protocolos síncronos podem ser suprimidos através da chave manual. Esta facilidade se aplica na segurança de dados.

5.10. VERIFICAÇÃO DE ERROS

a) Paridade

- Impar
- Par
- Marca
- Espaço
- Nenhuma

b) "Block Error Checks"

- CRC 16
- CRC CCITT
- LRC

5.11. CARACTERES DE SINCRONISMO

- 1 ou 2 caracteres selecionáveis de maneira independente para as linhas de transmissão (TD) e recepção (RD).

5.12. FACILIDADES DE TRIGGER

a) Critérios

- Inicia a monitoração
- Finaliza a monitoração
- Contagem de Trigger
- Número de caracteres após trigger: 1 a 4095

b) Eventos

- Time-out 1 a 65535ms
- Erro de paridade
- Erro de frame
- Erro de CRC/FCS
- Sequência de até 8 caracteres (incluindo NOT e DON'T CARE)
- Buffer cheio
- "Abort"

5.13. CAPACIDADE DE ARMAZENAMENTO

- Alocação de espaço no disco
 Face A - Programas de sistema e área para armazenamento de arquivos - 189 K,
 Face B - Buffer - 189 K

- Duas áreas com 512 bytes onde pode-se armazenar dados através das funções STO e recuperar os dados através da função RCL.

- Gravação ON-LINE no disquete de dados monitorados/simulados até 19200bps full-duplex.

5.14 FUNÇÕES NO USER MODE

- EDIT

Editor de textos tipo "wordstar" para edição de programas de simulação/monitoração.

- COMPILE

Compilador para linguagem de simulação STEP/BASIC.

- LIST

Lista na tela do STEP2 o arquivo solicitado.

- PRINT

Imprime numa impressora serial o arquivo solicitado.

- STEP/DOS

Modo em que o usuário dispõe de um sistema operacional compatível com o CP/M.

- DIR

Mostra o diretório do disquete.

- COPY

Copia arquivos de um disquete para outro.

- RENAME

Altera nome de arquivos.

- ERASE

Apaga arquivos.

- FORMAT

Formata disquetes.

- SYSGEN

Copia o sistema do STEP2 para um disquete recém formatado.

6. LINGUAGEM DE SIMULAÇÃO

6.1. Desenvolvimento

Na fase inicial do desenvolvimento da linguagem procedeu-se o estudo de uma série de analisadores de protocolo, obtendo-se então uma lista das características das linguagens de simulação destes equipamentos. Com esta lista, mais as características de monitoração do STEP-2 definiu-se então quais as características que a linguagem de simulação deveria suportar, além disso definiu-se também que ela seria baseada no BASIC em virtude da ampla divulgação que esta linguagem possui.

A fase seguinte foi a de definição da sintaxe da linguagem e do ambiente de simulação. Nesta fase também optou-se pela implementação da linguagem através de um compilador em razão, principalmente, da maior velocidade de execução que é possível obter no código gerado por compilador do que utilizando um interpretador.

A última fase constou da implementação do compilador e do ambiente de desenvolvimento de software de simulação. A análise sintática-semântica do compilador foi baseada na metodologia "top-down", algoritmo descendente-recursivo, sendo a geração de código dirigido pela sintaxe. O ambiente de desenvolvimento consta de um editor de texto e de um programa gerenciador de arquivos (USER MODE). Tanto o compilador quanto os programas do ambiente de desenvolvimento foram implementados em PASCAL.

6.2. Descrição do STEP-BASIC

Um programa em STEP-BASIC é formado de um Módulo de Definição de Parâmetros e uma lista de linhas de comandos:

O módulo de Definição de Parâmetros define o ambiente da simulação. Este módulo inicia com o comando PARAM MODULE e termina com END PARAM. A separação entre os parâmetros é realizada por um ":".

Neste módulo é possível definir os seguintes parâmetros:

- Protocolo simulado: assíncrono, síncrono orientado a caracter ou síncrono orientado a bit.
- Código: ASCII, EBCDIC ou hexadecimal 5 bits a hexadecimal 8 bits.
- Equipamento simulado: DTE ou DCE.
- Canal monitorado
- Velocidade
- Paridade
- Tipo de CRC

- Fonte do relógio de transmissão
- Caracteres de sincronismo
- Protocolo de nível 1 ativo ou não
- Comportamento da portadora
- Protocolo de nível 2 ativo ou não (somente em protocolo orientado a bit)

Com relação a lista de linhas de comandos é possível subdividir estes comandos em 4 categorias: comandos voltados para a simulação de protocolos, comandos de interface com o usuário, comandos de controle de fluxo e comandos de movimentação e operação de dados.

Os comandos voltados para a simulação de protocolos de comunicação são basicamente três.

- . comando de controle do protocolo de nível 1:

```

SET <signal da interface>      ON
                                OFF

```

- . comando de transmissão de dados.

```
SEND <lista de strings>
```

- . comando de "trigger" de evento:

```
WHEN <evento> THEN <lista de comandos>
```

Onde <evento> pode ser:

- Sequência de Caracteres (incluindo NOT e DON'T CARE)
- Erros
- TIME-OUT
- Estados dos sinais da interface
- Teclado
- BUFFER FULL

Os comandos de interface com o usuário são:

- . TEXT MODE - chaveia para a tela texto.
- . CONT MODE - chaveia para a tela contadores e timers.
- . MONIT MODE - chaveia para a tela de monitoração.

- | | |
|---|---|
| <p>. INPUT - permite a entrada de caracteres ASCII, EBCDIC ou HEXA.</p> <p>. PRINT</p> <p>. SET { <contador> } = <valor>
 { <timers> }</p> <p>. SET <variável> = { <contador> }
 { <timers> }</p> | <p>} operam na tela TEXTO</p> <p>} operam na tela contadores e timers</p> |
|---|---|

Os comandos de controle de fluxo são encontrados na maioria das implementações de BASIC encontradas no mercado, constando de IF THEN ELSE, GOTO, GOSUB e RETURN.

Por último, os comandos de movimentação e operação de dados, que constam do comando de atribuição e de uma série de operadores e funções também comuns a outras implementações de BASIC.

6.3. Exemplos:

Implementação de um terminal assíncrono em STEP-BASIC:

```
PARAM MODULE
  SIMULATE DTE:
  PROTOCOL ASYNC:
  CODE ASCII % velocidade pode ser definida na tela %
```

```
END PARAM
```

```
10 WHEN KEY# THEN A# = KEY#: SEND A#
20 END
```

Simulação de parte do protocolo BSC:

```
PARAM MODULE
  SIMULATE DTE:
  PROTOCOL SYNC:
  CODE EBCDIC:
  SPEED 9600:
  CRC CRC16
```

```
END PARAM
```

```
10 SEND C"ET"
20 SEND H"40407F7F" + C"EQ"
30 WHEN RX# = C"ET" THEN GOTO 20 OR
   WHEN RX# = C" SX" THEN GOTO 40 OR
   WHEN TIMEOUT = 300 THEN GOTO 20
40 WHEN ERR CRC ON RX# THEN GOTO 60 OR
   WHEN RX# = C"EX DC DC DC" THEN GOTO 50
50 SEND C"DL" + "/": GOTO 20
60 SEND C"NK": GOTO 30
70 END
```