

UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

IMPLEMENTAÇÃO DE UMA SOLUÇÃO MODULAR E ESCALÁVEL DAS
FUNÇÕES DAED PARA O NÍVEL 2 DO SISTEMA DE SINALIZAÇÃO
POR CANAL COMUM NÚMERO 7 USANDO DISPOSITIVOS DE LÓGICA
PROGRAMÁVEL

HILLERMANN FERREIRA OSMÍDIO LIMA

MANAUS

2012

UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

HILLERMANN FERREIRA OSMÍDIO LIMA

IMPLEMENTAÇÃO DE UMA SOLUÇÃO MODULAR E ESCALÁVEL DAS
FUNÇÕES DAED PARA O NÍVEL 2 DO SISTEMA DE SINALIZAÇÃO
POR CANAL COMUM NÚMERO 7 USANDO DISPOSITIVOS DE LÓGICA
PROGRAMÁVEL

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para a obtenção do título de Mestre em Engenharia Elétrica, área de concentração Sistemas Embarcados

Orientador: Prof. Dr. Cícero Ferreira Fernandes Costa Filho

MANAUS

2012

Ficha Catalográfica
(Catalogação realizada pela Biblioteca Central da UFAM)

L732i	<p>Lima, Hillermann Ferreira Osmídio Implementação de uma solução modular e escalável das funções daed para o nível 2 do sistema de sinalização por canal comum número 7 usando dispositivos de lógica programável / Hillermann Ferreira Osmídio Lima. - Manaus: UFAM, 2012. 137 f.; il. color.</p> <p>Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal do Amazonas. Orientador: Prof. Dr. Cícero Ferreira Fernandes Costa Filho</p> <p>1. MTP2-Message Transfer Part level 2. SS7-Sistema de Sinalização número 7 3. VHDL- Linguagem de descrição de hardware 4. Simulação I. Costa Filho, Cícero Ferreira Fernandes (Orient.) II. Universidade Federal do Amazonas III. Título</p> <p>CDU (2007): 621.39(043.2)</p>
-------	---

HILLERMANN FERREIRA OSMÍDIO LIMA

**IMPLEMENTAÇÃO DE UMA SOLUÇÃO MODULAR E ESCALÁVEL
DAS FUNÇÕES DAED PARA O NÍVEL 2 DO SISTEMA DE
SINALIZAÇÃO POR CANAL COMUM NÚMERO 7 USANDO
DISPOSITIVOS DE LÓGICA**

Dissertação apresentada ao Programa de Pós-Graduação
em Engenharia Elétrica da Universidade Federal do
Amazonas, como requisito parcial para obtenção do
título de Mestre em Engenharia Elétrica na área de
concentração Controle e Automação de Sistemas.

Aprovado em 18 de Dezembro de 2012.

BANCA EXAMINADORA

Cícero F. F. Costa Filho

Prof. Dr. Cícero Ferreira Fernandes Costa Filho, Presidente

Universidade Federal do Amazonas- UFAM

Marly Guimarães

Marly Guimarães Fernandes Costa, Membro

Universidade Federal do Amazonas- UFAM

Jozias Parente de Oliveira

Prof. Dr. Jozias Parente de Oliveira, Membro

Universidade do Estado do Amazonas- UEA

Dedico este trabalho à minha esposa Karina Lima,
meu filho João Victor e minha mãe Maria das
Graças, como uma forma de reconhecer todo o
esforço, ajuda e compreensão.

AGRADECIMENTOS

Primeiramente a Deus, pela saúde, sabedoria, perseverança e paciência dadas a mim durante o desenvolvimento deste trabalho.

À minha maravilhosa esposa Karina Lima por todo suporte, carinho, amor, cuidado incentivo para que eu pudesse me dedicar ao término deste trabalho.

Ao meu filho amado João Victor, cuja presença neste mundo foi um dos maiores incentivos para que eu chegasse a esse momento.

À minha mãe Maria das Graças Lima pelo seu grande amor, cuidado e pela presença constante na minha vida.

Ao meu pai Laudimiro Lima pelo incentivo, companheirismo, esforço e investimento em minha educação.

Ao meu orientador, professor Dr. Cícero Ferreira Fernandes Costa Filho, que sempre me incentivou nos momentos mais desanimadores desta caminhada e pela sua dedicação para que o trabalho fosse feito da melhor maneira possível.

Ao meu grande amigo MSc. Victor Afonso Valenzuela Diaz pelo suporte em todas as fases deste trabalho. Pela sua disponibilidade, pelos e-mails com palavras de incentivo e por abrir mão de passar tempo com sua família para me ajudar a vencer mais esta etapa na minha vida.

Ao Instituto Federal de Educação, Ciência e Tecnologia (IFAM-CMDI) pelo apoio.

Ao Centro de Tecnologia Eletrônica e da Informação (CETELI) pelo apoio com a infraestrutura para a realização deste trabalho.

À Financiadora de Estudos e Projetos, FINEP e à empresa Trópico Telecomunicações, por acreditar e apoiar este trabalho.

Ao meu grupo de oração por todas as suas súplicas a Deus para que eu conseguisse me manter firme diante de tantas dificuldades.

Ao professor Dr. Daniel Guzman pela grande ajuda na revisão da dissertação.

A todos, AGRADEÇO

Resumo

Esta dissertação apresenta uma implementação em VHDL da camada MTP-2 da SS7, parte *Low*, em conjunto com uma Matriz de Comutação Programável de modo a constituir, da forma mais genérica possível, uma solução modular, portátil (reutilizável) e escalável para poder ser usada em várias tecnologias e em equipamentos de telecomunicações com diferentes arquiteturas e capacidades. Como contribuições paralelas do trabalho destacam-se: o desenvolvimento de uma metodologia para implementação em VHDL de circuitos digitais a partir de uma descrição visual com o uso de fluxogramas; a proposta de uma técnica de geração de vetores de forma aleatória usando o *software* MATLAB para simulação e validação de circuitos digitais usando linguagem de descrição de *hardware*, permitindo a detecção de condições de falha que dificilmente seriam avaliadas com vetores gerados de forma manual. Como resultado, este trabalho gerou artefato de utilização prática, apresentando um considerável aumento na capacidade de tratamento de Enlaces SS7 de equipamentos de telecomunicações, quando comparado com trabalhos realizados anteriormente.

Palavras-chave: MTP-2, SS7, linguagem de descrição de *hardware*, VHDL, simulação

Abstract

This dissertation presents an implementation in VHDL of the MTP-2 layer of SS7, Low part, together with a Programmable Switching Matrix, to reach, as more generic as possible, a modular, portable (reusable) and scalable solution to be used in various technologies and telecommunications equipments with different architectures and capabilities. As parallel contributions, this work includes: the development of a methodology for implementing digital circuits in VHDL based on a visual description using flowcharts; the proposing of a technique for generating random vectors using the MATLAB software for simulation and validation of digital circuits using hardware description language, allowing the detection of fault conditions that would hardly be evaluated with manually generated vectors. As a result, this work generated practical use artifact, presenting a substantial increase capacity on treatment of SS7 links in telecommunications equipments, when compared with previous related works.

Keywords: MTP-2, SS7, hardware description language, VHDL, simulation

ÍNDICE DE ILUSTRAÇÕES

FIGURA 1.1 – BLOCOS FUNCIONAIS DA MTP-2L E SUAS INTERFACES.....	18
FIGURA 2.1 – ESTRUTURA DA SS7 NA CENTRAL TRÓPICO RA.....	25
FIGURA 2.2 – ARQUITETURA DA PLACA CCO.....	25
FIGURA 2.3 – CONFIGURAÇÃO DA CENTRAL JSU-16000.....	27
FIGURA 2.4 – ESTRUTURA DO MSCC#7.....	27
FIGURA 2.5 – ESTRUTURA DO SLCB.....	28
FIGURA 3.1 – QUADRO DE UM ENLACE PCM DE 2,048 MBIT/S.....	30
FIGURA 3.2 – SEQUÊNCIA DE QUADRO USANDO CAS.....	31
FIGURA 3.3 – TIPOS DE PONTOS DE SINALIZAÇÃO.....	36
FIGURA 3.4 – CONJUNTO DE ENLACES DE SINALIZAÇÃO.....	36
FIGURA 3.5 – GRUPOS DE ENLACES DE SINALIZAÇÃO.....	37
FIGURA 3.6 – RELAÇÃO DE SINALIZAÇÃO.....	37
FIGURA 3.7 – ROTAS DE SINALIZAÇÃO.....	38
FIGURA 3.8 – ORGANIZAÇÃO EM CAMADAS DA SS7.....	38
FIGURA 3.9 – NÍVEIS FUNCIONAIS DA SS7.....	39
FIGURA 3.10 – ESTRUTURA DE CADA TIPO DE SU.....	45
FIGURA 3.11 – INSERÇÃO DO CRC NA SU.....	46
FIGURA 3.12 – VALIDAÇÃO DO CRC DA SU.....	47
FIGURA 3.13 – TECNOLOGIAS PARA CIRCUITOS DIGITAIS.....	47
FIGURA 3.14 – ESQUEMA SIMPLIFICADO DE UMA PAL.....	50
FIGURA 3.15 – ESQUEMA SIMPLIFICADO DE UMA PLA.....	51
FIGURA 3.16 – VÁRIOS SPLDs INTEGRADOS EM UM ÚNICO CIRCUITO INTEGRADO.....	52
FIGURA 3.17 – ESTRUTURA BÁSICA DE UM FPGA.....	53
FIGURA 3.18 – CLB DE GRANULOSIDADE GROSSA.....	54
FIGURA 3.19 – EXEMPLO DE UMA LUT.....	55
FIGURA 3.20 – ESTRUTURA DE ROTEAMENTO EM UM FPGA.....	56
FIGURA 3.21 – TECNOLOGIA DE PROGRAMAÇÃO SRAM.....	59
FIGURA 3.22 – COMUTADOR PROGRAMÁVEL BASEADO EM EPROM.....	60
FIGURA 3.23 – NÍVEIS DE ABSTRAÇÃO.....	61
FIGURA 3.24 – EXEMPLO DE DESCRIÇÃO ESTRUTURAL E FUNCIONAL.....	62
FIGURA 4.1 – CONVERSÃO DE DIAGRAMA SDL PARA FSM E VHDL.....	66

FIGURA 4.2 – EXEMPLO DE FLUXOGRAMA DE CIRCUITO SÍNCRONO	68
FIGURA 4.3 – MODELO DE UM FLUXOGRAMA SÍNCRONO EM VHDL.....	68
FIGURA 4.4 – MODELO DE UM FLUXOGRAMA ASSÍNCRONO EM VHDL.....	69
FIGURA 4.5 - MODELO DE UM FLUXOGRAMA MISTO EM VHDL.....	69
FIGURA 4.6 – EXEMPLO DE FLUXOGRAMA	70
FIGURA 4.7 – CRIAÇÃO DA ENTIDADE PARA O FLUXOGRAMA CONTADOR.....	71
FIGURA 4.8 – CRIAÇÃO DA ARQUITETURA PARA O FLUXOGRAMA CONTADOR.....	71
FIGURA 4.9 – CRIAÇÃO DO PROCESSO PARA O FLUXOGRAMA CONTADOR	72
FIGURA 4.10 – METODOLOGIA DE PROJETO USANDO DLPS	72
FIGURA 4.11 – PROCESSO DE SIMULAÇÃO COM ARQUIVO DE TESTE.....	74
FIGURA 4.12 – POSICIONAMENTO DO CIRCUITO NO DLP	75
FIGURA 5.1 – INTERFACES DO DAEDT	77
FIGURA 5.2 – MÁQUINA DE ESTADOS DO DAEDT	77
FIGURA 5.3 – MÁQUINA DO DAEDT NO ESTADO IN SERVICE	78
FIGURA 5.4 – MÁQUINA DO DAEDT NO ESTADO S1	79
FIGURA 5.5 – MÁQUINA DO DAEDT NO ESTADO S2	79
FIGURA 5.6 – FLUXOGRAMA DA TRANSMISSÃO DE <i>FLAG</i>	80
FIGURA 5.7 – DIAGRAMA DE TEMPO DA TRANSMISSÃO DE <i>FLAG</i>	81
FIGURA 5.8 – FLUXOGRAMA DA TRANSMISSÃO DE OCTETOS	81
FIGURA 5.9 – DIAGRAMA DE TEMPO DA TRANSMISSÃO DE OCTETOS	82
FIGURA 5.10 – DIAGRAMA DE TEMPO DA TRANSMISSÃO DO ÚLTIMO OCTETO.....	83
FIGURA 5.11 – FLUXOGRAMA DA TRANSMISSÃO DO CRC	84
FIGURA 5.12 – DIAGRAMA DE TEMPO DA TRANSMISSÃO DO CRC	84
FIGURA 5.13 – DIAGRAMA DE TEMPO DA GERAÇÃO DA SU COM <i>FLAG</i> , OCTETOS E CRC	85
FIGURA 5.14 – INSERÇÃO DE BIT DE PREENCHIMENTO	86
FIGURA 5.15 – SEQUÊNCIA 11111 EM OCTETOS DISTINTOS.....	86
FIGURA 5.16 – CONTADOR DE BITS CONSECUTIVOS EM NÍVEL ALTO.....	87
FIGURA 5.17 – FLUXOGRAMA DO GERADOR DE CRC.....	88
FIGURA 5.18 – DIAGRAMA DE TEMPO DO GERADOR DE CRC	89
FIGURA 5.19 – CIRCUITO LFSR PARA CÁLCULO CRC NA TRANSMISSÃO	89
FIGURA 5.20 – ESTRUTURA DO ENLACE PCM	90
FIGURA 5.21 – GERAÇÃO DO OCTETO PARA O E1	90
FIGURA 5.22 – DIAGRAMA DE TEMPO DO SINAL OCTETO E1	91

FIGURA 5.23 – MARCAÇÃO DE INÍCIO DE QUADRO NO ENLACE PCM.....	91
FIGURA 5.24 – INSERÇÃO DOS BITS DAS SU NO INTERVALO DE CANAL DO E1	92
FIGURA 5.25 – DIAGRAMA DE TEMPO DA INSERÇÃO DO OCTETO NO E1	92
FIGURA 5.26 – COMUNICAÇÃO ENTRE DAEDT E DAEDR	93
FIGURA 5.27 – INTERFACES DO DAEDR	93
FIGURA 5.28 – MÁQUINA DE ESTADOS DO DAEDR	94
FIGURA 5.29 – ATIVAÇÃO DO DAEDR PELO RC.....	94
FIGURA 5.30 – INSERÇÃO DOS BITS DA SU NO INTERVALO DE CANAL DO E1	96
FIGURA 5.31 – DIAGRAMA DE TEMPO DA EXTRAÇÃO DO CONTEÚDO DO CANAL.....	96
FIGURA 5.32 – GERAÇÃO DA SU NA FORMA SERIAL A 64 KBIT/S	97
FIGURA 5.33 – DIAGRAMA DE TEMPO DA GERAÇÃO DO SINAL SU64K.....	97
FIGURA 5.34 – DIAGRAMA EM BLOCO DO DETECTOR DE FLAG E ERRO	98
FIGURA 5.35 – IDENTIFICAÇÃO DA CONDIÇÃO DE <i>FLAG</i>	98
FIGURA 5.36 – IDENTIFICAÇÃO DE SETE BITS CONSECUTIVOS EM NÍVEL ALTO.....	99
FIGURA 5.37 – DELIMITAÇÃO DOS OCTETOS DE UMA SU.....	100
FIGURA 5.38 – <i>FLAGS</i> CONSECUTIVOS ANTES DO PRIMEIRO OCTETO DA SU.....	100
FIGURA 5.39 – FLUXOGRAMA DO FILTRO DE <i>FLAG</i>	101
FIGURA 5.40 – DIAGRAMA DE TEMPO DO FILTRO DE FLAG.....	102
FIGURA 5.41 – FIM DE RECEPÇÃO DOS OCTETOS CBP	102
FIGURA 5.42 – CONTAGEM DE BITS CONSECUTIVOS EM NÍVEL ALTO NA SU	103
FIGURA 5.43 – FLUXOGRAMA DO EXTRATOR DE BIT DE PREENCHIMENTO.....	104
FIGURA 5.44 – DIAGRAMA DE TEMPO EXTRATOR DE BIT DE PREENCHIMENTO	105
FIGURA 5.45 – TOTAL DE OCTETOS E BITS RECEBIDOS NO FIM DA SU.....	106
FIGURA 5.46 – FLUXOGRAMA DO VERIFICADOR DE CRC.....	107
FIGURA 5.47 – DIAGRAMA DE TEMPO DO VERIFICADOR DE CRC	108
FIGURA 5.48 - CIRCUITO LFSR PARA CÁLCULO CRC NA RECEPÇÃO.....	108
FIGURA 5.49 – EXEMPLO DE DETECÇÃO DE FALHA NO CRC POR ERRO DE TRANSMISSÃO.....	109
FIGURA 5.50 – DIAGRAMA EM BLOCO DO COMUTADOR DAED→E1	112
FIGURA 5.51 – CIRCUITO DE HABILITAÇÃO PARA O DAED NO COMUTADOR	113
FIGURA 5.52 – DIAGRAMA DE TEMPO DE HABILITAÇÃO DO DAED NO COMUTADOR	114
FIGURA 5.53 – GERAÇÃO DOS E1S DE TRANSMISSÃO NO COMUTADOR	114
FIGURA 5.54 – GERAÇÃO DOS E1S DE RECEPÇÃO NO COMUTADOR	115
FIGURA 5.55 – DECODIFICADOR DE ENDEREÇOS DA MATRIZ DE COMUTAÇÃO	116

FIGURA 5.56 – REGISTROS A E B DE CONFIGURAÇÃO PARA UM DAED	116
FIGURA 6.1 – FORMATO DO ARQUIVO DE SUS PARA SIMULAÇÃO DO DAEDT.....	119
FIGURA 6.2 – ARQUITETURA DE SIMULAÇÃO DO DAEDT	120
FIGURA 6.3 – FLUXOGRAMA PARA GERAÇÃO DO SINCRONISMO DE QUADRO	121
FIGURA 6.4 – GERAÇÃO DO RELÓGIO DE 2 MBIT/S E SINCRONISMO DE QUADRO.....	122
FIGURA 6.5 – TRANSMISSÃO DA PRIMEIRA SU DO ARQUIVO DE VETORES.....	122
FIGURA 6.6 – TRANSMISSÃO DO <i>FLAG</i> DE ABERTURA.....	122
FIGURA 6.7 – TRANSMISSÃO DO PRIMEIRO OCTETO DA SU.....	122
FIGURA 6.8 – TRANSMISSÃO DO ÚLTIMO OCTETO DA SU.....	122
FIGURA 6.9 – TRANSMISSÃO DOS 16 BITS DE CRC DA SU	123
FIGURA 6.10 – INSERÇÃO DOS BITS DA SU NO CANAL 1 DO E1	123
FIGURA 6.11 – ARQUITETURA DE SIMULAÇÃO DO DAEDR.....	123
FIGURA 6.12 – EXTRAÇÃO DOS BITS DA SU NO CANAL 1 DO E1	124
FIGURA 6.13 – ATIVAÇÃO DO DAEDR	124
FIGURA 6.14 – DETECTOR DE <i>FLAG</i>	124
FIGURA 6.15 – RECEPÇÃO DO PRIMEIRO OCTETO SBP DA SU.....	124
FIGURA 6.16 – RECEPÇÃO DE TODOS OS OCTETOS DE UMA SU	124

ÍNDICE DE TABELAS

TABELA 5.1 – EQUIVALÊNCIA DE OPERAÇÕES ENTRE DAEDT E DAEDR.....	95
TABELA 7.1 – RECURSOS UTILIZADOS NO FPGA PARA A MTP-2L.....	126

SUMÁRIO

1	INTRODUÇÃO	15
1.1	MOTIVAÇÃO	16
1.2	OBJETIVO	18
1.2.1	Objetivo Geral	18
1.2.2	Objetivos Específicos	18
1.3	ORGANIZAÇÃO DA DISSERTAÇÃO	19
2	REVISÃO BIBLIOGRÁFICA	21
3	FUNDAMENTOS	29
3.1	SINALIZAÇÃO TELEFÔNICA	29
3.1.1	Sinalização por Canal Associado	30
3.1.2	Sinalização por Canal Comum número 7	31
3.1.2.1	Objetivos e aplicações da SS7	33
3.2	SINALIZAÇÃO POR CANAL COMUM NÚMERO 7	34
3.2.1	Características Gerais	34
3.2.2	Componentes de uma rede SS7	35
3.2.2.1	Pontos de Sinalização	35
3.2.2.2	Enlaces de Sinalização	36
3.2.2.3	Relação de sinalização	37
3.2.2.4	Rotas de Sinalização	37
3.2.3	Arquitetura da SS7	38
3.2.3.1	A camada MTP	39
3.2.3.2	A Camada de Aplicação	40
3.3	O NÍVEL 2 DA CAMADA MTP DA SS7	40
3.3.1	Funções da MTP-2	41
3.3.1.1	Delimitação da SU	41
3.3.1.2	Alinhamento de SU	42
3.3.1.3	Detecção de Erro	42
3.3.1.4	Correção de Erro	42
3.3.1.5	Alinhamento Inicial	43

3.3.1.6	Supervisão de Erro no Enlace de Sinalização	43
3.3.1.7	Controle de Fluxo.....	43
3.3.2	Mensagem de Sinalização	44
3.4	TECNOLOGIAS PARA IMPLEMENTAÇÃO DE CIRCUITOS DIGITAIS ..	47
3.4.1	ASIC	48
3.4.1.1	<i>Full-custom</i>	48
3.4.1.2	<i>Standard-cell</i>	49
3.4.1.3	<i>Gate-array</i>	49
3.4.2	PLA e PAL	50
3.4.3	CPLD	51
3.4.4	FPGA	52
3.4.4.1	Arquitetura de um FPGA	53
3.4.4.2	Blocos Lógicos Configuráveis	53
3.4.4.3	Bloco de I/O Configurável	55
3.4.4.4	Interconexões Programáveis	55
3.4.5	Considerações de projeto.....	57
3.4.5.1	Sinais de <i>Clock</i>	58
3.4.5.2	Tecnologias de programação.....	58
3.4.6	Linguagem de Descrição de <i>Hardware</i>	60
3.4.6.1	Níveis de abstração de um Circuito Digital	61
3.4.6.2	VHDL.....	62
4	METODOLOGIA	64
4.1	RECURSOS	75
5	IMPLEMENTAÇÃO.....	76
5.1	DETALHAMENTO DO DAEDT	76
5.1.1	Transmissão de <i>Flag</i>	80
5.1.2	Transmissão de Octeto.....	81
5.1.3	Transmissão de CRC	83
5.1.4	Geração da SU com <i>flags</i>, octetos e CRC.....	85
5.1.5	Contador de bits consecutivos em nível alto na SU.....	85
5.1.6	Gerador de CRC.....	87
5.1.7	Interface entre o DAEDT e o Nível 1.....	89

5.2	DETALHAMENTO DAEDR.....	93
5.2.1	Interface entre o Nível 1 e o DAEDR.....	95
5.2.2	Detector de <i>Flag</i> e Erro	97
5.2.3	Filtro de <i>Flag</i>	99
5.2.4	Contador de bits consecutivos em nível alto na SU.....	103
5.2.5	Extrator de bit de preenchimento.....	103
5.2.6	Verificador de CRC	107
5.3	COMUTADOR DAED ↔ E1.....	109
5.3.1	Bloco de habilitação de um par DAETD/DAEDR.....	112
5.3.2	Circuito de comutação no sentido DAEDT → E1	114
5.3.3	Circuito de comutação no sentido E1 → DAEDR.....	114
5.3.4	Interface entre o Nível 3 e a Matriz de Comutação	115
6	SIMULAÇÃO.....	117
6.1	SIMULAÇÃO DO DAEDT	118
6.2	SIMULAÇÃO DO DAEDR.....	123
7	ANÁLISE DOS RESULTADOS, CONCLUSÕES E TRABALHOS FUTUROS.....	125
7.1	ANÁLISE DOS RESULTADOS.....	125
7.2	CONCLUSÕES	127
7.3	TRABALHOS FUTUROS.....	128
	REFERÊNCIAS BIBLIOGRÁFICAS	130
	ANEXO 1 - DIAGRAMA FUNCIONAL DA MTP-2	133
	ANEXO 2 - DIAGRAMA SDL DO DAEDT.....	134
	ANEXO 3 - DIAGRAMA SDL DO DAEDR.....	135

1 INTRODUÇÃO

A rede de telefonia pode ser considerada como uma rede mundial com bilhões de acessos e controlada por diversos processadores distribuídos em várias centrais telefônicas e outros elementos da rede. Sinalização, de uma forma geral, pode ser definida como a troca de informações entre os diversos elementos. No caso de uma rede de telecomunicações, sinalização é a troca de informações relacionada ao estabelecimento e controle de uma conexão, incluindo informações de gerência e serviços prestados sobre essa conexão. Vários sistemas de sinalização foram desenvolvidos ao longo dos anos para permitir que a troca de mensagens entre os elementos da rede de telefonia fosse realizada da melhor forma possível. Idealmente, a sinalização entre os elementos deve ser flexível o bastante para possibilitar a inclusão de novas funcionalidades sem a necessidade de mudanças na estrutura da rede.

O Sistema de Sinalização Número 7, ou SS7 (*Signalling System Number 7*), foi projetado inicialmente com o objetivo de controlar eventos relacionados a chamadas telefônicas de um modo mais eficiente que os sistemas de sinalização até então existentes. A evolução da arquitetura dos protocolos da SS7 permitiu que uma série de novos serviços relacionados a chamadas telefônicas pudessem ser oferecidos. Um exemplo recente que teve início no Brasil em 2009 é o serviço de portabilidade numérica, que permite ao usuário manter o número do telefone fixo ou móvel independente da operadora a que estiver vinculado e da central telefônica em que está conectado.

Apesar da evolução das telecomunicações e, em particular, da telefonia para uma rede centrada em dados, a SS7 continua a ter nessas redes um papel muito importante, permitindo que os sistemas telefônicos funcionem como verdadeiras redes, interligando os processadores que as controlam nos níveis regional, nacional e internacional. Essa continuação da relevância da SS7 vale tanto para as redes de telefonia fixa e móvel já existentes, como para novos elementos de rede. A eficiência, versatilidade e tolerância a falhas da SS7 continuam a lhe garantir um papel importante tanto nas redes de telecomunicações atuais como em novas redes que possam vir a surgir no futuro.

Os elementos de rede tais como Centrais Telefônicas, Pontos de Transferência de Sinalização e Servidores de Sinalização têm sofrido constantes evoluções desde o advento das centrais digitais e da SS7. Alavancados pela evolução da tecnologia de componentes eletrônicos, incluindo, entre outros, processadores cada vez mais poderosos e componentes de lógica programável (CPLDs, FPGAs, ASICs, etc), esses elementos de rede vêm atingindo

níveis de compactação, eficiência e consumo que eram impossíveis de serem alcançados com as tecnologias dos anos 80 e 90, quando a SS7 começou a ser difundida.

1.1 MOTIVAÇÃO

Em Agosto de 2009, a Fundação de Apoio Institucional Rio Solimões - UNISOL, que tem como objetivo apoiar a Universidade Federal do Amazonas - UFAM em suas atividades de pesquisa, ensino, extensão e desenvolvimento institucional, apresentou à Financiadora de Estudos e Projetos - FINEP, uma proposta para obtenção de apoio financeiro do Fundo para o Desenvolvimento Tecnológico das Telecomunicações - FUNTTEL no âmbito do Plano Nacional de Ciência e Tecnologia para o desenvolvimento do projeto *Hardware* Universal Inovador – HUI. Aprovado dentro da chamada pública MCT/FINEP/MC/FUNTTEL/Áreas Temáticas Prioritárias 01/2009, o projeto tem como meta a modernização e evolução tecnológica e industrial de produtos de telecomunicações, sendo a empresa Trópico Telecomunicações a interveniente.

A Trópico Telecomunicações desenvolve equipamentos de telecomunicações para a rede pública tendo como principais clientes as operadoras de telefonia. Além de serem equipamentos (por exemplo, centrais telefônicas) que envolvem uma diversidade muito grande de funções *hardware* e *software*, os mesmos têm a característica de possuir tempos de vida comerciais muito extensos (dezenas de anos). Por causa da obsolescência dos componentes eletrônicos com o passar do tempo, esses equipamentos necessitam de uma contínua atualização, tendo em vista as demandas de manutenção, evolução e expansão do produto.

Os equipamentos da empresa Trópico Telecomunicações vêm sendo usados desde o final dos anos 80 e, portanto, possuem várias gerações de *hardware*. Quanto mais antigas são essas gerações, mais difícil e custoso torna-se a obtenção de seus componentes. A possibilidade de resolver essas partes de legado tecnológico com placas de nova tecnologia resulta em benefícios nas várias fases do processo produtivo e consequentemente na competitividade da empresa.

O foco do projeto HUI é o desenvolvimento de uma nova geração de *hardware* com tecnologia inovadora que permite resolver esses problemas de obsolescência. Para tal, faz uso de componentes programáveis (FPGAs), cujos conteúdos são desenvolvidos usando linguagem de descrição de *hardware*, como VHDL, de tal forma que a solução possa ser usada em componentes de diferentes fabricantes e que possa ser replicada de forma modular e

escalável, atendendo toda a gama de produtos da empresa, desde aqueles de pequena capacidade e desempenho até os de maior complexidade. Sendo equipamentos utilizados no núcleo da rede telefônica, eles são responsáveis pelo controle de dezenas de milhões de chamadas por dia e, portanto, os aspectos de qualidade preservada ao longo do tempo de vida do equipamento são essenciais.

As placas eletrônicas desenvolvidas no projeto HUI, na medida em que utilizam tecnologias de componentes de ponta, como FPGAs, possuem a importante característica de serem configuráveis. A característica de configurabilidade confere às mesmas uma dupla finalidade: a primeira é que uma nova placa do HUI pode ser utilizada para substituir um grande número de placas do legado, bastando apenas que se mude o código programável; a segunda é que uma nova placa do HUI pode ser utilizada para implementação de novas funcionalidades, ainda não existentes.

O trabalho de pesquisa aqui apresentado, em conjunto com o trabalho de pesquisa de CARVALHO (2012), ambos desenvolvidos dentro do projeto HUI, se complementam para a obtenção de uma solução modular, portátil e escalável de uma parte essencial do *hardware* dos equipamentos de telefonia nos quais se inserem os equipamentos da Trópico Telecomunicações, que é o Nível 2 (ou camada MTP-2) da SS7. Além dos resultados práticos a serem utilizados no projeto HUI, este trabalho insere-se dentro de um contexto de aproximação Universidade – Empresa, gerando tanto benefícios acadêmicos, para a UFAM, quanto tecnológicos, para a empresa Trópico Telecomunicações.

Para a divisão do conteúdo das duas dissertações foi usado o conceito mostrado por DIAZ (2009), onde a camada MTP-2 da SS7 é dividida em duas partes denominadas MTP-2 *High* (ou MTP-2H) e MTP-2 *Low* (ou MTP-2L), sendo esta última o foco deste trabalho.

A MTP-2L corresponde aos blocos funcionais da MTP-2 que manipulam as mensagens de sinalização no nível de bit, tanto na transmissão como na recepção. Por um lado, a MTP-2L faz interface com a MTP-2H e por outro lado faz interface com o Nível 1 (ou MTP-1), que corresponde à última camada, que inclui o meio físico de transmissão e recepção. A Figura 1.1 mostra os blocos funcionais da MTP-2L. O digrama funcional completo da MTP-2 encontra-se no Anexo 1.

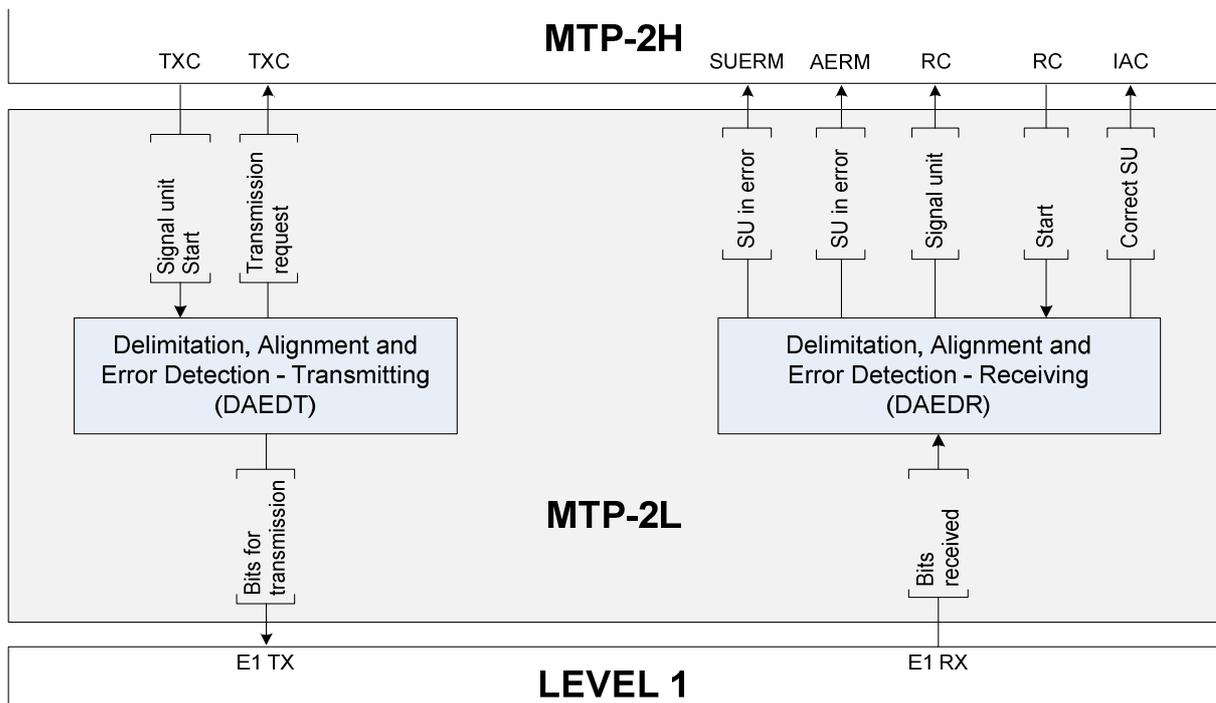


Figura 1.1 – Blocos funcionais da MTP-2L e suas interfaces

1.2 OBJETIVO

1.2.1 Objetivo Geral

O objetivo deste trabalho de pesquisa é implementar as funções DAEDT (*Delimitation, Alignment and Error Detection – Transmitting*) e DAEDR (*Delimitation, Alignment and Error Detection – Receiving*) da MTP-2 do sistema de Sinalização por Canal Comum Número 7 usando uma linguagem de descrição de *hardware* para que a solução possa ser replicada de forma modular, portátil e escalável em equipamentos de telecomunicações com múltiplos enlaces de sinalização.

1.2.2 Objetivos Específicos

- Implementar em lógica programável as funções DAED – *Delimitation, Alignment and Error Detection* da MTP-2, incluindo:
 - Inserção e detecção de *Flags* de abertura e fechamento de SU;
 - Geração e verificação de CRC;
 - Inserção e extração de bits de preenchimento (*stuffing bits*);

- Interface com a função TXC (*Transmission Control*);
 - Interface com o RC (*Reception Control*) da MTP-2;
 - Interface com o Nível 1 – Transmissão e Recepção.
-
- Incluir na implementação mecanismos que permitam que as funções DAED sejam replicadas de forma modular, portátil e escalável em equipamentos de várias capacidades;
 - Implementar uma Matriz de Comutação Programável entre os canais dos enlaces E1 que transportam os enlaces SS7 e os tratadores DAED;
 - Caracterizar a implementação mostrando os recursos necessários para a sua realização em FPGA;
 - Desenvolver uma metodologia para implementar em VHDL a recomendação internacional RFC Q.703 a partir das descrições textuais e diagramas em SDL.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Este trabalho foi organizado de forma a facilitar o entendimento do conteúdo tanto por profissionais da área de telecomunicações como da área de *hardware*. Está dividido em 7 capítulos assim distribuídos:

- Capítulo 1 – Apresenta o trabalho de uma forma geral, mostrando sua relevância, objetivos e como o mesmo pode contribuir para o desenvolvimento dos equipamentos de telecomunicações que fazem uso da SS7;
- Capítulo 2 – Apresenta alguns dos trabalhos relacionados, enfatizando as diferentes formas de implementação do Nível 2 da SS7, comparando-as com a solução aqui apresentada;
- Capítulo 3 – Apresenta os fundamentos teóricos, mostrando a evolução da sinalização telefônica até chegar à SS7. Também mostra os detalhes da MTP-2, principal foco deste trabalho, assim como algumas tecnologias de *hardware* que podem ser usadas para sua implementação;

- Capítulo 4 – Apresenta a metodologia usada para que fosse possível alcançar todos os objetivos propostos. Descreve o método desenvolvido para a geração de código em VHDL a partir de uma descrição do circuito em fluxograma. Esse capítulo também apresenta os critérios usados para formalizar a divisão da camada MTP-2 nas partes *Low* e *High*;
- Capítulo 5 – Apresenta a descrição detalhada dos blocos DAEDT e DAEDR através do uso de fluxogramas e diagramas de tempo. Também apresenta a implementação da Matriz de Comutação Programável que permite a escalabilidade da solução;
- Capítulo 6 – Apresenta arquitetura usada na simulação dos blocos DAEDT e DAEDR, mostrando a forma como foram gerados os vetores de teste de entrada e saída com o auxílio do *software* MATLAB®;
- Capítulo 7 – Apresenta as conclusões do trabalho, destacando suas contribuições acadêmicas e tecnológicas. São mostrados os recursos necessários para implementar a solução proposta em um FPGA da família Cyclone IV para diversos cenários, com quantidades diferentes de E1s e Enlaces SS7. Ainda neste capítulo são apresentadas algumas sugestões de trabalhos futuros que podem tornar a solução mais eficiente.

2 REVISÃO BIBLIOGRÁFICA

Existem diversos trabalhos publicados na área de sinalização entre centrais telefônicas, e muitos deles relacionados diretamente com a Sinalização por Canal Comum Número 7, porém apenas uma pequena parte trata dos detalhes de implementação deste tipo de sinalização em *hardware*.

Alguns trabalhos foram analisados de forma mais detalhada pelo fato de tratarem diretamente sobre a implementação da SS7, a saber:

- ✓ “Anteprojeto de Implementação *Hardware* da MTP-2 *Lower part* da SS7”, do autor Diaz (2009);
- ✓ “Study and Simulation of CCS#7 Protocol Using System Verilog”, dos autores Bhart *et al* (2010);
- ✓ “Contribuições para a Implementação do Sistema de Sinalização por Canal Comum nº 7 em uma Central Telefônica Baseada em Ambiente de Processamento Distribuído”, do autor d’Ávila (1998);
- ✓ “Implementing Signalling System No.7 in Tropic RA System”, dos autores NETO *et al* (1990);
- ✓ “The Implementation of SS7 in JSU-16000”, dos autores SHAOREN *et al* (1996).

O trabalho de DIAZ (2009) apresenta como pode ser implementada parte das funções da MTP-2 em *hardware*. O autor mostra detalhes de implementação principalmente para o que denomina MTP-2L (*Low*), incluindo as funções a seguir:

- ✓ Delimitação das mensagens (inserção de *flag*);
- ✓ Inserção e extração de bits de preenchimento (*stuffing bits*);
- ✓ Detecção de Erros (CRC);
- ✓ Inserção e filtro automático de FISU;
- ✓ Parte do Alinhamento Inicial de Enlace (contador AERM).

Além das funções supracitadas, o autor discute a interface entre a MTP-2 e a MTP-3, além de alguns aspectos de testabilidade. As idéias de implementação são apresentadas em um nível intermediário, utilizando diagramas de blocos funcionais tais como registradores e contadores, destacando os principais sinais que os interconectam. Em alguns casos, o autor discute a base teórica por trás de algumas das suas sugestões de implementação. Para as funções da MTP-2H (*High*) em alguns casos, cita como as mesmas poderiam ser implementadas caso se decida fazê-lo em *hardware*. Para as funções de supervisão de erros, o autor trata da implementação dos contadores AERM e SUERM que participam das fases de Alinhamento e Operação do Enlace de Sinalização e discute sua participação na função de Alinhamento de Enlace.

O trabalho de pesquisa aqui apresentado tem uma forte relação com o trabalho de DIAZ (2009), uma vez que as funções definidas na MTP-2L são, em sua maioria, as funções desempenhadas pelo bloco DAED. A diferença está principalmente na forma como essas funções são modeladas. Enquanto que DIAZ (2009) usa a modelagem com elementos de eletrônica digital (portas lógicas, registradores, memórias, etc), o trabalho aqui proposto fará uso de um nível mais alto de abstração, com fluxogramas e diagramas de tempo, de maneira que a implementação possa facilmente ser realizada através do uso de uma linguagem de descrição de *hardware*, como VHDL, permitindo um alto grau de portabilidade da solução entre diferentes fabricantes e tecnologias de dispositivos de lógica programável.

O trabalho de BHART *et al* (2010) compreende o estudo e a simulação da SS7 usando a linguagem de descrição de *hardware System Verilog*. A ferramenta de simulação usada para a validação do código foi o *Verilog Compiler Simulator (VCS)* da empresa *Synopsys*. Foram apresentadas diversas formas de onda como parte dos resultados, porém pouco explicativas. Foram criados dois módulos em *System Verilog*, sendo um para simular o PS (Ponto de Sinalização) de origem transmitindo uma mensagem (Módulo *Transmitter*) e outro para simular o PS de destino recebendo a mensagem (Módulo *Receiver*). Foram criados também dois módulos para simular a interface entre a aplicação e a camada MTP e vice-versa. O código possui poucos comentários, dificultando sobremaneira seu entendimento. Como a proposta dos autores era apenas de simulação da SS7, não foi apresentada nenhuma sugestão de implementação em *hardware*. Vale ressaltar que o trabalho de BHART *et al* (2010) refere-

se a uma dissertação de mestrado e ainda está em fase de elaboração. Os comentários aqui colocados foram retirados do documento de qualificação apresentado em 2010.

O trabalho de BHART *et al* (2010) também possui uma forte relação com este trabalho de pesquisa, inclusive na modelagem das funções, pois também é usada uma linguagem de descrição de *hardware*, neste caso, o *System Verilog*. As funções do bloco DAED foram “embutidas” nos módulos *Transmitter/Receiver*. O trabalho apresentado nesta dissertação pretende tornar a solução escalável com a implementação, entre outros recursos, de uma Matriz de Comutação Programável.

O trabalho de D’ÁVILA (1998) descreve o processo de implementação da SS7 na central telefônica ELCOM, desenvolvida e produzida pela empresa Batik Equipamentos S/A. O autor fornece uma descrição detalhada sobre a estrutura da central ELCOM, mostrando os diversos módulos que a compõem. A implementação da SS7 na central ELCOM foi realizada parte através de *hardware* e *software* comerciais e parte através de *hardware* e *software* desenvolvidos inteiramente pela Batik. Segundo o autor, a motivação para o uso de produtos comerciais na central ELCOM, foi devido à sua arquitetura, que sendo baseada em computadores do tipo PC, boa parte das soluções disponíveis para esta plataforma tão comum poderia ser facilmente incorporada à estrutura da central. O autor comenta sobre o cuidado na escolha de produtos comerciais em relação aos seguintes aspectos:

- ✓ Quanto mais completa a solução comercial oferecida, menor o nível de detalhamento do produto, o que dificulta qualquer tipo de alteração ou até mesmo de reparos;
- ✓ Dependendo da origem do produto, pode ser necessário uma ou mais adaptações para uso nas redes de telecomunicações do Brasil;
- ✓ O custo de soluções comerciais é normalmente muito elevado.

Segundo D’ÁVILA (1998), a implementação da SS7 na central ELCOM foi feita através de uma placa comercial com barramento ISA para computadores PC, chamada terminal SS7, que realiza grande parte das funções da MTP-2 em *hardware*. Também foi usado um *software* comercial para realizar as funções da MTP-3 e das funções da MTP-2 que não foram implementadas em *hardware*. Este *software* é responsável pela configuração e

inicialização das placas SS7, bem como uma série de tarefas de gerência da MTP-2 e interface com a MTP-3.

A implementação das funções da MTP-2L apresentadas nesta dissertação, foi realizada da forma mais genérica possível para que não se tornasse dependente de um fabricante específico de dispositivos de lógica programável. Segundo D'ÁVILA (1998), esta característica, quando não atendida, pode trazer problemas de continuidade e evolução do produto. Para isso pretende-se usar uma linguagem de descrição de *hardware* na implementação, de forma que a mesma possa ser compilada em qualquer ferramenta de síntese para dispositivos de lógica programável.

O trabalho de NETO *et al* (1990) apresenta a implementação da SS7 na central telefônica Trópico RA, sucessora da central telefônica Trópico R. De uma forma muito clara, o autor descreve toda a estrutura da central Trópico, mostrando os módulos que compõem o equipamento, em particular, o Módulo de Sinalização por Canal Comum, MC.

Segundo NETO *et al* (1990), o nível 1 da camada MTP é implementado por uma placa eletrônica chamada TDT (Tronco Digital Temporal), e tem capacidade para tratar dois enlaces PCM a 2,048 Mbits/s (no restante deste trabalho, por simplicidade, essa taxa será referenciada como 2 Mbit/s). A central Trópico RA possui uma matriz de comutação que permite que qualquer canal de qualquer dos enlaces PCM possa ser direcionado para qualquer tratador de canal comum. As funções do nível 2 da MTP são implementadas em uma placa eletrônica chamada CCO (Controladora de Canal Comum), com capacidade para tratamento de dois enlaces de canal comum. Parte do nível 3 da MTP é implementado por uma placa eletrônica chamada CTE (Controladora de Terminais), com capacidade para tratamento das mensagens de até quatro enlaces de canal comum, ou seja, de duas placas CCO. A outra parte do nível 3 da MTP, que corresponde às funções de gerenciamento da rede de sinalização e a aplicação (nível 4) são implementadas em *software* em um módulo chamado MA (Módulo Auxiliar). A estrutura da SS7 na central Trópico RA está ilustrada na Figura 2.1.

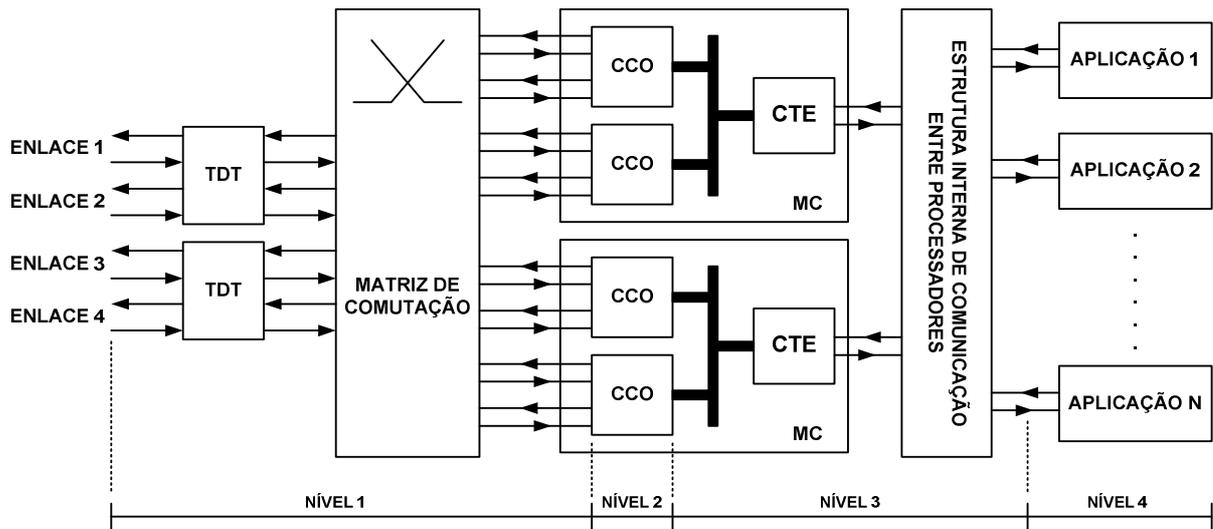


Figura 2.1 – Estrutura da SS7 na central Trópico RA

A comunicação entre as placas CCO e CTE é feita através de uma memória comum. Ambas as placas possuem processadores que podem acessar a memória comum, ficando o controle de acesso por conta de um árbitro. Quando o processador da placa CCO estiver acessando a memória comum e o processador da placa CTE também realizar um acesso, este último será colocado no estado de espera (*wait state*) até que o primeiro finalize o acesso. O mesmo acontece para o caso em que o processador da placa CTE esteja acessando a memória comum e o processador da placa CCO tentar acessá-la. A memória comum também serve como *buffer* de recepção e transmissão dos dois enlaces de sinalização tratados pela placa CCO, cuja arquitetura é mostrada na Figura 2.2.

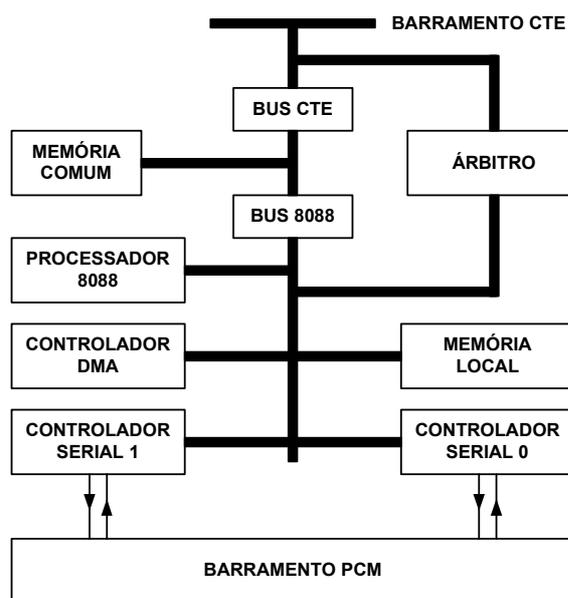


Figura 2.2 – Arquitetura da placa CCO

O controlador DMA é responsável pela transferência das mensagens do barramento PCM para a memória comum, no caso de uma recepção ou o contrário, no caso de uma transmissão. Segundo NETO *et al* (1990), os Controladores Seriais 0 e 1 foram desenvolvidos em *hardware* pelo Centro de Pesquisa e Desenvolvimento da Telebrás - CPqD, mas não menciona a tecnologia utilizada, e é responsável pelas seguintes funções do nível 2: detecção de erro de transmissão e recepção (*Check bits*), inserção e detecção de *flags* (delimitação das mensagens), inserção e extração de bits de preenchimento, inserção e filtro automático de FISU (*Fill In Signal Unit*).

NETO *et al* (1990) afirma que foram realizados testes de comunicação entre as placas CCO e CTE através da memória comum a uma taxa de 660 mensagens por segundo, sendo 330 em cada direção (transmissão e recepção), divididas da seguinte maneira: 70% das mensagens com 16 octetos, 20% com 64 octetos e 10% com 272 octetos. Segundo NETO *et al* (1990), os resultados foram satisfatórios, considerando que os processadores, tanto da placa CTE como da CCO são de 8 bits (Intel 8088).

Com exceção das funções de inserção e filtro automático de FISU (*Fill-In Signal Unit*), o bloco Controlador Serial da placa CCO possui as mesmas funções do DAED que se pretende implementar neste trabalho de pesquisa. A matriz de comutação, na arquitetura mostrada por NETO *et al* (1990), fica fora do tratador de canal comum, enquanto que nesta proposta a mesma deve ser integrada à solução, permitindo que tanto o número de enlaces E1 quanto o número de DAEDs sejam configuráveis.

O trabalho de SHAOREN *et al* (1996) mostra como a SS7 foi implementada na central telefônica JSU-16000, desenvolvida pelos próprio autores. Os autores comentam sobre a necessidade da evolução dos equipamentos de telecomunicações que fazem uso da SS7 devido ao grande número de novos serviços oferecidos a cada dia pelas empresas de telecomunicações. O controle na central telefônica JSU-16000 é totalmente distribuído nos módulos que a compõem, que consiste de 1 a 8 módulos de comutação e 1 módulo de sinalização por canal comum número 7. A conexão para os sinais de voz entre os módulos é feita através de enlaces PCM, sendo que cada módulo possui um enlace PCM ligado com todos os outros módulos da central, como mostra a Figura 2.3.

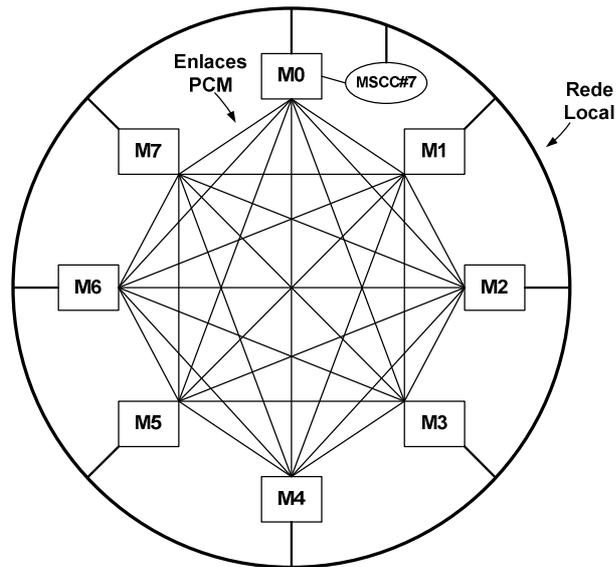


Figura 2.3 – Configuração da Central JSU-16000

O Módulo de Sinalização por Canal Comum Número 7 (MSCC#7) pode ser conectado a qualquer módulo de comutação. Na Figura 2.3, o MSCC#7 está conectado ao módulo de comutação M0. A informação de sinalização é passada para os outros módulos de comutação através de uma rede local de 10Mbps, representada na figura pela circunferência.

O MSCC#7 implementa as funções dos níveis 1, 2 e 3 da MTP e tem capacidade para tratar até 128 enlaces de sinalização. Sua estrutura é mostrada na Figura 2.4.

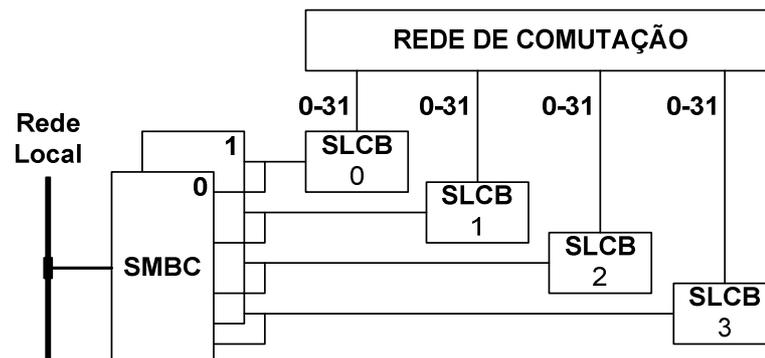


Figura 2.4 – Estrutura do MSCC#7

O bloco SLCB (*Signalling Link Control Board*) é responsável pelas funções da MTP-2. Segundo SHAOREN *et al* (1996), nesta implementação foi usado o circuito integrado Bt8071 da empresa Brooktree, que tem capacidade para tratar até 32 enlaces de sinalização (equivalente a 32 pares DAEDT/DAEDR). A Figura 2.5 mostra a estrutura do SLCB.

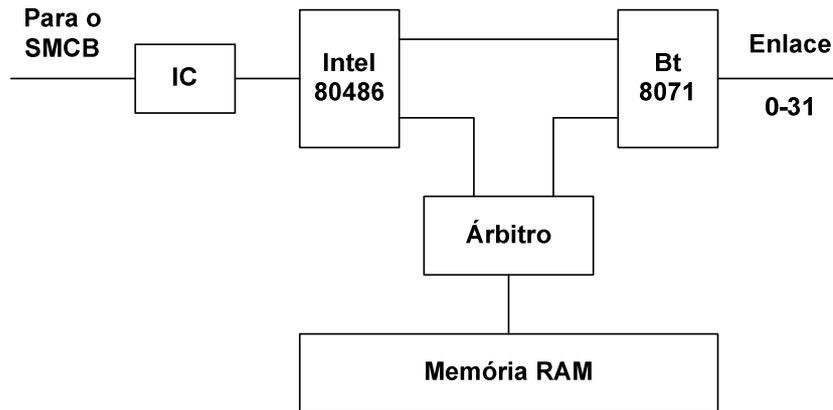


Figura 2.5 – Estrutura do SLCB

O bloco SMCB (*Signalling Master Control Board*) realiza as funções do nível 3 da MTP e mais algumas funções de operação e manutenção e através dele o MSCC#7 pode ter acesso à rede local para comunicação com qualquer outro módulo. Como o foco desta dissertação é na implementação do nível 2, o bloco SMCB não foi comentado.

Segundo SHAOREN *et al* (1996), o módulo MSCC#7 foi implementado com sucesso na central JSU-16000 para a aplicação em controle de chamadas telefônicas (TUP – *Telephone User Part*).

A empresa Brooktree foi comprada pela Conexant Systems e nas pesquisas realizadas não foi encontrado o *datasheet* do Bt8071, assim como nenhum outro componente equivalente. Esta é uma das questões importantes levantadas por D’AVILA (1998) e que pode levar o produto à descontinuidade quando a tecnologia empregada no projeto é muito específica ou possui um número muito limitado de fabricantes.

3 FUNDAMENTOS

3.1 SINALIZAÇÃO TELEFÔNICA

A rede de telefonia é uma rede mundial com bilhões de acessos e controlada por milhões de processadores distribuídos entre milhões de centrais telefônicas e outros elementos da rede. Para que essa rede funcione de forma adequada precisa haver troca de informações entre esses elementos.

Sinalização, de uma forma geral, pode ser definida como a troca de informações entre os diversos elementos e usuários que fazem parte de uma rede, seja ela qual for. No caso de uma rede de telecomunicações, sinalização é a troca de informações relacionada ao estabelecimento e controle de uma conexão e de serviços prestados sobre a mesma, incluindo gerência de rede.

Para que uma chamada telefônica seja estabelecida entre dois usuários, uma série de tarefas são executadas antes mesmo que a conversa seja iniciada. Algumas delas são:

- ✓ Detecção de telefone fora do gancho;
- ✓ Envio de tom de linha para o telefone de origem indicando que mesmo pode informar o número do telefone de destino;
- ✓ Alocação de recurso na central telefônica para tratamento da chamada;
- ✓ Verificação se o telefone chamado está livre ou ocupado;
- ✓ Envio de tom de chamada para o telefone de destino, caso esteja livre, ou envio do tom de ocupado para o telefone de origem, informando que o destino está ocupado;
- ✓ Se o destino estiver livre e atender a chamada, realização da comutação entre os circuitos para que a conversa ocorra.

Cada uma dessas tarefas é controlada por uma ou mais centrais telefônicas nas quais os usuários de origem e destino estão conectados e nas centrais que as interligam. Para isso, faz-se necessária a troca de mensagens internas ou entre centrais para que essas tarefas sejam executadas de forma correta.

A seguir serão descritos os dois principais tipos de sinalização entre centrais telefônicas: Sinalização por Canal Associado (SCA) e Sinalização por Canal Comum (SCC), sendo o último o foco deste trabalho.

A SCA foi usada por muitos anos nas centrais telefônicas e mostrou-se bastante eficaz para atender aos serviços básicos, como chamadas internacionais e interurbanas, tarifação, etc. Com o surgimento das Redes Inteligentes (RI), a quantidade e complexidade dos serviços oferecidos tornaram-se muito maior, e a SCA, da forma como foi desenvolvida, mostrou-se muito limitada e deficiente. Foi então que surgiu a SCC, que não somente superou as limitações e deficiências da SCA, como também mostrou-se flexível o bastante para atender novos serviços sem a necessidade de grandes mudanças nos equipamentos da rede.

3.1.1 Sinalização por Canal Associado

Este tipo de sinalização é transportada sobre enlaces (*links*) PCM (*Pulse Code Modulation*) com taxas de 2,048 Mbit/s (ou simplesmente 2 Mbit/s), formados por seqüências de quadros (*frames*) com duração de 125 μ s, divididos em 32 intervalos de tempo (*time slots*), também conhecidos como canais, como mostra a Figura 3.1.

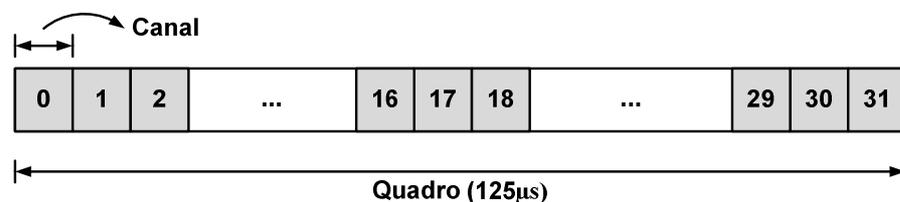


Figura 3.1 – Quadro de um enlace PCM de 2,048 Mbit/s

Apesar de cada quadro possuir 32 canais, somente 30 são dedicados para transporte de dados do usuário (principalmente voz digitalizada). O canal 0 é dedicado para sincronismo de quadro e o canal 16 para sinalização. Cada canal transporta uma informação digital de 8 bits. Assim, para se determinar a taxa de um canal em bit/s, podemos usar a equação

$$T_c = \frac{8}{125\mu} = 64\text{k bits/s}$$

Considerando um quadro completo com 32 canais de 64 kbit/s cada, a taxa de um enlace PCM será de

$$T_E = 32 \times 64\text{k} = 2,048\text{M bits/s}$$

Na SCA, o canal 16 de cada quadro transporta informações referentes a dois canais do enlace, sendo 4 bits para cada canal, conforme mostra a Figura 3.2. No primeiro quadro, a informação transportada no canal 16 é de sincronismo de multiquadro, indicando o início de uma nova sequência de quadros de sinalização dos canais 1 a 15 e 17 a 31.

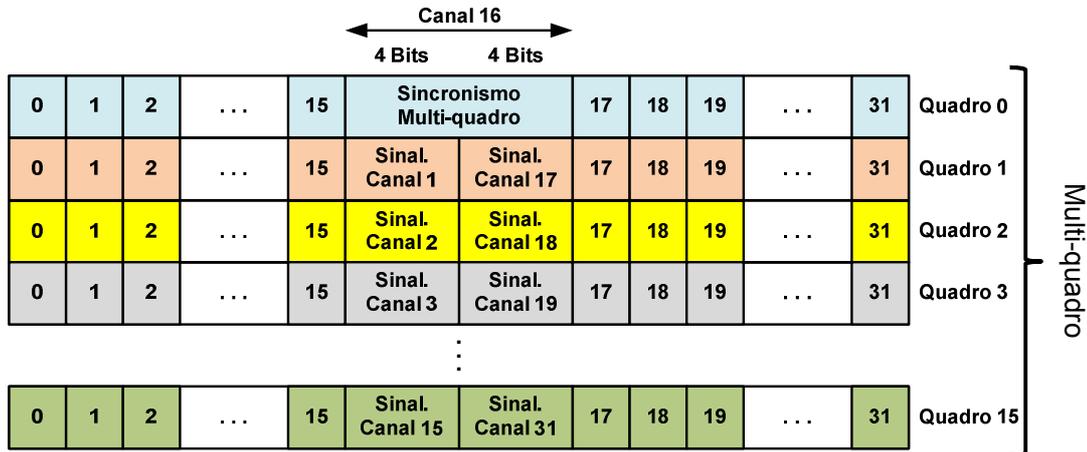


Figura 3.2 – Sequência de quadro usando CAS

Na SCA, o canal 16 de cada enlace PCM é dedicado para sinalização dos canais pertencentes somente ao enlace em questão. Quando ocorre uma falha em um enlace PCM que usa SCA ela afeta unicamente os canais do mesmo enlace e sua sinalização.

Uma das desvantagens da SCA é que o tempo necessário para a troca de mensagens é muito elevado. Isto se deve ao fato de que a sinalização de um canal só pode ocorrer no quadro destinado a ele, mesmo que nenhum dos outros canais esteja sendo usado no momento. Por isso, esse tipo de sinalização é conhecido como canal associado. Como um quadro tem a duração de 125µs, no pior caso, para se transmitir um sinal pode-se levar até 16 quadros (ou 2ms) em um sentido e mais 16 quadros para receber uma confirmação. Com a evolução dos sistemas de telecomunicações, a SCA deixou de ser utilizada pela sua forma ineficiente de tratar as mensagens de sinalização.

3.1.2 Sinalização por Canal Comum número 7

A sigla SS7 vem do inglês, *Signalling System Number 7* e possui outras denominações, em particular, no nome por extenso, onde se usa o termo “Canal Comum”. Uma dessas denominações é SCC7 (Sinalização por Canal Comum Número 7) ou CCS7 (*Common Channel Signalling Number 7*). Neste trabalho foi usada a sigla SS7 para denotar Sinalização por Canal Comum Número 7.

Há alguns anos atrás, uma central telefônica quase que em sua totalidade, tratava apenas chamadas telefônicas, cujo conteúdo era basicamente amostras digitalizadas de áudio, correspondentes à voz humana. Uma parcela bem menor ocupava a central telefônica para enviar e receber dados digitais de equipamentos como por exemplo, os *modems*.

Com a evolução das redes de telecomunicações e principalmente com a rede mundial de computadores (*Internet*), não somente a quantidade de dados trafegados aumentou, como também novos tipos de dados passaram a trafegar nessas redes, como por exemplo, vídeo. A própria rede de telefonia está convergindo cada vez mais para uma rede puramente de dados através da tecnologia VoIP (*Voice Over IP*). Para possibilitar o tratamento de todo esse volume e diversidade de dados foi necessário que o sistema de sinalização nas centrais telefônicas evoluísse.

O CCITT especificou alguns padrões internacionais de sistemas de sinalização, a saber, SS1 a SS7. Os sistemas SS1 a SS5 usam um modelo de sinalização chamado *in-channel*, onde dados e controle compartilham o mesmo canal, enquanto que os sistemas SS6 e SS7 usam a sinalização por canal comum, onde o controle e os dados trafegam em canais separados (BAGAD, 2009).

O SS6 foi o primeiro sistema de sinalização por canal comum a ser padronizado internacionalmente. Foi projetado para uso em redes analógicas e usado com taxas de 2,4 kbit/s e 4,8 kbit/s. As mensagens tinham um tamanho fixo de 28 bits, onde 20 eram usados para dados e 8 para verificação. Para tornar possível seu uso em redes digitais, foram adicionados mais 4 bits nas mensagens, compatibilizando com os canais de um enlace PCM, que possuem 8 bits (BAGAD, 2009).

A SS7 foi projetada inicialmente com o objetivo de controlar eventos relacionados a chamadas telefônicas de modo mais eficiente que os sistemas de sinalização até então existentes. A evolução da arquitetura dos protocolos da SS7 permitiu que uma série de serviços relacionados a chamadas telefônicas pudessem ser oferecidos, como por exemplo os serviços de ligação gratuita (0800) e o de lista negra, onde o usuário e/ou a operadora podem

restringir, tanto para receber como para originar chamadas de/para números de telefones previamente cadastrados.

Desde que a SS7 foi concebida, a mesma vem assumindo um papel cada vez mais importante, uma vez que as Centrais de Telefonia Pública migraram de sistemas de sinalização menos eficientes para implementações completas de SS7. Atualmente, a SS7 responde pelo roteamento de chamadas, não somente dentro de todos os países, como também entre os mesmos, além de ter uma grande contribuição para as redes de telefonia móvel.

3.1.2.1 Objetivos e aplicações da SS7

A SS7 foi criada para oferecer um sistema padronizado de Sinalização por Canal Comum que pudesse ser usado mundialmente em redes de telecomunicações de propósitos gerais. Este sistema deveria ter como principais características:

- Ser otimizado para uso em redes de telecomunicações digitais em conjunto com centrais com Controle por Programa Armazenado (CPA);
- Ser bastante flexível para poder atender não somente as necessidades atuais de comunicação entre os equipamentos, mas também permitir seu uso em novas funcionalidades para controle de fluxo de chamadas telefônicas, operação e/ou manutenção remota de equipamentos, execução e coleta de informações de tarifação, entre outros;
- Ser bastante confiável para garantir a transferência correta da informação entre os equipamentos, minimizando ao máximo os erros de comunicação causados por perda de pacotes, informações duplicadas ou fora de sequência. Técnicas como redundância do meio físico e funções para desvio automático de rota em caso de falha são usadas para garantir a confiabilidade do sistema de transporte.

A SS7 pode ser usada na sinalização de diversos serviços de telecomunicações, como por exemplo, controle de chamadas de Centrais Públicas de Telefonia e controle de serviços oferecidos na Rede Inteligente (IN ou RI), como bilhetagem especializada e lista negra.

A SS7 foi projetada para operar em canais com taxa de 64 kbits/s, mas segundo ITU-T (1993a), este sistema pode ser usado para operar também em canais analógicos e com taxas mais baixas. Para que a SS7 possa operar tanto em uma rede puramente de telefonia (para a

qual foi inicialmente criada), assim como em redes NGN e de telefonia móvel, onde a diversidade de serviços cresce a cada dia, o sistema foi projetado com uma estrutura modular para permitir que novas funcionalidades sejam agregadas sem causar impacto nas demais.

3.2 SINALIZAÇÃO POR CANAL COMUM NÚMERO 7

Esta seção é baseada principalmente na recomendação Q.700 da ITU-T (*International Telecommunication Union – Telecommunication Standardization Sector*) (ITU-T, 1993a) que descreve os elementos funcionais que compõem a SS7, assim como a relação entre os mesmos. A especificação completa é realizada em uma série de recomendações, a saber:

- Subsistema MTP (*Message Transfer Part*), descrito nas recomendações Q.701 – Q.710;
- Subsistema SCCP (*Signalling Connection Control Part*), descrito nas recomendações Q.711 a Q.719;
- Subsistema TUP (*Telephone User Part*), descrito nas recomendações Q.720 a Q.729;
- Serviços Suplementares para Rede Digital de Serviços Integrados, descrito nas recomendações Q.730 a Q.739;
- Subsistema DUP (*Data User Part*), descrito nas recomendações Q.740 a Q.749;
- Gerenciamento da SS7, descrito nas recomendações Q.750 a Q.759;
- Subsistema ISUP (*ISDN User Part*), descrito nas recomendações Q.760 a Q.769;
- Subsistema TCAP (*Transaction Capabilities Application Part*), descrito nas recomendações Q.770 a Q.779;
- Especificação de testes, descrita nas recomendações Q.780 a Q.799;

O foco deste trabalho será o subsistema de transferência de mensagens (MTP), mais especificamente as funções do Nível 2 (*Signalling Link*), descritas detalhadamente na recomendação Q.703 (ITU-T, 1993c).

3.2.1 Características Gerais

O termo “Sinalização por Canal Comum” refere-se a uma técnica de sinalização na qual um único canal (Canal Comum) transporta informações de sinalização de um ou mais usuários. São exemplos de usuários do sistema de sinalização por canal comum:

- Um assinante de uma central telefônica que deseja estabelecer uma conversação com outro assinante em uma central distante;
- Uma central de serviços de telecomunicações que está executando uma operação remota em um equipamento;
- Um equipamento que faz a coleta de informações referentes à tarifação de chamadas telefônicas.

Para que as mensagens de sinalização sejam transportadas entre os equipamentos que fazem parte da rede SS7, são usados enlaces de sinalização que incluem o meio físico usado para interligar os equipamentos. Algumas técnicas são usadas para garantir a transferência das mensagens de sinalização de forma confiável, mesmo na presença de perturbações no meio de transmissão ou de falhas na rede. Normalmente são usados no mínimo dois enlaces de sinalização para garantir um caminho alternativo para as mensagens caso um deles venha a falhar. Funções específicas no equipamento se encarregam de desviar o tráfego das mensagens automaticamente para o enlace que se encontra em serviço. A capacidade de tráfego de mensagens de sinalização em um equipamento, assim como o grau de confiabilidade do mesmo são dimensionados de acordo com a aplicação e podem exigir o uso de um grande número de enlaces de sinalização.

3.2.2 Componentes de uma rede SS7

Uma rede SS7 é composta por um conjunto de elementos (ou nós) interconectados entre si através de enlaces de transmissão e recepção. Para que a comunicação entre esses elementos da rede seja possível, os mesmos devem implementar as funções necessárias da SS7 de acordo com a sua aplicação na rede. Um equipamento na rede SS7 também é chamado de Ponto de Sinalização (PS). Como foi mencionado anteriormente, para que as mensagens de sinalização possam ser transportadas entre os diversos pontos de sinalização, faz-se uso dos enlaces de sinalização.

3.2.2.1 Pontos de Sinalização

Dependendo da função do PS, o mesmo pode ser chamado de Ponto de Fim de Sinalização (PFS) ou Ponto de Transferência de Sinalização (PTS). Um PFS é um ponto de sinalização onde os usuários estão diretamente conectados, como por exemplo, as linhas de telefone dos assinantes em uma central telefônica. O PTS é um ponto de sinalização usado

para transportar as mensagens entre PFSs, podendo inclusive passar por vários PTSs até chegar ao PFS de destino. A Figura 3.3 mostra um exemplo de rede SS7 com vários PFSs e PTSs. Para aumentar a confiabilidade na comunicação, um PS é interligado a pelo menos outros dois, permitindo que a comunicação seja estabelecida por mais de um caminho.

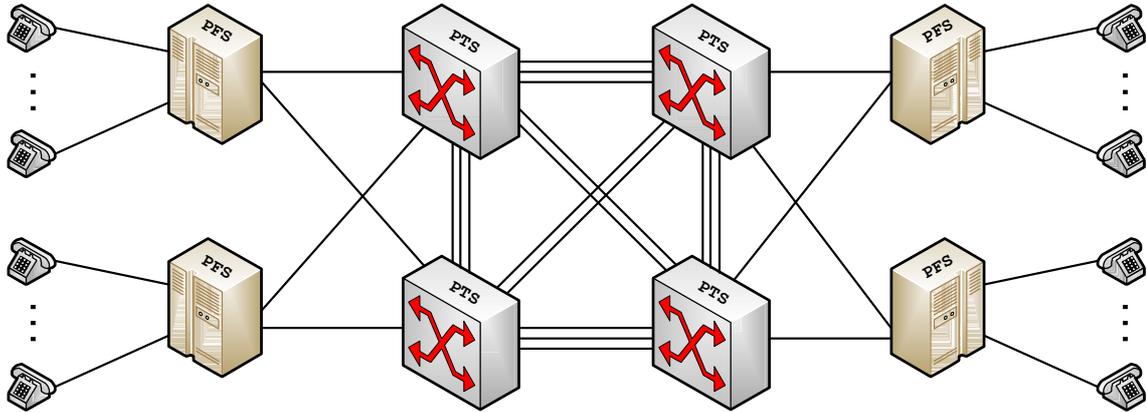


Figura 3.3 – Tipos de Pontos de Sinalização

Cada PS possui um número único, que o identifica na rede, seja ela nacional ou internacional. As mensagens que trafegam na rede possuem um campo para identificar o PS que originou a mensagem, conhecido como OPC (*Originating Point Code*) e um campo para identificar o PS de destino, chamado DPC (*Destination Point Code*).

3.2.2.2 Enlaces de Sinalização

Os Enlaces de Sinalização são usados para transportar as mensagens entre os diversos PSs. Um Enlace de Sinalização implementa a conexão entre um par de PSs, ou seja, consiste em uma conexão ponto a ponto bidirecional. Todos os enlaces de sinalização usados para interligar diretamente dois PSs formam um Conjunto de Enlaces de Sinalização, como mostra a Figura 3.4.

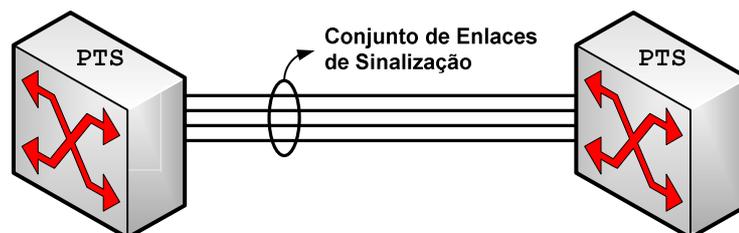


Figura 3.4 – Conjunto de Enlaces de Sinalização

Todos os enlaces dentro de um conjunto que possuem características idênticas, como por exemplo, a mesma velocidade em bits/s, fazem parte do mesmo Grupo de Enlace. Assim,

em um mesmo Conjunto de Enlaces de Sinalização é possível ter vários Grupos de Enlace, como mostra a Figura 3.5.

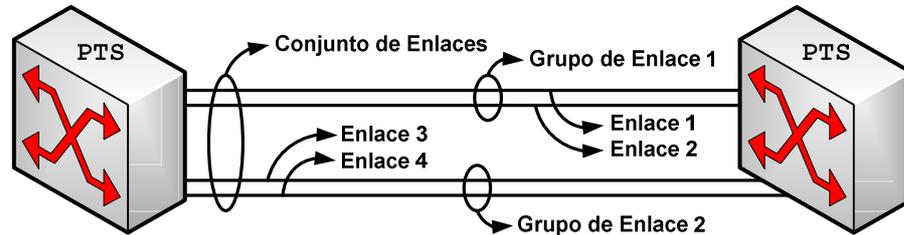


Figura 3.5 – Grupos de Enlaces de Sinalização

3.2.2.3 Relação de sinalização

Dois PSs em uma rede SS7, para os quais existe a possibilidade de comunicação, ainda que seja através de outros PSs, têm uma Relação de Sinalização, como mostra a Figura 3.6. Apesar do PS 1 não estar diretamente conectado ao PS 3, é possível a troca de mensagens entre eles através do PS 2, que para este caso, funciona como um PTS.

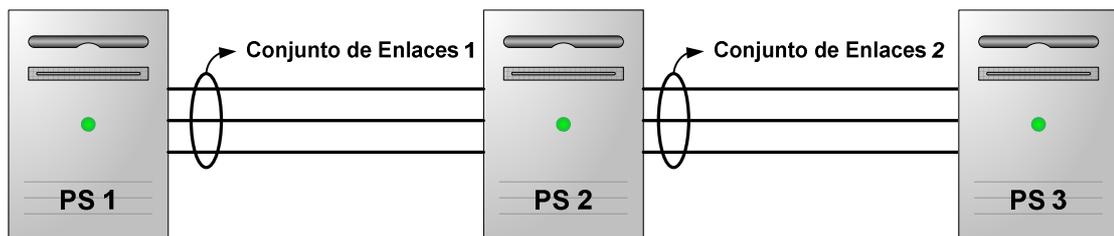


Figura 3.6 – Relação de Sinalização

3.2.2.4 Rotas de Sinalização

O caminho que uma mensagem percorre na rede desde o PS de origem até o PS de destino chama-se Rota de Sinalização. Para chegar ao PS de destino, uma mensagem poderá usar caminhos diferentes, de acordo com a distribuição dos enlaces de sinalização. O conjunto de todas as rotas possíveis para uma mesma relação de sinalização é chamado Conjunto de Rotas. A Figura 3.7 mostra o conceito de Rota de Sinalização e Conjunto de Rotas. Para ir do PS 1 ao PS 3, uma mensagem pode usar a Rota 1 ou a Rota 2.

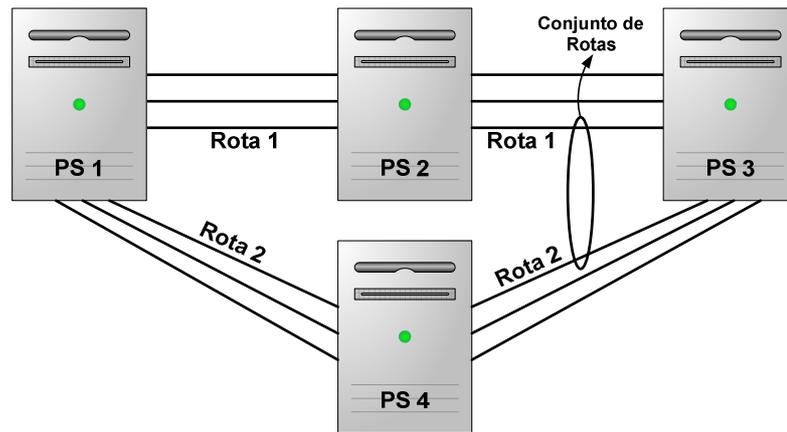


Figura 3.7 – Rotas de Sinalização

3.2.3 Arquitetura da SS7

Para lidar com a rápida evolução das redes de telecomunicações, o sistema de sinalização deve ser o mais flexível possível, sendo capaz de tratar não somente as aplicações existentes, como as futuras. Para que essa característica fosse alcançada na SS7, a mesma foi desenvolvida usando uma estrutura modular, organizada em camadas. A camada responsável pelo transporte das mensagens é chamada MTP (*Message Transfer Part*) e a camada que a utiliza é chamada UP (*User Part*). A SS7 foi desenvolvida de forma que todas as aplicações presentes na camada UP usem as funções da MTP para a entrega e recebimento das mensagens. A Figura 3.8 representa a conexão entre dois pontos de sinalização representada nas camadas da SS7.

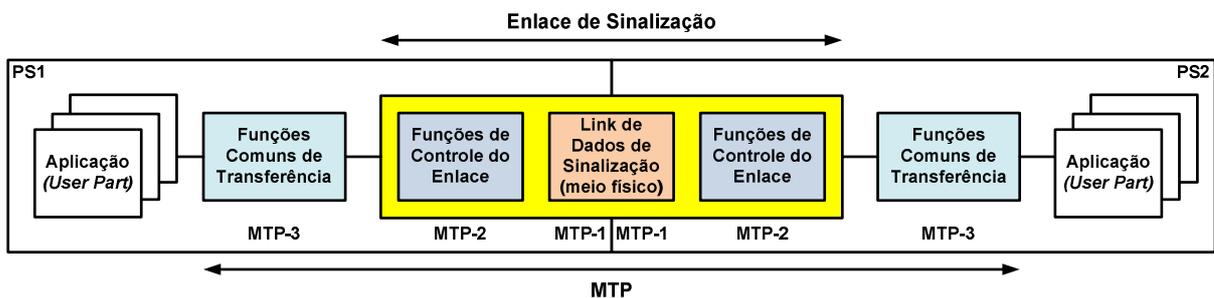


Figura 3.8 – Organização em camadas da SS7

Inicialmente, a SS7 foi desenvolvida para atender aos requisitos de controle de chamadas em centrais telefônicas, sendo dividida em quatro níveis, sendo os níveis um a três para a estrutura de transporte e o nível quatro para as aplicações, como mostra a Figura 3.9.

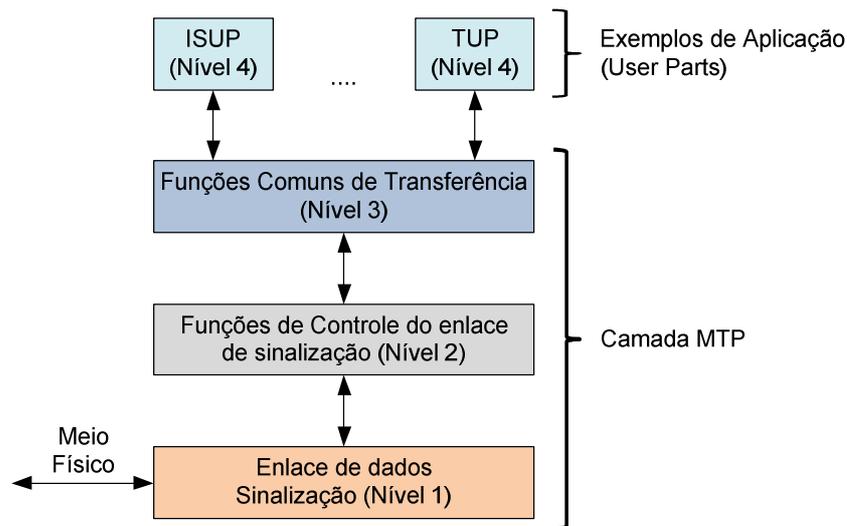


Figura 3.9 – Níveis funcionais da SS7

3.2.3.1 A camada MTP

A MTP, corresponde à camada mais baixa da SS7 e sua função é prover um mecanismo para transporte das mensagens na rede, desde a sua origem até o seu destino de uma forma confiável, garantindo a entrega sem perdas e na sequência correta. Cada conexão de um PS com outros PSs é implementada por um Enlace de Sinalização e cada Enlace de Sinalização possui uma MTP em cada um dos seus extremos. A camada MTP é dividida em três níveis:

- **MTP-1 ou nível 1:** Corresponde à parte física e elétrica do enlace de sinalização usado entre dois pontos de sinalização. É especificado na recomendação Q.702;
- **MTP-2 ou nível 2:** Possui funções para garantir a transmissão das mensagens na rede sem erros dentro de um determinado enlace de sinalização. Dentre essas funções inclui-se a detecção e correção de erro para garantir a entrega de mensagens na sequência correta, entre outros. É especificado em recomendação Q.703 (ITU-T, 1993c);
- **MTP-3 ou nível 3:** Possui duas categorias de funções. A primeira possui funções para tratamento de mensagens de sinalização, que basicamente são responsáveis por entregar a mensagem para o enlace de sinalização correto, no caso de uma transmissão, ou passar para o nível 4 (*User Part*), no caso de uma recepção. A segunda categoria possui funções para gerenciamento da rede de sinalização,

controlando o roteamento das mensagens, e em caso de falha na rede, realiza as ações necessárias para manter e/ou restaurar a capacidade de transferência de mensagens.

De uma forma geral, o três níveis da camada MTP em conjunto, são responsáveis pela transmissão e recepção de todas as mensagens que trafegam na rede de sinalização. Pode-se assumir que a camada MTP é a conexão entre as aplicações dos diversos PSs, ficando completamente transparente para a aplicação a forma como esta comunicação é realizada.

3.2.3.2 A Camada de Aplicação

Também chamada de Nível 4, está imediatamente acima da camada MTP. O Nível 4 é formado por diversas aplicações, de acordo com a funcionalidade do PS. Todas as aplicações usam a camada MTP para transmitir e receber suas mensagens na sequência correta e teoricamente livre de falhas.

Cada aplicação possui um conjunto específico de mensagens de sinalização, todas padronizadas em uma recomendação da ITU-T. Assim, caso seja criada uma nova aplicação, a ITU-T deverá criar uma nova recomendação padronizando todas as suas mensagens de tal forma que equipamentos de diferentes fabricantes possam comunicar-se sem problemas de compatibilidade. Um exemplo de aplicação no nível 4 é a TUP (*Telephone User Part*), que é usada para sinalização de telefonia e está definida nas recomendações Q.720 a Q.729.

3.3 O NÍVEL 2 DA CAMADA MTP DA SS7

Conforme visto anteriormente, a camada MTP da SS7 é dividida em três níveis. Este capítulo trata especificamente do Nível 2, ou simplesmente MTP-2.

A principal função da MTP-2 é garantir a transferência das mensagens de sinalização de forma confiável entre dois PSs diretamente conectados. É a MTP-2, através de funções específicas, que garante que uma mensagem originada em um determinado PS alcançará seu destino livre de erros, na sequência correta e sem perda de dados.

Esta seção é baseada principalmente na recomendação Q.703 da ITU-T (*International Telecommunication Union – Telecommunication Standardization Sector*) (ITU-T, 1993c) que descreve as funções do Enlace de Sinalização.

A recomendação Q.703 especifica todas as funções e os procedimentos da MTP-2, mas não fornece detalhes para a sua implementação. O principal objetivo deste trabalho, é propor uma implementação das funções da MTP-2 usando *hardware* de lógica programável.

Os PSs comunicam-se através de mensagens de sinalização, ou simplesmente SU (*Signal Unit*) de tamanhos variados. Além dos dados da aplicação, as SUs possuem informações adicionais para a correta operação do enlace de sinalização, como por exemplo, identificação do PS de origem e de destino para que a mensagem seja roteada corretamente.

3.3.1 Funções da MTP-2

Para garantir que as SUs sejam entregues em seu destino na sequência correta e livres de erros e perdas, várias funções são desempenhadas pela MTP-2:

- ✓ Delimitação da SU;
- ✓ Alinhamento de SU;
- ✓ Detecção de Erro;
- ✓ Correção de Erro;
- ✓ Alinhamento Inicial;
- ✓ Supervisão de Erro no Enlace de Sinalização;
- ✓ Controle de Fluxo.

3.3.1.1 Delimitação da SU

Como as SUs possuem tamanhos diferentes, foi especificado um padrão único de 8 bits (sequência 01111110) que serve como um marcador (*flag*) de início e fim de uma SU. Uma técnica conhecida como preenchimento de bit (*bit stuffing*) é aplicada na SU para evitar que este padrão seja confundido com algum dado da SU (que seria entendido como um falso marcador de fim de mensagem).

3.3.1.2 Alinhamento de SU

Existem duas formas de acontecer uma perda de alinhamento de SU entre dois PSs: quando uma sequência de mais de seis bits iguais a 1 é recebida ou quando a SU possui um tamanho acima do máximo especificado, que é de 272 octetos. Em ambos os casos, o enlace de sinalização passa a operar em um modo chamado Contagem de Octetos. A perda de alinhamento de SU é um dos parâmetros monitorados pela função de Supervisão de Erro no Enlace de Sinalização da MTP-2, e dependendo da taxa com que esses erros acontecem, o enlace pode ser colocado na condição “fora de serviço”, impedindo o tráfego de novas SUs.

3.3.1.3 Detecção de Erro

Ao final de cada SU (antes do marcador de fechamento) são inseridos dois octetos (16 bits) para possibilitar a detecção de erro no PS de destino usando o recurso de CRC (*Cyclic Redundance Code*). Os dois octetos de CRC são gerados através de um algoritmo específico que atua em todos os bits da SU, com exceção dos marcadores de inícios e fim de mensagem. No PS de destino o algoritmo é aplicado de forma inversa na SU e o resultado é comparado com os dois octetos transmitidos pelo PS de origem. Caso haja inconsistência nos resultados, a SU é descartada e o erro é contabilizado para medições da qualidade do enlace de sinalização, que dependendo da taxa, pode ser colocado na condição “fora de serviço”.

3.3.1.4 Correção de Erro

A função de Detecção de Erro da MTP-2 indica se uma SU recebida está correta ou não. Há também uma função para Correção de Erros. Quando uma SU é recebida com erro, a mesma é descartada e o PS de destino deve informar ao de origem sobre a ocorrência do erro para que a SU seja transmitida novamente. Na SS7 existem dois métodos usados para permitir a correção de uma SU com erro:

- ✓ **Básico:** aplicado em enlaces de sinalização que usam meios de transmissão terrestre não-intercontinental e para enlaces de sinalização intercontinentais cujo tempo de propagação seja inferior a 15ms;
- ✓ **Retransmissão Cíclica Preventiva:** usado para enlaces de sinalização intercontinentais cujo tempo de propagação é igual ou superior a 15ms e para todos os enlaces de sinalização estabelecidos através de satélites.

3.3.1.5 Alinhamento Inicial

O procedimento de alinhamento inicial é executado basicamente em dois casos: quando o equipamento é energizado (ligado) ou após uma falha no enlace, para retirá-lo da condição “fora de serviço” e retornar para a condição “em serviço”.

3.3.1.6 Supervisão de Erro no Enlace de Sinalização

A MTP-2 possui duas funções usadas para monitorar a taxa de erros em um enlace de sinalização. A primeira delas atua quando o enlace encontra-se na condição “em serviço” e chama-se Monitor de Taxa de Erro de SU, ou SUERM (*Signal Unit Error Rate Monitor*). Trata-se de um contador baseado no princípio do “balde furado”, onde cada SU com erro incrementa o contador em uma unidade e a cada 256 SUs recebidas decrementa o contador em uma unidade. O SUERM é um dos critérios usados para levar o enlace de sinalização para o estado “fora de serviço”. A segunda função de monitoramento de erro é usada durante a fase de alinhamento inicial e chama-se Monitor de Taxa de Erro de Alinhamento, ou AERM (*Alignment Error Rate Monitor*). Uma quantidade excessiva de erros durante o alinhamento inicial pode levar o enlace de sinalização para o estado “fora de serviço” ou reiniciar o procedimento de alinhamento.

3.3.1.7 Controle de Fluxo

Em determinadas condições, o PS de destino pode não conseguir tratar todas as SUs recebidas na mesma velocidade com que são transmitidas, causando um congestionamento de SUs. Nesse caso, a MTP-2 possui a função de controle de fluxo para que o PS de destino, por meio de SUs apropriadas, comunique ao PS de origem sobre o congestionamento, solicitando a interrupção na transmissão de novas SUs. Essas SUs enviadas pelo PS de destino são conhecidas como Mensagens de Sinalização de Estado do Enlace, ou LSSU (*Link State Signal Unit*). Enquanto a condição de congestionamento for mantida, o PS de origem será notificado periodicamente pelo PS de destino através de LSSUs. Se esta condição for mantida por um período de tempo muito longo, o PS de origem indicará falha no enlace de sinalização e poderá levá-lo à condição de “fora de serviço”.

3.3.2 Mensagem de Sinalização

Uma SU é formada por um conjunto de bytes (ou octetos) cujo conteúdo depende da aplicação (*User Part*). Existem três estruturas diferentes de SUs, que variam em tamanho e são usadas em ocasiões diferentes. São elas: a MSU (*Message Signal Unit*), a LSSU (*Link State Signal Unit*) e a FISU (*Fill-In Signal Unit*).

A MSU é usada para a comunicação entre as aplicações dos diversos PSs da rede SS7. O tamanho em octetos de uma MSU varia de acordo com as aplicações, e por isso existe um campo na estrutura da MSU para indicar o tamanho da mesma.

A LSSU é usada para informar o estado do enlace entre dois PSs. Diferente da MSU que pode passar por vários PSs desde a origem até o destino, a LSSU sempre é usada somente no enlace que conecta diretamente dois PSs. Conforme descrito anteriormente, em caso de congestionamento no enlace, uma LSSU é enviada continuamente ao PS de origem indicando a estado de ocupado, evitando que novas mensagens sejam enviadas até que o congestionamento no PS de destino seja resolvido. Durante a fase de alinhamento inicial, as LSSUs também são usadas para garantir a inicialização correta do enlace de sinalização.

Quando as aplicações não possuem dados para serem transmitidos na rede, mensagens sem conteúdo são enviadas continuamente para garantir atividade no enlace de sinalização. Essas mensagens “vazias” são chamadas FISU e são usadas para monitorar a integridade do enlace de sinalização.

A estrutura de cada tipo de SU é mostrada na Figura 3.10. Pode-se observar que os três tipos de SU possuem os campos F, BSN/BIB, FSB/FIB e LI em comum. Através do campo LI (*Length Indicator*) é possível identificar se a SU é uma MSU, uma LSSU ou uma FISU.

O campo F corresponde ao marcador de abertura e fechamento da SU. Como foi mencionado anteriormente, este campo é uma sequência binária fixa de 8 bits, igual a 01111110. Tanto a MSU, LSSU e FISU sempre iniciam e terminam com um marcador F.

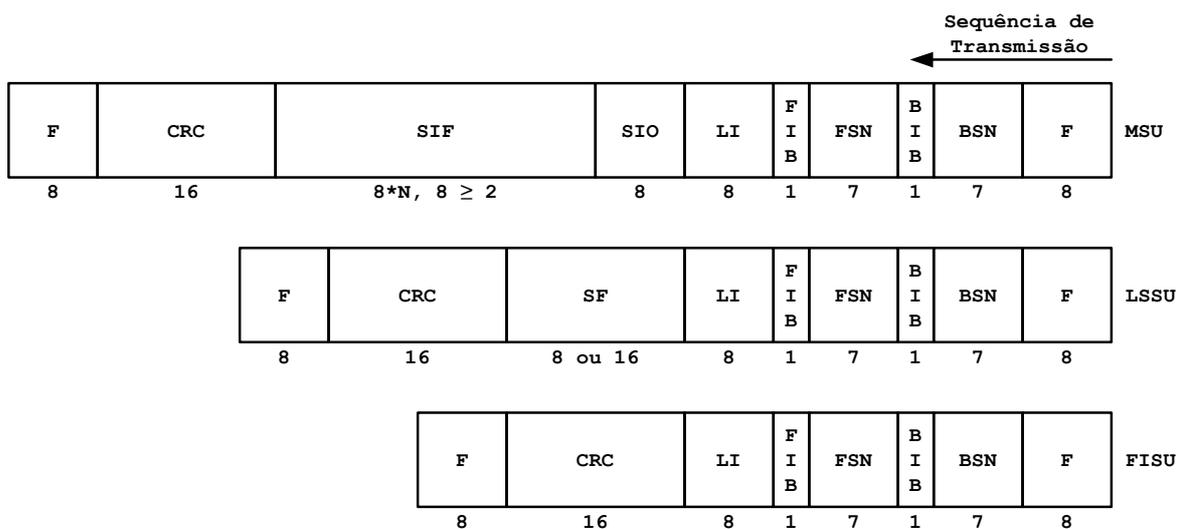


Figura 3.10 – Estrutura de cada tipo de SU

Os campos BSN/BIB e FSN/FIB foram criados para permitir que as SUs sejam transmitidas e recebidas na sequência correta. O campo FSN (*Forward Sequence Number*) possui 7 bits e corresponde ao número da SU que está sendo transmitida. Esse número é fornecido por um contador que conta ciclicamente de 0 a 127 e é utilizado pelo algoritmo de correção de erro no caso de perda ou deterioração de mensagem(ns) recebida(s). O campo BSN (*Backward Sequence Number*) também possui 7 bits e corresponde ao número da última mensagem reconhecida pelo PS. Os campos FIB e BIB possuem apenas um bit, e em conjunto com os campos FSN e BSN permitem que o controle de sequenciamento seja realizado corretamente através de pedidos de retransmissão. Este processo será detalhado mais adiante.

O campo LI (*Length Indicator*) serve para indicar o número de octetos entre o LI e o CRC. É através deste campo que se determina o tipo da SU, sendo uma FISU quando LI=0, uma LSSU quando LI = 1 ou 2 e uma MSU quando LI > 2. Apesar desse campo possuir 8 bits, apenas os 6 menos significativos são usados. Quando o conteúdo binário dos 6 bits do campo LI está entre 3 e 62, esse conteúdo indica o número de octetos depois do campo LI e antes do CRC. Quando o conteúdo binário é igual a 63, significa que o número de octetos é maior que 62. Isso significa que quando o tamanho do campo SIF está entre 63 e 273 o tamanho exato não faz parte da informação transmitida pelo Nível 2. Ele somente indica que o tamanho é maior que 62 e, por construção deve ser menor ou igual a 273.

O campo SIO (*Service Information Octet*) existe somente nas MSUs e é usado pelo nível 3 da camada MTP para indicar o tipo de protocolo usado na camada de aplicação (TUP, ISUP, TCAP, etc).

O campo SIF (*Signaling Information Field*) pode ter entre 2 e 272 bytes e é responsável por levar a informação propriamente dita da MSU, assim como informações de roteamento, que serão usadas pelo nível 3 da camada MTP para a distribuição correta das mensagens nos diversos enlaces de sinalização ou para recompor mensagens fragmentadas.

O campo SF (*Status Field*) é usado somente nas LSSUs e serve para indicar o estado do enlace de sinalização em que a LSSUs está sendo transmitida/recebida.

O campo CRC possui dois octetos e serve para detectar erros de transmissão em uma SU. Estes dois bytes são resultado de uma manipulação matemática com todos os bits da mensagem, com exceção dos marcadores, como mostra a Figura 3.11.

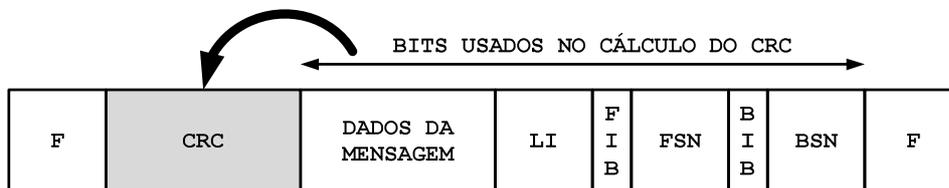


Figura 3.11 – Inserção do CRC na SU

No final da mensagem a ser transmitida, o PS insere os bits do campo CRC e termina a mensagem com o marcador, para então transmitir a mesma. Quando a mensagem chega ao PS de destino, o CRC é novamente calculado sobre os bits da mensagem recebida, incluindo os bits do CRC transmitido, usando o mesmo algoritmo do PS de origem. Essa aplicação do algoritmo na transmissão e na recepção é matematicamente calculado de modo que, na ausência de erro de transmissão os 16 bits de CRC obtidos na recepção correspondem a uma palavra fixa, ou seja, que sempre é a mesma. O valor do CRC calculado na recepção é comparado com essa palavra padrão, e caso sejam diferentes, a SU é descartada. Todo este procedimento está ilustrado na Figura 3.12.

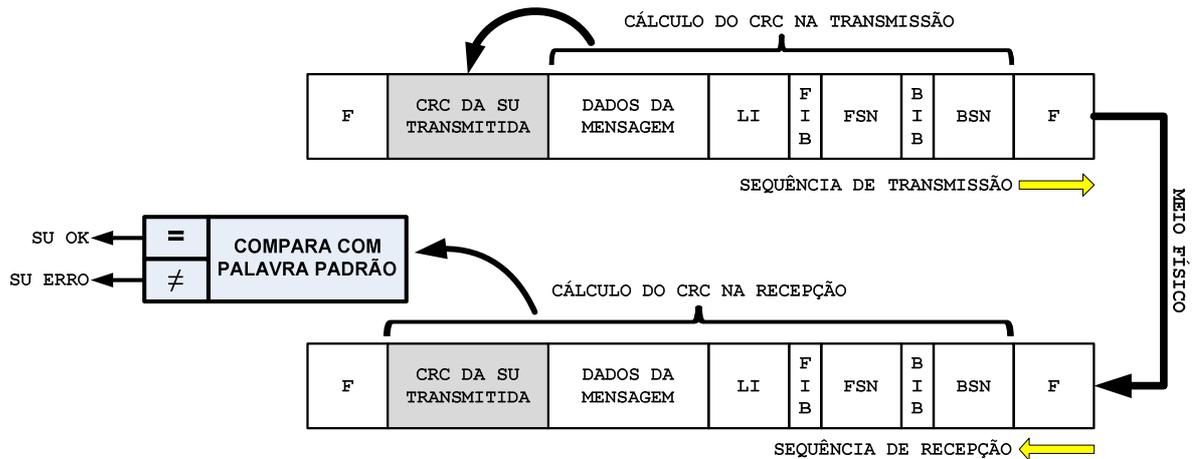


Figura 3.12 – Validação do CRC da SU

3.4 TECNOLOGIAS PARA IMPLEMENTAÇÃO DE CIRCUITOS DIGITAIS

É muito comum o uso de circuitos digitais nos mais diversos equipamentos da atualidade, que vão de brinquedos a equipamentos médicos. As variáveis analógicas são muitas vezes convertidas em sinais digitais através de transdutores adequados, para então serem tratadas por um circuito digital. Os sinais digitais podem ser mais facilmente manipulados e armazenados quando comparados com sinais analógicos. Assim, a quantidade de circuitos digitais presentes nos equipamentos é cada vez maior, exigindo que essa modalidade da eletrônica evoluísse rapidamente. Existe uma grande variedade de tecnologias disponíveis para implementação de circuitos digitais, como mostra a Figura 3.13.

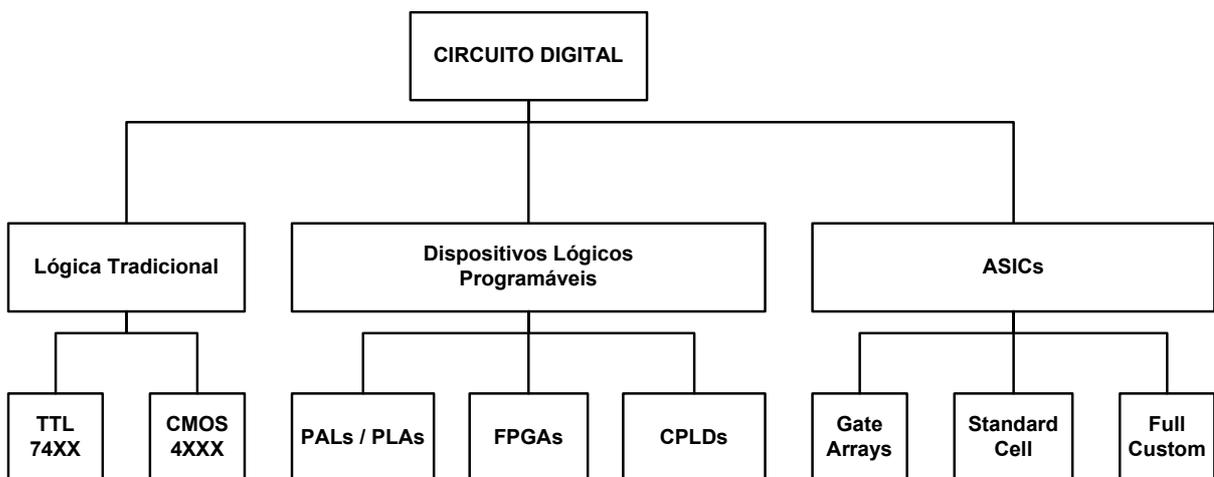


Figura 3.13 – Tecnologias para Circuitos Digitais

Em circuitos digitais de equipamentos mais antigos é muito comum encontrar-se uma grande quantidade de circuitos integrados da série 74, que usa tecnologia TTL (*Transistor-*

Transistor Logic) ou da série 40, que usa tecnologia CMOS (*Complementary Metal-Oxide Semiconductor*). Estes circuitos integrados possuem uma função definida pelo fabricante, e, pelo fato de cada circuito integrado possuir apenas poucas portas lógicas internamente, muitas vezes faz-se necessário conectar vários deles para se realizar uma simples função. Circuitos integrados mais avançados como ASICs (*Application Specific Integrated Circuit*), CPLDs (*Complex Programmable Logic Devices*) e FPGAs (*Field Programmable Gate Arrays*) possuem a característica de permitir que sua funcionalidade seja determinada pelo usuário final (HAMBLEN, 2008).

A decisão de qual tecnologia usar no desenvolvimento de produto depende de vários fatores, mas os principais são: volume de produção, velocidade do circuito ou frequência de operação, custo, tempo de desenvolvimento, área ocupada pelo circuito, consumo de energia, entre outros. A seguir será feita uma breve descrição de algumas dessas tecnologias.

3.4.1 ASIC

Os ASICs são circuitos integrados projetados especificamente para o produto onde serão usados. Os principais tipos de ASICs são: *full-custom*, *standard-cell* e *gate-array*. Cada um oferece um nível de customização diferente. As etapas de fabricação de um ASIC variam de acordo com o tipo escolhido, podendo exigir somente camadas de metalização, no caso do ASIC *gate-array* ou um conjunto maior de camadas, no caso do ASIC *full-custom*. Este processo é caro, complexo, e só pode ser feito em fábricas específicas, chamadas *foundry*.

Pelo fato de um ASIC necessitar de processos industriais de fabricação, deve-se considerar o tempo e custos adicionais de desenvolvimento neste processo, que são bastante relevantes. Normalmente essa tecnologia é usada em produtos que possuem uma vida muito longa no mercado e um alto volume de produção, e neste caso, o custo por unidade produzida é menor que um CPLD ou FPGA.

3.4.1.1 Full-custom

A tecnologia *full-custom* permite que o circuito seja desenvolvido a nível de transistor e, conseqüentemente, exige uma equipe com um alto nível de conhecimento da tecnologia e normalmente requer um grande esforço durante a fase de desenvolvimento e teste.

Os ASICs *full-custom* permitem o maior nível de customização possível pelo fato de permitir o controle total do circuito, inclusive sobre a disposição dos transistores de forma a atender requisitos de área e desempenho. Por causa da grande complexidade no desenvolvimento de ASICs *full-custom*, essa tecnologia normalmente é usada para projetos de lógicas básicas, para então serem integradas em um sistema maior, como por exemplo, memórias e somadores, onde uma vez projetado e validado um único bit, o mesmo pode ser replicado de acordo com a quantidade desejada (CHU, 2006).

Exemplos de circuitos integrados com essa tecnologia são as memórias RAM e microprocessadores usados na produção de computadores pessoais (HAMBLEN, 2008).

3.4.1.2 *Standard-cell*

No caso de ASICs do tipo *standard-cells*, o usuário seleciona os blocos funcionais (ou células) que deseja em seu circuito integrado, como memórias, microcontroladores, Unidade Lógica e Aritmética (ULA). Esses blocos funcionais são componentes de uma biblioteca do fabricante do ASIC. Quando comparada à tecnologia *full-custom*, a complexidade e o tempo de desenvolvimento são reduzidos significativamente uma vez que o projeto é feito em nível de blocos funcionais e não mais em nível de transistores.

Quando todas as células estiverem posicionadas e devidamente conectadas, segue-se com o processo de fabricação, que é semelhante ao usado na tecnologia *full-custom*, sendo necessária a elaboração das máscaras de cada camada.

3.4.1.3 *Gate-array*

Em um ASIC do tipo *gate-array* o circuito integrado já possui um conjunto de células idênticas, distribuídas em linhas e colunas na forma de uma matriz. Ao contrário da tecnologia *standard-cell*, todas as células no *gate-array* possuem a mesma funcionalidade, que pode ser desde uma simples porta lógica a um bloco lógico mais avançado, chamado *macro cell*. Como todas as células são idênticas e já estão posicionadas, as mesmas podem ser pré-fabricadas. A funcionalidade final do circuito é alcançada de acordo com as conexões realizadas entre as células. Comparada à tecnologia *standard-cell*, o processo de fabricação de um ASIC do tipo *gate-array* é mais simples, sendo necessária somente a fabricação das camadas de metalização para a conexão das células (CHU, 2006).

3.4.2 PLA e PAL

Esses DLPs (Dispositivos Lógicos Programáveis) têm sido usados por vários anos para substituir circuitos integrados que utilizam a tecnologia TTL (série 74) ou CMOS (série 40). A vantagem desses DLPs em relação esses circuitos integrados é que a sua funcionalidade pode ser programada pelo próprio usuário, além de permitir uma redução significativa da área ocupada pelo circuito.

A PAL (*Programmable Array Logic*) consiste em um conjunto programável de portas AND, cujas entradas podem ser ligadas diretamente aos terminais do circuito integrado ou através de uma porta lógica inversora. As saídas das portas AND são conectadas a um conjunto fixo de portas OR, cujas saídas são ligadas diretamente aos terminais do circuito integrado. A Figura 3.14 mostra o esquema simplificado de uma PAL. A programação da PAL consiste em selecionar quais sinais serão conectados às entradas das diversas portas AND, de forma que o circuito responda à lógica esperada.

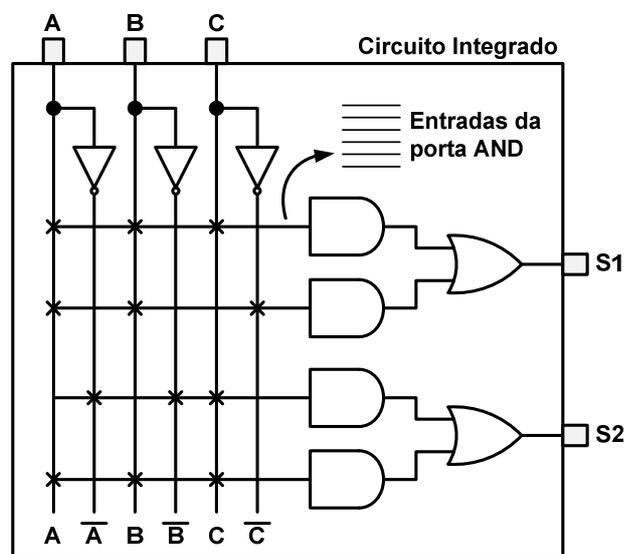


Figura 3.14 – Esquema simplificado de uma PAL

A PLA (*Programmable Logic Array*) é muito semelhante à PAL no que diz respeito às entradas. A diferença entre essas duas tecnologias é que na PLA o conjunto de portas OR de saída também é programável, aumentando consideravelmente a flexibilidade na implementação de funções lógicas. A Figura 3.15 mostra um exemplo de PLA.

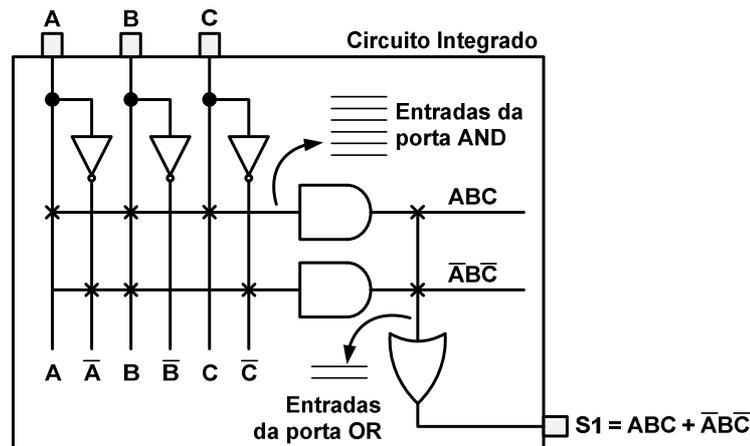


Figura 3.15 – Esquema simplificado de uma PLA

Devido a sua estrutura, as PLAs são adequadas para a implementação de funções lógicas na forma de soma de produtos e apresentam-se muito versáteis, pois tanto os termos AND como os termos OR podem ter muitas entradas. As desvantagens dessa tecnologia são o alto custo de fabricação e o baixo desempenho em termos de velocidade, devido aos dois níveis de lógica configurável, que introduzem atrasos significativos de propagação dos sinais elétricos (TEIXEIRA, 2002). As PALs, PLAs e similares são agrupados em uma única categoria denominada SPLDs (*Simple Programmable Logic Devices*).

3.4.3 CPLD

A dificuldade em aumentar a capacidade da arquitetura de um DLP é que a estrutura de planos lógicos programáveis aumenta muito rapidamente com o aumento do número de entradas. Uma maneira viável de se produzir dispositivos com maior capacidade é integrando múltiplos SPLDs em um único *chip* e prover interconexões programáveis para conectá-los internamente (TEIXEIRA, 2002), como mostra a Figura 3.16.

Os DLPs que utilizam a estrutura indicada na Figura 3.16, apresentando um número elevado de entradas e saídas, juntamente com vários planos de lógica configurável, são chamados CPLDs. Eles são compostos por um conjunto de portas AND e de portas OR programáveis. Na maioria dos casos, possuem memórias embutidas chamadas EAB (*Embedded Array Block*) para implementar funções lógicas complexas (LIMA, 1999).

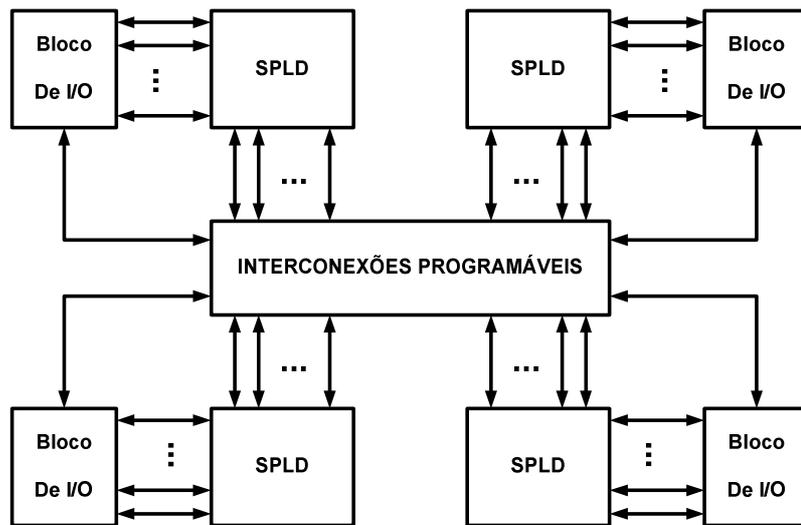


Figura 3.16 – Vários SPLDs integrados em um único circuito integrado

Cada fabricante possui sua arquitetura de CPLD, mas no geral, possuem basicamente os mesmos blocos mostrados na Figura 3.16. A forma de programação dos CPLDs depende da tecnologia utilizada na fabricação dos mesmos, podendo ser células de EPROM, EEPROM ou *Flash*. A interconexão entre os blocos funcionais do CPLD é feita através de uma grande matriz de chaves, permitindo a conexão de todos os sinais do dispositivo.

3.4.4 FPGA

Os DLPs descritos anteriormente permitem a implementação de uma grande variedade de circuitos lógicos. Porém, com exceção dos CPLDs, estes componentes possuem pequena capacidade lógica, sendo indicados para aplicações relativamente pequenas. Até mesmo para os CPLDs, apenas circuitos moderadamente grandes podem ser acomodados em um único circuito integrado (ZEIDMAN, 2002).

Com o objetivo de aumentar a quantidade de portas lógicas internas ao circuito integrado e possibilitar a implementação de circuitos digitais mais complexos, como microprocessadores, controladores de vídeo, entre outros, a empresa Xilinx Inc. no ano de 1985, lançou uma nova tecnologia de DLP, o FPGA (*Field Programmable Gate Array*).

Os FPGAs são assim chamados por possuírem uma estrutura muito similar a dos ASICs do tipo *gate-array*. Esta similaridade faz com que os FPGAs sejam muito bons para protótipos de ASICs, ou até mesmo para substituir um ASIC onde seria eventualmente usado. Por exemplo, um FPGA pode ser usado em um projeto que precisa chegar ao mercado

rapidamente, mesmo que a um custo mais alto. Posteriormente, um ASIC pode ser usado para substituir o FPGA caso o volume de produção justifique a troca de tecnologia (ZEIDMAN, 2002). Com o grande aumento no uso de FPGAs, o custo por unidade tem reduzido significativamente, tornando-os economicamente viáveis quando comparado aos ASICs.

3.4.4.1 Arquitetura de um FPGA

Cada fabricante de FPGA possui a sua própria arquitetura, mas em termos gerais esta arquitetura é uma variação da arquitetura mostrada na Figura 3.17. A arquitetura consiste de blocos lógicos configuráveis, blocos de I/O configuráveis e interconexões programáveis. Também há circuitos de relógio com a finalidade de distribuir este sinal para cada bloco lógico. Recursos lógicos adicionais como ULAs, memórias e decodificadores também podem estar disponíveis (ZEIDMAN, 2002).

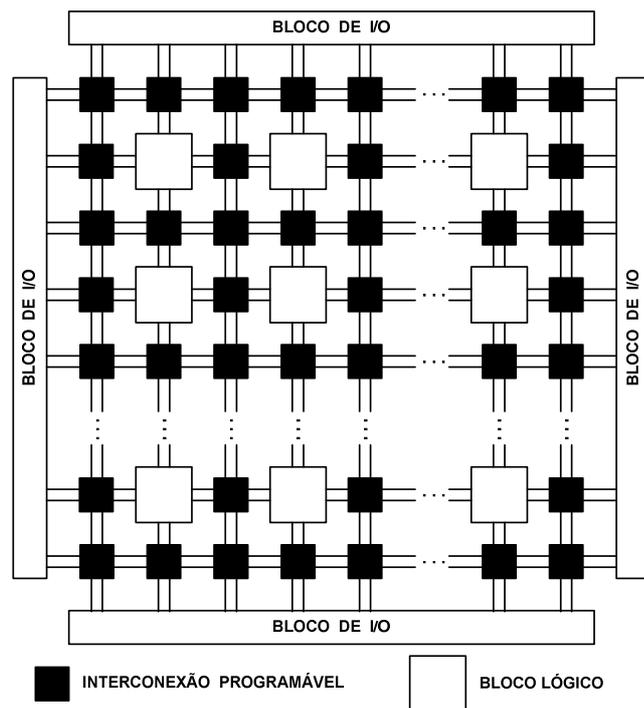


Figura 3.17 – Estrutura básica de um FPGA

3.4.4.2 Blocos Lógicos Configuráveis

O Bloco Lógico Configurável ou CLB (*Configurable Logic Block*) é responsável pela implementação da lógica combinacional e sequencial de um circuito no FPGA (LIMA, 1999). Como será visto mais adiante, existem os blocos de granulosidade fina e os blocos de granulosidade grossa. Em um bloco de granulosidade grossa, os CLBs possuem lógica

suficiente para criar máquinas de estado. Em um bloco de granulosidade fina, os CLBs contêm apenas elementos básicos de lógica. O diagrama mostrado na Figura 3.18 pode ser considerado como um CLB de granulosidade grossa.

Este CLB contém uma memória RAM que pode ser usada como uma simples memória ou como uma Tabela de Respostas (LUT – *Look-up Table*). Também contém *flip-flops* para circuitos com lógica sequencial, e multiplexadores para que seja possível fazer desvios da lógica dentro do bloco e para recursos externos. Neste caso, os multiplexadores também permitem a seleção de polaridade e seleção das entradas *reset* e *clear*.

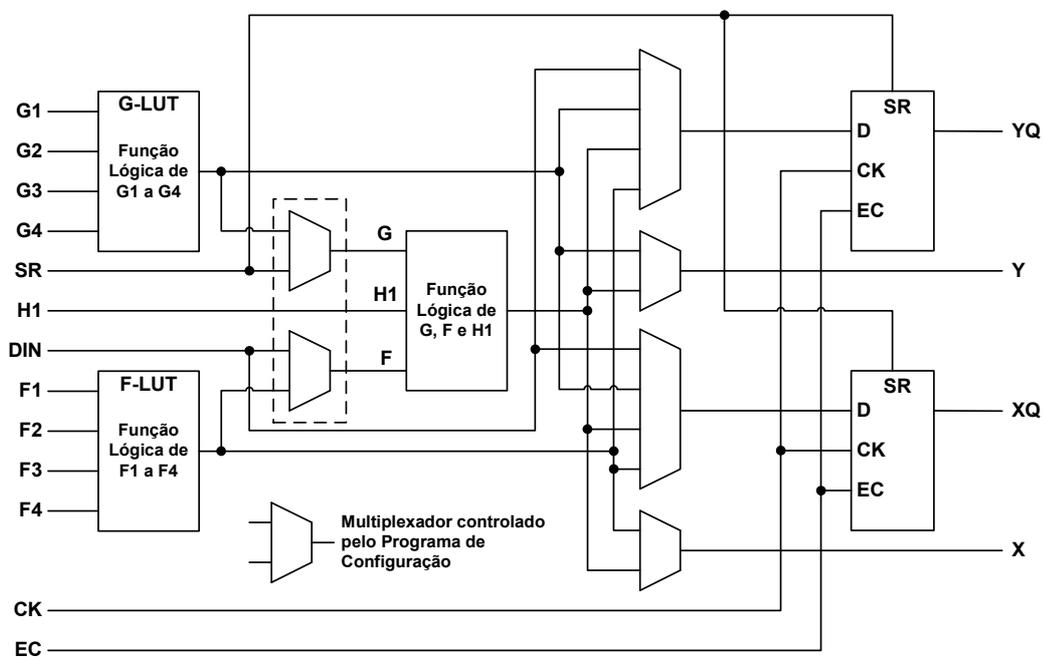


Figura 3.18 – CLB de granulosidade grossa

Os fabricantes de FPGA utilizam diferentes tipos de CLBs. O mais comum possui uma LUT como unidade básica. Este tipo de bloco lógico contém células de armazenamento (memórias) que são utilizadas para implementar pequenas funções lógicas, na forma de tabela da verdade. As entradas da LUT servem como os endereços para a memória e o dado armazenado naquele endereço é a saída da LUT, como mostra a Figura 3.19.

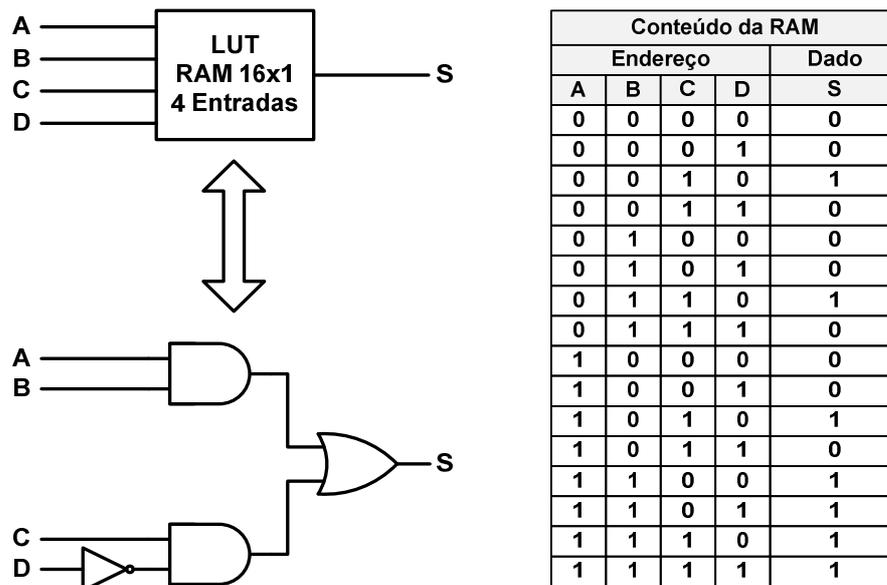


Figura 3.19 – Exemplo de uma LUT

3.4.4.3 Bloco de I/O Configurável

É usado para trazer sinais para dentro e para fora do circuito integrado e consiste de um buffer de entrada, um buffer de saída que permite modo de alta impedância (*tri-state*) e coletor aberto. Tipicamente há resistores de *pull-up* nas saídas e às vezes resistores de *pull-down*. A saída pode ser usualmente programada para ser ativa em nível alto ou baixo, assim como a velocidade de transição (*slew rate*) pode ser programada para ter um rápido ou lento tempo de subida e descida. Além disso, geralmente existe um *flip-flop* na saída de tal forma que sinais de relógio (*clock*) podem sair diretamente nos pinos sem atraso significativo (ZEIDMAN, 2002).

3.4.4.4 Interconexões Programáveis

Os FPGAs possuem comutadores programáveis que em conjunto com os segmentos de trilha, permitem realizar as conexões entre os CLBs dentro do circuito integrado. Os segmentos de trilha são camadas de metal introduzidas no processo de fabricação do circuito integrado. A Figura 3.20 mostra a estrutura básica de roteamento em um FPGA, mas alguns conceitos são necessários para um melhor entendimento:

- **Pinos:** correspondem às entradas e saídas dos CLBs;
- **Bloco de Conexão:** tem a função de conectar os pinos dos CLBs às trilhas, possibilitando a conexão entre CLBs;

- **Bloco de Comutação:** tem a função de fazer a conexão elétrica entre dois segmentos de trilha;
- **Segmento de Trilha:** São as trilhas entre os blocos de conexão e os blocos de comutação e as trilhas entre os blocos de comutação;
- **Trilha:** corresponde a um segmento de metal usado para transportar os sinais entre os diversos CLBs dentro do circuito integrado;
- **Canal de Roteamento:** É por onde passam as trilhas. Um canal contém um conjunto de trilhas paralelas.

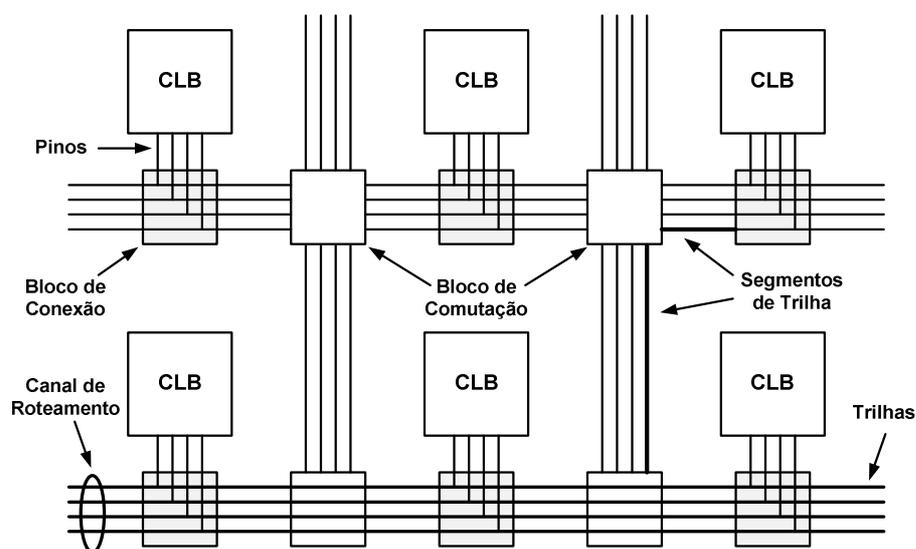


Figura 3.20 – Estrutura de roteamento em um FPGA

Cada bloco lógico possui um conjunto de entradas e saídas que podem ou não ser conectadas a outros CLBs, ou até mesmo ao próprio CLB. O bloco de conexão é quem permite a conectividade das entradas e saídas dos CLBs com os segmentos de trilhas nos canais de roteamento. Já o bloco de comutação permite a conexão entre as trilhas horizontais e verticais, possibilitando a conexão entre os CLBs. É comum encontrar estruturas como mostra a Figura 3.20, que possuem o bloco de conexão e o bloco de comutação distintos, assim como é muito comum também estruturas em que os blocos de comutação e conexão estão combinados em um só bloco.

Uma das tarefas mais difíceis para os fabricantes é conseguir uma alta densidade (maior número de portas no circuito integrado) e ao mesmo tempo conseguir uma arquitetura que permita um roteamento completo, ou seja, que cada ponto do circuito integrado possa conectar-se a qualquer outro ponto, não importando a localização dos mesmos. Para permitir o

roteamento completo, deve-se inserir uma grande quantidade de blocos de conexão e comutação. Porém, esses blocos ocupam uma área relativamente grande do circuito integrado que compromete a densidade (TEIXEIRA, 2002).

3.4.5 Considerações de projeto

Quando um projeto for desenvolvido com CPLD ou FPGA, alguns tópicos devem ser levados em conta (ZEIDMAN, 2002):

- Tecnologia de programação
 - Esta característica determina que equipamentos serão necessários para a programação dos circuitos integrados e se eles podem ou não ser gravados mais de uma vez;
- Capacidade em termos de blocos funcionais
 - Quantos blocos funcionais existem no dispositivo?
 - Quantos produtos e soma de termos podem ser usados?
 - Qual é o mínimo e máximo tempo de propagação através da lógica?
 - Quais recursos adicionais existem nos blocos, além de simplesmente AND e OR?
 - Que controles de registros estão disponíveis, tais como *clock*, *reset*, *preset*, e *enable*? Quantas são as entradas locais dos blocos funcionais e quantas são as entradas globais do *chip*?
 - Qual o pior caso de “escorregamento” (*clock skew*) para o sinal de *clock* no *chip*? Isso ajuda a determinar a máxima frequência permitida;
- Capacidade de I/O do chip
 - Quantos pinos de I/O são independentes para qualquer função e quantos são dedicados para entrada de *clock*, *master reset*, etc.?
 - Quais são as características dos pinos no que diz respeito ao nível de tensão e corrente?
 - Que tipo de lógica está incluída no bloco de I/O e que pode ser usada para aumentar a flexibilidade do projeto?
- Escalabilidade
 - Existem na mesma família, componentes compatíveis na pinagem e que possuem diferentes quantidades de blocos funcionais? Isto é importante em casos como o deste trabalho onde se espera que uma mesma solução possa ser replicada em uma gama

grande de tamanhos, dependendo da aplicação, e otimizada em custo para cada caso ou sub-conjunto de casos.

3.4.5.1 Sinais de *Clock*

Em FPGAs e CPLDs, os sinais de *clock* devem preferencialmente ser transportados em trilhas especiais que apresentam baixo “escorregamento” (*low-skew*), fazendo com que o sinal chegue em todos os flip-flops do *chip* praticamente ao mesmo tempo. Essas linhas são ligadas a pinos de entrada dedicados, chamados *Global Clocks*.

Caso os sinais de relógio do circuito sejam ligados em pinos de I/O comum, seu roteamento será feito internamente no circuito integrado da mesma forma que qualquer outro sinal. Desta forma é possível que os tempos de *hold* e *setup* exigidos pelos flip-flops sejam violados, causando um efeito conhecido como “meta-estabilidade”. Isso ocorre pelo fato do sinal de relógio chegar aos flip-flops em tempos diferentes devido aos atrasos causados pela lógica de interconexão e pode fazer com que o circuito não opere de forma correta (HAMBLEN, 2008).

3.4.5.2 Tecnologias de programação

Algumas propriedades dos comutadores dos CPLDs e FPGAs, tais como resistência, tamanho, capacitância e tecnologia de fabricação, afetam principalmente o desempenho e definem características como volatilidade e capacidade de reprogramação. Na escolha de um dispositivo reconfigurável, estes fatores devem ser considerados (TEIXEIRA, 2002). Existem basicamente três tipos de tecnologia de programação:

- SRAM (*Static Random Access Memory*), onde o comutador é um transistor de passagem controlado pelo estado de um bit;
- *Antifuse* é originalmente um circuito que quando programado, forma um caminho de baixa resistência;
- *Gate* flutuante, onde o comutador é um transistor com *gate* flutuante.

Tecnologia SRAM

A tecnologia de programação SRAM, ilustrada na Figura 3.21, usa uma célula de RAM estática para controlar transistores de passagem ou multiplexadores (TEIXEIRA, 2002).

Devido à volatilidade destas memórias, os FPGAs que utilizam essa tecnologia precisam de uma memória externa do tipo PROM, EPROM, EEPROM ou FLASH. Essa tecnologia ocupa muito espaço no *chip*, mas apresenta vantagens: é rapidamente reprogramável e requer apenas a tecnologia padrão de circuitos integrados para a sua fabricação (TEIXEIRA, 2002).

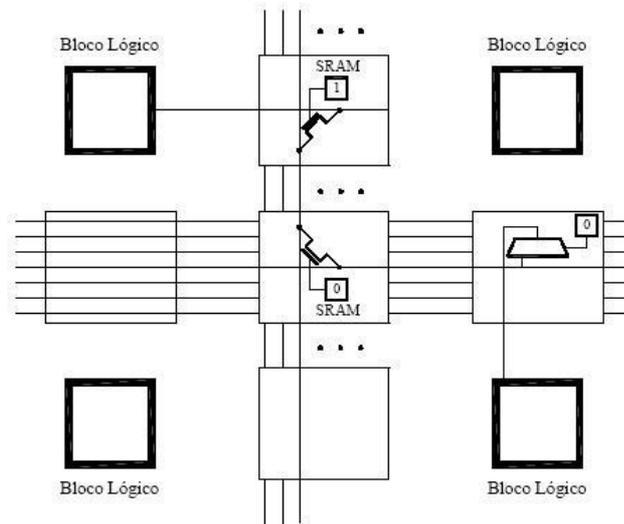


Figura 3.21 – Tecnologia de programação SRAM

Tecnologia Antifuse

O *antifuse* é um dispositivo de dois terminais que no estado não programado apresenta uma alta impedância (equivalente a um circuito aberto) entre seus terminais. Aplicando-se uma tensão adequada em seus terminais, o *antifuse* passa a apresentar um caminho de baixa impedância, equivalente a um curto-circuito (TEIXEIRA, 2002). Algumas vantagens do *antifuse* são tamanho reduzido, baixa capacitância quando não programado, baixa resistência quando programado e o fato de ser não-volátil, garantindo a configuração do circuito integrado mesmo na ausência de alimentação externa. As principais desvantagens dessa tecnologia são um complexo processo de fabricação e a necessidade de um programador externo para sua gravação, que uma vez realizada, não pode ser modificada.

Tecnologia Porta Flutuante

Nessa tecnologia, os comutadores programáveis são baseados em transistores com porta (*gate*) flutuante igual aos usados nas memórias EPROM e EEPROM. A Figura 3.22 ilustra um comutador programável do tipo EPROM (TEIXEIRA, 2002).

O transistor MOS (*Metal-Oxide Semiconductor*) é construído com duas portas. A porta flutuante armazena carga elétrica que é injetada através da aplicação de uma tensão alta entre o terminal da porta 1 e o terminal de fonte (*source*) do transistor. Essa carga faz com que a tensão de limiar (*threshold*) do transistor seja aumentada, impedindo a condução do mesmo. Ao expor o componente a raios ultravioleta, essa carga desaparece e o transistor volta a funcionar normalmente. Na prática, o transistor do tipo EPROM não é usado diretamente como comutador, mas como um *pull-down*. Desta forma, torna-se possível a construção de portas lógicas do tipo *wired-AND*, e conseqüentemente a realização de roteamento e lógica (TEIXEIRA, 2002).

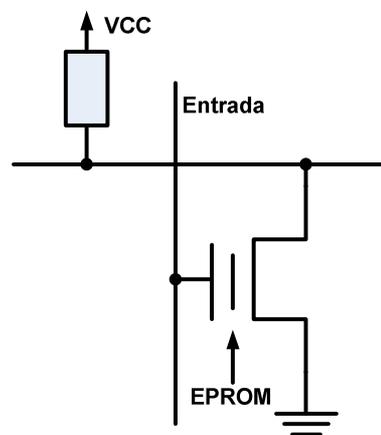


Figura 3.22 – Comutador programável baseado em EPROM

A maior vantagem da tecnologia EPROM é sua capacidade de reprogramação e a retenção dos dados. Como desvantagens, a tecnologia EPROM exige processos adicionais, além do processo normal de fabricação. Além disso, a resistência dos comutadores ligados não atinge valores muito baixos e o consumo total é maior. Quanto à tecnologia EEPROM, apesar de permitir a programação do circuito integrado diretamente na placa, cada célula ocupa o dobro da área de uma célula EPROM (TEIXEIRA, 2002), o que compromete a densidade do dispositivo.

3.4.6 Linguagem de Descrição de *Hardware*

O ponto de início para o desenvolvimento de um circuito digital é a definição da funcionalidade que o mesmo deve possuir. Dependendo da complexidade do circuito, os métodos tradicionais como extração de tabela verdade, obtenção de equações booleanas, simplificações com mapa de Veitch-Karnaugh passam a não ser indicados como metodologia,

pois a produtividade tende a ser muito baixa, exigindo muito tempo do projetista para se alcançar o resultado desejado.

Com o objetivo de aumentar a produtividade e elevar o nível de abstração no desenvolvimento de circuitos digitais, está se tornando cada vez mais comum o uso das Linguagens de Descrição de *Hardware* (HDL - *Hardware Description Language*).

VHDL, que significa VHSIC (*Very High Speed Integrated Circuit*) HDL e Verilog são as duas linguagens mais usadas atualmente para o desenvolvimento de sistemas digitais em DLPs. A estrutura e sintaxe dessas duas linguagens são bem diferentes, mas ambas oferecem basicamente os mesmos recursos. Neste trabalho será adotada a VHDL na fase de implementação das funções do nível 2 da camada MTP da SS7.

3.4.6.1 Níveis de abstração de um Circuito Digital

Um circuito digital pode ser representado em diferentes níveis de abstração e as HDLs permitem descrever sua funcionalidade em todos os níveis, como mostra a Figura 3.23.

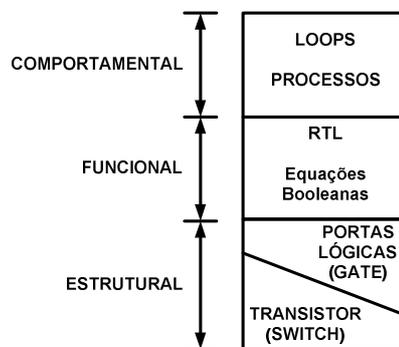


Figura 3.23 – Níveis de abstração

O nível mais baixo de abstração refere-se a circuitos digitais projetados a nível de transistor (*switch*), como no caso dos ASICs *full-custom*. Em um nível um pouco mais alto de abstração estão os circuitos projetados em nível de portas lógicas (*gate*). Esses dois níveis de abstração podem ser classificados como representações estruturais. A função da HDL neste caso é basicamente fazer a conexão entre os elementos (transistores ou portas lógicas).

No próximo nível de abstração, a HDL permite que a funcionalidade de um circuito seja descrita de forma textual, por exemplo, através de equações booleanas. A Figura 3.24 mostra o exemplo da descrição de um multiplexador 2:1 na forma estrutural e funcional.

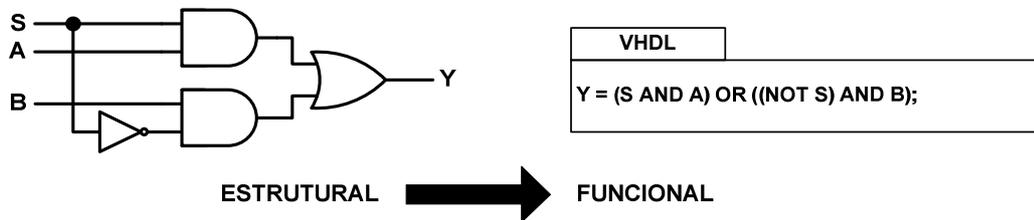


Figura 3.24 – Exemplo de descrição estrutural e funcional

Ainda no nível funcional, a HDL permite a construção de circuitos sequenciais, como registradores de deslocamento, contadores, flip-flops, entre outros. Com isso é possível a implementação de máquinas de estado e circuitos mais complexos.

O nível mais alto de abstração é conhecido como comportamental. Neste nível a HDL permite descrever o circuito baseado em seu comportamento, e para isso, faz uso de construções como *loops* e processos. Fica por conta das ferramentas de síntese a tarefa de identificar as construções usadas na HDL e transformá-la em portas lógicas, LUTs, etc, possibilitando a implementação em um FPGA, por exemplo.

3.4.6.2 VHDL

É uma linguagem de programação desenvolvida para modelagem de circuitos digitais. Com a VHDL é possível descrever as funcionalidades de um circuito digital através de sentenças, elevando o nível de abstração e permitindo que o projeto seja feito através de uma descrição funcional, ao contrário do método tradicional, onde o circuito era projetado em nível de portas lógicas. A VHDL possui recursos para simulação de circuitos digitais considerando as questões de temporização, permitindo que sejam detectados problemas como violação de tempos de *Hold* e *Setup* em flip-flops, memórias, entre outros. A VHDL é um padrão do IEEE, garantindo a portabilidade e longevidade de um projeto.

A linguagem VHDL foi criada pelo Departamento de Defesa dos Estados Unidos como um padrão para documentação de *hardware* na década de 80, e então foi transferida para o IEEE (*Institute of Electrical and Eletronics Engineers*), que em 1987 lançou o padrão 1076, mais conhecido como VHDL-87. Uma nova revisão do padrão foi feita em 1993 e

diversas características foram incorporadas à linguagem, que passou a ser conhecida como o padrão VHDL-93 (CHU, 2006).

O padrão mais recente é o IEEE 1076-2008, publicado em Janeiro de 2009, também conhecido como VHDL 2008. Algumas extensões da linguagem foram desenvolvidas para facilitar a modelagem de circuitos com características mais complexas.

6 SIMULAÇÃO

Em projetos que utilizam dispositivos lógicos programáveis é comum que até uma fase avançada do projeto o *hardware* exista unicamente em forma de arquivo em um computador, sem nenhum circuito físico associado ao mesmo (registros, lógica, memória, etc). Durante esta fase é importante que se possa validar o que está sendo desenvolvido. Como essa validação é feita ainda com os circuitos que existem apenas virtualmente no computador é necessário o uso de técnicas de simulação, com a criação de vetores de teste que simulam os sinais de entrada e verificam se as mudanças nas saídas ocorrem da forma esperada.

As ferramentas de simulação no desenvolvimento de *hardware* em dispositivos lógicos programáveis têm um papel muito importante, não somente para a validação funcional dos circuitos como também para a redução do tempo de desenvolvimento. Todos os fluxogramas e máquinas de estado mostrados anteriormente foram intensivamente simulados utilizando-se o *software* ModelSim-Altera 6.5b (Quartus II 10.1 SP1) Starter Edition®.

A simulação é um dos passos previstos no fluxo de desenvolvimento de *hardware* para Dispositivos Lógicos Programáveis e deve ser feita de maneira exaustiva para garantir a correta funcionalidade do circuito sob teste. Um dos grandes desafios na simulação de circuitos digitais é a geração de um conjunto de estímulos de entrada e saída (também chamados vetores de teste) que possam garantir sua correta funcionalidade. Sem o recurso de simulação, o *hardware* somente poderia ser validado depois de ter sido programado e, portanto depois de existir fisicamente no componente programável. Da mesma forma, todo o *hardware* desenvolvido internamente ao componente programável, para ser validado precisaria ser exercitado por sinais reais gerados externamente ou internamente a esse componente. Testes isolados de blocos seriam complexos e limitados.

A linguagem VHDL, além de ser usada para síntese de circuitos digitais também possui instruções poderosas para a simulação dos mesmos que permitem interação com arquivos, geração de sinais com atrasos programados, detecção de *glitches*, entre outros.

Neste trabalho foi usado o *software* MATLAB® para geração dos vetores de entrada e saída para os diversos blocos que compõem o DAEDT e o DAEDR, permitindo a detecção e correção de falhas em suas funcionalidades.

6.1 SIMULAÇÃO DO DAEDT

Para simulação do DAEDT, foi desenvolvido um algoritmo no MATLAB® para gerar um arquivo de texto contendo um número de SUs definido pelo usuário, sendo cada SU composta por um número aleatório de octetos.

Para gerar os octetos da SU foi usada a função **binornd**. Esta função gera um número ou uma matriz de números aleatórios usando a distribuição binomial. O exemplo a seguir gera uma matriz de tamanho $L \times C$ com distribuição binomial de parâmetros N e p , com $0 \leq p \leq 1$.

$$\text{Sequencia} = \text{binornd}(N, p, L, C)$$

Se $N = 1$ tem-se uma distribuição de Bernoulli e, no caso particular em que $p = 0,5$ é gerada uma sequência binária aleatória com 50% de probabilidade tanto para 1 como para 0. Se o valor de p for maior que 0,5 pode-se gerar uma sequência com maior probabilidade de sequências de 1s consecutivos gerando mais bits de preenchimento nas SUs.

O total de octetos de cada SU também é gerado de forma aleatória pela função **randi**, e assim é possível realizar simulações com SUs de tamanhos diferentes. O exemplo a seguir mostra como gerar um número inteiro aleatório no intervalo definido entre A e B .

$$\text{TotalOctetos} = \text{randi}([A \ B])$$

O exemplo a seguir mostra a geração de uma SU com 70% de probabilidade de 1s (30% de 0s) a partir do total de octetos obtido através da função **randi**. O resultado é uma matriz linha aleatória com $8 * \text{TotalOctetos}$ colunas, sendo esse valor o total de bits da SU.

$$\text{SU} = \text{binornd}(1, 0.7, 1, 8 * \text{TotalOctetos})$$

O processo de geração de SUs se repete de acordo com o total escolhido pelo usuário no momento da execução do algoritmo. A Figura 6.1 mostra um exemplo de arquivo gerado pelo MATLAB contendo três SUs. Para cada SU são geradas duas linhas, onde a primeira

contém o total de octetos em 9 bits, gerado pela função **randi** e a segunda contém os octetos separados por um espaço, gerados com a função **binornd**. O símbolo “*” após o último octeto indica o final de cada SU. O bit mais à esquerda de cada octeto corresponde ao LSB.

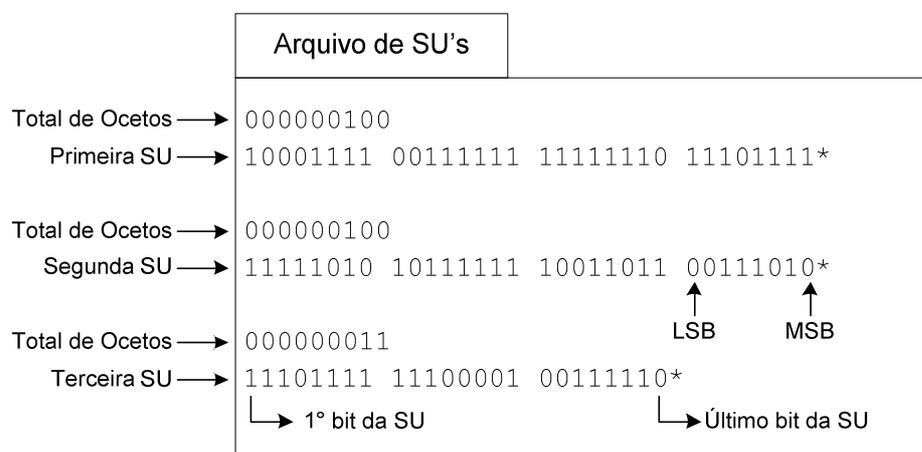


Figura 6.1 – Formato do arquivo de SUs para simulação do DAEDT

Além do arquivo com as SUs a serem transmitidas, também foi desenvolvido um algoritmo no MATLAB para gerar um arquivo com os resultados esperados para que seja possível comprovar a correta funcionalidade do DAEDT. Para obter esses resultados, para cada SU no arquivo de entrada o algoritmo realiza as operações de inserção de *flags*, inserção de bits de preenchimento e geração de CRC, da mesma forma como é realizado pelo DAEDT.

Usando a primeira SU da Figura 6.1 como exemplo, serão mostrados a seguir os passos executados pelo algoritmo gerador de resultados esperados. Estes passos são repetidos para cada SU do arquivo de entrada.

- ✓ **Passo 1:** Obtém os 16 bits de CRC a partir de uma função criada no MATLAB que recebe os octetos da SU (sem os bits de preenchimento) e realiza a divisão do polinômio que representa os bits da SU pelo polinômio gerador, da mesma forma como é feito no Gerador de CRC visto no DAEDT.

Octetos da SU: 10001111 00111111 11111110 11101111

CRC da SU: 1010011101110010

- ✓ **Passo 2:** Unifica os octetos da SU com os 16 bits de CRC.

10001111 00111111 11111110 11101111 1010011101110010

- ✓ **Passo 3:** Insere os bits de preenchimento (sublinhados). A SU usada neste exemplo possui 3 bits de preenchimento. Para isso foi criada uma função no MATLAB que recebe a SU com o CRC e retorna a mesma com os bits de preenchimento. É importante observar que devido aos bits de preenchimento o total de bits resultante pode não ser múltiplo de 8. As sequências de cinco bits em nível alto estão destacadas em negrito.

10001111 001**11110** **11110**11 101110**11** **111**00100 11101110 010

- ✓ **Passo 4:** Insere o *flag* de abertura. O *flag* de abertura da próxima SU servirá como *flag* de fechamento para esta SU, e não aparece na sequencia abaixo, e assim sucessivamente.

01111110 10001111 00111110 11111011 10111011 11100100 11101110 010

Durante o intervalo de tempo de canal definido na simulação, o sinal **SU_E1** gerado pelo DAEDT é comparado bit a bit com o arquivo de resultados esperados gerado pelo MATLAB e qualquer divergência é sinalizada nas janelas *Wave* e *Transcript* do ModelSim.

Esta forma de simulação garantiu uma excelente confiabilidade nos resultados obtidos e uma excelente cobertura de diversas condições de tamanhos e conteúdos de mensagens, uma vez que os octetos das SUs são gerados de forma aleatória pelo MATLAB. A Figura 6.2 mostra a arquitetura da simulação usada para o DAEDT.

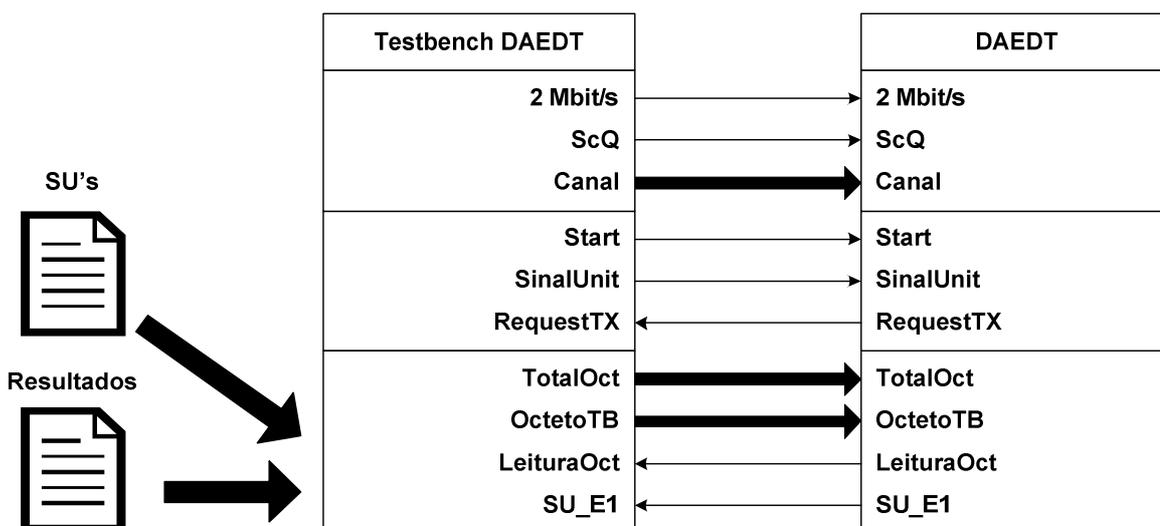


Figura 6.2 – Arquitetura de simulação do DAEDT

O bloco **Testbench** é um módulo escrito em VHDL que tem a função de gerar estímulos e analisar as respostas do DAEDT, garantindo assim seu correto funcionamento.

Ele faz o papel de uma jiga de teste virtual que não vem a ser sintetizada em circuito. O sinal de relógio de 2 Mbit/s (mais precisamente 2,048 Mbit/s) é gerado em uma única linha de VHDL,

```
Clk2M_s <= not Clk2M_s after 244.140625 ns;
```

O fluxograma da Figura 6.3 mostra o procedimento para geração do sinal de sincronismo de quadro, **ScQ**. Nos equipamentos de telecomunicações, tanto o relógio de 2 Mbit/s como o sincronismo de quadro são extraídos do nível 1.

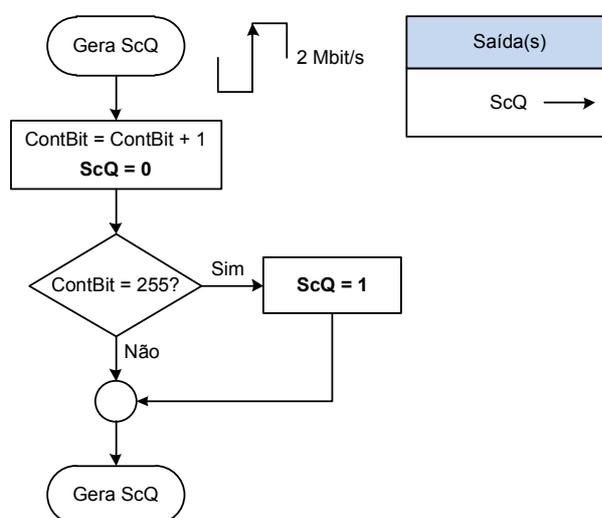


Figura 6.3 – Fluxograma para geração do sincronismo de quadro

O sinal **ContBit** é do tipo **std_logic_vector** com 8 bits e por isso a contagem volta para 0 após o valor 255. O valor inicial do sinal **ContBit** pode ser qualquer um entre 0 e 255 e com isso elimina-se a necessidade de um sinal de reset. O sinal **ScQ** tem a duração de um ciclo do relógio de 2 Mbit/s, sendo ativado durante o tempo em que o sinal **ContBit** é 255.

O sinal **Canal** possui cinco bits e indica o número do canal dentro do enlace PCM em que o DAEDT irá transmitir os octetos. Na simulação do DAEDT este sinal foi definido como uma constante de valor 5. Para mudar o número do canal, essa constante deve ser alterada e deve ser feita uma nova simulação.

```
constant Canal_cte : std_logic_vector(4 downto 0) := "00101";
```

Os sinais **OctetoTB** e **TotalOct** são obtidos através da leitura do arquivo de SUs mostrado na Figura 6.1, e o sinal **SU_E1**, como dito anteriormente, é comparado com o

conteúdo do arquivo de resultados esperados. As figuras a seguir, extraídas do simulador ModelSim, mostram os pontos mais relevantes da simulação do DAEDT usando os vetores da Figura 6.1 como entrada.

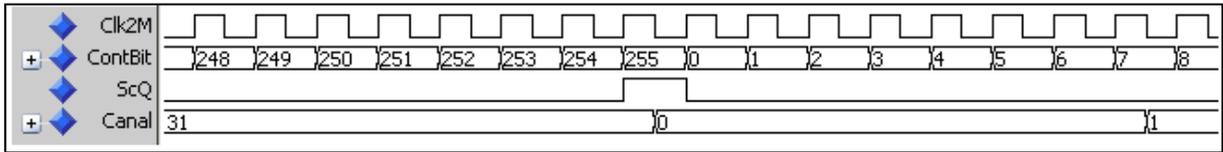


Figura 6.4 – Geração do relógio de 2 Mbit/s e Sincronismo de Quadro

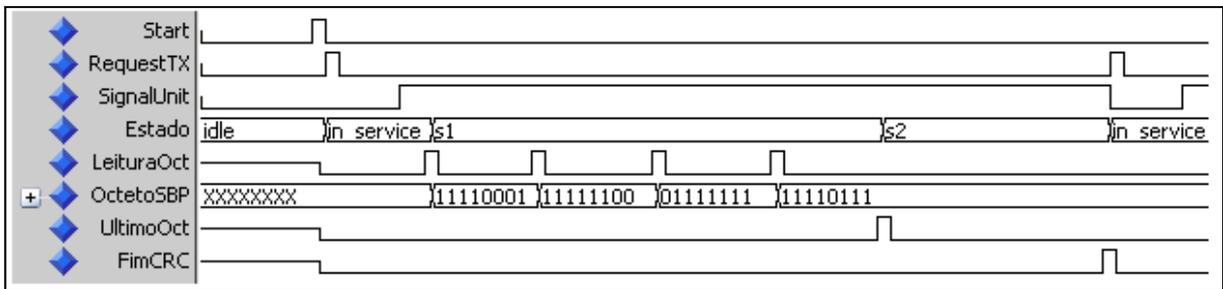


Figura 6.5 – Transmissão da primeira SU do arquivo de vetores

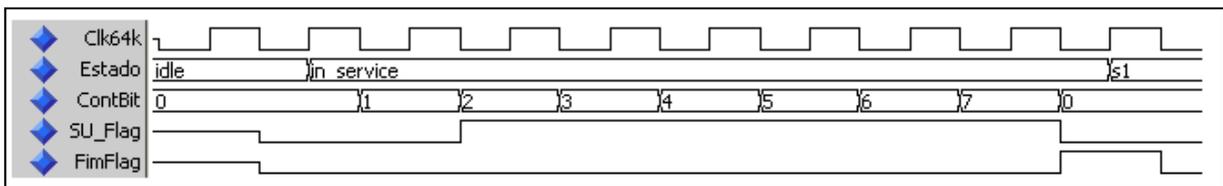


Figura 6.6 – Transmissão do flag de abertura

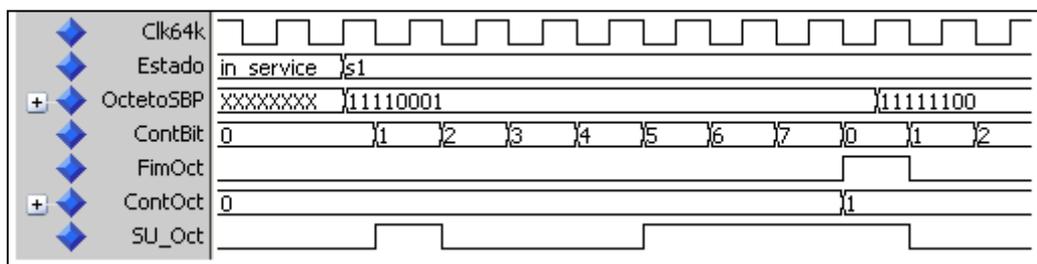


Figura 6.7 – Transmissão do primeiro octeto da SU

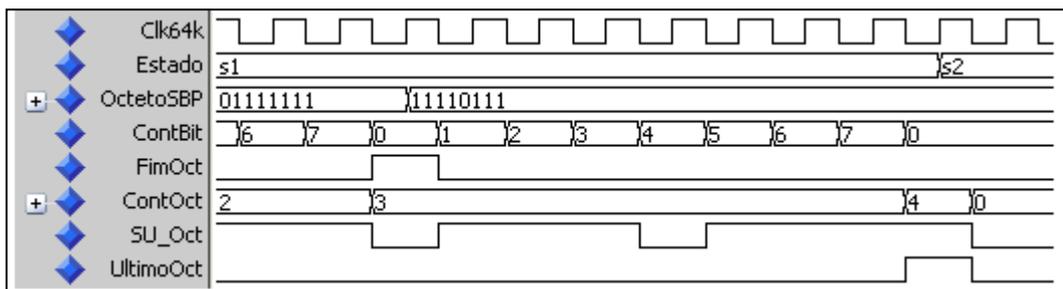


Figura 6.8 – Transmissão do último octeto da SU

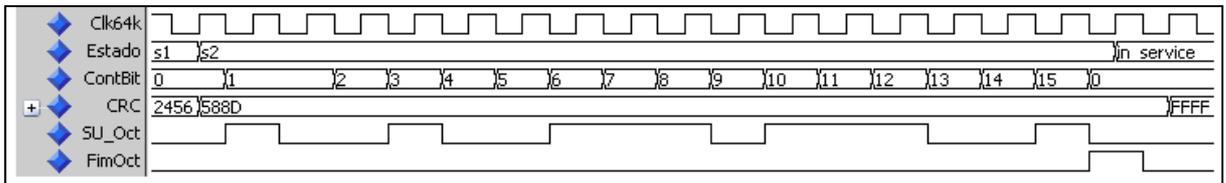


Figura 6.9 – Transmissão dos 16 bits de CRC da SU

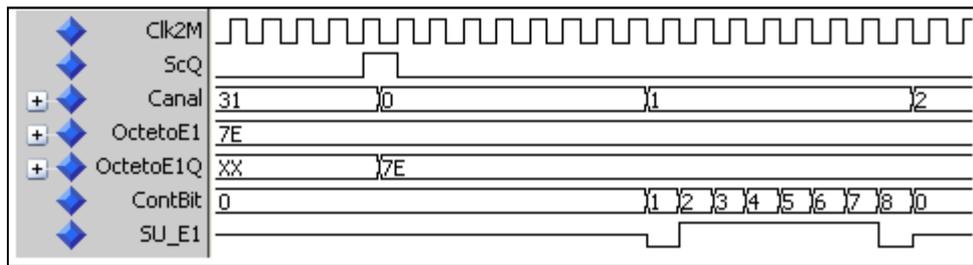


Figura 6.10 – Inserção dos bits da SU no canal 1 do E1

6.2 SIMULAÇÃO DO DAEDR

A função do DAEDR é recuperar os octetos originais (sem bits de preenchimento) a partir do sinal **SU_E1** recebido do DAEDT do PS de origem. Desta forma, para a simulação do DAEDR foi usado o módulo DAEDT para gerar o sinal **SU_E1**, dispensando a necessidade da criação de um arquivo de entrada. Da mesma forma o arquivo de SUs usado como entrada para o DAEDT foi usado como arquivo de resultados esperados pelo DAEDR. A Figura 6.11 mostra a arquitetura da simulação usada para o DAEDR.

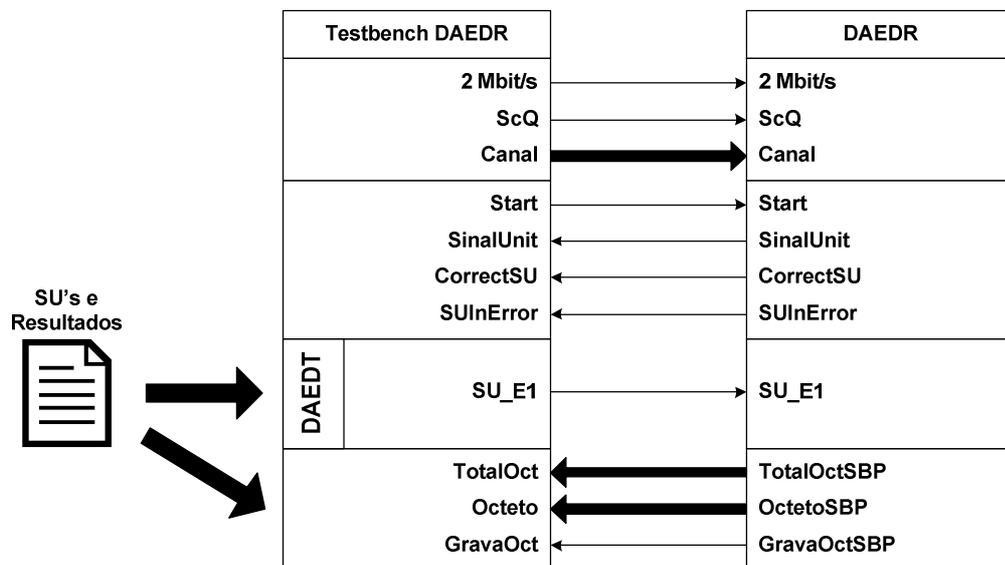


Figura 6.11 – Arquitetura de simulação do DAEDR

Cada vez que o sinal **GravaOctSBP** é ativado, o conteúdo do sinal **OctetoSBP** é comparado com o um octeto do arquivo de SUs (sinal **Octeto** do testbench). Quando todos os octetos da SU são recebidos, o sinal **SignalUnit** é ativado pelo DAEDR e o sinal **TotalOctSBP** é comparado com o sinal **TotalOct** do *testbench* obtido do arquivo de SUs para garantir que a quantidade de octetos recebidos está correta.

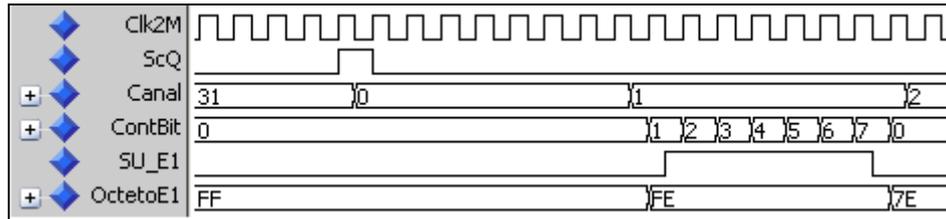


Figura 6.12 – Extração dos bits da SU no canal 1 do E1

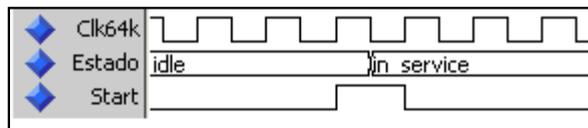


Figura 6.13 – Ativação do DAEDR

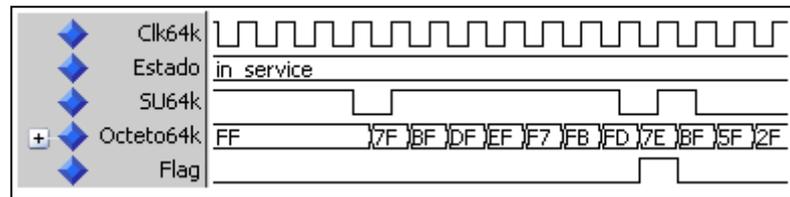


Figura 6.14 – Detector de flag

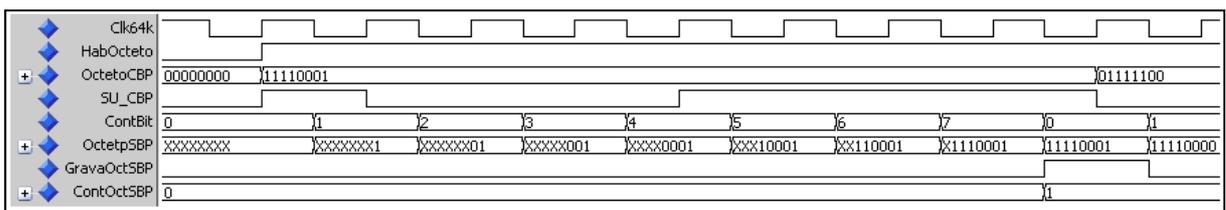


Figura 6.15 – Recepção do primeiro octeto SBP da SU

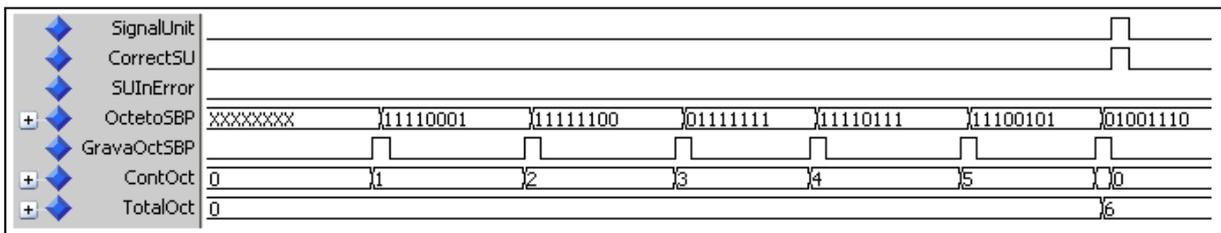


Figura 6.16 – Recepção de todos os octetos de uma SU

7 ANÁLISE DOS RESULTADOS, CONCLUSÕES E TRABALHOS FUTUROS

7.1 ANÁLISE DOS RESULTADOS

As simulações das funções DAEDT e DAEDR realizadas com auxílio de vetores aleatórios gerados pelo MATLAB permitiram comprovar seu correto funcionamento. A técnica de geração de vetores de forma aleatória usando o MATLAB é uma contribuição deste trabalho como metodologia para simulação/validação de circuitos digitais usando linguagem de descrição de *hardware* e permitiu a detecção de condições de falha que dificilmente seriam avaliadas com vetores gerados de forma manual. Os resultados obtidos nas simulações serviram para refinar a implementação da MTP-2 *Low* garantindo assim um alto grau de confiabilidade para que a mesma seja usada em equipamentos de telecomunicações.

Tanto a MTP-2 *Low* como a Matriz de Comutação Programável apresentadas neste trabalho foram codificadas em VHDL. A especificação da MTP-2 em (ITU-T, 1993c) consiste em uma descrição textual, figuras e diagramas em SDL. A metodologia para que a partir dessas especificações fosse possível chegar a uma implementação em VHDL é também uma contribuição deste trabalho.

A portabilidade da solução entre Dispositivos Lógicos Programáveis de diferentes famílias e fabricantes, e a conseqüente possibilidade de reuso em outros equipamentos, foi alcançada através do uso somente de instruções nativas da VHDL e que são possíveis de serem sintetizadas. Após a compilação no ambiente de desenvolvimento da ALTERA, o Quartus II 10.1 SP1 *Full Version*, verificou-se que não houve nenhuma mensagem relacionada a instruções não sintetizáveis. Este resultado traz o benefício de facilitar a atualização do *hardware*, permitindo não somente a mudança para outros FPGAs do mesmo fabricante, mas também o uso de FPGAs de outros fabricantes. Com este resultado tem-se uma maior garantia de continuidade e evolução da solução quando comparado com as soluções apresentadas por D'ÁVILA (1998) e SHAOREN *et al* (1996), que fazem uso de equipamentos comerciais ou de tecnologias, que quando obsoletos, podem levar um produto à descontinuidade.

A escalabilidade da solução foi alcançada não somente pela escolha da tecnologia de Dispositivos Lógicos Programáveis na implementação, mais especificamente, FPGA, que permite o crescimento do projeto sem alterações externas no *hardware*, mas também com a implementação de uma Matriz de Comutação Programável entre os pares DAEDT/DAEDR e o Nível 1, podendo ser usada não somente no projeto HUI, mas em qualquer equipamento de telecomunicações com um número maior ou menor de enlaces E1 para diferentes quantidades de pares DAEDT/DAEDR. A implementação utilizada para essa matriz é também uma contribuição deste trabalho. Quando comparado com o trabalho de NETO *et al* (1990), este resultado mostra a flexibilidade alcançada com a solução através do uso de FPGA.

A análise dos recursos necessários para a implementação da solução em FPGA mostrada a seguir refere-se somente à MTP-2 *Low*. A solução completa será avaliada em conjunto com o trabalho de Mitsuyoshi Nishi de Carvalho. Foram criados alguns cenários com quantidades variadas de interfaces E1 e pares DAEDT/DAEDR. Esses cenários foram compilados usando a ferramenta Quartus II Web Edition, versão 9.1 SP2 da ALTERA e o FPGA usado na compilação foi o EP4CE30F23C7, da família Cyclone IV com 28.848 Elementos Lógicos, também usado no projeto HUI. A mostra a quantidade de elementos lógicos e pinos necessários no FPGA para cada cenário.

Cenário	Número de interfaces E1	Número de pares DAEDT/DAEDR	Total de Elementos Lógicos	Total de Pinos
1	4	2	837 (3%)	36 (11%)
2	4	4	1635 (6%)	36 (11%)
3	4	8	3254 (11%)	36 (11%)
4	4	16	6446 (22%)	36 (11%)
5	16	16	6989 (24%)	60 (18%)

Tabela 7.1 – Recursos utilizados no FPGA para a MTP-2L

Analisando os dados obtidos na , pode-se observar que a quantidade de elementos lógicos usada no FPGA é proporcional ao número de pares DAEDT/DAEDR para a mesma quantidade de E1s. Para 16 pares DAEDT/DAEDT (Cenário 4) o consumo de elementos lógicos do FPGA foi menor que 25% de sua capacidade quando a solução possui 4 enlaces E1 (caso do projeto HUI), restando 75% para a implementação da parte *High* da MTP-2. É

importante observar que a mudança de 4 para 16 EIs, para a mesma quantidade de DAEDT/DAEDR, causou um aumento de apenas 2% no uso de elementos lógicos. Isto se deve ao fato de que esta mudança tem impacto somente na matriz de comutação. Os 24 pinos adicionais no cenário 5 se devem aos 12 enlaces E1 de transmissão e 12 de recepção que foram adicionados em relação aos 4 usados nos cenários anteriores.

Fazendo-se uma comparação quantitativa com o trabalho de NETO *et al* (1990) que mostra a implementação da SS7 nas centrais da empresa Trópico Telecomunicações (interveniente do projeto HUI) na placa CCO, pode-se concluir que a implementação da SS7 utilizando-se o FPGA EP4CE30F23C7 permite um aumento considerável na capacidade do equipamento. Enquanto que a placa CCO tem capacidade para tratar dois enlaces SS7, a mostra que, dependendo dos resultados da MTP-2H, a solução aqui apresentada poderá chegar a um número maior.

Outros FPGAs, com maior quantidade de elementos lógicos podem ser utilizados para alcançar um número maior de tratadores de enlaces SS7. É comum que para um mesmo fabricante e família de FPGAs existam componentes com pinagem compatíveis para diferentes quantidades de elementos lógicos. No caso do EP4CE30F23C7, que possui 30k LEs, pode-se realizar a migração vertical para 40k (EP4CE40F23C7) ou 55k LEs (EP4CE55F23C7).

Com a longevidade da tecnologia de Dispositivos Lógicos Programáveis, aliada à portabilidade e escalabilidade alcançadas, entende-se que a solução desenvolvida está preparada para atender atuais e futuras evoluções tecnológicas e demandas em novos equipamentos ou redes.

7.2 CONCLUSÕES

As principais contribuições deste trabalho de pesquisa são: 1) uma proposta de implementação em linguagem VHDL da parte *Low* da camada MTP-2 da SS7 em conjunto com uma Matriz de Comutação Programável da forma mais genérica possível de modo a constituir uma solução modular, portátil (reutilizável) e escalável para poder ser usada em várias tecnologias e em equipamentos de telecomunicações com diferentes arquiteturas e capacidades; 2) desenvolvimento de uma metodologia para implementação em VHDL de

circuitos digitais a partir de uma descrição visual com o uso de fluxogramas. Ressalta-se também que esse trabalho, vinculado ao projeto HUI, descrito anteriormente, desenvolvido em parceria entre a Universidade Federal do Amazonas - UFAM e a empresa Trópico Telecomunicações, com subsídios da Financiadora de Estudos e Projetos, FINEP, resultou em artefatos de utilização prática.

Como já salientado, este trabalho se complementa com o de Mistuyoshi Nishi de Carvalho (CARVALHO, 2012) que também é vinculado ao projeto HUI, que consiste no desenvolvimento de uma nova geração de *hardware* para equipamentos da empresa Trópico Telecomunicações que inclui partes que deverão incorporar versões particulares dos resultados destes trabalhos de pesquisa.

7.3 TRABALHOS FUTUROS

O uso de um único canal do enlace E1 permite a comunicação entre dois Pontos de Sinalização a uma taxa de 64 kbit/s. Ao se fazer o uso de mais de um canal para a comunicação entre dois elementos em uma rede SS7, é possível aumentar essa taxa para 128 kbit/s, 192 kbit/s, 256 kbit/s, e assim sucessivamente, permitindo que a solução seja utilizada em *links* de alta velocidade (HSL – *High Speed Links*). A implementação da parte *Low* da MTP-2 apresentada neste trabalho de pesquisa pode ser modificada de forma a permitir o uso de mais de um canal para que a mesma seja usada em soluções HSL, também previstas nas recomendações do ITU-T.

Como descrito anteriormente, a implementação da MTP-2 foi dividida nas partes *Low* e *High*. Um próximo trabalho é realizar a integração das partes e avaliar a solução em termos de recursos utilizados no FPGA. Também serão realizados testes práticos da solução nos equipamentos da empresa Trópico Telecomunicações de acordo com os critérios e regras estabelecidos em ITU-T (1993d) e ITU-T (1993e) para comprovar seu correto funcionamento.

A solução pode ser enriquecida através da implementação de recursos adicionais de testabilidade, incluindo a possibilidade de fechamento da malha de transmissão e recepção (*loopback*), geradores e detectores de sequências pseudo aleatórias e insersor programável de taxa de erro, permitindo avaliar parâmetros importantes na rede, como por exemplo, a qualidade do enlace E1.

A Matriz de Comutação Programável pode ser modificada para incorporar Compensadores de Atraso, de modo que os enlaces E1 possam ser recebidos em qualquer fase, dispensando o uso de Memórias Elásticas. Para isso, os enlaces de transmissão e recepção devem receber seus sinais de sincronismo de forma independente.

Um trabalho que pode ser realizado e que parece ser bastante promissor é a alteração da arquitetura da solução apresentada para permitir o uso “multiplexado” de um único par DAED para tratar mais de um Enlace SS7. Esta característica pode ser alcançada usando-se uma frequência maior que 64 kbit/s para tratar outros canais nos intervalos de tempo em que o circuito está em “repouso”. Dependendo da complexidade desta nova arquitetura, pode-se alcançar uma redução importante na quantidade de recursos utilizados no FPGA, permitindo o uso de componentes mais baratos ou o aumento da quantidade de Enlaces SS7 na solução.

REFERÊNCIAS BIBLIOGRÁFICAS

[ITU93a] ITU-T. Recommendation Q.700: Introduction to CCITT Signalling System No. 7. Revisada pelo ITU-T Study Group XI (1988-1993), Helsinki, 1993.

[ITU93b] ITU-T. Recommendation Q.701: Functional Description of The Message Transfer Part (MTP) of Signalling System No. 7. Revisada pelo ITU-T Study Group XI (1988-1993), Helsinki, 1993.

[ITU93c] ITU-T. Recommendation Q.703: Signalling System No. 7 - Signalling Link. Revisada pelo ITU-T Study Group XI (1988-1993), Helsinki, 1993.

[ITU93d] ITU-T. Recommendation Q.780: Signalling System No. 7 Test Specification – General Description. Revisada pelo ITU-T Study Group XI (1988-1993), Helsinki, 1993.

[ITU93e] ITU-T. Recommendation Q.781: Signalling System No. 7 – MTP Level 2 Test Specification. Revisada pelo ITU-T Study Group XI (1988-1993), Helsinki, 1993.

[ITU93f] ITU-T. Recommendation X.141: General Principles for the Detection and Correction of Errors in Public Data Networks. Extraída do fascículo VIII.3 do Livro Azul, 1993.

[ITU98] ITU-T. Recommendation G.704: Synchronous frame structures used at 1544, 6312, 2048, 8448 and 44 736 kbit/s hierarchical levels. Revisada pelo ITU-T Study Group 15 (1997-2000). Aprovada em Outubro de 1998.

[ITU07] ITU-T. Recommendation Z.100: Specification and Description Language SDL. Revisada pelo ITU-T Study Group XI (2005-2008), 2007.

DIAZ, V. A. V. Relatório Técnico: Anteprojeto de Implementação Hardware da MTP-2 Lower part do Nível 2 da SS7. Campinas, 2009.

D'ÁVILA , M. H. C. Contribuições para a Implementação do Sistema de Sinalização por Canal Comum nº 7 em uma Central Telefônica Baseada em Ambiente de Processamento Distribuído. 1998. 68 f. Dissertação (Mestrado em Ciência da Computação), Universidade Federal de Minas Gerais - UFMG, Belo Horizonte. 1998.

CARVALHO , M. N. Proposta de Implementação das Funções de Controle para o Nível 2 do Sistema de Sinalização por Canal Comum Número 7 Usando Dispositivos Lógicos Programáveis. Dissertação (Mestrado em Engenharia Elétrica), Universidade Federal do Amazonas - UFAM, Manaus. 2012. A publicar.

LIMA, F. G. Projeto com Matrizes de Células Lógicas Programáveis. 1999. 126 f. Dissertação (Mestrado em Ciência da Computação), Universidade Federal do Rio Grande do Sul - UFRGS, Porto Alegre. 1999.

TEIXEIRA, M. A. Técnicas de Reconfigurabilidade dos FPGAs da Família APEX 20K – Altera. 2001. 142 f. Dissertação (Mestrado em Ciência da Computação), Universidade de São Paulo - USP, São Carlos. 2002.

BHART, S.; GANDHI, J. Study and Simulation of CCS#7 Protocol Using System Verilog. 2010. 40 f. Documento de Qualificação (Mestrado em Engenharia Elétrica), Universidade do Estado da Califórnia, Sacramento. 2010.

HAMBLEN, J. O.; HALL T. S.; FURMAN, M. D. Rapid Prototyping Of Digital Systems: SOPC Edition. 1ª Edição. Springer, 2008. 411p.

BAGAD, V. S. Telecommunication Switching Systems and Networks. 1ª Edição. Technical Publications, 2009. 344p.

CHU, P. P. RTL Hardware Design Using VHDL: Coding for Efficiency, Portability and Scalability. 1ª Edição. Wiley-Interscience, 2006. 669p.

ZEIDMAN, B. Designing with FPGAs and CPLDs. CMP Book, 2002. 220p.

NETO, I. L. da S.; SILVEIRA, L. M. Implementing Signalling System No. 7 In Trópico RA System. Telecommunications Symposium, 1990. ITS '90 Symposium Record., SBT/IEEE International, p.137-142, 3-6 Set 1990.

SHAOREN, Z.; JIANFENG, Z.; JUN, Z. The implementation of SS7 in JSU-16000, Communication Technology Proceedings, 1996. ICCT'96. International Conference on, v. 2, p. 970-972, 5-7 May, 1996.

BONATTI, I.S.; FIGUEIREDO, R. J. O. Stoht - An SDL-to-Hardware Translator, Design Automation Conference. Proceedings of the ASP-DAC '95/CHDL '95/VLSI '95., IFIP International Conference on Hardware Description Languages; IFIP International Conference on Very Large Scale Integration., Asian and South Pacific, pp.33-36, 29 Aug - 1 Sep, 1995.

MING-HWA, S.; MING-DER, S.; SHENG-WEI, L.; CHIE DOW. An efficient hardware design approach from system-level specification. Circuits and Systems. Proceedings of the 40th Midwest Symposium on, vol.2, pp. 1213-1216, 3-6 Aug. 1997.

MORTADA, M.; ABDEL-HAMID, A.T. SDL to VHDL conversion rules: A case study. Microelectronics, 2007. ICM 2007. Internatonal Conference on, pp.249-252, 29-31 Dec. 2007.

ANEXO 1 - DIAGRAMA FUNCIONAL DA MTP-2

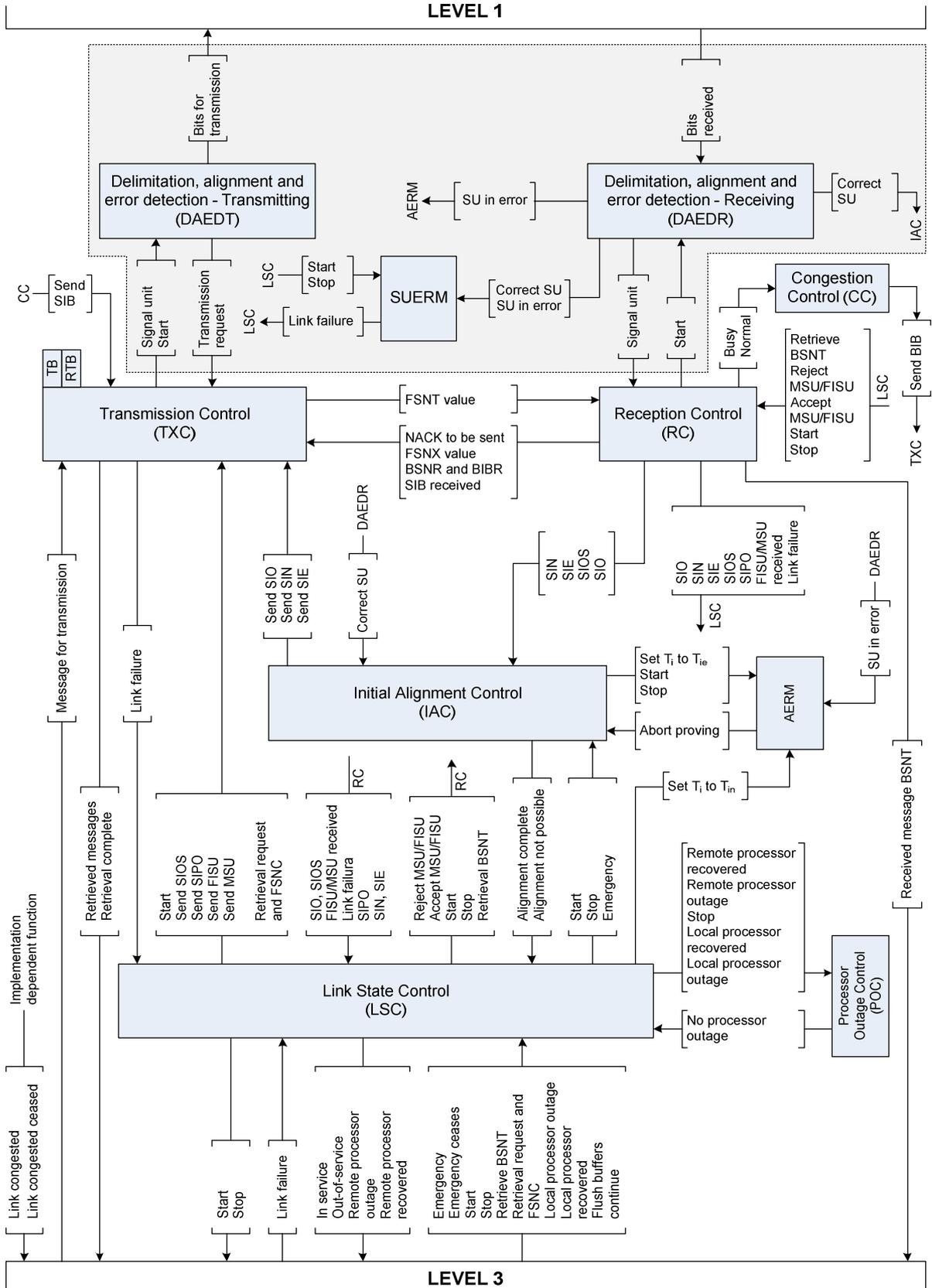


Figure 7/Q.703 – Level 2 – Functional block diagram

ANEXO 2 - DIAGRAMA SDL DO DAEDT

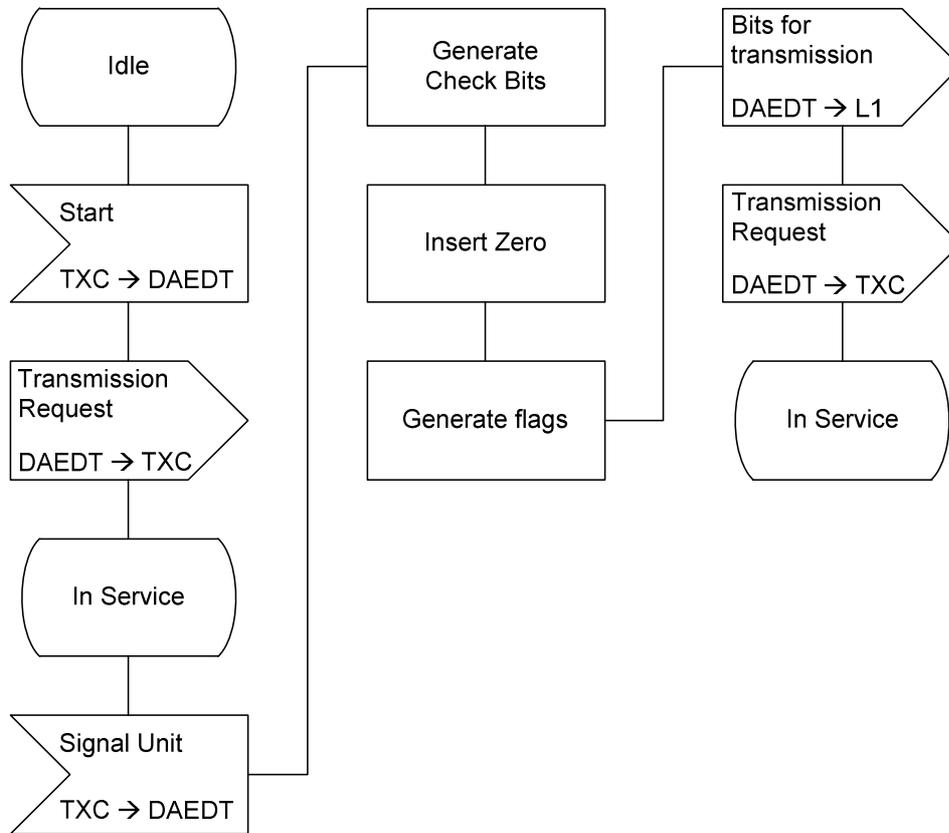


Figure 12/Q.703 - Delimitation, alignment and error detection (transmitting)

ANEXO 3 - DIAGRAMA SDL DO DAEDR

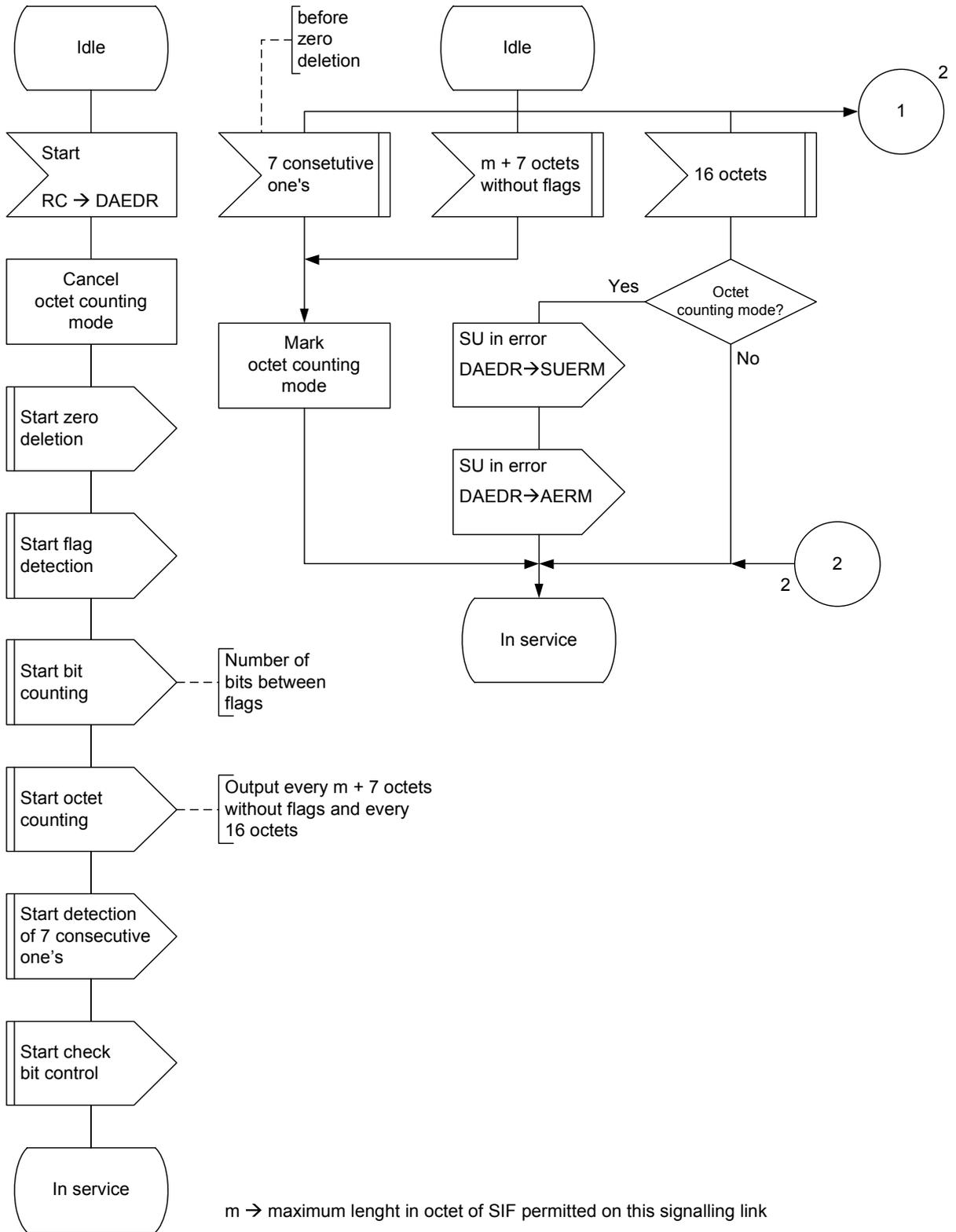


Figure 11/Q.703 (sheet 1 of 2) – Delimitation, alignment and error detection (receiving)

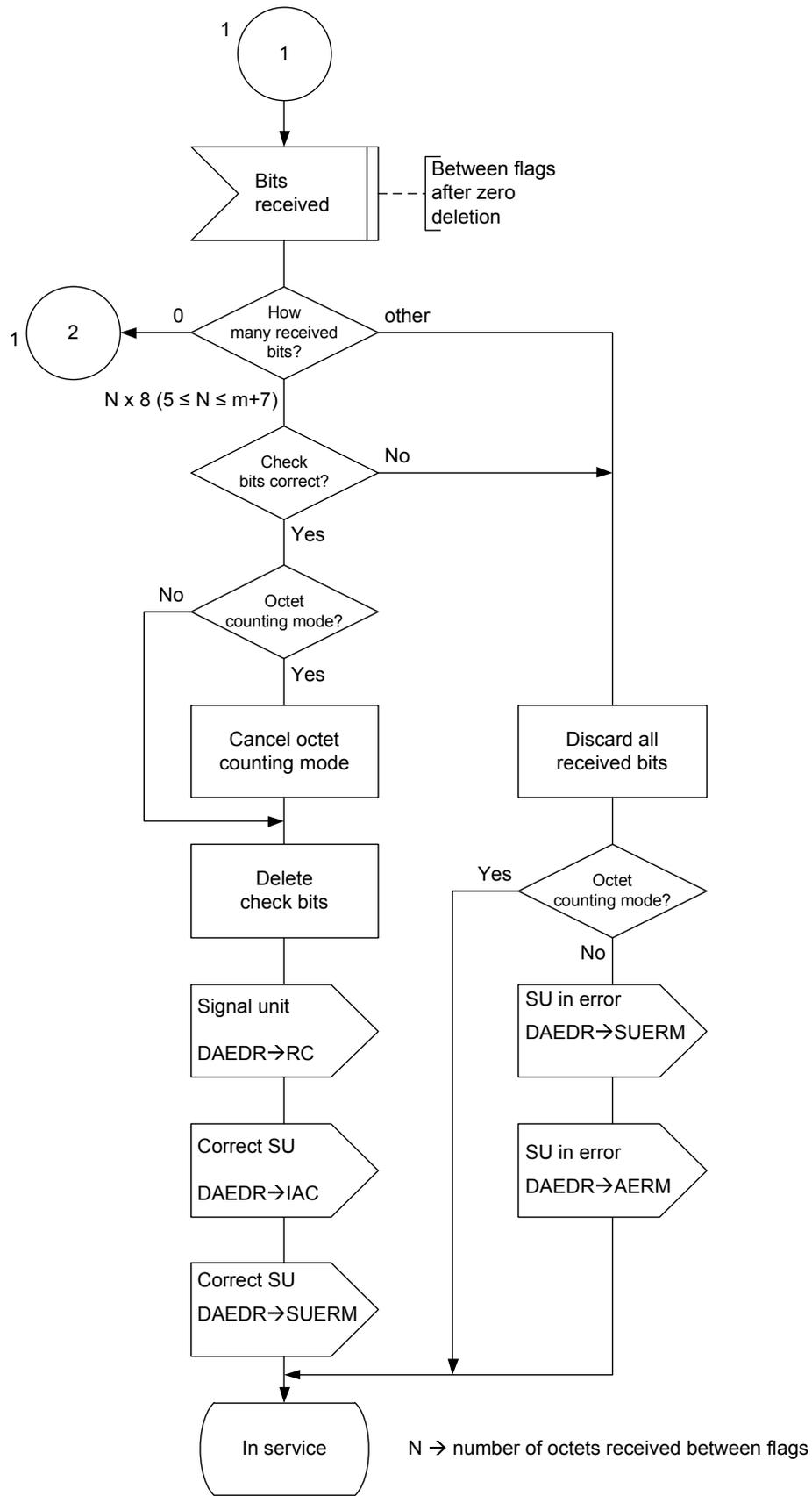


Figure 11/Q.703 (sheet 2 of 2) – Delimitation, alignment and error detection (receiving)