



Ministério da  
**Ciência, Tecnologia  
e Inovação**



sid.inpe.br/mtc-m19/2013/10.04.19.58-NTE

## MODELAGEM DO MÓDULO DE COMUNICAÇÃO DO SATÉLITE ITASAT SEGUNDO A METODOLOGIA COFI

Arineiza Cistina Pinheiro

Ana Maria Ambrosio

Estudo de caso realizado no contexto do projeto ITASAT com a colaboração do prof. Dr. Adenilso da Silva Simão (ICMC-USP), da prof<sup>a</sup> Dr<sup>a</sup>. Emília Villani (ITA) e do aluno de mestrado Waldo Acioli Falcão de Alencar (ITA).

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3EUF338>>

INPE

São José dos Campos

2013

## **PUBLICADO POR:**

Instituto Nacional de Pesquisas Espaciais - INPE

Gabinete do Diretor (GB)

Serviço de Informação e Documentação (SID)

Caixa Postal 515 - CEP 12.245-970

São José dos Campos - SP - Brasil

Tel.:(012) 3208-6923/6921

Fax: (012) 3208-6919

E-mail: pubtc@sid.inpe.br

## **CONSELHO DE EDITORAÇÃO E PRESERVAÇÃO DA PRODUÇÃO INTELLECTUAL DO INPE (RE/DIR-204):**

### **Presidente:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

### **Membros:**

Dr. Antonio Fernando Bertachini de Almeida Prado - Coordenação Engenharia e Tecnologia Espacial (ETE)

Dr<sup>a</sup> Inez Staciarini Batista - Coordenação Ciências Espaciais e Atmosféricas (CEA)

Dr. Gerald Jean Francis Banon - Coordenação Observação da Terra (OBT)

Dr. Germano de Souza Kienbaum - Centro de Tecnologias Especiais (CTE)

Dr. Manoel Alonso Gan - Centro de Previsão de Tempo e Estudos Climáticos (CPT)

Dr<sup>a</sup> Maria do Carmo de Andrade Nono - Conselho de Pós-Graduação

Dr. Plínio Carlos Alvalá - Centro de Ciência do Sistema Terrestre (CST)

### **BIBLIOTECA DIGITAL:**

Dr. Gerald Jean Francis Banon - Coordenação de Observação da Terra (OBT)

### **REVISÃO E NORMALIZAÇÃO DOCUMENTÁRIA:**

Marciana Leite Ribeiro - Serviço de Informação e Documentação (SID)

Yolanda Ribeiro da Silva Souza - Serviço de Informação e Documentação (SID)

### **EDITORAÇÃO ELETRÔNICA:**

Maria Tereza Smith de Brito - Serviço de Informação e Documentação (SID)

Luciana Manacero - Serviço de Informação e Documentação (SID)



Ministério da  
**Ciência, Tecnologia  
e Inovação**



sid.inpe.br/mtc-m19/2013/10.04.19.58-NTE

## MODELAGEM DO MÓDULO DE COMUNICAÇÃO DO SATÉLITE ITASAT SEGUNDO A METODOLOGIA COFI

Arineiza Cistina Pinheiro

Ana Maria Ambrosio

Estudo de caso realizado no contexto do projeto ITASAT com a colaboração do prof. Dr. Adenilso da Silva Simão (ICMC-USP), da prof<sup>a</sup> Dr<sup>a</sup>. Emília Villani (ITA) e do aluno de mestrado Waldo Acioli Falcão de Alencar (ITA).

URL do documento original:

<<http://urlib.net/8JMKD3MGP7W/3EUF338>>

INPE

São José dos Campos

2013



## LISTA DE FIGURAS

	<u>Pág.</u>
1.1 Arquitetura do satélite ITASAT-1 (ITASAT, 2011). . . . .	3
2.1 S1_Normal: Máquina de estados finitos do modo de operação normal do CM Receiver. . . . .	8
2.2 S1_Ex13: Máquina de estado finitos para exceções descritas nos itens 1 e 3 da Tabela 2.3. . . . .	9
2.3 S1_Ex2: Máquina de estados finitos para exceção descrita no item 2 da Tabela 2.3. . . . .	10
2.4 S1_Ex4: Máquina de estados finitos para exceção descrita no item 4 da Tabela 2.3. . . . .	11
2.5 S1_Ex5: Máquina de estados finitos para exceção descrita no item 5 da Tabela 2.3. . . . .	12
3.1 S1_Normal: Máquina de estados finitos do modo de operação normal do CM Receiver. . . . .	20
3.2 S1_Ex13: Máquina de estado finitos para exceções descritas no item 1 da Tabela 3.3. . . . .	21
3.3 S1_Ex2: Máquina de estados finitos para exceção descrita no item 2 da Tabela 3.3. . . . .	22
A.1 Conjunto de testes gerado pelo método SPY para a MEF S1_N do Receiver.	29
A.2 Conjunto de testes gerado pelo método SPY para a MEF S1_Ex13 do Receiver. . . . .	30
A.3 Conjunto de testes gerado pelo método SPY para a MEF S1_Ex2 do Receiver. . . . .	31
A.4 Conjunto de testes gerado pelo método SPY para a MEF S1_Ex4 do Receiver. . . . .	32
A.5 Conjunto de testes gerado pelo método SPY para a MEF S1_Ex5 do Receiver. . . . .	33
B.1 Conjunto de testes gerado pelo método UIO para a MEF S1_N do Trans- mitter. . . . .	34
B.2 Conjunto de testes gerado pelo método UIO para a MEF S1_Ex1 do Transmitter. . . . .	35
B.3 Conjunto de testes gerado pelo método UIO para a MEF S1_Ex2 do Transmitter. . . . .	35

## LISTA DE TABELAS

	<u>Pág.</u>
2.1 Lista de entradas do SUT ( <i>CM Receiver</i> ). . . . .	6
2.2 Lista de saídas e Ações Observáveis do SUT ( <i>CM Receiver</i> ). . . . .	6
2.3 Relação Requisitos por MEFs. . . . .	13
2.4 Conjuntos de casos de teste gerados por método para cada MEF do <i>CM Receiver Software</i> . . . . .	14
2.5 Número de testes obtidos no conjunto total e mínimo por MEF. . . . .	14
2.6 Número de mutantes gerados e Score de Mutação por MEF. . . . .	15
3.1 Lista de entradas do SUT ( <i>Transmitter</i> ). . . . .	17
3.2 Lista de saídas e Ações Observáveis do SUT ( <i>Transmitter</i> ). . . . .	17
3.3 Exceções especificadas. . . . .	19
3.4 Casos de teste gerados pelos métodos da JPlavisFSM para cada MEF do <i>CM Transmitter</i> . . . . .	19
3.5 Número de casos de teste obtido no conjunto total e mínimo por MEF. . . . .	23
3.6 Número de mutantes gerados e Score de Mutação por MEF. . . . .	23
4.1 Resumo dos testes. . . . .	24

## LISTA DE ABREVIATURAS E SIGLAS

ACDH	–	<i>Attitude Control and Data Handing</i>
ACK	–	<i>Positive Acknowledgement</i>
CoFI	–	<i>Conformance and Fault Injection</i>
CM	–	<i>Communication Module</i>
EESC	–	Escola de Engenharia de São Carlos
ICMC	–	Instituto de Ciências Matemáticas e de Computação
INT	–	Interrupção de hardware
ITA	–	Instituto Tecnológico Aeronáutica
MEF	–	Máquina de Estados Finitos
OBC	–	<i>Onboard Computer</i>
SUT	–	<i>System Under Testing</i>
TC	–	Telecomando
TM	–	Telemetria
USP	–	Universidade de São Paulo

## SUMÁRIO

	<u>Pág.</u>
<b>1 INTRODUÇÃO</b> . . . . .	<b>2</b>
<b>2 O Software CM Receiver</b> . . . . .	<b>5</b>
2.1 Identificação . . . . .	5
2.2 Criação dos Modelos Formais . . . . .	7
2.2.1 Geração de Casos de Teste . . . . .	13
<b>3 O Software CM Transmitter</b> . . . . .	<b>16</b>
3.1 Identificação . . . . .	16
3.2 Criação dos Modelos Formais . . . . .	18
3.2.1 Geração de Casos de Teste . . . . .	19
<b>4 CONCLUSÕES</b> . . . . .	<b>24</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> . . . . .	<b>26</b>
<b>APÊNDICE A - Conjunto de testes para o Software CM Receiver</b> .	<b>28</b>
<b>APÊNDICE B - Conjunto de testes para o Software CM Transmitter</b>	<b>34</b>



# 1 INTRODUÇÃO

Este relatório apresenta os resultados obtidos da aplicação da metodologia de teste CoFI (Conformance Test and Fault Injection) (AMBROSIO, 2005; AMBROSIO et al., 2005; AMBROSIO et al., 2006) para a validação do Módulo de Comunicação do satélite universitário ITASAT-1 (SATO et al., 2011).

O Projeto ITASAT (SATO et al., 2011) tem como objetivo a formação de estudantes universitários para a área espacial, em especial no que se refere a satélites, utilizando para isso o projeto, a construção, o lançamento e a operação de um satélite universitário tecnológico com a missão de testar a concepção do projeto em situação real no espaço e de testar cargas úteis compostas também por experimentos. Este projeto está sendo desenvolvido no Instituto Tecnológico de Aeronáutica (ITA), em parceria com o Instituto Nacional de Pesquisas Espaciais (INPE), a Agência Espacial Brasileira (AEB) e as universidades USP (ICMC e EESC), Unicamp, Universidade Estadual de Londrina, Universidade de Brasília, Universidade Federal do Rio Grande do Norte, Faculdade de Engenharia de Garatinguetá (UNESP) e Universidade Técnica de Berlin.

O satélite ITASAT-1 é composto por cinco subsistemas: (1) Estrutura mecânica; (2) Controle Térmico; (3) Suprimento de energia (em inglês, *Energy Power System*, EPS); (4) Controle de Atitude e Supervisão de Bordo (em inglês, *Attitude Control and Data Handling*, ACDH) e (5) Transceptor de Telemetria e de Telecomandos (TMTC). O satélite ainda conta com quatro cargas úteis experimentais (payloads). Na Figura 1.1, é apresentada a arquitetura proposta para o ITASAT-1.

O Módulo de Comunicação, do inglês *Communication Module* (CM), faz parte do Sub-sistema ACDH e é composto por um receptor (*Receiver*), um transmissor (*Transmitter*) e um software responsável pela comunicação.

O software do Módulo de Comunicação (CM) é dividido em duas partes: o CM Receiver Software, que recebe os telecomandos da estação terrestre e os encaminha para processamento (direto ou para o computador de bordo, OBC); e o CM Transmitter Software, que recebe as telemetrias geradas pelo OBC e pelos demais módulos do satélite e os encaminha para o transmissor.

O objetivo principal deste relatório é descrever as atividades de modelagem e geração de testes para o Módulo de Comunicação do satélite ITASAT-1. A modelagem foi feita em modelos de estados de acordo com a metodologia CoFI (AMBROSIO, 2005).

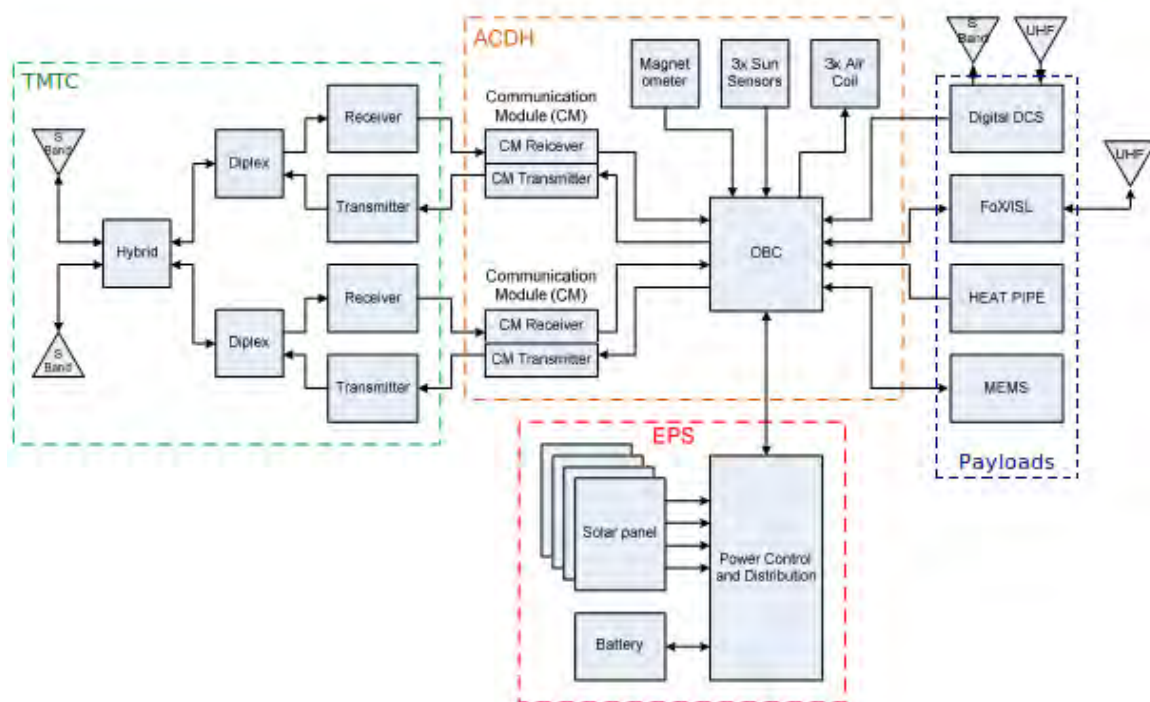


Figura 1.1 - Arquitetura do satélite ITASAT-1 (ITASAT, 2011).

A modelagem é descrita separadamente para ambos os módulos (Receiver e Transmitter) de forma a simplificar o seu entendimento. A ferramenta de teste baseada em Máquinas de Estados Finitos, denominada JPlavisFSM (PINHEIRO, 2012), desenvolvida no mestrado da aluna Arineiza Cristina Pinheiro, é utilizada para auxiliar a criação dos modelos e para gerar automaticamente os casos de teste. A ferramenta JPlavisFSM consta de uma evolução do projeto Plavis descrito em (SIMÃO et al., 2005).

Quatro métodos diferentes foram usados para a geração de casos de teste: W (CHOW, 1978), UIO (VUONG et al., 1989), HSI (PETRENKO et al., 1993) e SPY (SIMÃO et al., 2009). Neste estudo de caso, os melhores conjuntos de casos de teste foram obtidos com os métodos SPY e UIO.

Este relatório está organizado da seguinte forma. No Capítulo 2, são apresentadas as atividades executadas para a modelagem do CM Receiver, bem como os resultados obtidos a partir do processo de geração de casos de testes apoiado pela ferramenta JPlavisFSM. No Capítulo 3, são descritas as atividades de modelagem e a geração de casos de testes para o CM Transmitter. No Apêndice A, são apresentados os conjuntos de testes gerados a partir das MEFs dos módulos CM Receiver e CM

Transmitter usando-se métodos SPY e UIO, respectivamente, sendo que estes conjuntos correspondem aos menores conjuntos gerados pelos métodos disponíveis na JPlavisFSM.

## 2 O Software CM Receiver

Esta seção apresenta os resultados obtidos na atividade de modelagem do CM Receiver Software a partir da utilização da metodologia CoFI. O CM Receiver Software é considerado o sistema alvo de teste, do inglês *System Under Test* (SUT), nesta seção.

### 2.1 Identificação

Na fase de identificação, a metodologia recomenda extrair dos documentos de requisitos do sistema dados como: serviços providos, possíveis falhas de hardware e a lista de entradas e saídas do sistema. A seguir, são apresentados cada um destes itens.

#### a) *Serviços*

Baseado na especificação de requisitos fornecida (ITASAT, 2011) foram identificados os serviços que um usuário reconhece e pode usar do SUT. Os serviços identificados foram:

- Transmitir TC ou TC error log para o OBC (FRq11152000-13)
- Executar comandos do OBC (FRq11152000-13)

Esses serviços podem ser definidos, em resumo, como Comunicação com o OBC.

#### b) *Falhas físicas que podem ocorrer no hardware*

No caso deste projeto, a modelagem envolve apenas as características do software do módulo de comunicação do ITASAT-1, que não inclui o tratamento de falhas de hardware. Desta forma, não existem falhas de hardware a serem consideradas neste contexto.

#### c) *Lista de Entradas do SUT*

Os eventos de entrada que podem ser percebidos pelo SUT e que deverão ser exercitados nos casos de testes são listados na Tabela 2.1.

#### d) *Lista de saídas dos SUT*

Os eventos de saída que podem ser percebidos pelo usuário e que deverão ser verificados ao final da execução dos casos de testes são listados na Tabela 2.2.

Tabela 2.1 - Lista de entradas do SUT (*CM Receiver*).

ENTRADAS	DESCRIÇÃO
ACKOBC	ACK enviado pelo OBC
NPCS pin	Requisição do módulo para comunicação com o OBC
FimTimerC	Indicação que o intervalo de tempo de TC segundos expirou. Este tempo corresponde à espera pela requisição do OBC
Stored TC	Comando que indica que existe TC armazenados a serem enviados
CmdOBCOK	Comando que indica que existe comando do OBC a ser executado
FimTimerA	Indicação que o intervalo de tempo de TA segundos expirou. Este tempo corresponde à espera pelo recebimento do ACK do OBC
CmdOBCNOK	Comando recebido está inconsistente
INT1SurvFlag	Interrupção gerada pelo CM Transmitter
INT2AddrNOK	Interrupção - TC com endereço incompatível com o do módulo
INT2EDACNOK	Interrupção - TC com EDAC inconsistente
INT2DirectTC	Interrupção - TC direto
INT2TC	Interrupção - TC não direto
INT3UART	Interrupção gerada pela UART

É importante observar que para elaborar as MEFs de forma a viabilizar a geração de casos de testes a partir dos métodos incorporados na JPlavisFSM, foi necessária a inclusão de abstrações nos possíveis eventos de saídas. Desta forma, algumas saídas foram compostas e outras derivadas em mais de uma, adicionando o contexto em que ocorrem. Um exemplo é a saída *StoreTC1Wait*, que foi derivado a partir do evento *StoreTC*, quando ele ocorre durante a fase de armazenamento dos TC e a espera pelo início de seu envio. Uma outra abstração necessária foi a abstração do buffer que armazena os TC, sendo nesta MEF modelado com tamanho 3.

Tabela 2.2 - Lista de saídas e Ações Observáveis do SUT (*CM Receiver*).

SAÍDAS	DESCRIÇÃO
ACK (sent via SPI)	Saída gerada para confirmar requisição da CPU
DisparaTimerC	Ação que dispara um contador de TC segundos
DisparaTimerA	Ação que dispara um contador de TA segundos (ACK)
Terminate	Quando expira o tempo de espera de 'TA' segundos
NoTCmsg	Saída gerada para indicar o fim da comunicação com a OBC
Execute	Ação que indica a execução do TC
ACK(2)	Saída gerada para confirmar execução do comando do OBC
OFFdevices	Saída gerada para desligar o OBC e os payloads
DiscardTC	Saída gerada que descarta o TC enviado com endereço diferente do módulo
LogERROR	Saída que gera log de erro para TC com EDAC inconsistente
Execute	Saída que solicita a execução do TC direto
StoreTC	Ação que solicita armazenamento do TC no buffer
BeaconON	Ação que indica para ligar o Modo Beacon
UmbTestRoutine	Ação que solicita execução da rotina de teste umbilical

## 2.2 Criação dos Modelos Formais

Para cada serviço foram definidos modelos parciais para o comportamento (a) Normal e o comportamento frente às (b) Exceções especificadas do Software CM Receiver.

*Serviço 1 (S1): Comunicação com o OBC*

### a) Normal

O modo de execução normal do CM Receiver considera a ocorrência da interrupção de Hardware gerada pelo TMTC receiver (INT2), uma vez que o envio de um TC depende do seu recebimento prévio. A interrupção gerada pela Survival Flag (INT1) também foi considerada no modelo normal. Na Figura 2.1, pode-se observar a máquina de estados finitos que modela o comportamento normal do CM Receiver. O estado inicial é o Wait0, indicado na figura com um pequeno triângulo.

### b) Exceções especificadas

As exceções identificadas no documento ITASAT-1: ACDH SOFTWARE SPECIFICATION (ITASAT, 2011) e que são modeladas são apresentadas na Tabela 2.3. Na primeira coluna da Tabela 2.3, são enumeradas as exceções; na segunda coluna, é apresentado o código da referência de acordo com o documento de especificação; na terceira coluna, encontra-se a descrição do requisito de forma a esclarecer qual a exceção modelada; na quarta coluna, é indicado qual o acrônimo que identifica a Máquina de Estados Finitos (MEF) na qual a exceção descrita foi modelada.

As exceções especificadas pelos requisitos Frq11152000-12 e Frq11152000-16 tratam de eventos de *timeout*; desta forma, foram modeladas em apenas uma MEF, a (S1\_Ex13). O modelo final pode ser observado na Figura 2.2. Na Figura 2.3, observa-se a máquina S1\_Ex2, que modela a exceção em que é finalizada a execução caso o comando do OBC não esteja correto. O modelo S1\_Ex4, apresentado na Figura 2.4, descreve o comportamento do requisito Frq11152000-27 (endereçamento incompatível com o CM Receiver). Por fim, na Figura 2.5, é exibido o modelo S1\_Ex5 que apresenta o comportamento do requisito Frq11152000-29 (*Log Error*).



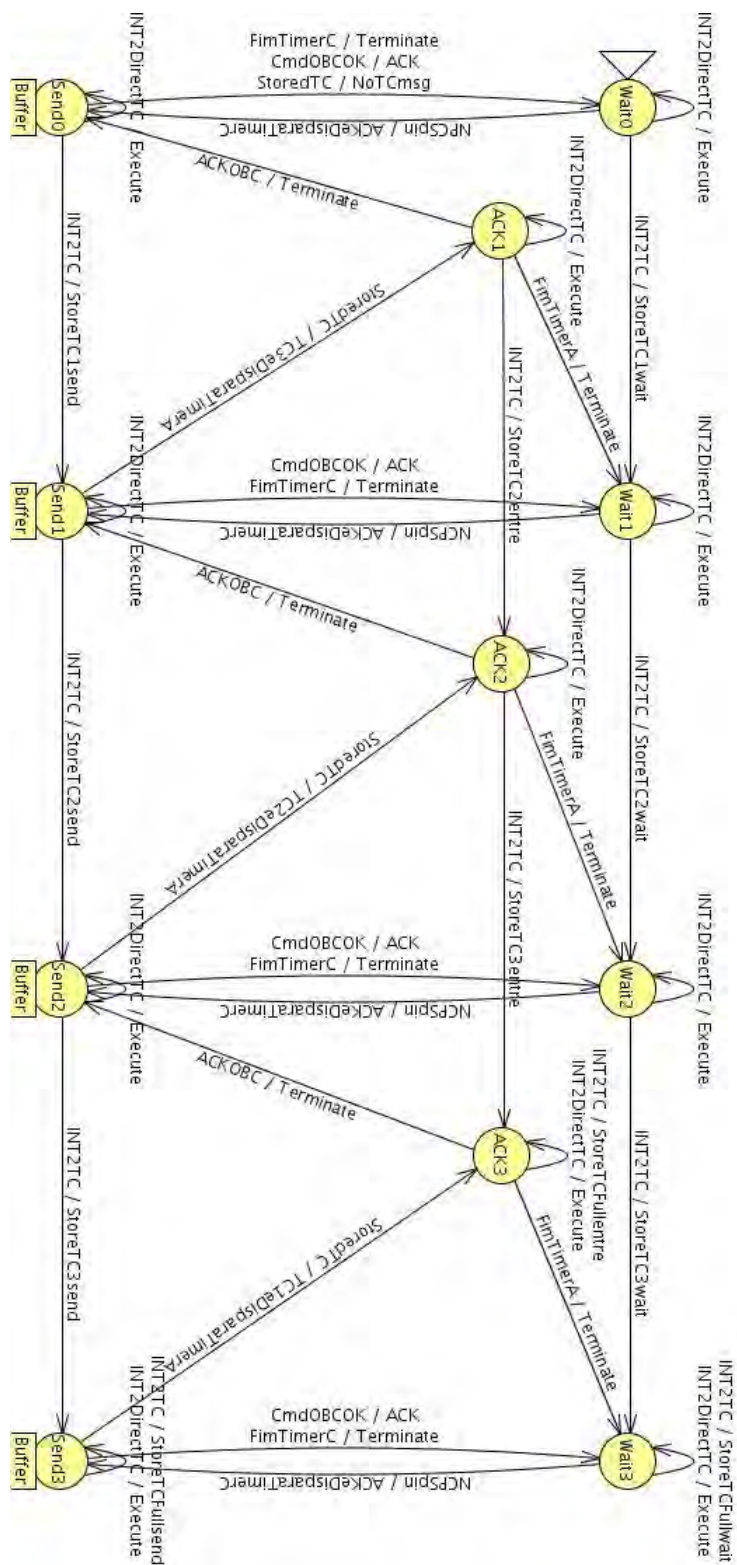


Figura 2.2 - S1\_Ex13: Máquina de estado finitos para exceções descritas nos itens 1 e 3 da Tabela 2.3.



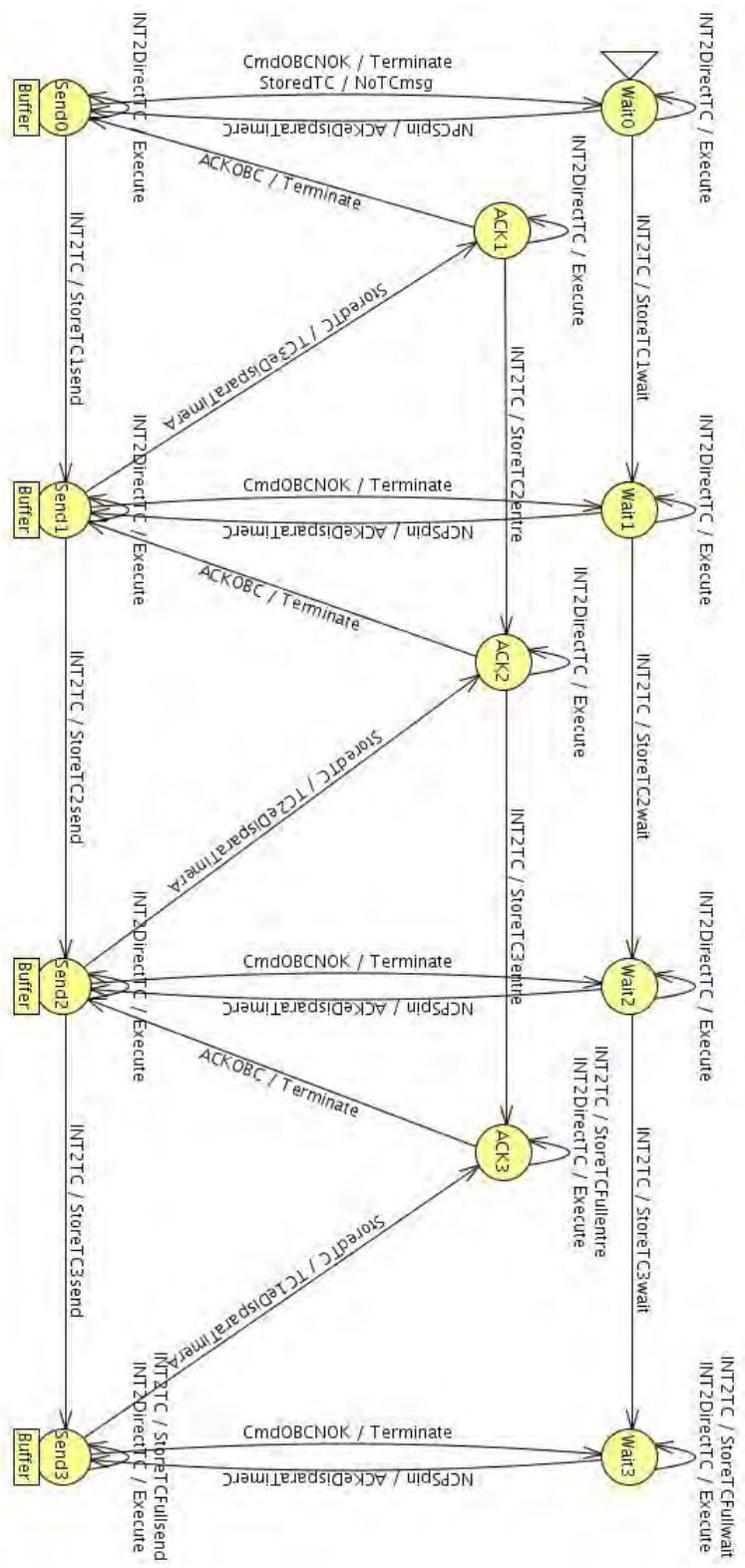


Figura 2.3 - S1\_Ex2: Máquina de estados finitos para exceção descrita no item 2 da Tabela 2.3.



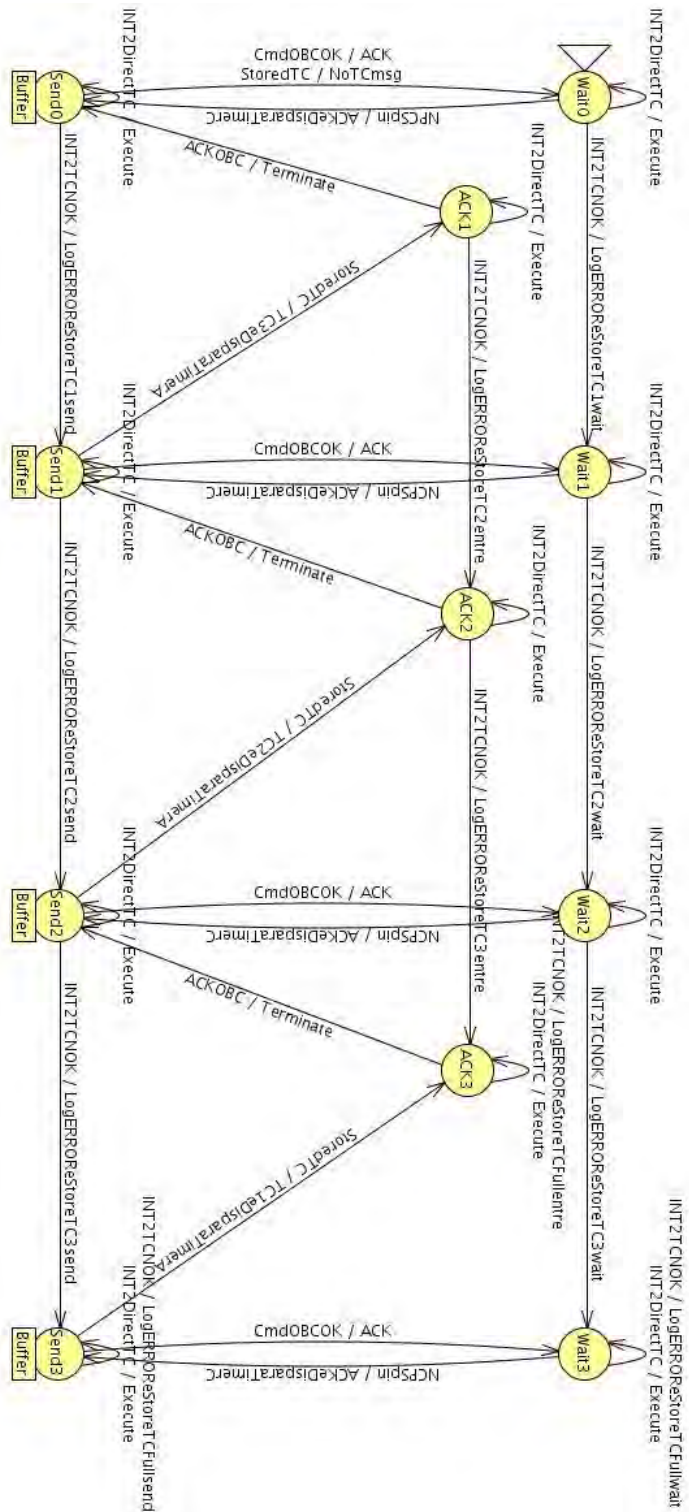


Figura 2.5 - S1\_Ex5: Máquina de estados finitos para exceção descrita no item 5 da Tabela 2.3.

Tabela 2.3 - Relação Requisitos por MEFs.

<b>Exceções especificadas</b>			
<b>#</b>	<b>Requisito</b>	<b>Descrição</b>	<b>Modelo (MEF)</b>
1	FRq11152000-12	“When the OBC does not send a request in TC seconds, the CM receiver finalizes the communication routine”	S1_Ex13
2	FRq11152000-13	“The request of the OBC shall be one of the following: the transmission of TC or TC error log from CM receiver to OBC, or the execution of a (correct) command from OBC.” Obs: Terminar a execução caso o comando não esteja OK.	S1_Ex2
3	FRq11152000-16	“If the CPU does not send the acknowledge in TA seconds (value TBD), the CM receiver shall terminate the communication routine.”	S1_Ex13
4	FRq11152000-27	“The CM receiver shall check the first 16 bits, used for synchronization and packet addressing, and shall discard the TC packet if its address data does not match with the module address.”	S1_Ex4
5	FRq11152000-29	“If the verification described in Frq11152000-28 (The CM receiver module shall check the mode, the repeated data and EDAC field ) results in failure, the CM receiver shall store a TC error log ( as specified in [AD3]) in the TC buffer”	S1_Ex5

### 2.2.1 Geração de Casos de Teste

Para a fase de geração de casos de testes, foi utilizada a ferramenta JPlavisFSM, desenvolvida no trabalho de mestrado de Arineiza Cristina Pinheiro (PINHEIRO, 2012). A ferramenta provê uma interface para desenvolvimento de MEF e uma área dedicada para criação e manipulação de sessões de teste, que disponibiliza quatro métodos de geração baseados em MEFs, sendo: W (CHOW, 1978), UIO (VUONG et al., 1989), HSI (PETRENKO et al., 1993) e SPY (SIMÃO et al., 2009).

O Método W é aplicável apenas em MEFs completamente especificadas, i.e., MEFs que tratam todas as entradas pertencentes ao domínio de entrada em todos os seus estados. Caso contrário, as MEFs são ditas parciais. Como as MEFs do CM Receiver Software são parciais, uma alternativa fornecida pela ferramenta JPlavisFSM é a utilização da funcionalidade 'auto-completar' para que seja possível aplicar o Método W em MEFs originalmente parciais. A funcionalidade 'auto-completar' insere transições nos estados parciais (ou seja, aqueles que não possuem transições previstas para todas as possíveis entradas) com origem e destino no próprio estado para cada uma das entradas não especificadas, inserindo um evento de saída nulo (ou

vazio), para simbolizar que a transição não gera nenhum efeito interno ou externo para a MEF.

Para as MEFs do CM Receiver Software foram gerados casos de testes com os Métodos UIO, HSI e SPY originais e com o Método W para as MEFs pré-processadas pelo ‘auto-completar’. O conjunto assim gerado é representado por W\*, na Tabela 2.4. Foram gerados também conjuntos de teste com o recurso ‘auto-completar’ para os métodos HSI e SPY para efeito de compação (HSI\* e SPY\*). O número de casos de testes gerados a partir de cada uma das MEFs pelos diversos métodos, implementados na JPlavisFSM é apresentado na Tabela 2.4.

Tabela 2.4 - Conjuntos de casos de teste gerados por método para cada MEF do *CM Receiver Software*.

MEF	MÉTODOS						Total
	W*	UIO	HSI	HSI*	SPY	SPY*	
S1_N	234	48	38	78	22	40	460
S1_Ex13	356	44	34	89	26	42	591
S1_Ex2	201	37	27	67	21	40	393
S1_EX4	201	37	27	67	21	40	393
S1_Ex5	201	37	27	67	19	37	388
Total	1193	203	153	368	109	199	2225

A ferramenta JPlavisFSM disponibiliza também a opção de gerar um conjunto mínimo de testes, composto pela união de todos os conjuntos de testes gerados por cada um dos métodos. O conjunto final elimina as sequências de teste idênticas e as que correspondem a prefixo <sup>1</sup> de outras. Na Tabela 2.5, é exibido o número total de casos de teste gerado para cada uma das MEFs, composto pela soma dos casos de teste gerados por cada método e o respectivo número mínimo de casos de teste.

Tabela 2.5 - Número de testes obtidos no conjunto total e mínimo por MEF.

MEF	Conjunto Total	Conjunto Mínimo
S1_N	342	285
S1_Ex13	460	408
S1_Ex2	286	264
S1_EX4	286	263
S1_Ex5	284	265

<sup>1</sup>Dadas duas sequências  $\alpha$  e  $\beta$ ,  $\alpha$  é dita prefixo de  $\beta$ , denotado  $\alpha = \beta$ , se  $\beta = \alpha.\omega$ , para algum  $\omega$ .

A ferramenta JPlavisFSM disponibiliza o Critério de Mutação (FABBRI et al., 1999) como meio de avaliação de um dado conjunto de casos de teste. O Critério de Mutação, através de Operadores de Mutação, inserem pequenas mudanças sintáticas no artefato de teste (a MEF). Os artefatos com esses defeitos são chamados de Mutantes e o objetivo do critério é 'matar' todos os mutantes, ou seja, revelar tais mudanças. A JPlavisFSM executa um conjunto de casos de teste contra os mutantes a fim de realizar esta avaliação. A avaliação final é feita por meio do cálculo do Score de Mutação, que para MEFs é dado por:

$$\text{SCORE} = \text{Número de mutantes mortos} / \text{Número total de mutantes}$$

Para as MEFs do CM Receiver Software, foi aplicado o Critério de Mutação e os resultados obtidos são descritos na Tabela 2.6, em que são apresentados número de mutantes gerados e o seu respectivo score de mutação alcançado pelo conjunto de casos de teste formado pela união de todos os casos de testes gerados a partir dos métodos W, HSI, UIO e SPY da JPlavisFSM.

Tabela 2.6 - Número de mutantes gerados e Score de Mutação por MEF.

MEF	Mutantes	Score de Mutação
S1_N	1852	1.0
S1_Ex13	1720	1.0
S1_Ex2	1320	1.0
S1_EX4	1357	1.0
S1_Ex5	1357	1.0

Como estratégia de teste, pode-se adotar a aplicação do conjunto mínimo de casos de teste gerado pela ferramenta JPlavisFSM ou escolher um conjunto gerado por um método particular. Para as MEFs do CM Receiver, o Método SPY gerou o menor conjunto de testes e ainda sim possui score de mutação igual a 1.0. A princípio pode-se determinar a aplicação do menor conjunto e, posteriormente, de acordo com a necessidade e o tempo disponibilizado para a fase de execução dos testes, aplicar os demais conjuntos eliminando os testes redundantes, de forma a explorar uma maior combinação de teste.

### 3 O Software CM Transmitter

Esta seção apresenta os resultados obtidos na atividade de modelagem do CM Transmitter Software a partir da utilização da metodologia CoFI. Nesta seção, o sistema em teste, do inglês *System Under Test* (SUT) é o CM Transmitter Software.

#### 3.1 Identificação

Nesta fase de identificação foram levantados os serviços, as falhas de hardware e a lista de entradas e saídas possíveis no sistema, sob o ponto de vista do CM Transmitter.

a) *Serviços*

Baseado na especificação fornecida, os serviços que um usuário reconhece e pode usar do SUT são:

- Transmitir Pacotes para o Transciever (Serviço 1)
- Transmitir Pacotes para o Umbilical (Test Routine; Serviço 2)

b) *Falhas físicas que podem ocorrer no hardware*

No caso deste projeto, a modelagem envolve apenas as características do software do Módulo de Comunicação do ITASAT-1. Desta forma, não são consideradas as falhas de hardware neste contexto.

c) *Lista de Entradas do SUT*

Os eventos de entrada que podem ser percebidos pelo SUT e que deverão ser exercitados nos casos de testes são listados na Tabela 3.1.

d) *Lista de saídas dos SUT*

Os eventos de saída que podem ser percebidos pelo usuário e que deverão ser verificados ao final dos casos de testes são listados na Tabela 3.2.

Para o CM Transmitter, foi necessário definir um indicador de saída vazia (EmptyOut) uma vez que era necessário acompanhar o comportamento da comunicação com o Transciever/Umbilical em ordens diferentes de entrada com resultados distintos na saída. Desta forma, a saída vazia foi inserida artificialmente para que fosse possível avaliar o caminho percorrido durante os testes.

Tabela 3.1 - Lista de entradas do SUT (*Transmitter*).

ENTRADAS	DESCRIÇÃO
P1.6 pin ON	Entrada que identifica a conexão com a Umbilical
P1.6 pin OFF	Entrada que indica a comunicação normal com o Transceiver
EndTimerB	Indicação que o intervalo de tempo de TB segundos expirou. Este tempo corresponde à intervalos para 'beacon telemetries'
EndTimerV	Indicação que o intervalo de tempo de TV segundos expirou. Este tempo é utilizado para determinar a verificação de canais analógicos
FimTimerBeDBs	Indicação que o intervalo de tempo de TB+DB segundos expirou. Este tempo corresponde aos intervalos para mandar pacotes de TM
BeaconON	Quando o Beacon flag está ligado
BeaconOFF	Quando o Beacon flag está desligado
OBCONReqNotFromCPU	OBC ligado e requisição não vem da CPU
OBCONRequestFromCPU	OBC ligado e requisição vem da CPU
OBCOFF	OBC desligado
TMConfig	TM está configurado
TMnotConfig	TM não configurado
AnChVerNOK	Analog Channels Verification NOK
AnChVerOK	Analog Channels Verification OK
PackSurvTM	Indicação de pacote Survival TM pronto para envio
TM	Telemetria a ser enviada
CmdOBC	Comando da OBC a ser executado
CmdConfigFromOBC	Comando da OBC para realizar configuração no Transceiver
INT3UART	Interrupção gerada pela UART

Tabela 3.2 - Lista de saídas e Ações Observáveis do SUT (*Transmitter*).

ENTRADAS	DESCRIÇÃO
StartTimerB	Ação que dispara um contador de TB segundos (beacon telemetries)
StartTimerV	Ação que dispara um contador de TV segundos (analog channel verification)
ConfigTMTCTransm	Inficação da necessidade de realizar a configuração do TMTTC Transmitter
SurvFlag	Ação que dispara um pulso na Survival Flag
AnChDtCollect	Indicação do início da Analog channels data collection
SendToTransc/SendToUmb	Envio de Pacote para o Transceiver (Serviço 1) ou o Umbilical (Serviço 2)
ACK	ACK de confirmação de resposta do OBC
SendTMeACK	Envia Telemetria e ACK de confirmação para a CPU
SendDtToCPU	Envia dados de confirguração para CPU
TranscConfig	Indicação de que foi realizado a configuração do Transceiver
TranscConfigUART	Transceiver Configuration (Recepção de dados)
Empty	Saída vazia



## 3.2 Criação dos Modelos Formais

Para cada serviço foram definidos modelos parciais para o comportamento (a) Normal e o comportamento frente às (b) Exceções especificadas do Software CM Transmitter.

Nas duas primeiras linhas da Tabela 3.1, são descritas duas entradas do SUT que são referentes à fase de pré-configuração do software para iniciar a transmissão. Estas entradas apenas distinguem o comportamento da máquina de estados quanto ao destinatário do envio de pacotes de *Survival TM*, que indica que o software continua ativo. Optou-se por omitir a verificação das entradas e assumir dois modelos distintos para o envio dos pacotes.

Dada a semelhança entre os dois serviços do CM Transmitter, o procedimento de modelagem de cada serviço segue os mesmos passos, diferindo apenas no destinatário a receber o pacote de *Survival TM*. Por esse motivo, a descrição e os argumentos de cada subitem a seguir são detalhados apenas no Serviço 1.

*Serviço 1 (S1) – Transmitir Pacotes para o Transciever*

### a) Normal

O modo de execução normal do CM Transmitter inclui: (1) a verificação dos canais analógicos (Analog Channels Verification); (2) envio dos pacotes de *Survival TM* para o Transciever (no Serviço 2 vai para o Umbilical); e (3) a comunicação com o OBC, além da possibilidade de ocorrência da interrupção de Hardware gerada pelo UART (INT3). Na Figura 3.1, pode-se observar a máquina de estados finitos que modela o comportamento normal do CM Transmitter. Obs.: A INT3 pode ocorrer em qualquer um dos estados modelados. Por clareza do modelo, optou-se por não inserir as transições correspondentes.

### b) Exceções especificadas

A seguir, são descritas as exceções identificadas no documento ITASAT-1: ACDH SOFTWARE SPECIFICATION (ITASAT, 2011), que serão tratadas neste teste são apresentadas na Tabela 3.3. Na primeira coluna da Tabela 3.3, são enumeradas as exceções; na segunda coluna, é apresentado o código da referência de acordo com o documento de especificação; na terceira coluna, encontra-se a descrição do requisito de forma a esclarecer qual a exceção modelada; na quarta coluna, é indicado qual o acrônimo que

identifica a Máquina de Estados Finitos (MEF) na qual a exceção descrita foi modelada.

Tabela 3.3 - Exceções especificadas.

Exceções especificadas para o Transmitter			
#	Requisito	Descrição	Modelo (MEF)
1	FRq11152000-52	“When any analog channel verified present unexpected results, the CM Transmitter shall generate a pulse in the survival flag pin (which is read by the CM Receiver).”	S1_Ex1
2	FRq11152000-58	“When the OBC does not send a request in TC seconds, the CM Transmitter finalizes the communication routine.”	S1_Ex2

Na Figura 3.2, observa-se a máquina S1\_Ex1, que modela a exceção responsável pela geração da *survival flag*. O modelo S1\_Ex2, apresentado na Figura 3.3, descreve o comportamento do requisito Frq11152000-58, que determina a finalização da rotina de comunicação quando nenhum TC é recebido.

### 3.2.1 Geração de Casos de Teste

Análogo ao descrito na Seção 2.2.1, para a fase de geração de casos de testes, foi utilizada a ferramenta JPlavisFSM. Neste caso, a funcionalidade 'auto-completar' foi aplicada também aos Métodos HSI e SPY devido a inexistência de sequências de separação necessárias para a execução dos métodos.

Para as MEFs do CM Transmitter Software foram gerados casos de testes com o Método UIO original e com os Métodos W, HSI e SPY para as MEFs pré-processadas pelo 'auto-completar' (simbolizado por \*). O número de casos de testes gerados para as MEFs por métodos é apresentado na Tabela 3.4.

Tabela 3.4 - Casos de teste gerados pelos métodos da JPlavisFSM para cada MEF do *CM Transmitter*.

MEF	MÉTODOS				
	W*	UIO	HSI*	SPY*	Total
S1_N	316	21	184	72	593
S1_Ex1	316	21	184	72	593
S1_Ex2	335	19	158	61	573
Total	967	61	526	205	1759

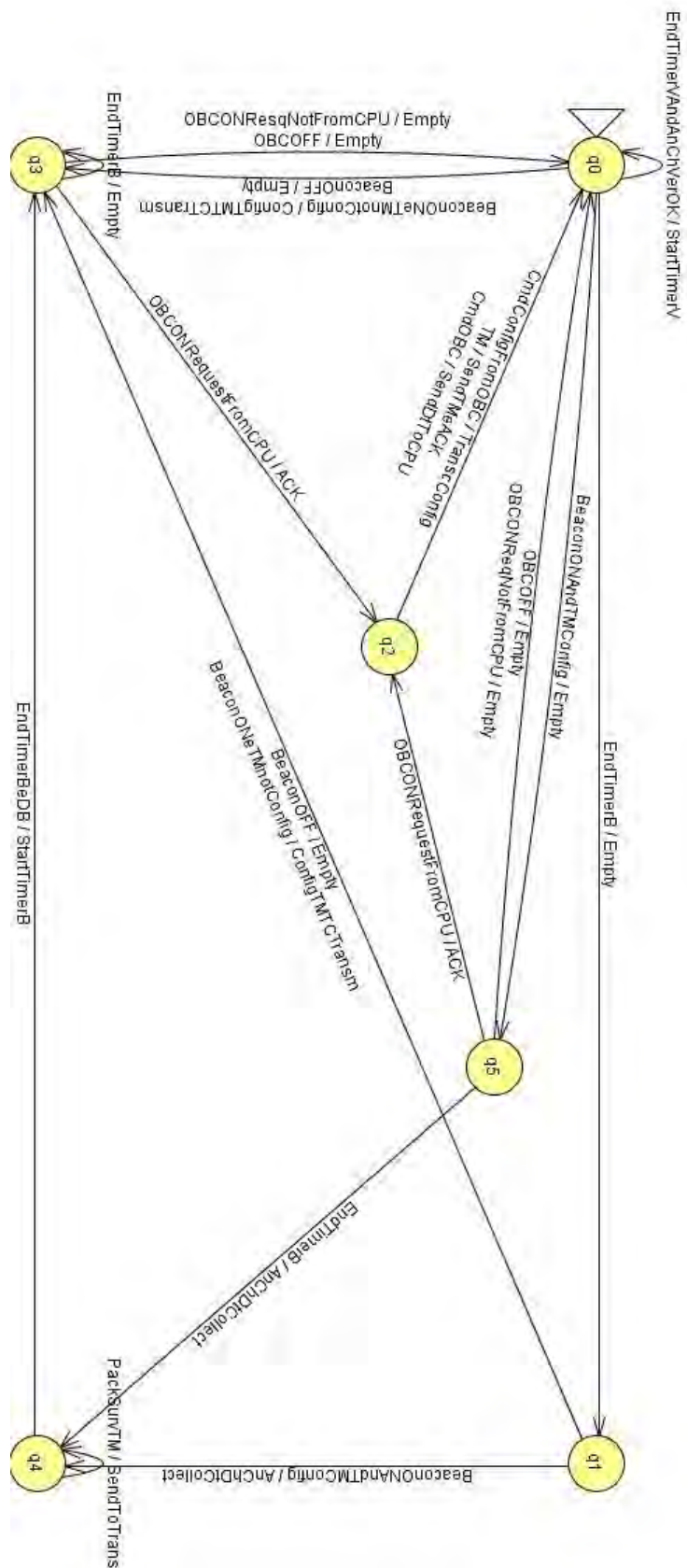


Figura 3.1 - S1\_Normal: Máquina de estados finitos do modo de operação normal do CM Receiver.

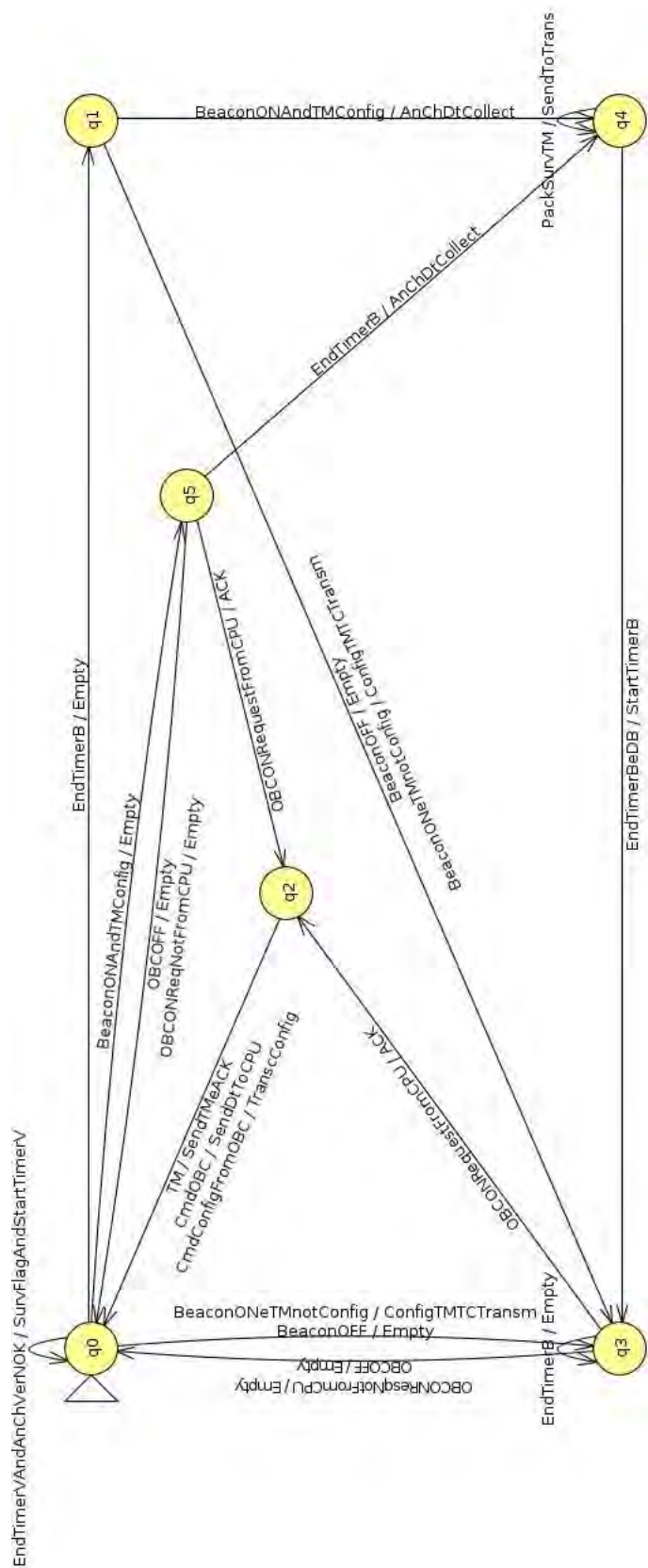


Figura 3.2 - S1\_Ex13: Máquina de estado finitos para exceções descritas no item 1 da Tabela 3.3.

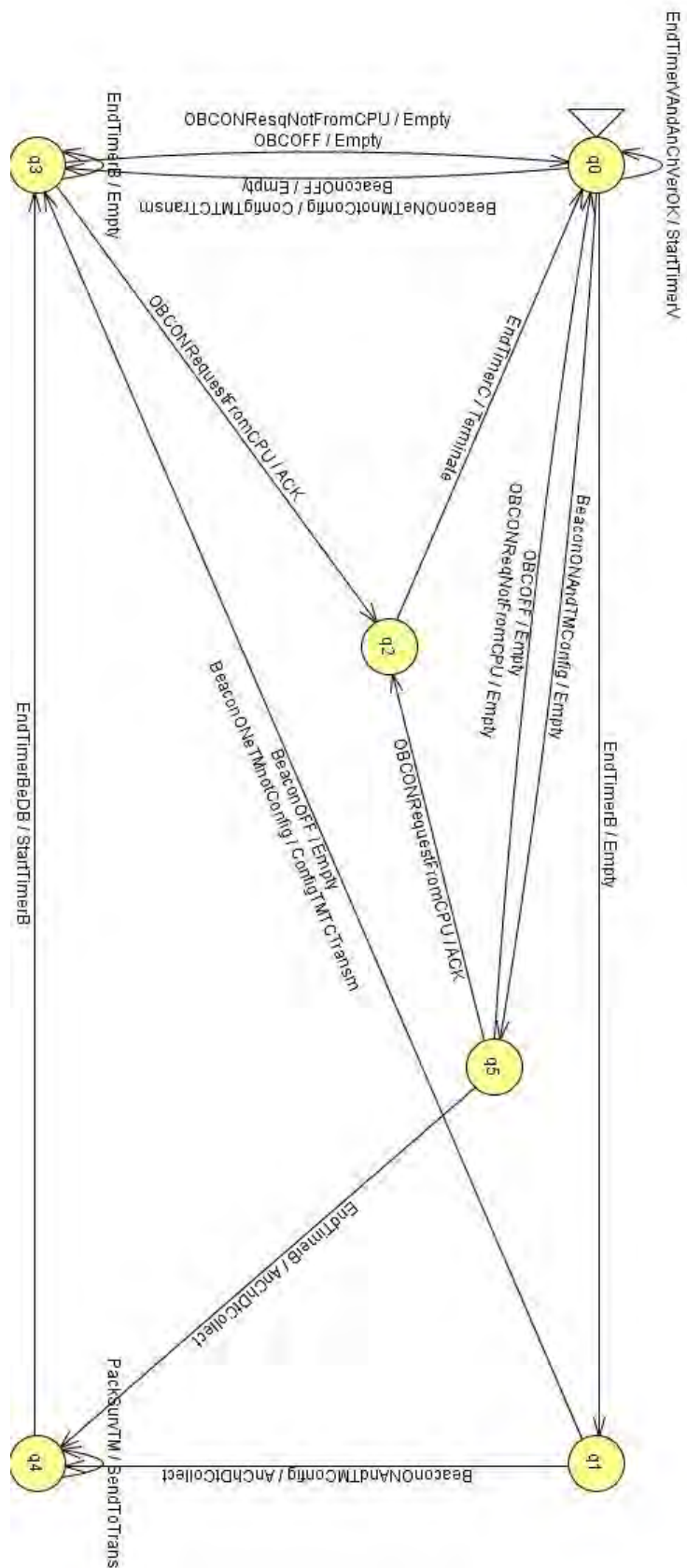


Figura 3.3 - S1\_Ex2: Máquina de estados finitos para exceção descrita no item 2 da Tabela 3.3.

Na Tabela 3.5, são exibidos o número total de teste gerado para cada uma das MEFs, composto pela soma dos testes gerados por cada método e o respectivo número mínimo de teste.

Tabela 3.5 - Número de casos de teste obtido no conjunto total e mínimo por MEF.

<b>MEF</b>	<b>Conjunto Total</b>	<b>Conjunto Mínimo</b>
S1_N	593	432
S1_Ex1	593	432
S1_Ex2	573	454

Quanto ao Critério de Mutação, os resultados obtidos para as MEFs do CM Transmitter Software são descritos na Tabela 3.6, em que são apresentados o número de mutantes gerados e o escore de mutação alcançado pelo conjunto de casos de teste formado pela união de todos os casos de teste gerados a partir de todos os métodos de geração implementados na JPlavisFSM.

Tabela 3.6 - Número de mutantes gerados e Score de Mutação por MEF.

<b>MEF</b>	<b>Mutantes</b>	<b>Score de Mutação</b>
S1_N	593	1.0
S1_Ex1	593	1.0
S1_Ex2	573	1.0

Como estratégia de teste, pode-se adotar a aplicação do conjunto mínimo de testes gerado pela ferramenta JPlavisFSM ou escolher um dos métodos, como por exemplo o Método UIO, que gerou o menor conjunto de testes e ainda possui score de mutação igual a 1.0. A princípio pode-se determinar a aplicação do menor conjunto e, posteriormente, de acordo com a necessidade e o tempo disponibilizado para a fase de testes, aplicar os demais conjuntos eliminando os testes redundantes, de forma a explorar uma maior combinação de teste.

## 4 CONCLUSÕES

O Método W, apesar de ser o mais tradicional dentre os métodos presentes na literatura, pode ser aplicado apenas em MEFs completamente especificadas, o que restringe a possibilidade da sua aplicação prática. Os demais métodos puderam ser aplicados na maioria das MEFs, exceto nas MEFs do *CM Transmitter* que não possuem sequências de separação e, conseqüentemente, não permitem a execução dos métodos HSI e SPY originais (sem a utilização do ‘auto-completar’).

A JPlavisFSM permite o cálculo do conjunto mínimo de teste composto pela combinação dos conjuntos gerados por todos os métodos. A ferramenta elimina todas sequências idênticas ou que sejam prefixos de outras. Na Tabela 4.1, é possível observar o número total de testes representado pela soma dos testes gerados por todos os métodos da ferramenta, o conjunto mínimo calculado, o número de mutantes para cada uma das MEFs e o seu respectivo escore de mutação atingido.

Tabela 4.1 - Resumo dos testes.

MEF	Conjunto Total	Conjunto Mínimo	Mutantes	Escore de Mutaç�o
R_N	342	285	1852	1.0
R_Ex1	460	408	1720	1.0
R_Ex2	286	264	1320	1.0
R_Ex3	286	263	1357	1.0
R_Ex4	284	265	1357	1.0
T_N	593	432	775	1.0
T_Ex1	593	432	775	1.0
T_Ex2	573	454	591	1.0

O n mero de testes gerados automaticamente a partir de uma MEF   consideravelmente grande; desta forma, cabe ao testador decidir, baseado no objetivo dos testes, qual a metodologia mais conveniente e qual m todo de gera o satisfar  o objetivo dos testes. Por exemplo, o testador pode escolher executar o conjunto m nimo de testes ou um conjunto gerado apenas por um dos m todos da ferramenta. A an lise do escore de muta o atingido pode ser usada como um indicador da adequa o do conjunto ao objetivo de teste definido. Ser  responsabilidade da equipe de testes do Software do M dulo de Comunica o determinar quanto o software deve ser testado.

Para a viabilização da modelagem baseada em MEFs, muitos detalhes do software foram abstraídos. Ao final do processo proposto pela Metodologia CoFI, obtém-se um conjunto ‘abstrato’ de teste que devem passar ainda por uma fase de pós-processamento para se tornar executável. Algumas entradas devem ser substituídas por um conjunto de entradas reais, como a entrada ‘*INT2DirectTC*’ que corresponde a 23 diferentes telecomandos que devem ser tratados pelo módulo de comunicação. A geração de casos de testes é automática, mas o pós-processamento e a execução dos testes não é uma atividade trivial.

Este trabalho permitiu identificar que a fase de modelagem para geração de casos de teste não é simples de ser desenvolvida; porém, com apoio da metodologia CoFI e da ferramenta JPlavisFSM pode-se reduzir o tempo para realização desta atividade, uma vez que foi traçada uma estratégia inicial para a modelagem. Observou-se que existe a necessidade de uma fase adicional após a geração dos casos de teste, para o pós-processamento da informação gerada, de forma a traduzir os casos de testes abstratos gerados pela JPlavisFSM para casos de testes concretos. A implementação de um *script* de teste pode ser útil para automatizar a fase de execução e análise dos resultados ao final da atividade de teste.



## REFERÊNCIAS BIBLIOGRÁFICAS

- AMBROSIO, A. M. **CoFI - uma abordagem combinando teste de conformidade e injeção de falhas para validação de software em aplicações espaciais**. [S.l.: s.n.], 2005. 2
- AMBROSIO, A. M.; MARTINS, E.; VIJAYKUMAR, N. L.; CARVALHO, S. A methodology for designing fault injection experiments as an addition to communication systems conformance testing. In: **Proceedings of the First Workshop on Dependable Software - Tools and Methods**. Yokohama, Japan: [s.n.], 2005. 2
- AMBROSIO, A. M.; MARTINS, E.; VIJAYKUMAR, N. L.; CARVALHO, S. V. d. A conformance testing process for space applications software services. **Journal of Aerospace Computing, Information, and Communication**, v. 3, n. 4, p. 146–158, Apr. 2006. ISSN 1542-9423. 2
- CHOW, T. S. Testing software design modeled by finite-state machines. **IEEE Transactions on Software Engineering**, IEEE Press, Piscataway, NJ, USA, v. 4, p. 178 – 187, 1978. ISSN 0098-5589. 3, 13
- FABBRI, S. C. P. F.; MALDONADO, J. C.; MASIERO, P. C.; DELAMARO, M. E. Proteum/FSM: A tool to support finite state machine validation based on mutation testing. In: **Proceedings of the 19<sup>th</sup> International Conference of the Chilean Computer Science Society**. Washington, DC, USA: IEEE Computer Society, 1999. p. 96 – 104. 15
- ITASAT, P. **ITASAT-1: ACDH Software Specification**. 2011. Disponível em: <<http://www.itasat.ita.br/index.php/pt-br/institucional/missao>>. ii, 3, 5, 7, 18
- PETRENKO, A.; YEVTUSHENKO, N.; LEBEDEV, A.; DAS, A. Nondeterministic state machines in protocol conformance testing. In: **Proceedings of the 6<sup>th</sup> International Workshop on Protocol Test systems VI (IFIP TC6/WG6.1)**. Amsterdam, The Netherlands, The Netherlands: North-Holland Publishing Co., 1993. p. 363–378. 3, 13
- PINHEIRO, A. **Subsídios para a Aplicação de Métodos de Geração de Casos de Testes Baseados em Máquinas de Estados Finitos**. Dissertação (Mestrado) — Universidade de São Paulo, São Carlos, SP, 2012. 3, 13

SATO, L. S.; SAOTOME, O.; TIMM, C.; FERNADES, D.; YAMAGUTI, W.  
Itasat-1: Brazilian university microsatellite for payload test and validation in low earth orbit. In: **Proceeding of the 8<sup>th</sup> Symposium on Small satellites for Earth Observation**. Berlin, Germany: [s.n.], 2011. 2

SIMÃO, A. d. S.; AMBRÓSIO, A. M.; FABBRI, S. C. P.; AMARAL, A. S. M. S. d.; MARTINS, E.; MALDONADO, J. C. Plavis/fsm: na environment to integrate fsm-based testing tool. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 19., 3-7 out., Uberlândia, MG. **Proceedings...** INPE, 2005. p. 06. Disponível em: <http://urlib.net/sid.inpe.br/iris@1916/2005/11.21.18.33>. Acesso em: 05 set. 2013. 3

SIMÃO, A.; PETRENKO, A.; YEVTUSHENKO, N. Generating reduced tests for fsm's with extra states. In: **Proceedings of the 21<sup>st</sup> IFIP WG 6.1 International Conference on Testing of Software and Communication Systems and 9th International FATES Workshop**. Berlin, Heidelberg: Springer-Verlag, 2009. (TESTCOM '09/FATES '09), p. 129 – 145. 3, 13

VUONG, S.; CHAN, W.; ITO, M. The uiiov-method for protocol test sequence generation. In: **Proceedings of the 2<sup>nd</sup> International Workshop Protocol Test Systems**. Berlin, Germany: [s.n.], 1989. p. 161 – 175. 3, 13

## **APÊNDICE A - Conjunto de testes para o Software CM Receiver**

Este apêndice apresenta os conjuntos de casos de testes obtidos com a execução do Método SPY, quando as MEFs criadas para representar o comportamento do Software CM Receiver, lhe foram submetidas. Estas MEFs são ilustradas na Seção 2. O menor conjunto de casos de teste gerados automaticamente a partir das MEFs foi o método SPY.

As figuras A.1, A.2, A.3, A.4 e A.5 correspondem, respectivamente, as MEFs S1-Normal, S1-Ex13, S1-Ex2, S1-Ex4 e S1-Ex5.

Método SPY – S1_N	
Sequências de Entrada	Sequências de Saída Esperada
INT1SurvFlag INT2TC INT1SurvFlag INT2TC INT1SurvFlag INT2TC INT2DirectTC INT2TC	OFFdeviceBeaconON StoreTC1wait OFFdeviceBeaconON StoreTC2wait OFFdeviceBeaconON StoreTC3wait Execute StoreTCFullwait
INT2DirectTC INT2TC INT2DirectTC INT2TC INT2DirectTC INT2TC	Execute StoreTC1wait Execute StoreTC2wait Execute StoreTC3wait
INT2TC INT2TC INT2TC INT2TC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait StoreTCFullwait StoreTCFullwait
INT2TC INT2TC INT2TC NCPSpin INT2TC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKandStartTimerC StoreTCFullsend StoreTCFullsend
INT2TC INT2TC INT2TC NCPSpin StoredTC INT2TC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKandStartTimerC TC1andStartTimerA StoreTCFullentre StoreTCFullentre
INT2TC INT2TC INT2TC NCPSpin StoredTC ACKOBC INT2TC INT2DirectTC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKandStartTimerC TC1andStartTimerA Terminate StoreTC3send Execute StoreTCFullsend
INT2TC INT2TC INT2TC NCPSpin CmdOBCOK INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKandStartTimerC ACK StoreTCFullwait
INT2TC INT2TC NCPSpin INT2TC INT2TC	StoreTC1wait StoreTC2wait ACKandStartTimerC StoreTC3send StoreTCFullsend
INT2TC INT2TC NCPSpin StoredTC INT2TC INT2TC	StoreTC1wait StoreTC2wait ACKandStartTimerC TC2andStartTimerA StoreTC3entre StoreTCFullentre
INT2TC INT2TC NCPSpin StoredTC ACKOBC INT2TC	StoreTC1wait StoreTC2wait ACKandStartTimerC TC2andStartTimerA Terminate StoreTC2send
INT2TC INT2TC NCPSpin CmdOBCOK INT2TC INT1SurvFlag INT2TC	StoreTC1wait StoreTC2wait ACKandStartTimerC ACK StoreTC3wait OFFdeviceBeaconON StoreTCFullwait
INT2TC NCPSpin INT2TC INT2TC INT1SurvFlag INT2TC	StoreTC1wait ACKandStartTimerC StoreTC2send StoreTC3send OFFdeviceBeaconON StoreTCFullsend
INT2TC NCPSpin StoredTC INT1SurvFlag INT2TC INT1SurvFlag INT2TC INT2DirectTC INT2TC	OFFdeviceBeaconON StoreTC2entre OFFdeviceBeaconON StoreTC3entre Execute StoreTCFullentre
INT2TC NCPSpin StoredTC INT2DirectTC INT2TC INT2DirectTC INT2TC	StoreTC1wait ACKandStartTimerC TC3andStartTimerA Execute StoreTC2entre Execute StoreTC3entre
INT2TC NCPSpin StoredTC INT2TC INT2TC INT1SurvFlag INT2TC	StoreTC2entre StoreTC3entre OFFdeviceBeaconON StoreTCFullentre
INT2TC NCPSpin StoredTC ACKOBC INT2TC	StoreTC1wait ACKandStartTimerC TC3andStartTimerA Terminate StoreTC1send
INT2TC NCPSpin CmdOBCOK INT2TC	StoreTC1wait ACKandStartTimerC ACK StoreTC2wait
NCPSpin INT1SurvFlag INT2TC INT1SurvFlag INT2TC INT2DirectTC INT2TC	ACKandStartTimerC OFFdeviceBeaconON StoreTC1send OFFdeviceBeaconON StoreTC2send Execute StoreTC3send
NCPSpin INT2DirectTC INT2TC INT2DirectTC INT2TC	ACKandStartTimerC Execute StoreTC1send Execute StoreTC2send
NCPSpin INT2TC INT2TC INT1SurvFlag INT2TC	ACKandStartTimerC StoreTC1send StoreTC2send OFFdeviceBeaconON StoreTC3send
NCPSpin StoredTC INT2TC	ACKandStartTimerC NoTMsg StoreTC1wait

Figura A.1 - Conjunto de testes gerado pelo método SPY para a MEF S1\_N do Receiver.

Método SPY – S1_Ex13	
Sequências de Entrada	Sequências de Saída Esperada
INT2DirectTC INT2TC INT2DirectTC INT2TC INT2DirectTC INT2TC INT2TC INT2TC INT2TC INT2TC INT2TC	Execute StoreTC1wait Execute StoreTC2wait Execute StoreTC3wait StoreTC1wait StoreTC2wait StoreTC3wait StoreTCFullwait StoreTCFullwait
INT2TC INT2TC INT2TC NCPSpin INT2TC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKandStartTimerC StoreTCFullsend StoreTCFullsend
INT2TC INT2TC INT2TC NCPSpin StoredTC INT2TC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKandStartTimerC TC1andStartTimerA StoreTCFullentre StoreTCFullentre
INT2TC INT2TC INT2TC NCPSpin StoredTC FimTimerAINT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKandStartTimerC TC1andStartTimerA empty StoreTCFullentre
INT2TC INT2TC INT2TC NCPSpin StoredTC ACKOBC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKandStartTimerC TC1andStartTimerA Terminate StoreTC3send
INT2TC INT2TC INT2TC NCPSpin CmdOBCOK INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKandStartTimerC ACK StoreTCFullwait
INT2TC INT2TC INT2TC NCPSpin FimTimerC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKandStartTimerC empty StoreTCFullsend
INT2TC INT2TC NCPSpin INT2TC INT2TC	StoreTC1wait StoreTC2wait ACKandStartTimerC StoreTC3send StoreTCFullsend
INT2TC INT2TC NCPSpin StoredTC INT2TC INT2TC	StoreTC1wait StoreTC2wait ACKandStartTimerC TC2andStartTimerA StoreTC3entre StoreTCFullentre
INT2TC INT2TC NCPSpin StoredTC FimTimerAINT2TC	StoreTC1wait StoreTC2wait ACKandStartTimerC TC2andStartTimerA empty StoreTC3entre
INT2TC INT2TC NCPSpin StoredTC ACKOBC INT2TC	StoreTC1wait StoreTC2wait ACKandStartTimerC TC2andStartTimerA Terminate StoreTC2send
INT2TC INT2TC NCPSpin CmdOBCOK INT2TC INT2DirectTC INT2TC	StoreTC1wait StoreTC2wait ACKandStartTimerC ACK StoreTC3wait Execute StoreTCFullwait
INT2TC INT2TC NCPSpin FimTimerC INT2TC	StoreTC1wait StoreTC2wait ACKandStartTimerC empty StoreTC3send
INT2TC NCPSpinINT2TC INT2TC INT2DirectTC INT2TC	StoreTC1wait ACKandStartTimerC StoreTC2send StoreTC3send Execute StoreTCFullsend
INT2TC NCPSpin StoredTC INT2DirectTC INT2TC INT2DirectTC INT2TC	StoreTC1wait ACKandStartTimerC TC3andStartTimerA Execute StoreTC2entre Execute StoreTC3entre
INT2TC NCPSpin StoredTC INT2TC INT2TC INT2DirectTC INT2TC	StoreTC1wait ACKandStartTimerC TC3andStartTimerA StoreTC2entre StoreTC3entre Execute StoreTCFullentre
INT2TC NCPSpin StoredTC FimTimerAINT2TC	StoreTC1wait ACKandStartTimerC TC3andStartTimerA empty StoreTC2entre
INT2TC NCPSpin StoredTC ACKOBC INT2TC	StoreTC1wait ACKandStartTimerC TC3andStartTimerA Terminate StoreTC1send
INT2TC NCPSpin CmdOBCOK INT2TC	StoreTC1wait ACKandStartTimerC ACK StoreTC2wait
INT2TC NCPSpin FimTimerC INT2TC	StoreTC1wait ACKandStartTimerC empty StoreTC2send
NPCSpin INT2DirectTC INT2TC INT2DirectTC INT2TC	ACKandStartTimerC Execute StoreTC1send Execute StoreTC2send
NPCSpin INT2TC INT2TC INT2DirectTC INT2TC	ACKandStartTimerC StoreTC1send StoreTC2send Execute StoreTC3send
NPCSpin StoredTC INT2TC	ACKandStartTimerC NotCmsg StoreTC1wait
NPCSpin CmdOBCOK INT2TC	ACKandStartTimerC ACK StoreTC1wait
NPCSpin FimTimerC INT2TC	ACKandStartTimerC empty StoreTC1send

Figura A.2 - Conjunto de testes gerado pelo método SPY para a MEF S1\_Ex13 do Re-  
ceiver.

Método SPY – S1_Ex2	
Sequências de Entrada	Sequências de Saida Esperada
INT2DirectTC INT2TC INT2DirectTC INT2TC INT2DirectTC INT2TC INT2TC INT2TC INT2TC INT2TC INT2TC	Execute StoreTC1wait Execute StoreTC2wait Execute StoreTC3wait StoreTC1wait StoreTC2wait StoreTC3wait StoreTCFullwait StoreTCFullwait
INT2TC INT2TC INT2TC NCPSpin INT2DirectTC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKeDis paraTimerC Execute StoreTCFulls end
INT2TC INT2TC INT2TC NCPSpin INT2TC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKeDis paraTimerC StoreTCFulls end StoreTCFulls end
INT2TC INT2TC INT2TC NCPSpin StoredTC INT2DirectTC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKeDis paraTimerC TC1eDis paraTimerA Execute StoreTCFullentre
INT2TC INT2TC INT2TC NCPSpin StoredTC INT2TC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKeDis paraTimerC TC1eDis paraTimerA StoreTCFullentre StoreTCFullentre
INT2TC INT2TC INT2TC NCPSpin StoredTC ACKOBC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKeDis paraTimerC TC1eDis paraTimerA Terminate StoreTC3s end
INT2TC INT2TC INT2TC NCPSpin CmdOBCNOK INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKeDis paraTimerC Terminate StoreTCFullwait
INT2TC INT2TC NCPSpin INT2TC INT2TC	StoreTC1wait StoreTC2wait ACKeDis paraTimerC StoreTC3s end StoreTCFulls end
INT2TC INT2TC NCPSpin StoredTC INT2TC INT2TC	StoreTC1wait StoreTC2wait ACKeDis paraTimerC TC2eDis paraTimerA StoreTC3entre StoreTCFullentre
INT2TC INT2TC NCPSpin StoredTC ACKOBC INT2TC	StoreTC1wait StoreTC2wait ACKeDis paraTimerC TC2eDis paraTimerA Terminate StoreTC2s end
INT2TC INT2TC NCPSpin CmdOBCNOK INT2TC INT2DirectTC INT2TC	StoreTC1wait StoreTC2wait ACKeDis paraTimerC Terminate StoreTC3wait Execute StoreTCFullwait
INT2TC NCPSpin INT2TC INT2TC	StoreTC1wait ACKeDis paraTimerC StoreTC2s end StoreTC3s end
INT2TC NCPSpin StoredTC INT2DirectTC INT2TC INT2DirectTC INT2TC	StoreTC1wait ACKeDis paraTimerC TC3eDis paraTimerA Execute StoreTC2entre Execute StoreTC3entre
INT2TC NCPSpin StoredTC INT2TC INT2TC	StoreTC1wait ACKeDis paraTimerC TC3eDis paraTimerA StoreTC2entre StoreTC3entre
INT2TC NCPSpin StoredTC ACKOBC INT2TC	StoreTC1wait ACKeDis paraTimerC TC3eDis paraTimerA Terminate StoreTC1s end
INT2TC NCPSpin CmdOBCNOK INT2TC	StoreTC1wait ACKeDis paraTimerC Terminate StoreTC2wait

Figura A.3 - Conjunto de testes gerado pelo método SPY para a MEF S1\_Ex2 do Receiver.

Método SPY – S1_Ex4	
Sequências de Entrada	Sequências de Saída Esperada
INT2AddrNOK INT2TC INT2AddrNOK INT2TC INT2AddrNOK INT2TC	DiscardTC StoreTC1wait DiscardTC StoreTC2wait DiscardTC StoreTC3wait
INT2TC INT2TC INT2TC INT2TC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait StoreTCFullwait StoreTCFullwait
INT2TC INT2TC INT2TC NCPSpin INT2TC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKeDisparaTimerC StoreTCFullsend StoreTCFullsend
INT2TC INT2TC INT2TC NCPSpin StoreTC INT2AddrNOK INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKeDisparaTimerC TC1eDisparaTimerA DiscardTC StoreTCFullentre
INT2TC INT2TC INT2TC NCPSpin StoreTC INT2TC INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKeDisparaTimerC TC1eDisparaTimerA StoreTCFullentre StoreTCFullentre
INT2TC INT2TC INT2TC NCPSpin StoreTC ACKOBC INT2TC INT2AddrNOK INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKeDisparaTimerC TC1eDisparaTimerA Terminate StoreTC3send DiscardTC StoreTCFullsend
INT2TC INT2TC INT2TC NCPSpin CmdOBCOK INT2TC	StoreTC1wait StoreTC2wait StoreTC3wait ACKeDisparaTimerC ACK StoreTCFullwait
INT2TC INT2TC NCPSpin INT2AddrNOK INT2TC	StoreTC1wait StoreTC2wait ACKeDisparaTimerC DiscardTC StoreTC3send
INT2TC INT2TC NCPSpin INT2TC INT2TC	StoreTC1wait StoreTC2wait ACKeDisparaTimerC StoreTC3send StoreTCFullsend
INT2TC INT2TC NCPSpin StoreTC INT2TC INT2TC	StoreTC1wait StoreTC2wait ACKeDisparaTimerC TC2eDisparaTimerA StoreTC3entre StoreTCFullentre
INT2TC INT2TC NCPSpin StoreTC ACKOBC INT2TC	StoreTC1wait StoreTC2wait ACKeDisparaTimerC TC2eDisparaTimerA Terminate StoreTC2send
INT2TC INT2TC NCPSpin CmdOBCOK INT2TC INT2AddrNOK INT2TC	StoreTC1wait StoreTC2wait ACKeDisparaTimerC ACK StoreTC3wait DiscardTC StoreTCFullwait
INT2TC NCPSpin INT2TC INT2TC	StoreTC1wait ACKeDisparaTimerC StoreTC2send StoreTC3send
INT2TC NCPSpin StoreTC INT2AddrNOK INT2TC INT2AddrNOK INT2TC	StoreTC1wait ACKeDisparaTimerC TC3eDisparaTimerA DiscardTC StoreTC2entre DiscardTC StoreTC3entre
INT2TC NCPSpin StoreTC INT2TC INT2TC	StoreTC1wait ACKeDisparaTimerC TC3eDisparaTimerA StoreTC2entre StoreTC3entre
INT2TC NCPSpin StoreTC ACKOBC INT2TC	StoreTC1wait ACKeDisparaTimerC TC3eDisparaTimerA Terminate StoreTC1send
INT2TC NCPSpin CmdOBCOK INT2TC	StoreTC1wait ACKeDisparaTimerC ACK StoreTC2wait
NCPSpin INT2AddrNOK INT2TC INT2AddrNOK INT2TC	ACKeDisparaTimerC DiscardTC StoreTC1send DiscardTC StoreTC2send
NCPSpin INT2TC INT2TC	ACKeDisparaTimerC StoreTC1send StoreTC2send
NCPSpin StoreTC INT2TC	ACKeDisparaTimerC NoTMsg StoreTC1wait
NCPSpin CmdOBCOK INT2TC	ACKeDisparaTimerC ACK StoreTC1wait

Figura A.4 - Conjunto de testes gerado pelo método SPY para a MEF S1\_Ex4 do Receiver.

Método SPY – S1_Ex5	
Sequências de Entrada	Sequências de Saída Esperada
INT2DirectTC INT2TCNOK INT2DirectTC INT2TCNOK INT2DirectTC INT2TCNOK	Execute LogERROReStoreTC1wait Execute LogERROReStoreTC2wait Execute LogERROReStoreTC3wait
INT2TCNOK INT2TCNOK INT2TCNOK INT2TCNOK INT2TCNOK	LogERROReStoreTC1wait LogERROReStoreTC2wait LogERROReStoreTC3wait LogERROReStoreTCFullwait LogERROReStoreTCFullwait
INT2TCNOK INT2TCNOK INT2TCNOK NCP Spin INT2TCNOK INT2TCNOK	LogERROReStoreTC1wait LogERROReStoreTC2wait LogERROReStoreTC3wait ACKeDisparaTimerC LogERROReStoreTCFullsend LogERROReStoreTCFullsend
INT2TCNOK INT2TCNOK INT2TCNOK NCP Spin CmdOBCOK INT2TCNOK	LogERROReStoreTC1wait LogERROReStoreTC2wait LogERROReStoreTC3wait ACKeDisparaTimerC ACK LogERROReStoreTCFullwait
INT2TCNOK INT2TCNOK INT2TCNOK NCP Spin StoredTC INT2TCNOK INT2TCNOK	LogERROReStoreTC1wait LogERROReStoreTC2wait LogERROReStoreTC3wait ACKeDisparaTimerC TC1eDisparaTimerA LogERROReStoreTCFullentre LogERROReStoreTCFullentre
INT2TCNOK INT2TCNOK INT2TCNOK NCP Spin StoredTC ACKOBC INT2TCNOK	LogERROReStoreTC1wait LogERROReStoreTC2wait LogERROReStoreTC3wait ACKeDisparaTimerC TC1eDisparaTimerA Terminate LogERROReStoreTC3send
INT2TCNOK INT2TCNOK NCP Spin INT2TCNOK INT2TCNOK	LogERROReStoreTC1wait LogERROReStoreTC2wait ACKeDisparaTimerC LogERROReStoreTC3send LogERROReStoreTCFullsend
INT2TCNOK INT2TCNOK NCP Spin CmdOBCOK INT2TCNOK INT2DirectTC INT2TCNOK	LogERROReStoreTC1wait LogERROReStoreTC2wait ACKeDisparaTimerC ACK LogERROReStoreTC3wait Execute LogERROReStoreTCFullwait
INT2TCNOK INT2TCNOK NCP Spin StoredTC INT2TCNOK INT2TCNOK	LogERROReStoreTC1wait LogERROReStoreTC2wait ACKeDisparaTimerC TC2eDisparaTimerA LogERROReStoreTC3entre LogERROReStoreTCFullentre
INT2TCNOK INT2TCNOK NCP Spin StoredTC ACKOBC INT2TCNOK	LogERROReStoreTC1wait LogERROReStoreTC2wait ACKeDisparaTimerC TC2eDisparaTimerA Terminate LogERROReStoreTC2send
INT2TCNOK NCP Spin INT2TCNOK INT2TCNOK INT2DirectTC INT2TCNOK	LogERROReStoreTC1wait ACKeDisparaTimerC LogERROReStoreTC2send LogERROReStoreTC3send Execute LogERROReStoreTCFullsend
INT2TCNOK NCP Spin CmdOBCOK INT2TCNOK	LogERROReStoreTC1wait ACKeDisparaTimerC ACK LogERROReStoreTC2wait
INT2TCNOK NCP Spin StoredTC INT2DirectTC INT2TCNOK INT2DirectTC INT2TCNOK	LogERROReStoreTC1wait ACKeDisparaTimerC TC3eDisparaTimerA Execute LogERROReStoreTC2entre Execute LogERROReStoreTC3entre
INT2TCNOK NCP Spin StoredTC INT2TCNOK INT2TCNOK INT2DirectTC INT2TCNOK	LogERROReStoreTC1wait ACKeDisparaTimerC TC3eDisparaTimerA LogERROReStoreTC2entre LogERROReStoreTC3entre Execute LogERROReStoreTCFullentre
INT2TCNOK NCP Spin StoredTC ACKOBC INT2TCNOK	LogERROReStoreTC1wait ACKeDisparaTimerC TC3eDisparaTimerA Terminate LogERROReStoreTC1send
NCP Spin INT2DirectTC INT2TCNOK INT2DirectTC INT2TCNOK	ACKeDisparaTimerC Execute LogERROReStoreTC1send Execute LogERROReStoreTC2send
NCP Spin INT2TCNOK INT2TCNOK INT2DirectTC INT2TCNOK	ACKeDisparaTimerC LogERROReStoreTC1send LogERROReStoreTC2send Execute LogERROReStoreTC3send
NCP Spin CmdOBCOK INT2TCNOK	ACKeDisparaTimerC ACK LogERROReStoreTC1wait
NCP Spin StoredTC INT2TCNOK	ACKeDisparaTimerC NoTMsg LogERROReStoreTC1wait

Figura A.5 - Conjunto de testes gerado pelo método SPY para a MEF S1\_Ex5 do Receiver.



## APÊNDICE B - Conjunto de testes para o Software CM Transmitter

Este apêndice apresenta os conjuntos de casos de testes obtidos com a execução do Método UIO, quando as MEFs criadas para representar o comportamento do Software CM Transmitter lhe foram submetidas. Estas MEFs são ilustradas na Seção 3. O método que gerou o menor conjunto de casos de teste para as MEFs foi o método UIO.

As figuras B.1, B.2 e B.3 correspondem, respectivamente, as MEFs S1-Normal, S1-Ex1 e S1-Ex2.

Método UIO – S1_N	
Sequências de Entrada	Sequências de Saída Esperada
EndTimerVAndAnChVerOK EndTimerVAndAnChVerOK	StartTimerV StartTimerV
EndTimerB BeaconONAndTMConfig	Empty AnChDtCollect
BeaconONeTMnotConfig OB CONRes qNotFromCPU	ConfigTMTCTrans m Empty
BeaconONAndTMConfig OB CONReqNotFromCPU	Empty Empty
BeaconOFF OB CONRes qNotFromCPU	Empty Empty
BeaconONeTMnotConfig EndTimerB OB CONRes qNotFromCPU	ConfigTMTCTrans m Empty Empty
BeaconONeTMnotConfig OB COFF EndTimerVAndAnChVerOK	ConfigTMTCTrans m Empty StartTimerV
BeaconONeTMnotConfig OB CONRes qNotFromCPU EndTimerVAndAnChVerOK	ConfigTMTCTrans m Empty StartTimerV
BeaconONeTMnotConfig OB CONRequestFromCPU CmdOBC	ConfigTMTCTrans m ACK SendDtToCPU
BeaconONAndTMConfig OB COFF EndTimerVAndAnChVerOK	Empty Empty StartTimerV
BeaconONAndTMConfig OB CONReqNotFromCPU EndTimerVAndAnChVerOK	Empty Empty StartTimerV
BeaconONAndTMConfig OB CONRequestFromCPU CmdOBC	Empty ACK SendDtToCPU
BeaconONAndTMConfig EndTimerB EndTimerBeDB	Empty AnChDtCollect StartTimerB
EndTimerB BeaconONeTMnotConfig OB CONRes qNotFromCPU	Empty ConfigTMTCTrans m Empty
EndTimerB BeaconONAndTMConfig EndTimerBeDB	Empty AnChDtCollect StartTimerB
EndTimerB BeaconOFF OB CONRes qNotFromCPU	Empty Empty Empty
BeaconONeTMnotConfig OB CONRequestFromCPU CmdOBC EndTimerVAndAnChVerOK	ConfigTMTCTrans m ACK SendDtToCPU StartTimerV
BeaconONeTMnotConfig OB CONRequestFromCPU TM EndTimerVAndAnChVerOK	ConfigTMTCTrans m ACK SendTMeACK StartTimerV
BeaconONeTMnotConfig OB CONRequestFromCPU CmdConfigFromOBC EndTimerVAndAnChVerOK	ConfigTMTCTrans m ACK Trans cConfig StartTimerV
EndTimerB BeaconONAndTMConfig EndTimerBeDB OB CONRes qNotFromCPU	Empty AnChDtCollect StartTimerB Empty
EndTimerB BeaconONAndTMConfig Pack SurvTM EndTimerBeDB	Empty AnChDtCollect SendToTrans StartTimerB

Figura B.1 - Conjunto de testes gerado pelo método UIO para a MEF S1\_N do Transmitter.

Método UIO – S1_Ex1	
Sequências de Entrada	Sequências de Saída Esperada
EndTimerVAndAnChVerNOK EndTimerVAndAnChVerNOK	SurvFlagAndStartTimerV SurvFlagAndStartTimerV
EndTimerB BeaconONAndTMConfig	Empty AnChDtCollect
BeaconONeTMnotConfig OBCONResqNotFromCPU	ConfigTMTCTransm Empty
BeaconONAndTMConfig EndTimerB	Empty AnChDtCollect
BeaconOFF OBCONResqNotFromCPU	Empty Empty
BeaconONeTMnotConfig EndTimerB OBCONResqNotFromCPU	ConfigTMTCTransm Empty Empty
BeaconONeTMnotConfig OBCONResqNotFromCPU	ConfigTMTCTransm Empty SurvFlagAndStartTimerV
EndTimerVAndAnChVerNOK	ConfigTMTCTransm Empty SurvFlagAndStartTimerV
BeaconONeTMnotConfig OBCOFF EndTimerVAndAnChVerNOK	ConfigTMTCTransm ACK SendDtToCPU
BeaconONeTMnotConfig OBCONRequestFromCPU CmdOBC	Empty AnChDtCollect SendToTrans
BeaconONAndTMConfig EndTimerB PackSurvTM	Empty Empty SurvFlagAndStartTimerV
BeaconONAndTMConfig OBCONReqNotFromCPU	Empty ACK SendDtToCPU
EndTimerVAndAnChVerNOK	Empty Empty SurvFlagAndStartTimerV
BeaconONAndTMConfig OBCOFF EndTimerVAndAnChVerNOK	Empty Empty Empty
EndTimerB BeaconOFF OBCONResqNotFromCPU	Empty ConfigTMTCTransm Empty
EndTimerB BeaconONeTMnotConfig OBCONResqNotFromCPU	Empty AnChDtCollect SendToTrans
EndTimerB BeaconONAndTMConfig PackSurvTM	ConfigTMTCTransm ACK SendDtToCPU SurvFlagAndStartTimerV
BeaconONeTMnotConfig OBCONRequestFromCPU CmdOBC	ConfigTMTCTransm ACK SendTMeACK SurvFlagAndStartTimerV
EndTimerVAndAnChVerNOK	ConfigTMTCTransm ACK Trans cConfig SurvFlagAndStartTimerV
BeaconONeTMnotConfig OBCONRequestFromCPU TM	Empty AnChDtCollect SendToTrans SendToTrans
EndTimerVAndAnChVerNOK	Empty AnChDtCollect StartTimerB Empty
BeaconONeTMnotConfig OBCONRequestFromCPU CmdConfigFromOBC EndTimerVAndAnChVerNOK	
EndTimerB BeaconONAndTMConfig PackSurvTM PackSurvTM	
EndTimerB BeaconONAndTMConfig EndTimerBeDB OBCONResqNotFromCPU	

Figura B.2 - Conjunto de testes gerado pelo método UIO para a MEF S1\_Ex1 do Transmitter.

Método UIO – S1_Ex2	
Sequências de Entrada	Sequências de Saída Esperada
EndTimerVAndAnChVerOK EndTimerVAndAnChVerOK	StartTimerV StartTimerV
EndTimerB BeaconONAndTMConfig	Empty AnChDtCollect
BeaconONeTMnotConfig OBCONResqNotFromCPU	ConfigTMTCTransm Empty
BeaconONAndTMConfig OBCONReqNotFromCPU	Empty Empty
BeaconOFF OBCONResqNotFromCPU	Empty Empty
BeaconONeTMnotConfig OBCOFF EndTimerVAndAnChVerOK	ConfigTMTCTransm Empty StartTimerV
BeaconONeTMnotConfig EndTimerB OBCONResqNotFromCPU	ConfigTMTCTransm Empty Empty
BeaconONeTMnotConfig OBCONResqNotFromCPU	ConfigTMTCTransm Empty StartTimerV
EndTimerVAndAnChVerOK	ConfigTMTCTransm ACK Terminate
BeaconONAndTMConfig OBCOFF EndTimerVAndAnChVerOK	Empty Empty StartTimerV
BeaconONAndTMConfig OBCONReqNotFromCPU	Empty Empty StartTimerV
EndTimerVAndAnChVerOK	Empty AnChDtCollect SendToTrans
BeaconONAndTMConfig EndTimerB PackSurvTM	Empty ACK Terminate
BeaconONAndTMConfig OBCONRequestFromCPU EndTimerC	Empty AnChDtCollect SendToTrans
EndTimerB BeaconONAndTMConfig PackSurvTM	Empty ConfigTMTCTransm Empty
EndTimerB BeaconONeTMnotConfig OBCONResqNotFromCPU	Empty Empty Empty
EndTimerB BeaconOFF OBCONResqNotFromCPU	Empty AnChDtCollect SendToTrans SendToTrans
EndTimerB BeaconONAndTMConfig PackSurvTM PackSurvTM	Empty AnChDtCollect StartTimerB Empty
EndTimerB BeaconONAndTMConfig EndTimerBeDB OBCONResqNotFromCPU	ConfigTMTCTransm ACK Terminate StartTimerV
BeaconONeTMnotConfig OBCONRequestFromCPU EndTimerC	
EndTimerVAndAnChVerOK	

Figura B.3 - Conjunto de testes gerado pelo método UIO para a MEF S1\_Ex2 do Transmitter.