

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE ENGENHARIA DE COMPUTAÇÃO

ALBERTO WILTGEN JUNIOR

**Projeto do Processamento Digital de TAG  
RFID Adequado à Norma ISO/IEC 18000-2**

Trabalho de Diplomação.

Prof. Dr. Fernanda G. de Lima Kastensmidt  
Orientador

Porto Alegre, junho de 2010.



UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Profa. Valquíria Link Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do ECP: Prof. Gilson Inácio Wirth

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro



# SUMÁRIO

<b>LISTA DE ABREVIATURAS E SIGLAS</b> .....	<b>7</b>
<b>LISTA DE FIGURAS</b> .....	<b>8</b>
<b>LISTA DE TABELAS</b> .....	<b>10</b>
<b>RESUMO</b> .....	<b>11</b>
<b>ABSTRACT</b> .....	<b>12</b>
<b>1 INTRODUÇÃO</b> .....	<b>13</b>
1.1 <b>RFID</b> .....	<b>14</b>
1.2 <b>Padronização</b> .....	<b>15</b>
1.2.1 Importância das Normas Internacionais.....	16
1.2.2 NORMALIZAÇÃO E RFID.....	16
1.3 <b>Motivação</b> .....	<b>17</b>
1.4 <b>Objetivos e Metodologia</b> .....	<b>18</b>
1.5 <b>Estrutura do Trabalho</b> .....	<b>18</b>
<b>2 RFID: FUNDAMENTOS E TRABALHOS RELACIONADOS</b> .....	<b>19</b>
2.1 <b>Básico da RFID</b> .....	<b>19</b>
2.1.1 TAG.....	19
2.1.2 Leitor.....	20
2.2 <b>RFID em Detalhes</b> .....	<b>20</b>
2.2.1 Fonte de Energia.....	20
2.2.2 Freqüência.....	21
2.2.3 Memória.....	22
2.3 <b>Critérios de seleção para sistemas RFID</b> .....	<b>22</b>
2.3.1 Freqüência de Operação.....	23
2.3.2 Distância.....	23
2.3.3 Requerimentos de Segurança.....	24
2.3.4 Capacidade de Memória.....	24
2.4 <b>Trabalhos Relacionados</b> .....	<b>24</b>
2.5 <b>Projeto “CHIP CALOR”</b> .....	<b>25</b>
2.5.1 Definição das Especificações.....	25
2.5.2 Especificação.....	26
2.5.3 Diagrama de Blocos.....	27
<b>3 ESTUDO DA NORMA ISO/IEC 18000-2</b> .....	<b>29</b>
3.1 <b>Série ISO/IEC 18000</b> .....	<b>30</b>
3.2 <b>Freqüência de trabalho</b> .....	<b>30</b>
3.3 <b>Modulação e Fluxo de Dados do Leitor para o TAG</b> .....	<b>31</b>
3.4 <b>Modulação e fluxo de Dados do TAG para o Leitor</b> .....	<b>31</b>
3.5 <b>Protocolo de Comunicação</b> .....	<b>32</b>
3.6 <b>Identificação do TAG</b> .....	<b>33</b>
3.7 <b>Comandos</b> .....	<b>33</b>
3.7.1 Classificação.....	33
3.7.2 Formato.....	34
3.7.3 Flags.....	34
3.8 <b>Respostas aos Comandos</b> .....	<b>35</b>
3.9 <b>Diagrama de Estados</b> .....	<b>36</b>

<b>4</b>	<b>BLOCO DE PROCESSAMENTO DIGITAL .....</b>	<b>38</b>
4.1	Exigências.....	38
4.2	Comandos Implementados .....	39
4.2.1	Stay Quiet .....	39
4.2.2	Inventory.....	39
4.2.3	Read Single Block .....	40
4.2.4	Read Multiple Blocks .....	41
4.3	Entradas e Saídas do circuito .....	41
4.4	Arquitetura proposta .....	42
4.4.1	Décodeur.....	42
4.4.2	Codeur.....	43
4.4.3	Gestion des horloges .....	43
4.4.4	CRC .....	43
4.4.5	Sequenceur.....	44
<b>5</b>	<b>VALIDAÇÃO E IMPLEMENTAÇÃO DO BLOCO DE PROCESSAMENTO DIGITAL .....</b>	<b>46</b>
5.1	Testbench .....	46
5.2	Resultados Obtidos.....	47
5.2.1	Partes 1 a 5: A funcionalidade dos Comandos.....	47
5.2.2	Parte 6: Testes de seqüência de envio.....	49
5.2.3	Parte 7 e 8: Reação do TAG a comandos indesejados .....	49
5.2.4	Parte 9 : Os estados do TAG.....	50
5.3	Síntese do HDL .....	50
5.4	Fluxo <i>standard cell</i> .....	51
5.5	Fluxo FPGA .....	52
5.6	Simulação Pós Síntese .....	54
5.7	Static Timing Analysis (STA).....	55
5.8	Análise de consumo .....	55
5.9	Síntese Física.....	55
5.9.1	Posicionamento Floorplanning e Roteamento .....	57
5.9.2	Roteamento da árvore de relógio .....	57
5.9.3	Extração de Parasitas .....	57
5.9.4	Análise de temporização .....	58
5.9.5	Análise de Ruído, de Queda da alimentação e Eletromagnetismo.....	58
5.9.6	Posicionamento dirigido à temporização .....	58
5.9.7	Análise de Consumo .....	58
5.9.8	Correção das Regras de Desenho (DRC).....	58
5.9.9	Layout Versus Schematic .....	59
<b>6</b>	<b>CONCLUSÃO .....</b>	<b>61</b>
	<b>REFERÊNCIAS.....</b>	<b>63</b>
	<b>ANEXO &lt;RESULTADOS DO TESTBENCH&gt;.....</b>	<b>65</b>

## LISTA DE ABREVIATURAS E SIGLAS

ANSI	American National Standards Institute
ASIC	Application Specific Integrated Circuit
CRC	Código de Redundância Cíclica
EOF	<i>End Of Frame</i>
FDX	<i>full-duplex</i>
FSM	<i>Finite States Machine</i>
HDX	half-duplex
HF	altas-freqüências
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
LF	baixas freqüências
RFID	<i>Radio Frequency Identification</i>
SOF	<i>Start Of Frame</i>
UHF	ultra altas freqüências

## LISTA DE FIGURAS

Figura 1.1: Sistema RFID.....	15
Figura 3.1: Codificação PIE .....	31
Figura 3.2: Codificação TAG => Leitor.....	32
Figura 3.3: UID e SUID .....	33
Figura 3.4: Comando definido pela norma ISO/IEC 18000-2.....	34
Figura 3.5: Resposta do TAG.....	36
Figura 3.6: Diagrama de Estados do TAG .....	37
Figura 4.1: Comando Stay Quiet .....	39
Figura 4.2: Inventory quando "INV flag" = '1' .....	39
Figura 4.3: Resposta Inventory quando "INV flag" = '1'.....	40
Figura 4.4: Inventory quando "INV flag" = '0' .....	40
Figura 4.5: Resposta Inventory quando "INV flag" = '0'.....	40
Figura 4.6: Comando Read Single Block .....	40
Figura 4.7: Resposta Read Single Block sem erro .....	40
Figura 4.8: Resposta Read Single Block com erro.....	41
Figura 4.9: Comando Read Multiple Blocks .....	41
Figura 4.10: Resposta Read Multiple Blocks sem erro .....	41
Figura 4.11: Resposta Read Multiple Blocks com erro.....	41
Figura 4.12: Arquitetura proposta .....	42
Figura 4.13: Estrutura do Décodeur .....	42
Figura 4.14: Estrutura do Codeur .....	43
Figura 4.15: Estrutura do Gestion des horloges .....	43
Figura 4.16: Estrutura do CRC .....	43
Figura 4.17: Estrutura do Séquenceur .....	44
Figura 4.18: FSM de leitura de um comando .....	44
Figura 4.19: FSM de resposta a um comando .....	45
Figura 5.1: Comando Read Single Block .....	47
Figura 5.2: Comando Enviado.....	48
Figura 5.3: O tempo de Resposta Tap1 .....	48
Figura 5.4: Resposta ao Read Single Block .....	48
Figura 5.5: Comando « Stay Quiet » com pedido de CRC .....	49
Figura 5.6: O TAG continua no estado Ready .....	49
Figura 5.7: "Stay Quiet" entre bits aleatórios .....	49
Figura 5.8: O TAG passa ao estado Quiet.....	49
Figura 5.9: Envio de um comando custom não implementado .....	50
Figura 5.10: "Error Code" é 0x001.....	50
Figura 5.11: Teste dos Estados do TAG.....	50
Figura 5.12: CLB presente no Virtex 4 .....	53



Figura 5.13: Arquitetura de um LE presente no Cyclone IV .....	54
Figura 5.14: Fluxo de projeto síntese de leiaute.....	56
Figura 5.15: Placement das células do processador digital .....	59
Figura 5.16: Roteamento das células do processador digital.....	60

## LISTA DE TABELAS

Tabela 1.1: Sistemas de Identificação .....	13
Tabela 2.1: Frequências usadas na RFID .....	21
Tabela 2.2: Especificações iniciais do CHIP CALOR .....	26
Tabela 3.1: ISO/IEC 18000-2.....	30
Tabela 3.2: Tempos de símbolo PIE.....	31
Tabela 3.3: Tipos de comando.....	33
Tabela 3.4: Bits de flag.....	35
Tabela 3.5 .....	35
Tabela 3.6 .....	35
Tabela 3.7: Códigos de erro.....	36
Tabela 5.1: Partes do Banco de Teste.....	46
Tabela 5.2: Dados de área do circuito digital .....	51
Tabela 5.3: Dados de Power do Processador Digital.....	52
Tabela 5.4: Comparação entre síntese em FPGA .....	53

## RESUMO

Atualmente, os processos automáticos de identificação se tornam cada vez mais populares e necessários por fornecerem de forma rápida informações sobre pessoas, animais ou mercadorias.

Muitos métodos de identificação são utilizados, em especial o sistema de código de barras, devido ao seu baixo custo. Algumas aplicações, entretanto, onde a segurança da informação ou uma grande quantidade de memória são requisitos, outras soluções de identificação, com a identificação por radiofrequência (RFID), se tornam vantajosas.

Uma, entre as várias, aplicações onde o uso da RFID se torna vantajoso é no controle de qualidade de produtos que devem ser rigorosamente mantidos dentro de certa faixa de temperaturas, o que se chama a “Cadeia do Frio”. Isto se deve ao fato de sensores poderem ser facilmente integrados dentro de um chip deste tipo.

Por causa disso, o número de fabricantes de chips RFID não para de crescer. Esse fato tem gerado esforços de diversos organismos internacionais para regulamentar o funcionamento deste tipo de circuito e prover compatibilidade entre os chips. Modulações, delays e formatos de quadros são alguns dos pontos discutidos e padronizados pela ISO/IEC 18000-2. Esse documento regulamenta o funcionamento de chips que funcionam em baixas frequências.

Em relação a isso, a concepção de um módulo de processamento digital dentro de um circuito RF com um sensor de temperatura torna-se indispensável. Este módulo deve comandar várias ações dentro do circuito, como recepção e interpretação de comandos além da entrega de possíveis informações aos módulos que assim desejarem.

Será proposta, desenvolvida, testada, sintetizada e validada a estrutura de um módulo de processamento digital que comandará as ações de um TAG RFID com um sensor de temperatura integrado. Este módulo, bem como o circuito no qual ele será inserido, está de acordo com a norma ISO/IEC 18000-2, garantindo compatibilidade com diversos leitores disponíveis no mercado.

**Palavras-Chave:** RFID, ISO/IEC 18000-2, Simulação comportamental, síntese lógica.

# **DESIGN OF DIGITAL PROCESSING OF SUITABLE RFID TAG TO ISO/iec 18000-2**

## **ABSTRACT**

Nowadays, the automatic processes of identification has become popular and necessary for providing information about people, animals or goods. There are lots of identification methods used, in particular, the barcode system because of its low cost. However, some applications, when security of information and memory space are required, needs other type of solutions, such as the radio frequency identification (RFID).

An application where the use of RFID becomes advantageous is in the quality control of products that must be strictly kept within a certain range of temperatures, which is known as “The Cold Chain”. We can easily integrate different kinds of sensors inside an RFID chip.

Thus, the number of manufacturers of RFID chips keeps growing. This fact creates large efforts to regulate the operation of RFID circuits by International Organizations. Modulations, delays and frame formats are some of the points discussed and standardized by ISO/IEC 18000-2. This document regulates the chip operation of low frequency circuits.

In this regard, the design of a digital module inside an RF circuit contained a temperature sensor becomes indispensable. This module must receive and interpret several commands in addition to possible information delivery to the readers.

This work proposes, develops, test, synthesizes, and validates the structure of a digital module which controls the actions of an RFID tag with an integrated temperature sensor. This module, and the whole circuit, is in strictly accordance with ISO/IEC 18000-2, ensuring compatibility with many manufacturers available.

**Keywords:** RFID, ISO/IEC 18000-2, behavioral simulation, logical synthesis.

# 1 INTRODUÇÃO

Nos últimos anos, processos de identificação automáticos (Auto-ID) tornaram-se populares em vários setores, como na indústria de serviços, nas vendas, na logística de distribuição de mercadorias, na indústria de manufaturados e sistemas de fluxo de materiais. Esses processos têm a finalidade de fornecer informações sobre pessoas, animais, bens e produtos em trânsito.

Como podemos ver na tabela 1.1, [2], existem vários sistemas de identificação utilizados atualmente para as mais diversas aplicações. Uma das mais comuns, e mais simples, é a identificação por código de barras. Ela é largamente utilizada em redes de supermercados para identificar e ligar o preço a certo produto. Isso porque não são necessárias muitas informações para o sistema de processamento.

Em aplicações que exigem segurança, por exemplo, alguns métodos de identificação mais refinados, e também mais custosos, devem ser utilizados. Exemplos são os sistemas de reconhecimento vocal e identificação biométrica. Estes métodos têm a semelhança de terem um custo de operação muito alto e, por isso, só são utilizados em aplicações críticas.

Tabela 1.1: Sistemas de Identificação

Parâmetros do Sistema	Código de Barras	Reconhecimento Vocal	Biometria	Smart Card	RFID
Quantidade típica de dados ( <i>bytes</i> )	1-100	–	–	16-64 k	16-64 k
Densidade de dados	Baixa	Alta	Alta	Muito Alta	Muito Alta
Leitura por máquina	Boa	Cara	Cara	Boa	Boa
Leitura por pessoas	Limitada	Simples	Difícil	Impossível	Impossível
Influência de sujeira	Muito Alta	–	–	Possível (contatos)	Sem Influência
Ausência de contato Visual	Falha total	–	Possível	–	Sem Influência
Influência da direção e posição	Baixa	–	–	Unidirecional	Sem Influência
Degradação com o uso	Limitada	–	–	Contatos	Sem Influência
Custo da leitura	Muito	Muito Alto	Muito Alto	Baixo	Médio

<b>eletrônica</b>	Baixo				
<b>Custos operacionais</b>	Baixo	Sem custo	Baixo	Médio (contatos)	Sem custo
<b>Cópia/Modificação</b>	Fácil	Possível	Impossível	Impossível	Impossível
<b>Velocidade de Leitura</b>	Baixa ~4s	Muito Baixa >5s	Muito Baixa >5-10s	Baixa ~4s	Muito Rápida ~0.5 s
<b>Distância máxima de leitura</b>	0-50 cm	0-50 cm	Contato Direto	Contato Direto	0-5 m, Microondas

Alguns outros sistemas de identificação permitem um maior espaço para o armazenamento de dados. A melhor solução técnica para este problema seria o armazenamento dos dados em um chip de silício. A forma mais comum de dispositivo armazenador de dados em uso é o *smart card* baseado em campo de contato (cartões telefônicos e bancários). Entretanto, o contato mecânico utilizado em *smart cards* nem sempre é prático. A transferência de dados através de dispositivos que não precisam se conectar fisicamente é uma solução muito mais flexível. No caso ideal, a energia necessária para a operação do dispositivo que carrega os dados seria transmitida, também sem necessidade de contato, pelo leitor. Por causa dos processos usados para a transmissão de energia e dados, sistemas de identificação livres de contato são chamados de Identificação por Rádio Frequência (RFID). Exemplos deste tipo de sistema são os *smart cards* sem contato, tickets de pagamento em transporte coletivo, sistema de acesso e reconhecimento de pessoas, rastreamento e controle de qualidade de produtos entre outros.

## 1.1 RFID

Criada por volta dos anos 40 para a identificação de aviões militares, a RFID é uma tecnologia emergente pois permite a automação da tarefa de identificação com rapidez e segurança. Seu princípio de funcionamento é simples: um leitor envia um sinal eletromagnético para uma etiqueta, (TAG, em inglês) que, em seguida, retorna seu número de identificação junto com outras informações, dependendo da aplicação.

Num nível muito simples de abstração, as tecnologias RFID permitem a transmissão de um identificador único, mas podem ser enviadas outras informações, de forma *wireless* utilizando ondas de rádio [6]. As duas partes chave que compõe o sistema são o TAG e o leitor. Conectando um TAG RFID a um objeto qualquer, passamos a ver e monitorar este objeto por leitores presentes em redes de computadores existentes e sistemas de administração de *back-office* das empresas. A figura 1.1 mostra a configuração de um sistema RFID.

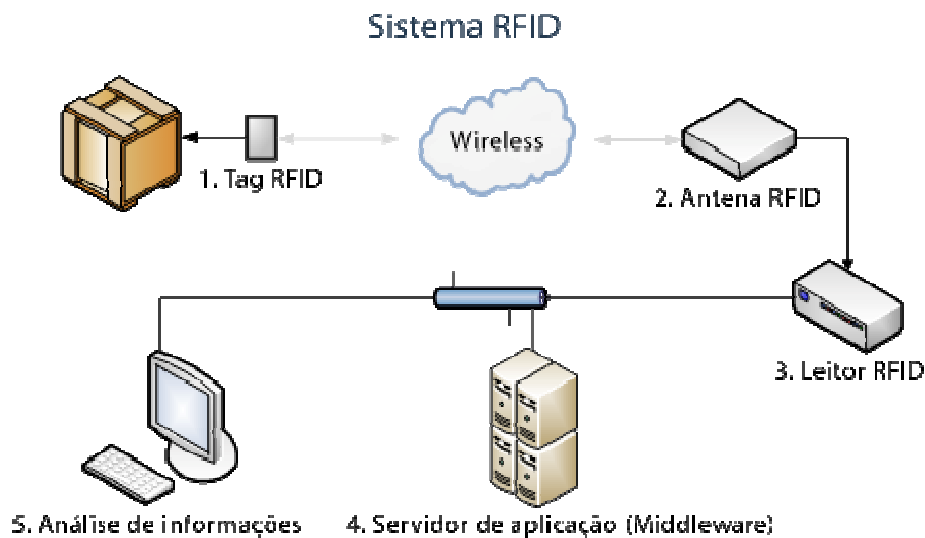


Figura 1.1: Sistema RFID

Estes sistemas podem ser classificados de acordo com diversos fatores. Podem ser classificados pela sua frequência de funcionamento (baixa: <135kHz; alta: 13,56MHz; ultra-alta (UHF): 868/915 MHz e microondas: 2,45 GHz), pelo seu modo de funcionamento (somente leitura ou leitura e escrita) e por seu modo de comunicação (*full duplex* (FDX), onde a comunicação do leitor para o TAG e do TAG para o leitor podem ser simultâneas, ou *half-duplex* (HDX), onde a comunicação ocorre do leitor para o TAG ou do TAG para o leitor, mas nunca os dois ao mesmo tempo). Os TAGs, de uma maneira geral podem ser classificados como passivos ou ativos. Essa classificação se baseia segundo a utilização ou não de uma bateria: se ela é utilizada para fazer o envio de uma informação então o TAG é dito ativo, caso contrário o TAG é passivo. Devido a essas características, muitas são as possibilidades de aplicação desse tipo de sistema.

## 1.2 Padronização

As normas internacionais são normas técnicas estabelecidas por um organismo internacional de normalização para aplicação em âmbito mundial. Existem diversos organismos internacionais de normalização, em campos específicos, como a ISO (a maioria dos setores), a IEC (área elétrica e eletrônica) e a ITU (telecomunicações).

As normas internacionais são reconhecidas pela Organização Mundial do Comércio (OMC) como a base para o comércio internacional, e o seu atendimento significa contar com as melhores condições para ultrapassar eventuais barreiras técnicas.

Essas barreiras podem ser entendidas como qualquer empecilho técnico que torne inviável a realização de uma transação comercial. Um exemplo simples é a diferença existente entre os *plugs* de tomadas elétricas existentes entre alguns países. Um produto que respeite um determinado leiaute de tomada não poderia ser vendido para um país com leiaute diferente.

### 1.2.1 Importância das Normas Internacionais

O Acordo de Barreiras Técnicas ao Comércio da OMC (TBT) estabelece uma série de princípios com o objetivo de eliminar entraves desnecessários ao comércio, em particular as barreiras técnicas, que são aquelas relacionadas com normas técnicas, regulamentos técnicos e procedimentos de avaliação da conformidade que podem dificultar o acesso de produtos aos mercados.

Um dos pontos essenciais do acordo é o entendimento de que as normas internacionais - aquelas elaboradas pelos organismos internacionais de normalização - constituem referência para o comércio internacional.

O acordo considera que as normas técnicas internacionais não constituem barreiras técnicas, e recomenda que estas normas sejam usadas como referência para os regulamentos técnicos e que também sejam adotadas como normas nacionais.

Por esta razão assiste-se a uma forte tendência de os organismos nacionais de normalização adotarem as normas internacionais integralmente como normas nacionais.

Assim, é hoje extremamente importante para os agentes econômicos que querem ser competitivos seguirem de perto os trabalhos de normalização internacional e procurarem que seus produtos, serviços e sistemas de gestão atendam aos requisitos das normas internacionais. Um exemplo desta tendência são as normas da série ISO 9000.

Pode-se mencionar que atualmente, nos países europeus, menos de 5% das normas adotadas anualmente são especificamente nacionais. Os outros 95% correspondem à adoção como normas nacionais de normas européias (EN, por exemplo) e de normas internacionais (ISO e IEC) [10].

### 1.2.2 NORMALIZAÇÃO E RFID

O número e o uso de padrões dentro da RFID e suas indústrias é bastante complexo, envolve um bom número de organismos e ainda se encontra em processo de desenvolvimento. Padrões têm sido produzidos para cobrir quatro áreas chave dentro das aplicações RFID e seus usos: padrões para as interfaces de ar (regulamentando como leitores e TAGs se comunicam), para o conteúdo e codificação dos dados, para os testes de conformidade com as normas e para a interoperabilidade entre aplicações e sistemas RFID [11].

Existem vários organismos de padronização envolvidos no desenvolvimento e definição das tecnologias RFID. Alguns deles são:

- International Organization of Standardization (ISO)
- EPCglobal Inc.
- European Telecommunication Standards Institute (ETSI)
- Federal Communications Commission (FCC)

#### 1.2.2.1 Padrões para Interface de Ar (Frequência)

As frequências utilizadas na RFID são regidas pela família de padrões ISO 18000-RFID *Air Interface*. Um conjunto completo de Padrões foi lançado em setembro de 2004.



Existem, também, alguns padrões anteriores a esses se referindo, por exemplo, aos sistemas de rastreamento e identificação de animais (ISO 11785), cartões de pagamento baseados em TAG (ISO 14443) e cartões de vizinhança (ISO 15693).

#### *1.2.2.2 Conteúdo dos Dados e Codificação*

Um componente crucial para o desenvolvimento da RFID foi a introdução do *Electronic Product Code* (EPC). Em resumo, ele é um código de identificação único que é embarcado na memória dos TAGs RFID [6]. Ele é um esquema genérico de numeração global para objetos físicos, similar ao esquema utilizado em códigos de barra.

#### *1.2.2.3 Testes de Conformidade ISO*

Padrões para o teste de conformidade de equipamentos RFID aos padrões em uso e para a medição da performance dos equipamentos são cobertos pela ISO 18047 e pela ISO 18046 respectivamente.

#### *1.2.2.4 Interoperabilidade entre Aplicações e Sistemas RFID*

Alguns padrões para garantir a interoperabilidade dos sistemas vêm sido propostos. Como exemplos, podemos citar o EPC Network Architecture, o Savant, o NOS e a Physical Mark Up Language (PML), [6].

### **1.3 Motivação**

Analisando as características dos sistemas RFID, percebemos que a sua utilização não se restringe à identificação. Podemos facilmente integrar em um TAG diferentes sensores (temperatura, pressão, umidade, luminosidade, etc.), tornando possível uma vasta gama de aplicações para estes circuitos.

A necessidade em termos de controle de qualidade de produtos da cadeia de produção de manufaturados é evidente e importante. É fundamental que estes produtos sejam submetidos a critérios de rastreabilidade desde sua origem até a chegada ao consumidor final. No mercado, existem certas mercadorias que precisam de ainda mais rigor no controle de sua temperatura, como medicamentos e alimentos. Estes produtos devem respeitar aquilo que chamamos de “Cadeia do Frio”, ou seja, é preciso ter a certeza de que o produto sempre esteve dentro da faixa de temperaturas que ele deve respeitar, assegurando a sua salubridade. Como consequência, a inserção de um meio capaz de controlar a temperatura dos ambientes de armazenamento e transporte destes produtos deve ser encarada como uma estratégia interessante para tal cadeia.

Um TAG contendo um sensor de temperatura pode, entre outros, estocar em memória um histórico das temperaturas a que foi submetido. Esta possibilidade satisfaz plenamente a necessidade descrita acima da “Cadeia do Frio”. Desta análise nasceu o projeto chamado de “CHIP CALOR”. Este projeto visa à concepção de um TAG RFID com um sensor de temperatura integrado. Este chip é um chip misto, envolvendo partes RF, analógicas e digitais.

Este trabalho trata, especificamente da concepção do Módulo Digital deste projeto. O projeto, em si, deve adequar-se a um padrão não proprietário para garantir sua compatibilidade com os diversos fabricantes disponíveis no mercado.

O Módulo Digital, que realiza o processamento de sinais do chip deve interpretar e responder às requisições do leitor além de comandar o funcionamento da memória. Memória, esta, que serve para o armazenamento do histórico de temperaturas a que o TAG foi submetido. Os dados sobre a temperatura vêm de um sensor integrado no chip.

Para o processo de desenvolvimento do módulo digital, um estudo da norma escolhida para balizar o projeto deve ser feita. Em seguida, será possível entender as necessidades do módulo para propor a estrutura adequada que possa executar todas as tarefas necessárias ao bom funcionamento do circuito como um todo.

## **1.4 Objetivos e Metodologia**

Este trabalho apresenta uma proposta para o processador de sinais do projeto CHIP CALOR. O projeto deve estar em acordo com a norma ISO 18000-2, que normaliza a comunicação entre leitores e TAGs para as baixas frequências, (menores que 135khz).

Primeiro, um estudo sobre sistemas RFID, seus componentes, suas exigências e suas utilizações será feita em detalhes. Um segundo passo será o estudo da norma que retida para o projeto, afinal, este documento prevê todas as restrições e condições de funcionamento do circuito a ser produzido.

Após esse estudo, é possível pensar na estrutura a ser proposta para a parte digital. Esta parte será feita utilizando uma linguagem de descrição de hardware, o VHDL, e sua síntese e implementações serão baseadas no fluxo de *standard cells* [13].

## **1.5 Estrutura do Trabalho**

Este trabalho se organiza como segue: A sessão 2 apresenta uma descrição dos sistemas usando RFID bem como algumas de suas aplicações. Na sessão 3 será apresentado um estudo detalhado da norma ISO 18000-2 no que se refere à parte digital do projeto CHIP CALOR. Na sessão 4 será apresentado o Bloco de Processamento Digital do circuito. A sessão 5 apresenta a validação do módulo e propõe algumas implementações possíveis. Finalmente, algumas conclusões são feitas.

## **2 RFID: FUNDAMENTOS E TRABALHOS RELACIONADOS**

Este capítulo apresenta a RFID, seus componentes, princípio de funcionamento, critérios de escolha para os diferentes sistemas existentes, entre outros. Em seguida, serão citados alguns trabalhos relacionados ao assunto que são interessantes à seqüência deste projeto.

### **2.1 Básico da RFID**

Um sistema RFID possui dois componentes principais. O leitor RF e o TAG RF. Quando um TAG RFID é colocado em algum objeto físico, este objeto pode ser identificado por um leitor através do uso da comunicação por rádio frequência.

Em princípio, e de forma muito básica, os TAGs permitem a um objeto dizer “Estou aqui e meu nome é...”. Com respeito à inteligência digital, deve ficar claro que, embora existam muitos tipos de TAGs com capacidades computacionais distintas, a inteligência do sistema reside na rede ou na aplicação conectada ao sistema RFID. A principal função de um TAG RFID é a de ligar objetos, pessoas e animais, com as redes de computadores existentes no ambiente, funcionando, assim, como uma “cola lógica”.

#### **2.1.1 TAG**

Existem dois principais componentes presentes em um TAG RFID. Primeiro, um pequeno chip de silício ou seja, um circuito integrado que contém um número de identificação único. Segundo, uma antena que pode enviar e receber ondas de rádio. Estes dois componentes podem ser muito pequenos, flexibilizando a utilização dos TAGs. A antena consiste em uma bobina metálica condutiva e plana enquanto o circuito integrado possui, na maior parte das vezes, menos de meio milímetro. Estes dois componentes são colocados em uma etiqueta de plástico que pode ser fixada a algum objeto. Estes TAGs podem ser silenciosos, finos e, cada vez mais, facilmente encapsulados em cartões plásticos, tickets, etiquetas de roupas, livros, etc. Existem dois tipos principais de TAGs: passivos e ativos, como será visto na sessão 2.2.1. TAGs passivos são geralmente os mais utilizados devido ao seu menor preço de fabricação.

### 2.1.2 Leitor

O leitor é normalmente uma unidade fixa ou que pode ser manipulada manualmente com a capacidade de ler TAGs e obter informações utilizando a comunicação RF. Quando um TAG passivo está no alcance de um leitor, sua antena absorve a energia emitida por esse leitor e a usa para ligar o seu circuito integrado e poder, então, ler e enviar o seu número de identificação. TAGs ativos não precisam absorver energia vinda do leitor para poder funcionar.

Existem duas classes principais de leitores RFID. *Read-only*, que atuam de forma passiva podendo apenas ler as informações contidas no TAG e os *read/write* que podem escrever novas informações em TAGs que possuam algum tipo de memória de leitura e escrita. Os leitores tem se tornado cada vez mais sofisticados, agindo como *gateways* para redes de empresas modernas suportando comunicação do tipo TCP/IP e tecnologias de rede como o DHCP, UDP/IP e Ethernet ou 802.11x. Muitos modelos de leitores parecem com as máquinas de preços dos supermercados ou com leitores de códigos de barras, mas podem também ser fixados em algum lugar, como em portas ou veículos, ou até mesmo escondidos em tetos ou paredes.

Alguns leitores também podem ser embarcados em dispositivos móveis como PDAs e telefones celulares. Além disso, existem dispositivos que atuam, ao mesmo tempo, como leitores e TAGs.

## 2.2 RFID em Detalhes

A parte de radiofrequência de sistemas RFID consiste no meio de comunicação entre TAGs e leitores. Com TAGs passivos, a rádio frequência também é usada para alimentar os TAGs, uma vez que eles não possuem sistemas de energia próprios.

Sistemas RFID são, essencialmente, assimétricos. Os leitores são caros e consomem bastante energia, enquanto TAGs são baratos e requerem, relativamente, menos energia. Além disso, existem 3 elementos chaves que devem ser levados em consideração em qualquer discussão sobre este tipo de sistema: fonte de energia, frequência e memória.

### 2.2.1 Fonte de Energia

TAGs RFID aparecem em uma grande variedade de tipos de acordo com a sua funcionalidade, e esses tipos foram definidos em estruturas de classe pelo Auto-ID Center [14]. Essa divisão tem sido constantemente refinada e reformulada. A estrutura básica define 5 classes diferentes de TAGs

*Identity TAGs* são TAGs puramente passivos, usados apenas para identificação. *Higher Functionality TAGs* por sua vez, também são puramente passivas mas possuem algumas funções mais refinadas. Nos *Semi-Passive TAGs*, uma bateria é adicionada ao circuito, embora uma parte da alimentação ainda venha do leitor. Nas *semi-active TAGs*, a alimentação é inteiramente fornecida pela bateria. *Active TAGs e Reader TAGs* podem se comunicar com outros TAGs e, além disso, estes últimos têm a capacidade de fornecer energia a outros TAGs. Vale a pena ressaltar que a nomenclatura para as classes tem mudado com o tempo e não é uma unanimidade para a RFID.

Sistemas de TAGs passivos não possuem uma fonte de energia *on-board*, logo eles precisam da energia vinda do leitor para poder ativar a lógica digital integrada em seus circuitos. Elas só podem, então, funcionar na presença de um leitor. A distância de comunicação é limitada pela necessidade de o leitor gerar sinais muito fortes para alimentar o TAG. Essa distância não ultrapassa 5 metros em UHF. Entretanto, como TAGs passivos não requerem uma alimentação contínua eles possuem um ciclo de vida bem longo, além de serem mais baratos para produzir. Isto significa que TAGs passivos são mais adaptados para etiquetar produtos individuais em aplicações de supermercado, industriais, médico e científicas, por exemplo.

Sistemas de TAG semi-passivos requerem o uso de uma bateria para alimentar a lógica digital contida no circuito, mas ainda assim eles aproveitam parte da energia vinda do leitor para comunicação. Este tipo de TAG é muito mais confiável e permite maiores distâncias entre leitores e TAGs, embora possuam uma vida útil menor, são mais frágeis e significativamente mais caros.

Sistemas de TAG ativos possuem um transmissor RF, ou seja, eles possuem a capacidade de realizar comunicações ponto a ponto, além de usarem baterias para alimentar a lógica do circuito e comunicar com o leitor. Isto quer dizer que este tipo de TAG não aproveita a energia do leitor. A distância de leitura pode aumentar, então, até vários quilômetros e a confiabilidade também aumenta. Esse tipo de TAG pode ser lido em movimento de mais de 160 km/h e os leitores podem fazer a leitura de milhares de TAGs por segundo. Esse tipo de TAG pode ser equipado com sensores (temperatura, humidade, etc.) Seu preço aumenta consideravelmente em relação aos outros tipos de TAG, além disso, sua capacidade de memória também tende a aumentar bastante.

### 2.2.2 Freqüência

A RFID é baseada fundamentalmente na comunicação sem fio, utilizando ondas de rádio que formam o espectro eletromagnético. Ela não é diferente de dois outros tipos de tecnologia wireless, nesse ponto, o Wifi e o Bluetooth.

O RFID opera no espaço livre do espectro, algumas vezes chamado de ISM (Industrial, Scientific and Medical) mas o conjunto de freqüências que formam a ISM pode variar de país para país. A Tabela 2.1 mostra as freqüências mais utilizadas para os sistemas RFID.

Tabela 2.1: Freqüências usadas na RFID

<b>Banda</b>	<b>LF Baixas Freqüências</b>	<b>HF Altas Freqüências</b>	<b>UHF Ultra altas freqüências</b>	<b>Microondas</b>
<b>Freqüência</b>	30-300khz	3-30mhz	300M-3GHz	2-30ghz
<b>Freqüências típicas para RFID</b>	125-134khz	13.56MHz	433 MHz ou 865-956mhz 2,45GHz	2.45 GHz
<b>Distância de leitura média</b>	Abaixo de 0,5m	Até 1,5m	433mhz, até 100m	Até 10m

			865-956 MHz = entre 0,5 e 5 m	
<b>Taxa de transmissão típica</b>	Abaixo de 1kbps	25kbps	433-956M, 30kbps 2,45G, 100kbps	Até 100 kbps
<b>Usos Típicos</b>	Identificação de animais, carros	Rótulos inteligentes, cartões de viagem sem contato, acessos e segurança	Logística de rastreamento	Veículos em movimento

Existem dois tipos de sistema RFID, cada um utilizando diferentes propriedades físicas para permitir a comunicação entre o leitor e os TAGs [15]. A física empregada é bastante complexa, mas é importante perceber que, em partes, ela determina a faixa de operação dos sistemas. Os sistemas baseados em LF e HF utilizam a propriedade física do acoplamento indutivo de um campo magnético. O leitor cria um campo magnético entre ele e os TAGs, e este campo induz uma corrente elétrica na antena do TAG, que é usada para alimentar o circuito integrado para a obtenção de informações. Essas informações são enviadas através da variação na carga da bobina da antena, o que muda a corrente pelo lado do leitor. Os sistemas RFID baseados em UHF e frequências mais elevadas, por sua vez, utilizam a propriedade da reflexão de energia. A comunicação é baseada em ondas de rádio elétricas. O leitor envia um sinal contínuo na frequência de base que é refletido de volta pela antena do TAG. Durante o processo, o TAG modula o sinal a ser refletido com a informação a ser transmitida. Normalmente utiliza-se a modulação de amplitude ou de fase.

### 2.2.3 Memória

TAGs podem apresentar várias formas diferentes com vários tipos de memória embarcadas, variando em processo de fabricação e capacidade de armazenamento. Elas podem ser *read-only*, quando as informações são permanentemente armazenadas na memória, *read-write*, quando os dados podem ser alterados pelo usuário. Ainda existe a possibilidade de se fazer uma combinação das técnicas, com um número de identificação permanente e algum espaço de armazenamento para os dados adicionais da aplicação.

TAGs passivos tem uma capacidade tipicamente variando entre 64 bits a 1 kilobit de memória não-volátil. TAGs ativos tem memórias maiores, tipicamente na ordem de 16 bytes a 128 kbytes.

## 2.3 Critérios de seleção para sistemas RFID

Desenvolvedores de sistemas RFID propuseram diversos sistemas para o mercado. Os parâmetros técnicos destes diferentes tipos de sistema são otimizados para vários campos de aplicação como, *ticketing*, identificação animal, automação industrial ou controle de acesso. Os requerimentos técnicos de cada uma dessas aplicações

frequentemente se intersectam, o que significa que uma classificação clara dos diferentes tipos de sistema não é trivial. Além disso, com algumas exceções (identificação de animais, e smart cards) ainda não existe uma padronização 100% aceita para esses sistemas.

Nas próximas sessões serão apresentadas algumas características a serem consideradas quando forem selecionados sistemas RFID segundo [2].

### 2.3.1 Frequência de Operação

Sistemas RFID que usam frequências entre 100kHz e 30MHz operam utilizando acoplamento indutivo. Em contraste, sistemas que utilizam microondas operando em frequências entre 2,45 e 5,8 GHz são acoplados utilizando campos eletromagnéticos.

A taxa de absorção específica (damping) para a água e materiais não condutivos é menor em baixas frequências. Sistemas de baixa frequência são utilizados devido a sua melhor penetração em objetos [16]. Exemplo disso são os TAGs colocados no rúmen do gado, que podem ser lidos externamente utilizando frequências abaixo de 135kHz.

Sistemas de microondas possuem uma maior distância de operação, tipicamente entre 2-15m. Entretanto, este tipo de sistema requer uma bateria extra pois a energia vinda dos leitores normalmente é insuficiente para abastecer os TAGs.

Outro fator importante a ser levado em consideração é a sensibilidade à interferência aos campos eletromagnéticos. TAGs indutivos estão em significativa desvantagem aqui. Os sistemas que utilizam microondas estão, então, bem estabelecidos nas linhas de produção e sistemas de pintura das linhas automotivas.

### 2.3.2 Distância

A distância necessária a uma aplicação pode depender de vários fatores:

- A precisão na posição do TAG
- A distância mínima entre diversos TAGs na aplicação prática
- A velocidade do TAG enquanto estiver na zona do leitor

Como exemplo, em aplicações de pagamento sem contato (por exemplo os tickets de ônibus) a velocidade do TAG é bastante lenta, uma vez que o cartão é levado ao leitor pela mão do usuário. A distância entre dois cartões é a mesma que a distância entre dois usuários entrando no ônibus. Para este tipo de sistema a zona de operação pode variar entre 5 e 10 cm. Uma zona maior poderia causar problemas, como a leitura simultânea de diversos cartões. Seria impossível saber quem pagou a passagem primeiro.

Outro exemplo são as linhas de montagem de veículos. Diversos modelos diferentes de um determinado veículo são feitos simultaneamente nas linhas de produção dessas montadoras. Então, é possível ocorrer uma grande variação na distância entre o TAG no veículo e o leitor. A distância de leitura/escrita do sistema RFID deve ser pensada para a maior distância que pode ocorrer entre o carro e o leitor. Isso para que o TAG possa sempre estar na zona de leitura do leitor. Neste tipo de contexto, o uso de sistemas de microondas mostra-se vantajoso sobre sistemas de acoplamento indutivo.

A velocidade dos TAGs, em relação ao leitor, juntamente com a distância de leitura/escrita, determina a quantidade de tempo que o TAG deve passar na zona de

leitura. Para a identificação de veículos, a distância é determinada de forma que, na máxima velocidade do veículo o TAG permaneça o tempo necessário para a sua leitura no campo do leitor para a transmissão de todos os dados.

### 2.3.3 Requerimentos de Segurança

Os requerimentos de segurança necessários à aplicação, como a autenticação e a criptografia dos dados, devem ser pensados ainda na fase de especificação do projeto de forma a evitar surpresas desagradáveis na fase de implementação. Para este propósito, é preciso avaliar o risco que o sistema RFID representa, tanto em perdas materiais como financeiras para a organização. Nós podemos, a grosso modo, dividir as aplicações em dois grupos:

- Aplicações fechadas ou industriais
- Aplicações públicas, conectadas a dinheiro ou bens materiais

No primeiro tipo de aplicação, supõe-se que apenas pessoas autorizadas possuem acesso ao sistema RFID, como é o caso de um sistema utilizado na linha de montagem de uma montadora, por exemplo. Neste contexto não seria necessário um sistema de segurança muito forte envolvendo a RFID.

Por outro lado, podemos citar o sistema de pagamentos do transporte público. Neste meio, a segurança deve ser bastante observada, uma vez que qualquer pessoa tem livre acesso ao sistema em uso.

### 2.3.4 Capacidade de Memória

O tamanho do chip do portador de dados, e também o preço, é determinado pela sua capacidade de memória. TAGs *read-only* com pouco espaço de memória são utilizados em aplicações que possuem restrições financeiras ou que podem ser utilizadas em objetos que não precisam reter muita quantidade de informações, onde a maior parte das informações é armazenada no sistema de controle. Se for necessário escrever novos dados no TAG, então uma memória do tipo EEPROM ou RAM é necessária.

Memórias do tipo EEPROM são encontradas, essencialmente, em sistemas que utilizam o acoplamento indutivo. Capacidades de armazenamento entre 16 bytes e 8 kbytes podem ser encontradas.

Memórias do tipo SRAM com bateria de backup, por outro lado, são predominantemente encontradas em sistemas de microondas. As capacidades de memória variam tipicamente entre 256 bytes e 64 kbytes.

## 2.4 Trabalhos Relacionados

Nos últimos anos, muitos são os trabalhos publicados no que se refere aos sistemas RFID. Esses esforços se devem, principalmente, aos novos domínios de aplicação e à necessidade de redução de custos deste tipo de sistema. Uma nova taxonomia para sistemas RFID foi proposta em [17]. É dito que sistemas que trabalham em baixas frequências são utilizados em aplicações que necessitam uma boa penetração do sinal em objetos.



Em [18], são discutidos circuitos de baixo consumo para a comunicação RF e também é proposto um sistema UHF com uma distância de leitura de até 4.5m. Sistemas *low Power* tem sido alvo de grandes esforços em pesquisa para esses sistemas, [19], [20].

Outro assunto de grande importância é a questão de segurança. Normalmente, TAGs RFID são as partes mais suscetíveis do sistema a ataques por partes de agentes mal-intencionados. [21] e [22] discutem soluções de segurança baseadas em curvas elípticas enquanto [23] propõe o uso de sistema de criptografia utilizando chaves simétricas, o AES.

Um trabalho que deve ser citado, também, é [12]. Ele descreve a implementação de uma TAG RFID na frequência UHF. São descritas a parte analógica e digital do circuito. A parte digital é responsável pela decodificação, codificação, controle de memória, execução do algoritmo anticóllisão e tratamento de comandos.

## 2.5 Projeto “CHIP CALOR”

Tendo analisado as principais características da RFID, será proposto, nesta sessão, um diagrama de blocos e uma tabela de especificações para um projeto utilizando a tecnologia RFID que visa o mercado de logística de produtos. Especificamente, para produtos que devem ser rigorosamente mantidos dentro de certa faixa de temperatura durante todo o seu processo de distribuição, o que é conhecido como a “Cadeia do Frio”. Exemplos de produtos que precisam obedecer esta restrição são as carnes, alguns medicamentos e outros produtos perecíveis sensíveis à temperatura.

O “CHIP CALOR” é um projeto que visa à concepção de um TAG RFID com um sensor de temperatura integrado, [3], [4], [5]. Este TAG é a evolução de um TAG criado para a identificação de bovinos, o CHIP DO BOI [ACCRUS Technology], já desenvolvido, testado e validado. O circuito funcionará em baixa frequência (<135khz), será do tipo FDX e deverá fornecer um histórico das temperaturas quando uma leitura for feita.

A comunicação entre o TAG e o leitor deverá ser feita de acordo com o padrão ISO/IEC 18000-2. Desta forma, O TAG será compatível com vários leitores atualmente disponíveis no mercado.

### 2.5.1 Definição das Especificações

#### 2.5.1.1 Alimentação

Para começar, para que as funções de leitura e armazenamento da temperatura do chip estejam ativas sem a presença do leitor, é necessário que o circuito seja alimentado por uma bateria. As principais diferenças entre as baterias oferecidas pelo mercado estão em suas dimensões físicas e suas capacidades em corrente e tensão. As especificações da bateria são as seguintes:

- Suas dimensões devem ser as mais ajustadas ao tamanho e forma do TAG;
- O valor nominal de tensão deve ser definido no início da fase de desenvolvimento;

- A capacidade de corrente deve ser suficiente para atender à demanda total do circuito e da duração esperada.

Com a finalidade de simplificar o primeiro protótipo deste projeto e avaliar o comportamento da bateria do circuito, será utilizado o método de alimentação dito **semi-ativa**, ou seja, o circuito não utilizará a potência do sinal RF para seu funcionamento. Como consequência, a única fonte de alimentação do circuito será a bateria.

### 2.5.1.2 Sensor

No “CHIP CALOR” será integrado um sensor de temperatura. De acordo com a especificação do projeto, o sensor deverá ser capaz de medir temperaturas variando entre  $-40^{\circ}\text{C}$  e  $120^{\circ}\text{C}$ . Desta forma, ele poderá ser usado em grande parte das aplicações comerciais. Além disso, o circuito deverá ser construído de forma a permitir a conexão de um sensor externo, seja de temperatura (para uma comparação de valores, por exemplo) seja de outras grandezas físicas.

### 2.5.1.3 Normalização

Atualmente, a normalização para o RFID em baixa frequência se resume a três protocolos:

- As normas ISSO 11784/85 para a identificação de animais;
- A norma ISSO 14223 normaliza os TAGS ditos “evoluídos” (compatíveis com as normas ISO 11784/85);
- O padrão ISSO/IEC 18000-2. Este padrão será apresentado em maiores detalhes nos próximos capítulos pois foi o escolhido para embasar o projeto “CHIP CALOR”

Esta norma foi a norma retida pelo projeto pois é uma norma bastante aceita pelos mais variados fabricantes e, portanto, será fácil encontrar leitores disponíveis que sejam compatíveis com o TAG desenvolvido. Esta norma estabelece os parâmetros que permitem a comunicação entre leitores e TAGs em frequências variando entre 125 e 135khz. Estes parâmetros incluem a frequência de funcionamento, as modulações e os formatos de quadro utilizados em comandos e respostas.

## 2.5.2 Especificação

A partir da norma escolhida, uma primeira tabela de especificações foi criada. Estas exigências são o ponto de partida da concepção do “CHIP CALOR”. A tabela 2.2 mostra as principais exigências que devem ser respeitadas por esse chip.

Tabela 2.2: Especificações iniciais do CHIP CALOR

Parâmetro	Valor
<b>Compatibilidade</b>	ISO/IEC 18000-2
<b>Método de Comunicação</b>	FDX
<b>Frequência de Operação</b>	125 kHz
<b>Modulação <i>uplink/downlink</i></b>	ASK/OOK
<b>Codificação dos Dados <i>uplink/downlink</i></b>	Manchester-Dual Pattern Encoding/PIE

<b>Tamanho da Memória</b>	1 kbits (max. 1 kbyte)
<b>Alimentação</b>	Bateria
<b>Faixa Mensurável de Temperatura</b>	De -40°C a 120°C
<b>Resolução do Sensor</b>	@ 8 bits → ~1°C/bit
<b>Sensor Externo</b>	Saída em Corrente

Como foi discutido, este chip é a evolução de um circuito para a identificação de bovinos, o “CHIP DO BOI”. Este chip também trabalha em baixa frequência e é do tipo FDX. Razão esta pela qual o tipo de comunicação para este projeto também será o FDX.

A partir desta decisão, muitos dos parâmetros da tabela acima foram definidos em função da norma, como as frequências de operação e os tipos de modulação utilizados. Além disso, o sensor utilizado deve ser capaz de suportar a maior parte das operações comerciais, o que acontece numa faixa entre -40° e +120°.

### 2.5.3 Diagrama de Blocos

De acordo com as especificações definidas, foi proposto o diagrama de blocos dos módulos que compoem o circuito do “CHIP CALOR”, [3]. A figura 2.1 ilustra este circuito.

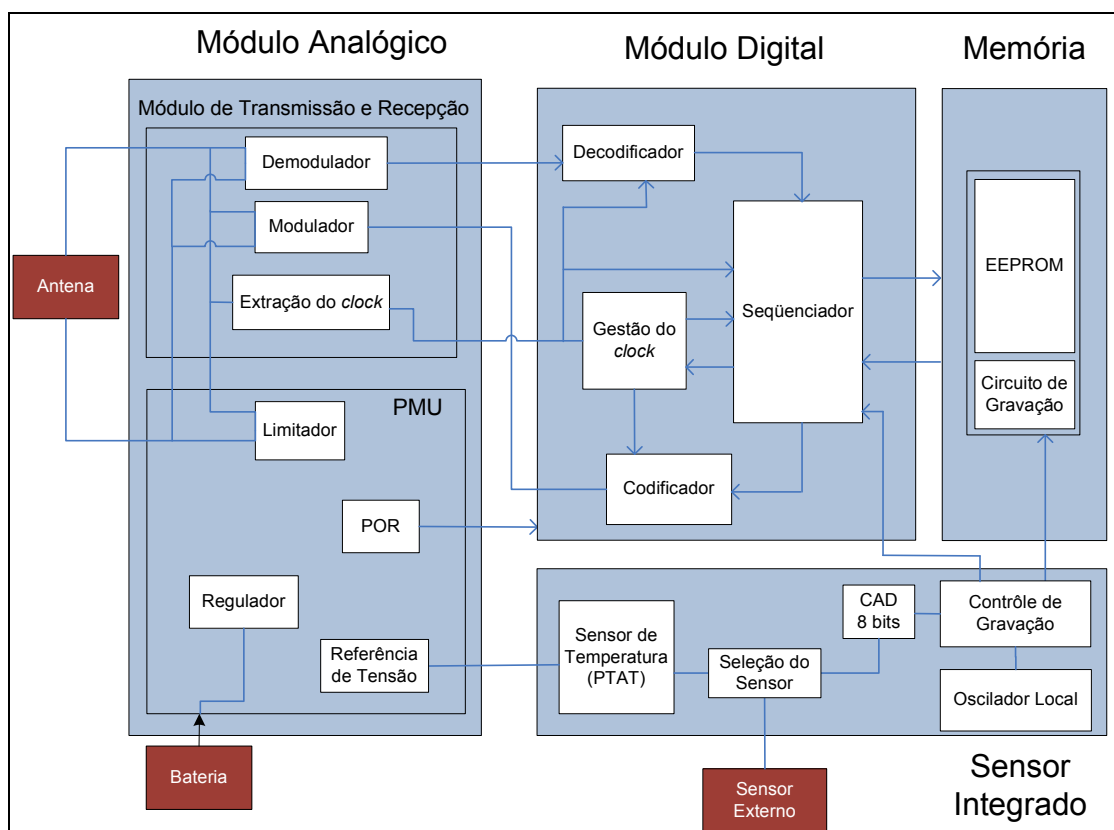


Figura 2.1: Diagrama de blocos do "CHIP CALOR"

A antena do transponder consiste em um circuito LC paralelo cuja frequência de ressonância corresponde à frequência do sinal RF emitido pelo leitor, neste caso 125 kHz. É através do acoplamento magnético entre as bobinas do circuito LC do TAG e do leitor, que os circuitos se comunicam.

Dentro do Módulo Analógico encontramos dois submódulos. O módulo de transmissão e recepção está em contato direto com o sinal RF. O modulador está conectado na saída do bloco de codificação e realiza a modulação do sinal na antena. Como as antenas se comunicam através de um acoplamento magnético, usa-se uma modulação de carga para realizar as mudanças na amplitude do sinal (modulação ASK). A extração do *clock* visa obter a partir do sinal RF o sinal de sincronização das tarefas realizadas pelo *tag* durante uma leitura.

O Power Management Unit (PMU) é responsável por fornecer uma referência de tensão ao restante do circuito e de gerenciar os processos de leitura com o sinal de começo de leitura (POR). O regulador de tensão é muito importante para se evitar que possíveis mudanças no valor nominal da bateria, devidas principalmente às descargas, possam prejudicar o correto funcionamento do circuito.

No módulo Sensor Integrado, encontramos o sensor de temperatura interno ao chip e também uma saída para um sensor externo. O bloco de seleção do sensor permite aplicar à entrada do conversor analógico digital (CAD) tanto o sensor interno como externo do TAG.

A memória serve para o armazenamento dos valores provindos da saída do CAD do circuito e é comandada pela lógica digital do chip. A parte digital do projeto, incluindo o Módulo Digital e o “Controle de Gravação” do SENSOR INTEGRADO será abordada em mais detalhes neste documento.

### 3 ESTUDO DA NORMA ISO/IEC 18000-2

Neste capítulo é apresentado em detalhes o padrão escolhido para guiar o projeto CHIP CALOR. O circuito a ser fabricado trabalhará em baixa frequência, garantindo assim uma boa penetração em objetos, sua comunicação será feita com base no acoplamento indutivo entre o leitor e o TAG e seu modo de comunicação será do tipo FDX. O tipo de comunicação foi escolhido em função deste circuito ser a evolução de outro chip RFID para identificação de bovinos que utiliza o mesmo tipo de comunicação.

Um padrão é um documento que estabelece especificações, critérios, métodos, processos ou práticas uniformes. Alguns padrões são praticamente obrigatórios enquanto outros são opcionais. Também existe o que chamamos *padrão de facto*, quando embora não exista um documento oficial, a norma é bastante aceita pela indústria. As organizações de padronização, tais quais a ISO ou a ANSI, são independentes das empresas que utilizam seus documentos.

A utilização de normas, principalmente internacionais, se mostra uma estratégia interessante para evitar possíveis barreiras técnicas em transações comerciais, tanto nacionais como internacionais. Neste contexto, a RFID também possui seus documentos de padronização. Como esta tecnologia ainda está em fase de desenvolvimento ainda não existem padrões 100% aceitos. A ISO, juntamente com a IEC, são duas organizações que fazem esforços para padronizar os sistemas RFID e algumas das normas propostas para as baixas frequências são:

- As normas ISO 11784/85 para a identificação de animais;
- A norma ISO 14223 normaliza os TAGS ditos “evoluídos” (compatíveis com as normas ISO 11784/85);
- O padrão ISO/IEC 18000-2.

A norma retida para guiar a concepção do projeto CHIP CALOR foi a norma ISO/IEC 18000-2. Este documento padroniza o modo de comunicação entre o leitor e o TAG, estabelecendo parâmetros como frequências de operação, modulações, conteúdo de dados entre outros parâmetros indispensáveis para o entendimento entre leitores e TAGs. Esta norma, lançada em setembro de 2004 é bastante aceita no mercado e também é uma norma bem completa, motivo este pelo qual ela será a norma utilizada no projeto.

Os parágrafos que seguem explicam algumas das exigências deste padrão no que se refere ao projeto desenvolvido. Serão ressaltados os tópicos referentes à inteligência digital do circuito para que possa ser proposta uma estrutura para o processador digital do chip. Maiores detalhes podem ser encontrados em [1].

### 3.1 Série ISO/IEC 18000

ISO/IEC 18000 é uma série de normas internacionais que descrevem os protocolos de comunicação comuns à RFID. A ISO/IEC 18000-2, adotada para este projeto, é ligada aos sistemas que trabalham em baixas frequências. Os parâmetros abrangidos comportam as frequências de transmissão, a velocidade de transmissão, a ocupação espectral, a estrutura dos comandos, as arquiteturas, as especificações do sistema e a interface de comunicação. Ela normaliza estes parâmetros para as comunicações de uma interface de menos de 135kHz.

A norma padroniza estes parâmetros para dois tipos de comunicação: FDX e HDX. A tabela 3.1 mostra algumas dessas características.

Tabela 3.1: ISO/IEC 18000-2

Parâmetro	FDX	HDX
<b>Frequência</b>	125kHz	134,2kHz
<b>Modulação Leitor =&gt; TAG</b>	OOK	OOK
<b>Modulação TAG =&gt; Leitor</b>	ASK	ASK
<b>Codificação de dados Leitor =&gt; TAG</b>	Pulse Interval Encoding (PIE)	
<b>Fluxo Leitor =&gt; TAG</b>	5.2kbps	1 a 2.3kbps
<b>Codificação de Dados TAG =&gt; leitor</b>	Manchester / Dual Pattern Encoding	NRZ
<b>Fluxo TAG =&gt; leitor</b>	Manchester 4kbps DPE 2kbps	NRZ "0" 8.2kbps NRZ "1" 7.7kbps
<b>Ordem de transmissão de bits</b>	LSB primeiro	

Como já foi discutido, O CHIP CALOR é do tipo FDX. As próximas sessões tratam dos parâmetros exigidos pela norma para este tipo de comunicação.

### 3.2 Frequência de trabalho

A frequência de transmissão do leitor em direção ao TAG deve ser  $f_{ac} = 125\text{kHz} = 1/T_{ac}$ .

### 3.3 Modulação e Fluxo de Dados do Leitor para o TAG

A comunicação é feita utilizando a codificação por intervalo de pulso (PIE). O tempo entre duas transições de descida determina se a informação corresponde a um bit '1', um bit '0', um "code violation" ou um "stop condition". Esses quatro símbolos são mostrados na figura 3.1. O fluxo será da ordem de 5.1kbps. A tabela 3.2 mostra os tempos que nos permitem diferenciar os 4 símbolos utilizados para esta codificação. Na tabela o símbolo  $T_{ac}$  significa o tempo do período do *clock* analógico utilizado. No nosso caso a frequência de operação é de 125kHz, ou seja,  $T_{ac} = (1/125k)s$ . O tempo *Carrier off*, na primeira linha se refere ao número de ciclos  $T_{ac}$  em que a portadora deve ter o sinal no nível lógico '0' para que o símbolo seja válido.

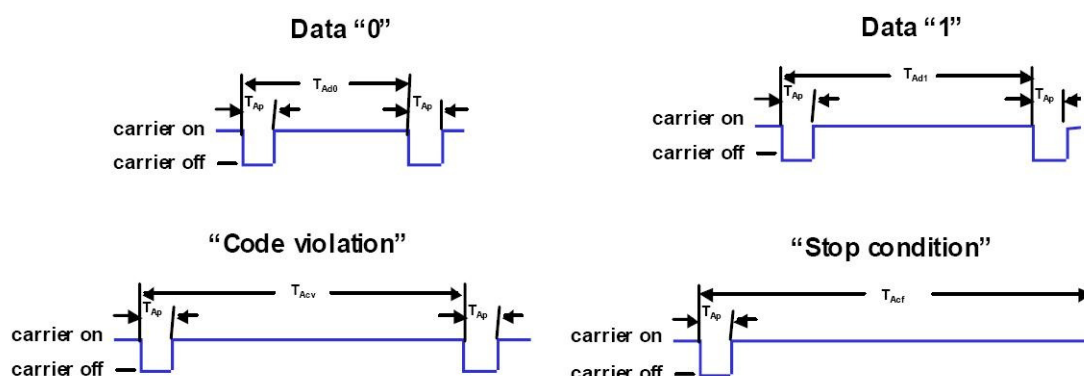


Figura 3.1: Codificação PIE

Tabela 3.2: Tempos de símbolo PIE

Meaning	Symbol	Min	Max
"Carrier off" time	$T_{ap}$	$4 * T_{ac}$	$10 * T_{ac}$
Data "0" time	$T_{ad0}$	$18 * T_{ac}$	$22 * T_{ac}$
Data "1" time	$T_{ad1}$	$26 * T_{ac}$	$30 * T_{ac}$
"code violation" time	$T_{acv}$	$34 * T_{ac}$	$38 * T_{ac}$
"stop condition" time	$T_{asc}$	$>42 * T_{ac}$	n/a

### 3.4 Modulação e fluxo de Dados do TAG para o Leitor

O TAG trabalha com dois tipos de codificação, a Manchester a 4 kbps e a *Dual Pattern Encoding* (DPE), a 2kbps. A escolha entre essas codificações é feita em função do comando transmitido pelo leitor. Quando for recebido um "INVENTORY" a codificação é a DPE. O *start of frame* (SOF) é codificado em Manchester para todos os comandos. No resto dos comandos a codificação Manchester é utilizada. Maiores detalhes sobre os comandos implementados são vistos na sessão 3.7. A figura 3.2 mostra os padrões das duas codificações utilizadas.

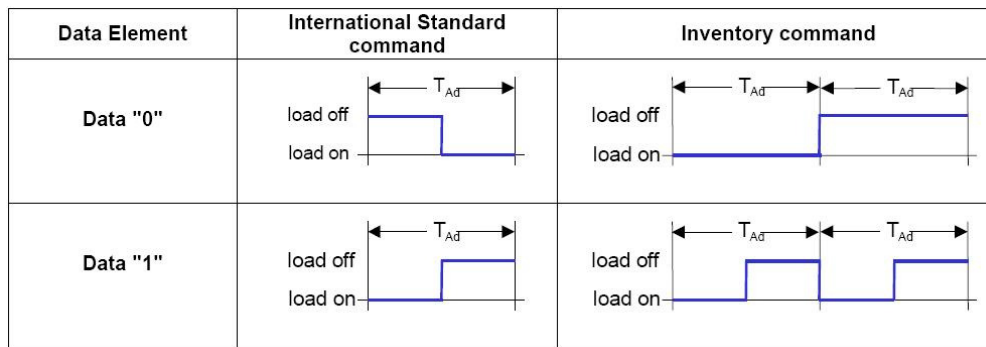


Figura 3.2: Codificação TAG => Leitor

Para a realização destas modulações, o TAG utilizara a modulação de carga. Essa modulação funciona da seguinte maneira: se a distância entre o leitor e o TAG é inferior a 16% do comprimento de onda do sinal, nós podemos considerar as duas bobinas como o primário e o secundário de um transformador. Nestas condições, a influência da carga do secundário sobre o primário pode ser modelada por uma impedância. Então, se fizermos variar a impedância do TAG, a impedância no leitor também mudará, o que corresponde a diferentes níveis de tensão. Basta comutar uma resistência no TAG para que o Leitor possa detectar os '0's e '1's.

### 3.5 Protocolo de Comunicação

A norma impõe um protocolo de transmissão para regulamentar a comunicação entre leitores e TAGs, nas duas direções. Os **leitores** devem ser capazes de se comunicar tanto com TAGs do tipo FDX como com TAGs do tipo HDX. O protocolo é baseado nos seguintes conceitos:

- O Leitor fala primeiro, o que significa que os TAGs devem esperar o recebimento de um comando para se manifestar.
- Cada TAG possui um número de identificação próprio chamado *unique identifier* (UID) codificado em 64 bits. Um segundo identificador, o *simplified unique identifier* (SUID) pode ser extraído a partir do UID do TAG. Ele é codificado em 48 bits.
- O protocolo é baseado em uma troca de:
  - Um comando do leitor para o(s) TAG(s)
  - Uma resposta do(s) TAG(s) de volta ao leitor
- O protocolo tem transmissão serial, ou seja, os quadros são transmitidos bit a bit. O número de bits transmitidos após um *Start Of Frame* (SOF) é variável e depende do comando em questão.
- A Transmissão sempre começa pelo bit menos significativo, o *Less Significant Bit* (LSB) e termina pelo *Most Significant Bit* (MSB).



Alguns bits de *flag* são utilizados para assegurar o controle dos comandos e das respostas. Cada *flag* possui seu próprio significado. Os *flags* enviados nos comandos e nas respostas são diferentes.

### 3.6 Identificação do TAG

A norma estipula que cada TAG deve ter um número de identificação único, o UID. O UID é representado em 64 bits que se dividem da seguinte forma:

- « E0 » em hexadecimal, em 8 bits
- Número de Identificação do Fabricante (MFG) em 8 bits
- Octeto de zeros, em 8 bits
- Número de série do TAG (MSN), em 40 bits

Para otimizar as performances do sistema, somente um pedaço do UID é transmitido na maior parte dos comandos e respostas do TAG durante o processo anticollisão. O UID simplificado é chamado de SUID e é codificado em 48 bits: MFG em 8 bits e MSN em 40 bits. A figura 3.3 mostra o formato destes identificadores.

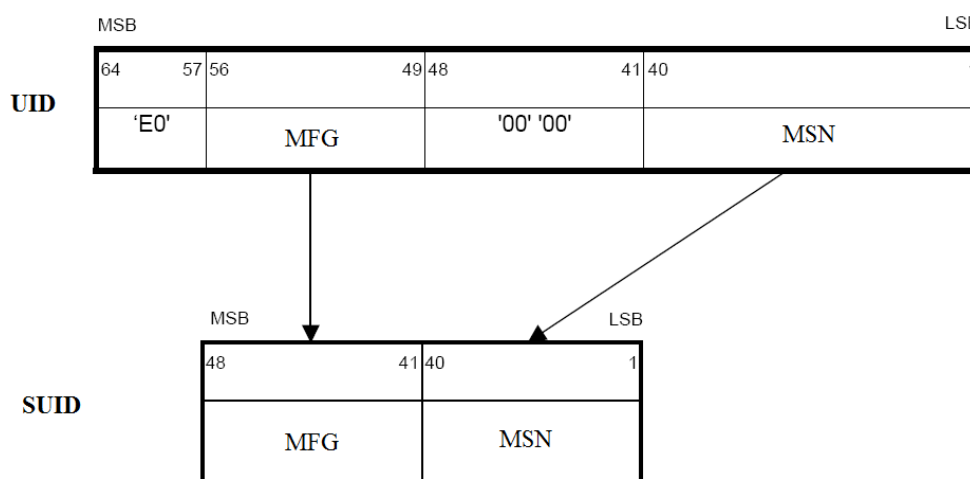


Figura 3.3: UID e SUID

### 3.7 Comandos

#### 3.7.1 Classificação

A norma estipula 4 tipos de comandos, todos codificados em 6 bits. Eles são apresentados na tabela 3.3.

Tabela 3.3: Tipos de comando

Code	Class
'00' – '0F'	Mandatory
'10' – '27'	Optional

'28' – '37'	<b>Customer</b>
'38' – '3F'	<b>Proprietary</b>

Podemos ver nesta tabela que a diferença entre as classes de comandos é feita pelo código do comando. Por exemplo, os comandos “Mandatory” sempre terão um código variando entre 0x00 e 0x0f e assim por diante. As principais características de cada grupo são

- **Mandatory:** os comandos deste grupo devem ser implementados tanto pelos leitores como pelos TAGs. Exemplos são o *Stay Quiet* e o *Inventory*.
- **Optional:** Os leitores devem implementar todos os comandos opcionais especificados no documento enquanto os TAGs podem implementá-los, mas não são obrigatórios. Exemplos são o *Read Multiple Blocks* e o *Select*.
- **Custom:** os comandos deste tipo não são especificados pela norma. Eles ficam a critério do fabricante.
- **Proprietary:** os comandos deste tipo não são especificados pela norma. Eles ficam a critério do fabricante.

### 3.7.2 Formato

A figura 3.4 mostra os formatos de um comando enviado pelo o leitor.

SOF	Flags	Command	Parameters	Data	CRC	EOF
-----	-------	---------	------------	------	-----	-----

Figura 3.4: Comando definido pela norma ISO/IEC 18000-2

Os comandos não possuem um tamanho fixo. Seu tamanho depende do comando em si e também do valor dos bits de flag. Um comando começa sempre por um *start of frame* (SOF) e termina sempre por um *end of frame* (EOF). Cada parâmetro é enviado a partir de seu LSB até seu MSB.

O código de redundância cíclica (CRC) é opcional e depende do leitor. Para o seu cálculo é utilizado o seguinte polinômio:  $P(x) = X^{16} + X^{12} + X^5 + 1$ . Uma proposta de arquitetura para o bloco de cálculo do CRC é apresentada em detalhes em [1].

### 3.7.3 Flags

São os flags que indicam quais parâmetros formarão as respostas. Em cada comando, enviamos 5 bits de flag. A função de cada flag depende do contexto. As tabelas 3.4,5 e 6 sintetizam a função de cada um desses bits. Os bits de *flag 4* e *5* dependem do valor do bit *inventory flag* como podemos perceber na análise da tabela.

Tabela 3.4: Bits de flag

Bit	FLAG	Valor	Descrição
B1	PEXT (Protocol Extension) flag	0	Reservado para aplicações futuras
B2	INV (Inventory flag)	0	Flags 4 e 5 – Tabela 3-5
		1	Flags 4 e 5 – Tabela 3-6
B3	CRCT	0	Resposta sem CRC
		1	Resposta sem CRC

Tabela 3.5

BIT	FLAG	Valor	Descrição
B4	SEL (Select) flag	0	Todos os TAGs devem responder em função do bit ADR.
		1	Somente os TAGs em estado “Selected” devem responder. Bit ADR deve estar em ‘0’ e sem SUID.
B5	ADR (address) flag	0	Comando não endereçado. Sem SUID. Todos os TAGs devem responder
		1	Comando endereçado com SUID. Somente o TAG escolhido deve responder

Tabela 3.6

BIT	FLAG	Valor	Descrição
B4	AFI flag	0	AFI ( <i>Application family identifier</i> ) não está presente.
		1	AFI está presente.
B5	NOS flag	0	Comando <i>INVENTORY</i> com 16 <i>slots</i> .
		1	Comando <i>INVENTORY</i> com 1 <i>slot</i> .

### 3.8 Respostas aos Comandos

Não são todos os comandos que exigem uma resposta por parte do TAG. Por exemplo, se for recebido o comando *Stay Quiet*, o TAG deve permanecer em silêncio, apenas mudando o seu estado, como será visto no capítulo 4.

Quando o leitor exigir uma resposta por parte do TAG, ela terá o formato genérico mostrado na figura 3.5. O *error flag* indica se houve, ou não, erro na recepção do comando exigido. Se o comando foi recebido corretamente então o TAG envia os dados

exigidos, como seu número de identificação por exemplo, e anexa no fim do quadro, se necessário, o código de redundância cíclica. Caso um erro seja percebido pelo TAG, então um código de erro será emitido pelo tag para que o leitor possa saber onde está o erro. A tabela 3.7 mostra os possíveis erros que podem ocorrer.

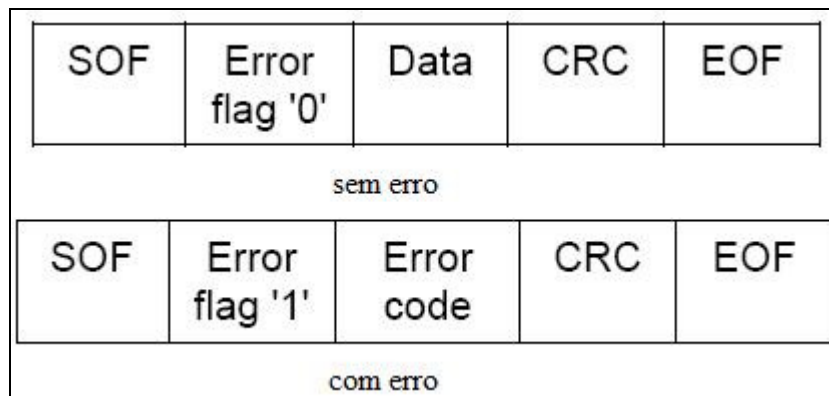


Figura 3.5: Resposta do TAG

Tabela 3.7: Códigos de erro

Código	Descrição
0	Sem erro
1	O comando não é suportado
2	O comando não é reconhecido, erro de formato
3	Bloco especificado não está disponível, não existe
4	O bloco especificado é um bloco seguro e não pode ser acessado
5	O bloco especificado não foi bem programado, fechado
6	RFU
7	Erro desconhecido

Uma tabela mais completa, mostrando a descrição de todos os comandos previstos pela norma pode ser encontrada em [1]. Ele explica, por exemplo, os parâmetros que devem ser enviados em cada comando e o formato da resposta do TAG para cada um entre eles.

### 3.9 Diagrama de Estados

A norma propõe um diagrama de estados para os TAGs. Um TAG pode possuir até 4 estados (Power-off, Ready, Selected e Quiet) como podemos ver na figura 3.6, juntamente com as condições de transição entre os estados. A presença dos estados Power-off Ready e Quiet é obrigatória enquanto a presença do estado Selected é opcional.

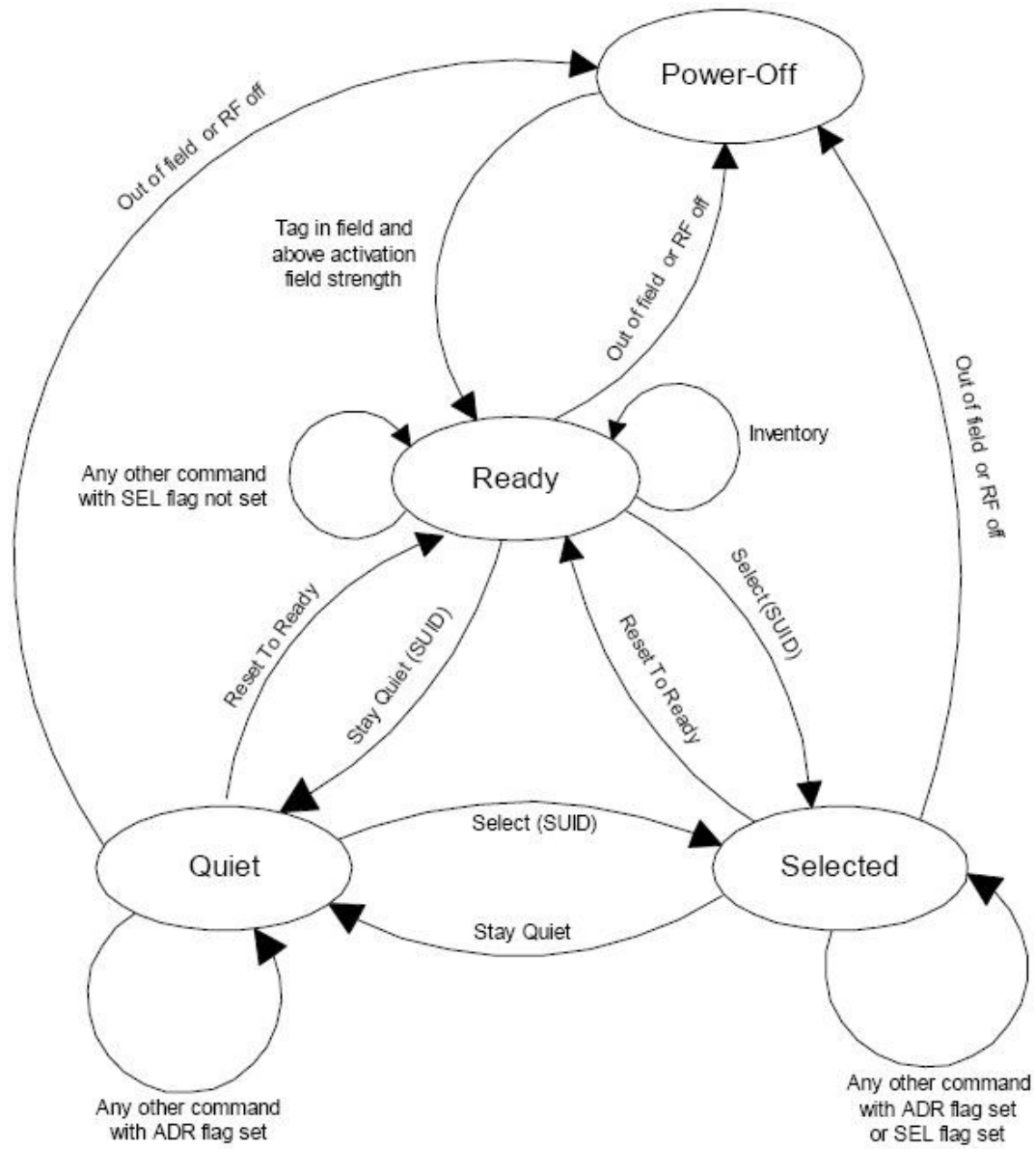


Figura 3.6: Diagrama de Estados do TAG

## 4 BLOCO DE PROCESSAMENTO DIGITAL

Esta sessão propõe a primeira versão do bloco de processamento digital do CHIP CALOR. Este bloco é o responsável pela lógica digital do circuito, a interpretação dos comandos implementados e a formação dos quadros de resposta que serão reenviados ao leitor. Além disso, é este bloco o responsável pela implementação da máquina de estados do TAG. É a norma ISO/IEC 18000-2 que define a forma como essas tarefas são feitas.

O circuito deve funcionar a 125khz e, a princípio, esta é a única restrição que será imposta ao projeto. Primeiramente, será feita uma implementação funcional que respeite a norma. Num segundo momento, estes fatores poderão ser levados em consideração na implementação da parte digital do CHIP CALOR.

A descrição da estrutura que será proposta no próximo parágrafo foi feita utilizando a linguagem de descrição de hardware VHDL. Com o VHDL pronto, foi possível validar seu comportamento através de um *testbench*, sintetizar a descrição comportamental e mapeá-la em um circuito que implemente a mesma função lógica.

### 4.1 Exigências

O módulo digital responderá a 4 diferentes comandos:

- Stay Quiet
- Inventory
- Read Single Block
- Read multiple Blocks

Os dois primeiros são obrigatórios e devem estar presentes em todos os TAGs que respeitam a norma enquanto os dois últimos são do tipo opcional. Eles serão implementados pela necessidade de interação com uma memória externa ao módulo que guardará o histórico de temperaturas pelo qual o TAG pode ser submetido durante seu tempo de vida.

Como uma consequência direta da escolha dos comandos implementados, apenas três estados serão implementados no TAG:

- Power Off
- Ready
- Quiet

Lembrando que o estado “Selected” é opcional, o que significa que os TAGs não são obrigados a implementá-lo. A escolha sendo feita pelos comandos opcionais que são implementados. A máquina de estados foi vista no capítulo 3.

## 4.2 Comandos Implementados

### 4.2.1 Stay Quiet

Diante da recepção de um comando *stay quiet*, se o TAG estiver no estado “READY”, ele deve passar para o estado “QUIET”. Não existe resposta para este comando. A figura 4.1 mostra o seu formato.

SOF	Flags	Command	Parameter	CRC	EOF
		STAY QUIET	SUID (optional)	(optional)	
	5 bits	6 bits	48 bits	16 bits	

Figura 4.1: Comando Stay Quiet (ISO/IEC 18000-2)

### 4.2.2 Inventory

Este comando é responsável por fazer um inventário dos TAGs presentes no campo magnético do leitor. Seu formato e o formato da resposta dependem do bit “*Inventory Flag*”. No caso de haver vários TAGs no campo do leitor, tanto o leitor como os TAGs devem ser capazes de executar uma seqüência anticolisão, para evitar que mais de um TAG transmita dados ao mesmo tempo. Em um TAG do tipo FDX, a resposta a este comando consiste em:

- Codificação “DPE” a 2 kbps se o bit “Inventory flag” estiver em ‘1’,
- Codificação “Manchester” a 4 kbps se o bit “Inventory flag” estiver em ‘0’

#### 4.2.2.1 Inventory quando o bit “Inventory flag” estiver em ‘1’

Com a recepção deste comando sem erro, os TAGs devem executar a seqüência anticolisão. O bit “NOS flag” determina se utilizaremos 1 ou 16 slots no algoritmo anticolisão. Caso o TAG detecte algum erro, ele deve ficar em silêncio. Detalhes sobre o funcionamento do algoritmo anticolisão pode ser visto em [1].

SOF	Flags	Command	Parameter 1	Parameter 2	Parameter 3	CRC	EOF
	01xxx	INVENTORY	AFI (optional)	Mask length(n) $0 \leq n \leq \text{SUID length}$	Mask value	(optional)	
	5 bits	6 bits	8 bits	6 bits	n bits	16 bits	

Figura 4.2: Inventory quando "INV flag" = '1' (ISO/IEC 18000-2)

SOF	Data	CRC	EOF
	Remaining section of the SUID (SUID without Mask value)	(optional)	
	48 - n bits	16 bits	

Figura 4.3: Resposta Inventory quando "INV flag" = '1' (ISO/IEC 18000-2)

#### 4.2.2.2 Inventory quando o bit "Inventory flag" estiver em '0'

Quando o bit "Inventory flag" estiver em '0', o bit "NOS flag" deve ser colocado em '1' para indicar apenas um slot na seqüência anticolisão. Isso permite ao TAG responder imediatamente o comando com o seu SUID.

Se o TAG receber o comando sem erro, ele deve transmitir seu SUID como resposta. As figuras 4.4 e 4.5 mostram os formatos padrão do comando e da respectiva resposta.

SOF	Flags	Command	Parameter 1	CRC	EOF
	00xx1	INVENTORY	AFI (optional)	(optional)	
	5 bits	6 bits	8 bits	16 bits	

Figura 4.4: Inventory quando "INV flag" = '0' (ISO/IEC 18000-2)

SOF	Data	CRC	EOF
	SUID	(optional)	
	48 bits	16 bits	

Figura 4.5: Resposta Inventory quando "INV flag" = '0' (ISO/IEC 18000-2)

#### 4.2.3 Read Single Block

O comando *read single block* é utilizado para efetuar a leitura de um bloco de 32 bits da memória. Sua estrutura, bem com a estrutura das respostas, são mostradas nas figuras 4.6, 4.7 e 4.8:

SOF	Flags	Command	Parameter 1	Parameter 2	CRC	EOF
		READ SINGLE BLOCK	SUID (optional)	Block address	(optional)	
	5 bits	6 bits	48 bits	6 bits	16 bits	

Figura 4.6: Comando *Read Single Block* (ISO/IEC 18000-2)

SOF	Error flag	Data	CRC	EOF
	0	User memory block data	(optional)	
	1 bit	32 bits	16 bits	

Figura 4.7: Resposta *Read Single Block* sem erro (ISO/IEC 18000-2)



SOF	Error flag	Error code	CRC	EOF
			(optional)	
	1 bit	3 bits	16 bits	

Figura 4.8: Resposta *Read Single Block* com erro (ISO/IEC 18000-2)

#### 4.2.4 Read Multiple Blocks

O comando *read multiple blocks* permite a leitura de diversos blocos contidos na memória de uma só vez. Sua estrutura, bem como as possíveis respostas, são mostradas nas figuras 4.9, 4.10 e 4.11:

SOF	Flags	Command	Parameter 1	Parameter 2	Parameter 3	CRC	EOF
		READ MULTIPLE BLOCKS	SUID (optional)	First block number	Number of blocks	(optional)	
	5 bits	6 bits	48 bits	8 bits	8 bits	16 bits	

Figura 4.9: Comando *Read Multiple Blocks* (ISO/IEC 18000-2)

SOF	Error flag	Data	CRC	EOF
	0	User memory block data (not present on error)	(optional)	
	1 bit	32 bits	16 bits	
		Repeated as needed		

Figura 4.10: Resposta *Read Multiple Blocks* sem erro (ISO/IEC 18000-2)

SOF	Error flag	Error code	CRC	EOF
	1		(optional)	
	1 bit	3 bits	16 bits	

Figura 4.11: Resposta *Read Multiple Blocks* com erro (ISO/IEC 18000-2)

### 4.3 Entradas e Saídas do circuito

O processador digital do CHIP CALOR é um bloco interno do circuito, por isso ele não apresenta *pads* para conexão com o exterior. Para que ele entre em operação, será fornecido pelo bloco "extração de *clock*" o relógio RF na frequência de 125khz. Também será fornecido o sinal de Power On Reset (POR), quando o TAG se encontrar na presença do campo magnético do leitor. Além disso, quando o leitor fizer contato com o TAG, o demodulador do bloco RF do circuito ira fornecer ao circuito digital os dados recebidos.

A única saída do circuito, será a saída do codificador de dados que fornecerá a resposta formada pelo TAG em Manchester ou DPE. Esta saída esta conectada ao modulador do circuito para poder reenviar ao leitor as informações exigidas.

## 4.4 Arquitetura proposta

Nesta sessão será apresentada a arquitetura da primeira versão do circuito digital do CHIP CALOR. Ela é composta por cinco diferentes blocos funcionais e uma memória. Esta memória será comprada na forma de um *intellectual property* (IP). A figura 4.12 apresenta esta arquitetura. Mesmo não fazendo parte do módulo digital apresentado no capítulo 2, a memória foi colocada na figura para representar sua ligação com os blocos do processador digital proposto.

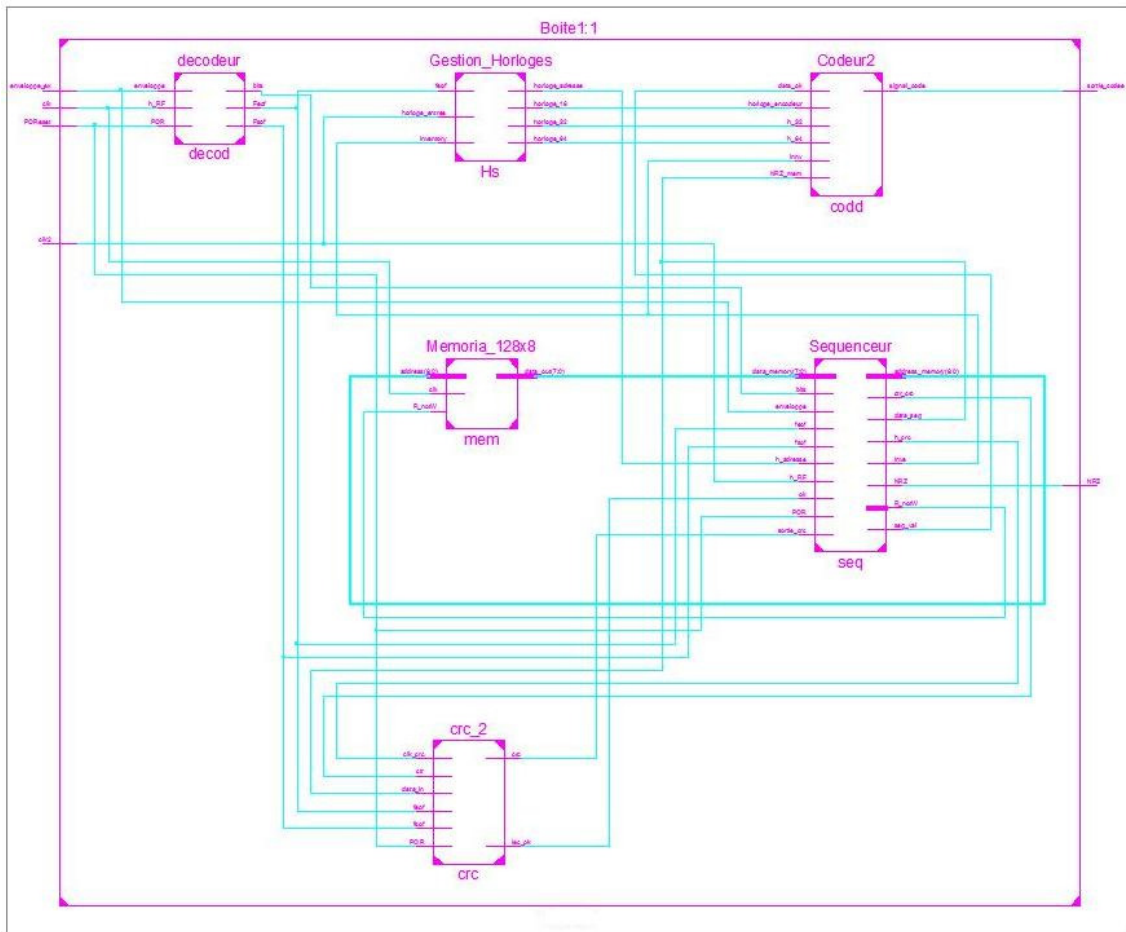


Figura 4.12: Arquitetura proposta

### 4.4.1 Décodeur

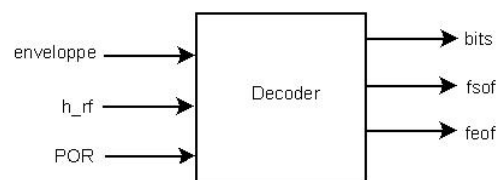


Figura 4.13: Estrutura do Décodeur

O bloco “décodeur” é o bloco responsável pela extração das informações enviadas pelo leitor. Ele identifica o SOF, na saída fsof, o EOF, na saída feof, e os bits enviados, na saída bits. Como já foi mencionado, a codificação utilizada pelo leitor é o PIE.

A entrada  $h_{rf}$  é o *clock* do bloco, ou seja, 125khz. O *Power On Reset* (POR) indica que o TAG encontra-se na presença de um leitor. O envelope vem da saída do demodulador do bloco analógico do circuito, com a informação codificada do leitor.

#### 4.4.2 Codeur

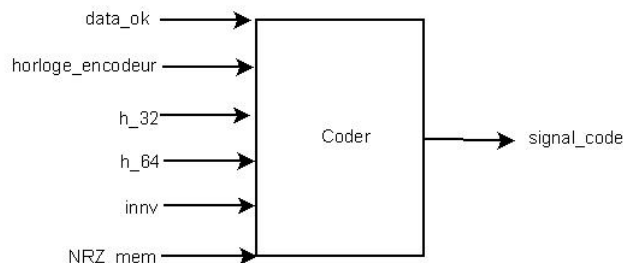


Figura 4.14: Estrutura do Codeur

Este bloco, quando “data\_ok” = ‘1’, codifica a entrada, no formato *non retour to zero* (NRZ) na entrada NRZ\_mem, para a codificação “Manchester” se “innv” = ‘0’ ou para DPE se “innv” = ‘1’. Os relógios fornecidos na entrada (h\_32 e h\_64) servem para determinar a velocidade da codificação, também em função do bit “innv”: 2kbps se “innv” = ‘1’ ou 4kbps caso “innv” = ‘0’. O horloge\_encodeur é o relógio do circuito, a 125khz.

#### 4.4.3 Gestion des horloges

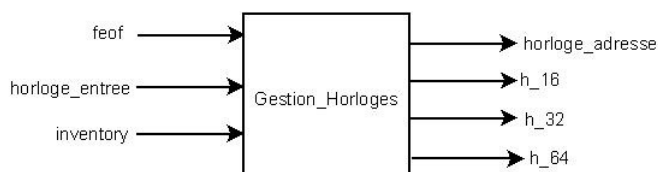


Figura 4.15: Estrutura do Gestion des horloges

O bloco “Gestion des Horloges” tem a finalidade de gerar as diferentes frequências necessárias ao funcionamento do sistema. As saídas deste bloco são o relógio RF a 125khz divididos por 16, 32 e 64. Uma quarta saída chamada “horloge\_adresse” foi adicionada a este bloco com a função de coordenar a codificação de dados segundo a velocidade exigida, a saber, 4kbps para a codificação “Manchester” e 2kbps para a “Dual Pattern Encoding”. Este sinal contém, ou o relógio dividido por 32, caso inventory = ‘0’, ou o relógio dividido por 64, caso inventory = ‘1’.

#### 4.4.4 CRC

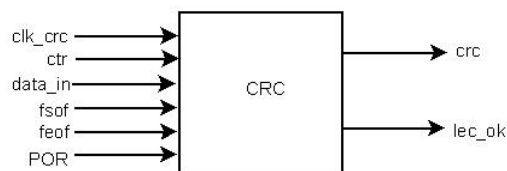


Figura 4.16: Estrutura do CRC

Este é o bloco responsável pelo cálculo do CRC, tanto para a verificação do comando recebido quanto pela resposta formada, se necessário. Uma proposta de arquitetura para este bloco pode ser encontrada em [1].

#### 4.4.5 Sequenceur

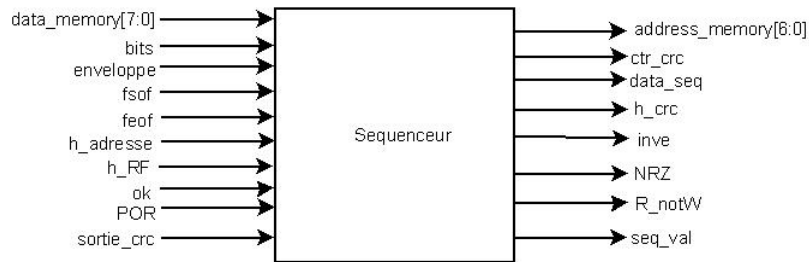


Figura 4.17: Estrutura do Séquenceur

Este bloco é o núcleo do módulo digital do projeto. Ele é o bloco mais complexo e é responsável pelo controle de todos os outros. Ele é responsável pela leitura de um comando, pelo envio de uma resposta e também pelo gerenciamento dos estados do TAG. Os parágrafos que seguem descrevem a realização de cada uma destas tarefas.

##### 4.4.5.1 Leitura de um Comando

Uma máquina de estados finitos (FSM) foi implementada para a realização desta tarefa. Como já foi visto, os parâmetros de um comando variam de acordo com os seus bits de flag.

A FSM começa em seu estado inicial e, ao detectar o recebimento de um SOF, o código do comando e seus flags são lidos. Em função destes dois parâmetros os outros dados que compõe o quadro são recebidos pelo seqüenciador.

Quando o comando é endereçado, o campo do SUID é comparado com o número de identificação do TAG. Em caso de diferença entre eles, a FSM entra num estado chamado "ERROR", o que leva à eliminação do quadro.

No caso de um comando com CRC anexado, os bits são lidos dentro do estado chamado "fin". A figura 4.18 mostra o diagrama de estados desta FSM.

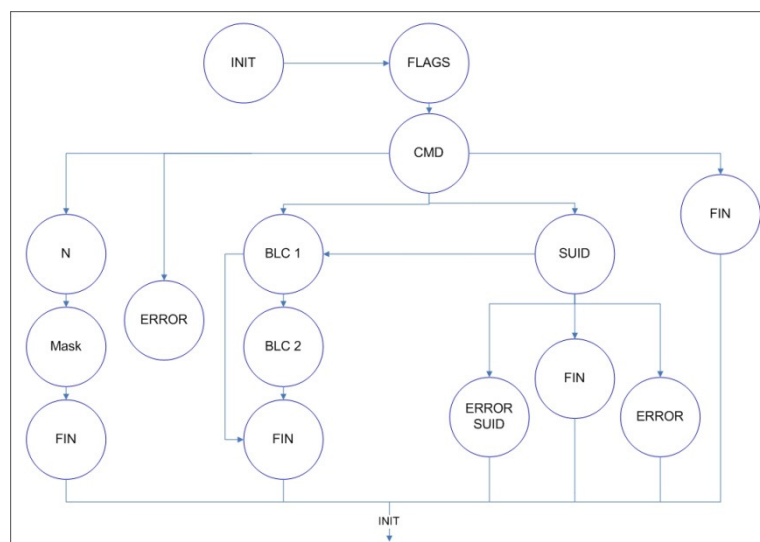


Figura 4.18: FSM de leitura de um comando

#### 4.4.5.2 Resposta a um comando

Outra FSM foi implementada para responder aos comandos recebidos. Se o comando exige uma resposta do TAG, os bits do quadro correspondente serão enviados em série para o leitor. Esses bits são colocados na saída “NRZ” do “Sequencer”.

Como foi discutido, uma resposta começa sempre pelo bit SOF. Em seguida, se o comando recebido não apresenta problemas, a FSM forma a resposta, seguindo as exigências do documento de normalização. Se, por outro lado, ocorrer um erro de recepção ou composição do comando, a FSM vai para o estado “ERROR” onde envia a mensagem de erro correspondente.

Por fim, se a resposta exige o cálculo de um CRC, ele é feito no estado “CRC” e será anexado ao fim do quadro, como previsto pela norma. A figura 4.19 ilustra os estados desta FSM.

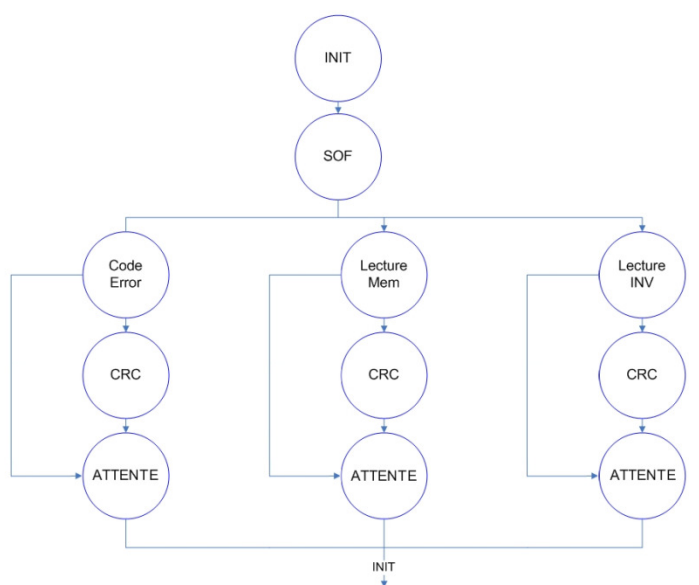


Figura 4.19: FSM de resposta a um comando

#### 4.4.5.3 Estados do TAG

Para que o TAG seja compatível com a norma ISO/IEC 18000-2, ele deve implementar uma máquina de estados com, no mínimo, 3 estados. São eles:

- Power-Off
- Quiet
- Ready

Com isso, foi apresentada uma arquitetura capaz de satisfazer as necessidades do processador digital do CHIP CALOR. Esta arquitetura possui 5 blocos funcionais, cada qual com uma função determinada com o objetivo de executar a lógica prevista pelo documento de padronização.

## 5 VALIDAÇÃO E IMPLEMENTAÇÃO DO BLOCO DE PROCESSAMENTO DIGITAL

Nesta sessão, será proposto e aplicado um banco de testes para a estrutura discutida no capítulo 4. Este teste deve mostrar que o VHDL responde como esperado aos comandos implementados, bem como leva em consideração os possíveis erros que podem ocorrer durante a comunicação com o leitor.

Tendo validado a versão comportamental do processador digital, é possível propor uma implementação para o circuito. O circuito será sintetizado em FPGA e também usando uma biblioteca de células no fluxo de concepção de um ASIC.

### 5.1 Testbench

É importante saber se o TAG responde corretamente a todas as variações possíveis de cada comando implementado e se ele respeita as características de seu estado. O testbench deverá ser escrito em acordo com a norma que guia o projeto. O banco de testes proposto foi dividido em 9 partes, como mostrado na tabela 5.1.

Tabela 5.1: Partes do Banco de Teste

Partes do <i>Testbench</i>	
Parte 1	Comando “STAY QUIET” e suas variações
Parte 2	Comando “INVENTORY” com “INV flag” = ‘1’ e suas variações
Parte 3	Comando “INVENTORY” com “INV flag” = ‘0’ e suas variações
Parte 4	Comando “READ SINGLE BLOCK” e suas variações
Parte 5	Comando “READ MULTIPLE BLOCKS” e suas variações
Parte 6	Testes de seqüências de envios
Parte 7	Excesso, Falta e Rotação de bits
Parte 8	Erros em flags e comandos não implementados
Parte 9	Estados do TAG

As cinco primeiras partes se referem aos comandos implementados. As três primeiras fazem referência aos comandos obrigatórios enquanto as partes 4 e 5 fazem referência aos dois comandos opcionais implementados. Serão testadas todas as

possíveis formas que esses comandos podem apresentar com todas as variações possíveis nos seus parâmetros e bits de flag. Os comandos são enviados, um de cada vez, para o TAG, que começa sempre no estado “POWER OFF”.

Em seguida, na parte 6, será testado o envio de alguns comandos em seqüência para provar que o TAG tem a capacidade de receber vários comandos e processá-los corretamente. Em todas as partes vistas até agora os comandos serão enviados corretamente em direção ao TAG. Validando o modelo até aqui, estaremos certos de que o TAG implementa e responde corretamente aos comandos que ele se propõe a interpretar.

Nas partes 7 e 8, serão testadas as habilidades do TAG em descartar comandos mal formados ou comandos que, embora corretos, não tenham sido implementados. Nestes casos, o TAG deve devolver ao leitor o código de erro correto, conforme previsto no documento de normalização.

Por fim, a parte 9 se consagra ao teste da máquina de estados do TAG. Em suma, esta parte realiza testes para saber se a máquina de estados funciona corretamente e se, dependendo do estado em que se encontra, o TAG respeita as restrições impostas na resposta dos comandos.

## 5.2 Resultados Obtidos

São discutidos, agora, os resultados obtidos após a aplicação do testbench. Como foi visto, ele foi dividido em 9 partes diferentes. Cada parte testa uma função específica que deve ser executada pelo TAG.

Como o testbench aplicado é bastante extenso, com muitos casos de teste, nós iremos discutir alguns exemplos analisando as formas de onda obtidas. Uma tabela com os resultados para todos os casos é apresentada em anexo. Os casos discutidos aqui foram pegos de forma arbitrária, apenas para fazer uma análise das formas de onda encontradas. Os casos visam mostrar os principais pontos exigidos pela norma, como a comparação dos SUIDs, dos TAGs, ou o cálculo do CRC, por exemplo.

### 5.2.1 Partes 1 a 5: A funcionalidade dos Comandos

O TAG obteve sucesso em todos os testes desta parte do *testbench*, ou seja, ele responde adequadamente a todos os comandos implementados e suas possíveis formas. Nós vamos, a título de exemplo, nos interessar ao comando *Read Single Block*.

#### 5.2.1.1 Read Single Block

Neste exemplo o quadro de bits enviados para o TAG é o seguinte :

Sof	Flags	RSB	SUID	Block Address	CRC	EoF
	10000	010000		00000000		

Figura 5.1: Comando *Read Single Block*

Isto quer dizer que o comando é endereçado e não temos a necessidade de fazer o cálculo do CRC, nem do comando nem da resposta. O bloco de memória que deve ser enviado é o bloco 0.

Como já foi visto, um comando endereçado só deve ser respondido por um TAG cujo SUID corresponde ao dado contido no parâmetro “SUID” do quadro enviado. Neste caso de teste, eles serão iguais, tanto para o TAG quanto para o quadro de comando.

A resposta esperada é, então, o envio do bloco de memória de endereço 0 para o leitor. A resposta de ser codificada em Manchester com uma velocidade de 4kbps.

A figura 5.2, mostra as formas de onda do comando enviado :



Figura 5.2: Comando Enviado

Nós podemos ver o quadro de bits enviado pelo leitor em direção ao TAG. Ele começa por um SOF e termina por um EOF com todos os parâmetros entre eles.

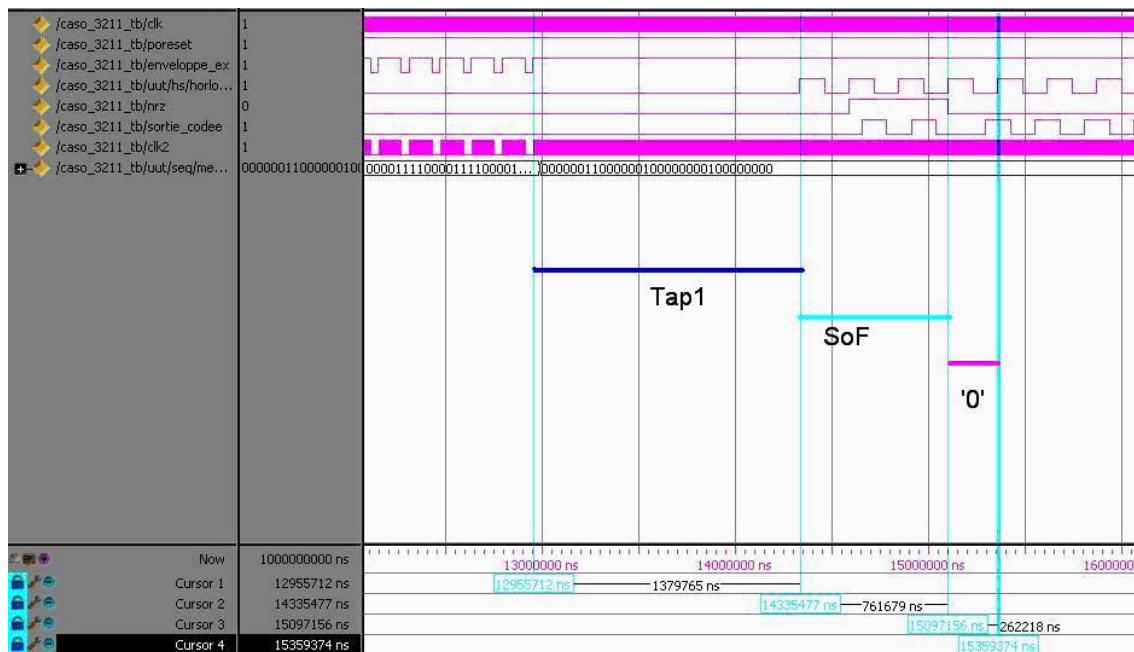


Figura 5.3: O tempo de Resposta Tap1

Na figura 5.3, uma observação deve ser feita. Antes de poder enviar uma resposta, o TAG deve obedecer a um tempo de espera igual a  $209 T_{ac}$ , [1]. Este tempo permite tanto ao leitor poder se preparar para receber a resposta como ao TAG de prepará-la. Como podemos ver, o TAG obedece esta imposição. Além disso, para todas as respostas, o TAG respeita o tempo.

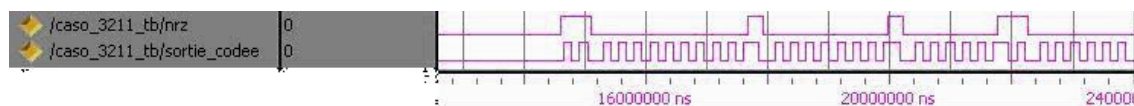


Figura 5.4: Resposta ao *Read Single Block*

Na figura 5.4, vemos a resposta do TAG ao *Read Single Block*. Ela está codificada em « Manchester » com uma velocidade de 4kbps como esperado. Finalmente, um bloco de memória de 32 bits é enviado.



### 5.2.1.2 Stay Quiet

Através de um segundo exemplo, nós vamos mostrar o cálculo do CRC pelo TAG. Será enviado um comando *Stay Quiet* endereçado (pela norma, todos os *Stay Quiet* devem ser endereçados) e com um pedido para que o cálculo do CRC seja feito. Dito isto, o “*CRCT flag*” e o “*ADR flag*” terão o valor 1. O comando formado pode ser visto na figura 5.5.

Sof	Flags	RSB	SUID	CRC	EoF
	10100	000001			

Figura 5.5: Comando « Stay Quiet » com pedido de CRC

Neste caso, o SUID do TAG coincidirá com o parâmetro SUID do comando. O parâmetro CRC, entretanto, contém um erro. A resposta esperada pelo TAG é o descarte do quadro. Assim, o TAG deve permanecer no seu estado de origem, que é o *Ready*.

Nós podemos ver na figura 5.6, a resposta do TAG a este comando. A comparação dos SUIDs obtém sucesso enquanto o cálculo do CRC não, o que leva o TAG a ignorar o pacote recebido, como já era esperado.

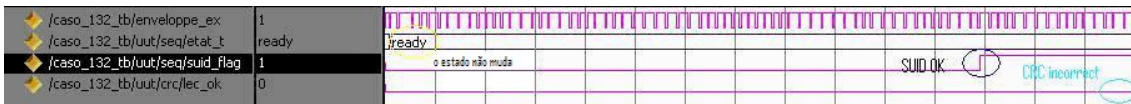


Figura 5.6: O TAG continua no estado *Ready*

### 5.2.2 Parte 6: Testes de seqüência de envio

Nós veremos um exemplo desta parte do *testbech* através do seguinte caso: um *Stay Quiet* será enviado após uma seqüência de bits aleatórios, como mostra a figura 5.7.

Randon	Sof	Flags	RSB	SUID	CRC	EoF
XXXXX		10000	000001			

Figura 5.7: “*Stay Quiet*” entre bits aleatórios

Como podemos ver na figura 5.8, o TAG ignora a seqüência de bits aleatórios e entra no estado *Quiet*, como esperado.

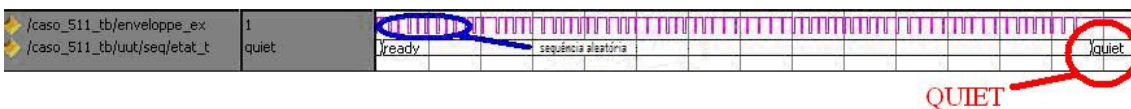


Figure 5.8: O TAG passa ao estado *Quiet*

### 5.2.3 Parte 7 e 8: Reação do TAG a comandos indesejados

O exemplo seguinte mostra o envio de um código de comando não implementado. Trata-se de um comando do tipo *custom*, o que deve levar o TAG a enviar a mensagem de erro « 001 » (*The command is not supported, i.e. the request code is not recognized*).

Sof	Flags	Custom	PN	CRC	EoF
	10100	101001			

Figura 5.9: Envio de um comando *custom* não implementado

Nós podemos ver na figura 5.10 que o TAG responde a este comando com o código de erro "001" conforme esperado, o que garante o sucesso do caso de teste.

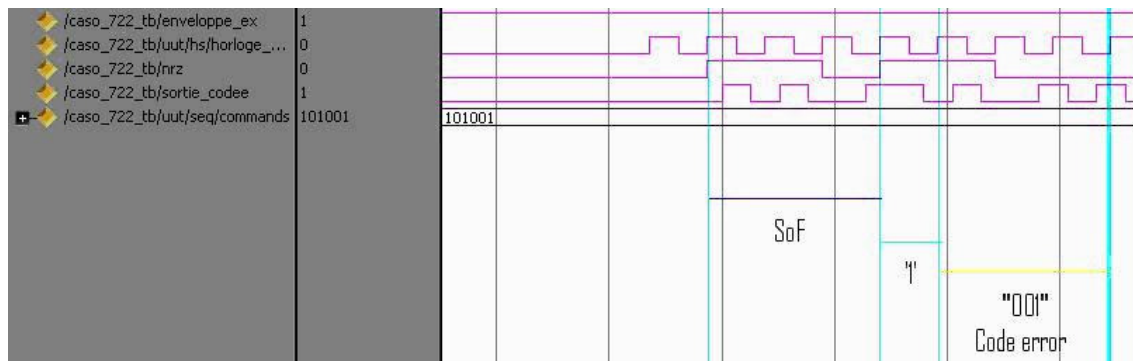


Figura 5.10: "Error Code" é 0x001

### 5.2.4 Parte 9 : Os estados do TAG

Tudo o que conseguimos provar até agora é que, saindo do estado *Ready*, o TAG responde aos comandos como deveria. Como foram implementados três estados (*Power-Off*, *Ready* e *Quiet*), para testar o funcionamento desta FSM, basta forçar o TAG ao estado *Quiet* e ver sua reação ao receber os diferentes comandos.

O exemplo seguinte ilustra esta situação. Forçamos o TAG ao estado *Quiet* através do comando *Stay Quiet* para, na seqüência, enviar um *Read Single Block* não endereçado. Isto deve levar o TAG a ignorar o quadro, sem enviar resposta ao leitor. A figura 5.11 mostra o quadro enviado pelo leitor.

Stay Quiet	Espera	RSB
	2ms	

Figura 5.11: Teste dos Estados do TAG

Com isso, terminamos a etapa de testes do Módulo Digital do CHIP CALOR.

## 5.3 Síntese do HDL

A síntese é o processo de transcrição de uma descrição comportamental de um arquivo HDL (VHDL, Verilog) em um arquivo estrutural. Em outras palavras, o comportamento que é capturado de um programa em VHDL, por exemplo, é sintetizado em um circuito que implemente a mesma função lógica. No fluxo de desenvolvimento utilizando uma biblioteca de células, o arquivo estrutural é construído a partir da células

disponíveis em uma biblioteca de *standard cells*. No fluxo utilizando circuitos programáveis, como FPGAs, são os componentes deste circuito que são estruturados de forma a implementar a função lógica desejada.

No fluxo de projeto de um ASIC usando *standard cell*, a biblioteca utilizada pode interferir bastante na qualidade do circuito gerado como resultado. Uma biblioteca com maior número de células, normalmente, tem a capacidade de gerar circuitos mais compactos e que consomem menos se forem escolhidas as células que melhor se adaptam as necessidades. Parte-se do princípio que as células implementadas funcionam corretamente e tenham desempenho compatível com outra biblioteca com um menor número de células.

No fluxo de projeto utilizando lógica programável, como os FPGAs, além do conjunto de bibliotecas utilizado na geração da *netlist* com os componentes, o dispositivo utilizado também pode interferir na performance final do circuito gerado. Um FPGA consiste em um grande arranjo de células lógicas ou blocos lógicos configuráveis contidos em um único circuito integrado. Cada célula contém capacidade computacional para implementar funções lógicas e realizar roteamento para comunicação entre elas.

Neste tipo de projeto, o tempo de desenvolvimento é relativamente curto mas o desempenho do circuito final gerado é menor se comparado com o mesmo circuito desenvolvido em ASIC. O fluxo em FPGA se mostra vantajoso quando poucas unidades do circuito serão produzidas ou em aplicações onde a performance não é um fator fundamental para o projeto. Se o circuito for produzido em larga escala ou performance for um fator crítico ao projeto então o fluxo de concepção de um ASIC deve ser utilizado.

Nesta sessão iremos sintetizar o VHDL do processador digital do CHIP CALOR utilizando FPGA e também usando uma biblioteca de células no fluxo *standard cell*.

#### 5.4 Fluxo *standard cell*

O VHDL comportamental foi sintetizado com a utilização de uma ferramenta comercial disponível no mercado. O circuito resultante possui 1985 células. A ferramenta não teve dificuldades para gerar o circuito tanto pela qualidade da biblioteca utilizada como pelas baixas restrições em termos de consumo, área e *timing* do projeto. Apenas lembrando que a única restrição do projeto é uma frequência mínima de 125kHz para o relógio do circuito. A tabela 5.2, abaixo, mostra alguns dos dados do circuito sintetizado. A coluna *Cell Área* apresenta a área ocupada pelas células enquanto a coluna *Net Área* apresenta a área ocupada pela rede de conexão entre elas.

Tabela 5.2: Dados de área do circuito digital

Instância	#Cells	Cell Área ( $\mu\text{m}^2$ )	Net Área ( $\mu\text{m}^2$ )
<b>Processador Digital</b>	2145	292867	31956
<b>Sequencer</b>	1892	257705	28913
<b>Crc</b>	89	14343	798

<b>Decodeur</b>	73	9047	527
<b>Gestion_horloges</b>	47	8499	251
<b>Codeur</b>	44	3273	188

Como era esperado o bloco de maior área, e com o maior número de células é o seqüenciador do circuito. Os outros blocos que compõe o processador digital são blocos simples, comandados e sincronizados pelo seqüenciador para a realização das tarefas.

A ferramenta de síntese também fez o cálculo do atraso máximo do circuito. O processo resultou em um atraso máximo de 7.99us, ou seja, frequência máxima de  $f_{max} = (1/7.99us) = 125,10kHz$ . O caminho crítico se encontra dentro do seqüenciador, o que é de se esperar devido às características do bloco.

Uma última análise que deve ser feita aqui é a de consumo do circuito. A tabela 5.3 mostra os dados extraídos da ferramenta de síntese. O circuito apresenta um *leakage* da ordem de 17 nW enquanto o consumo de chaveamento é da ordem de 6,5 mW respectivamente.

Apesar de o consumo de *leakage* ser significativamente menor que o consumo de chaveamento, é importante levá-lo em consideração pois ele está presente 100% do tempo enquanto o *switching power* apenas acontece quando o circuito está funcionando.

Tabela 5.3: Dados de Power do Processador Digital

<b>Instância</b>	<b>Leakage Power (nW)</b>	<b>Internal Power (nW)</b>	<b>Net Power (nW)</b>	<b>Switching Power (nW)</b>
<b>Processador Digital</b>	16.910	4568825.74	1979404.515	6548230.263
<b>Sequencer</b>	15,300	4253099,545	1771597,015	6024696,56
<b>Crc</b>	0.871	421154.950	167058.045	588212.995
<b>Decodeur</b>	0.534	88586.666	82627.875	171214.541
<b>Gestion_horloges</b>	0.447	5801.232	2423.025	8224.257
<b>Codeur</b>	0.21892	79950,8689	58004,65935	137955,5282

## 5.5 Fluxo FPGA

Nesta sessão, o processador digital proposto será sintetizado visando dispositivos reprogramáveis disponíveis comercialmente. Os dispositivos escolhidos são os FPGAs, devido a sua grande utilização. Serão comparados os resultados da síntese do VHDL em um dispositivo da XILINX e um dispositivo da Altera. Estes dois fabricantes detêm a maior parte do mercado neste ramo.

Primeiro, foi utilizado em dispositivo da família Virtex4 da Xilinx. Neste FPGA, a frequência máxima de funcionamento do circuito é de, aproximadamente 195 MHz, muito acima dos 125khz necessários. Sobre a utilização dos recursos, foram usadas 1050 *look-up tables* (LUTs) e 466 Slices. Uma LUT é um tipo de bloco lógico contém

células de armazenamento que são utilizadas para implementar pequenas funções lógicas. Normalmente um LUT possui 4 ou 5 entradas em FPGAs atuais.

No FPGA da Altera utilizado um dispositivo da família Cyclone IV, a frequência máxima de funcionamento encontrada foi de aproximadamente 135 Mhz. Sobre a utilização dos recursos, foram utilizados 1110 elementos lógicos (LÊS), o equivalente aos CLBs para a Xilinx e 466 registradores. A tabela 5.4 sintetiza os resultados obtidos.

Tabela 5.4: Comparação entre síntese em FPGA

	<b>Virtex 4 - XILINX</b>	<b>Cyclone IV- Altera</b>
<b>Fmax</b>	195Mhz	135Mhz
<b>CLBs/LEs</b>	1050	1110
<b>Flip-Flops</b>	466	466

Como podemos ver, os dois dispositivos apresentam desempenho semelhante. Os dois utilizam o mesmo número de *Flip-Flops* no circuito, a FPGA da Xilinx pode rodar numa frequência maior e ocupa um pouco menor de unidades lógicas.

Para a Xilinx, a principal unidade lógica para a implementação de lógica combinacional e sequencial é o CLB. Cada um desses CLBs é conectado à matriz de roteamento do FPGA. Cada CLB contém quatro Slices. Um Slice é composto por dois geradores de função, ou *look-up tables*, dois elementos de armazenamento, multiplexadores, e gates aritméticos [8]. A figura 5.14 mostra o formato de um CLB.

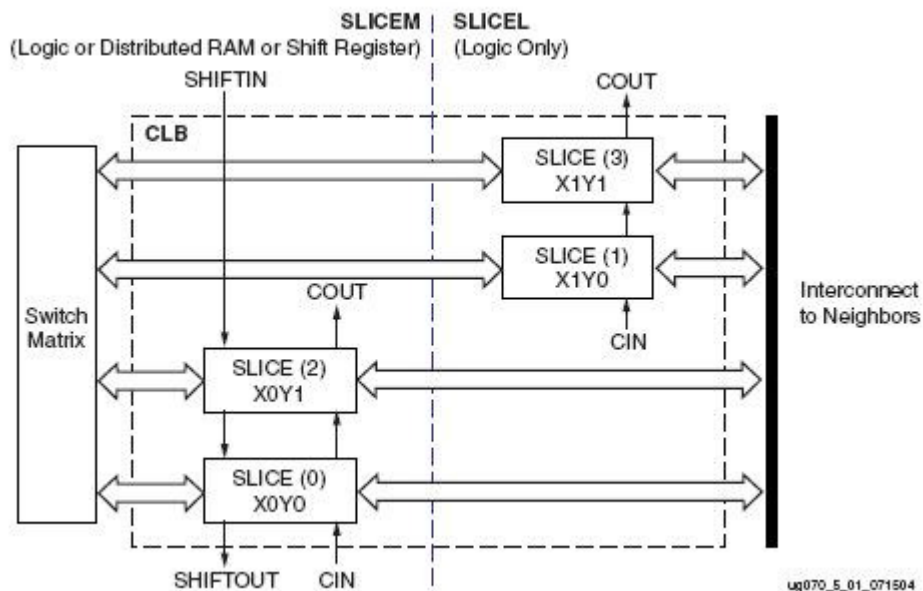


Figura 5.12: CLB presente no Virtex 4

Para a Altera, são os *logic elements* (LE), os menores elementos de lógica presentes no Cyclone IV. Eles são compactos e fornecem características avançadas para o uso de lógica eficiente. Cada LE possui 4 *look-up tables*, um registrador programável, além de

gates aritméticos. Além disso os LE são ligados à matriz de roteamento do FPGA [9]. A figura 5-13 mostra a arquitetura de um LE.

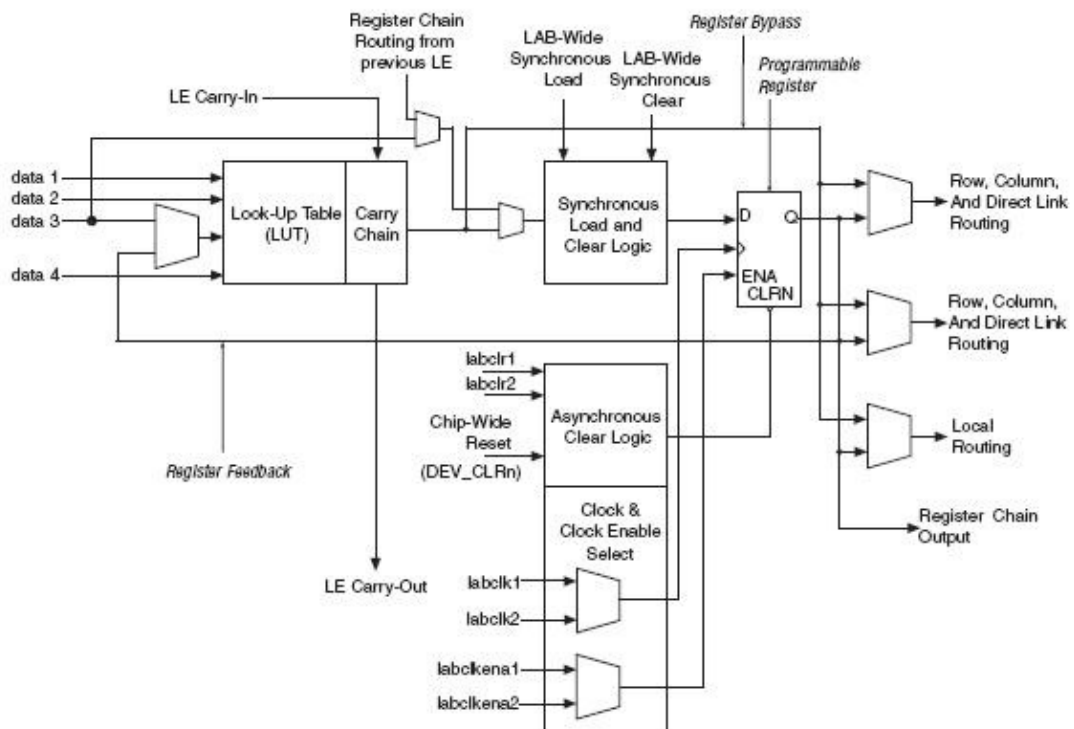


Figura 5.13: Arquitetura de um LE presente no Cyclone IV

## 5.6 Simulação Pós Síntese

Nesta parte, as simulações são feitas para verificar que o circuito mapeado realiza a mesma função lógica que sua descrição comportamental ou RTL. Isto porque é possível que ocorra um erro que cause mudança no comportamento do projeto. Na maioria das vezes, o erro é causado por uma descrição mal feita. Por exemplo, uma lista de sensibilidade de um processo VHDL mal formada.

Para esta verificação, é possível utilizar o mesmo *testbench* já utilizado. Se a resposta for a mesma, então a função não foi alterada. Com o aumento da complexidade do projeto, no entanto, são necessários cada vez mais vetores de teste e o tempo para esta abordagem aumenta bastante.

Outra opção é a utilização de técnicas de verificação formal. Esta estratégia compara diretamente a função lógica das duas descrições e prova, matematicamente, que elas são equivalentes. Este processo é mais complexo, mas ele permite ter certeza da equivalência entre as duas funções. Além disso, se o circuito for muito complexo, a verificação formal pode ser mais rápida do que a simulação normal.

Pode-se também fazer uma análise semântica e estrutural par verificar que algumas características mais simples não foram modificadas. Checa-se, por exemplo, se a largura de algum barramento foi modificada ou se existe alguma saída desconectada.

## 5.7 Static Timing Analysis (STA)

É preciso que o circuito, além de executar a função correta, execute essa função com a velocidade necessária. É nesta etapa que esta verificação é feita. Essencialmente, é testado se os tempos de *setup e hold* são respeitados. O *setup* especifica quanto tempo antes da borda de relógio a entrada deve estabilizar e o *hold* quanto tempo após a borda de relógio a entrada deve permanecer estável.

Esta é uma análise estática pois é independente dos vetores de entrada. Não são feitas simulações. Uma das conseqüências é a possibilidade de que não exista algum vetor de entrada capaz de sensibilizar o caminho que foi identificado pela ferramenta como sendo o caminho crítico. Esta situação é chamada de caminho falso. Neste caso, o atraso é irrelevante. Uma abordagem bastante utilizada é o projetista do circuito definir manualmente os caminhos falsos e, caso nenhum caminho deste tipo seja conhecido, então se considera que não há caminhos falsos.

Os atrasos das portas lógicas são fortemente influenciados pelo ambiente onde operam. Outra característica importante é a capacidade de analisar o comportamento para diferentes condições de funcionamento. Aumentos de temperatura e quedas na tensão de alimentação, por exemplo, diminuem a velocidade de um circuito. Se forem conhecidas a melhor e a pior condição de operação, é possível testar o circuito para estes dois casos e, caso ele respeite as especificações, assumir que o projeto opera corretamente nos demais casos.

## 5.8 Análise de consumo

O consumo de um circuito é uma característica extremamente importante, podendo ser, muitas vezes, fator decisivo na escolha entre dois produtos semelhantes. Portanto, deve-se avaliar se o desempenho do circuito esta dentro do aceitável.

Há dois tipos de consumo, o estático e o dinâmico. O estático é caracterizado pela energia gasta devido a parasitas, ou seja, mesmo que o circuito esteja parado existe consumo. O dinâmico corresponde à energia gasta devido ao chaveamento dos transistores, depende dos valores de entrada e do estado atual dos transistores, e é maior que o estático. A soma dos dois é o consumo total. De acordo com a tecnologia a relação entre os consumos, estático e dinâmico, pode ser tal que apenas o dinâmico é relevante para o circuito. No entanto, com a diminuição do tamanho dos transistores, a parcela correspondente ao consumo estático tende a ganhar importância para determinação do consumo total.

Como o consumo total depende da entrada do estado atual do circuito, uma análise precisa deve testar a maior quantidade possível de configurações. Obviamente, não é possível, a não ser em projetos pequenos, testar todas as possibilidades. É necessário escolher alguns vetores de entrada que avaliem adequadamente o consumo.

## 5.9 Síntese Física

O objetivo desta etapa é transformar o circuito mapeado em uma descrição que possa ser fabricada de acordo com as regras de uma tecnologia. Os passos desta fase são representados pelo fluxograma da figura 5.14. Não são mostradas as realimentações, mas cada fase que não obtiver os resultados esperados deve ser refeita.

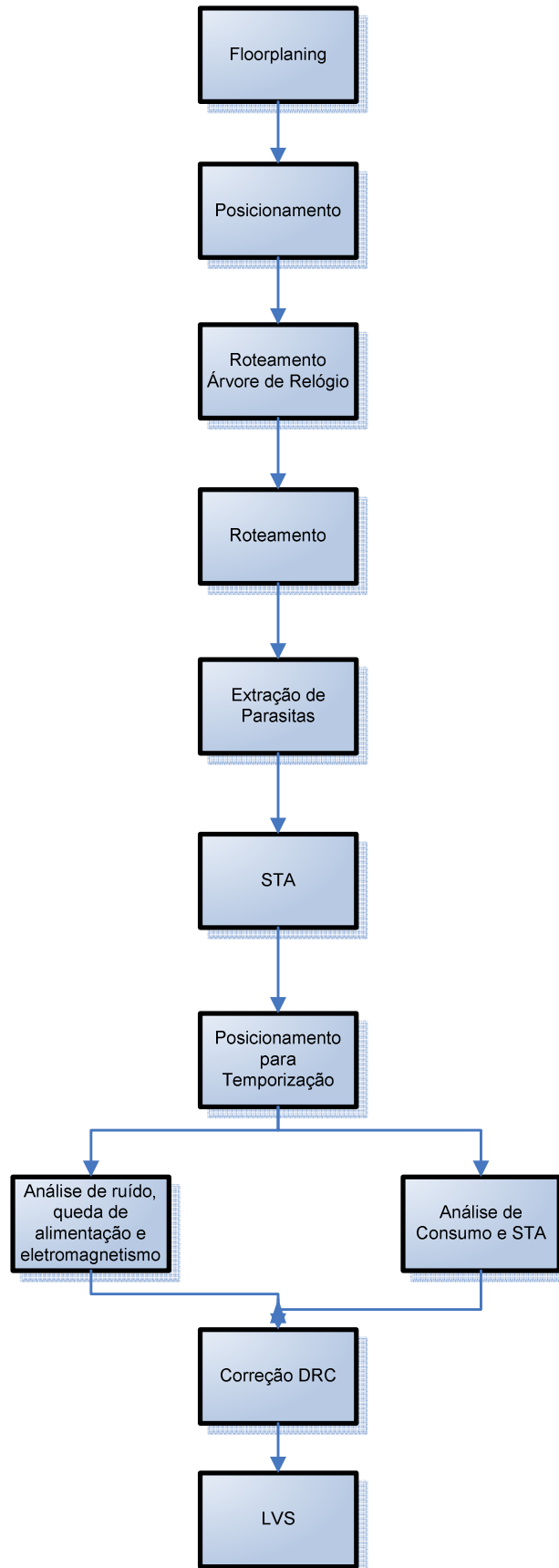


Figura 5.14: Fluxo de projeto síntese de leiaute



### 5.9.1 Posicionamento Floorplanning e Roteamento

Os módulos são colocados no chip de acordo com a sua necessidade de comunicação com os outros. Deseja-se que módulos que troquem informações estejam próximos de seus parceiros. Além disso, aqueles que têm conexões com o mundo exterior devem ficar próximos dos pinos de entrada e saída. O objetivo é fazer o posicionamento de maneira a diminuir o comprimento dos fios que conectarão as células.

A posição dos pinos de entrada e saída também precisa ser definida. É possível que estas localizações sejam definidas de acordo com a visão do sistema onde o circuito será usado.

As células são então posicionadas dentro dos módulos, nesta fase estima-se a quantidade de fio necessário para as conexões entre as células e tenta-se diminuí-la. Após, faz-se o roteamento, ou seja, realizam-se as conexões entre os elementos do projeto.

### 5.9.2 Roteamento da árvore de relógio

Os registradores presentes ficam em locais distintos dentro do circuito. Pode ocorrer, então, que o sinal de relógio não chegue a todos os elementos ao mesmo tempo. Este fenômeno é conhecido como *clock skew* e pode apresentar problemas de sincronização no circuito, especialmente quando a frequência do relógio é alta.

Nesta etapa, o objetivo é criar uma rede de distribuição do sinal que apresente o menor *skew* possível e que não sofra muitos problemas de degradação devido à propagação na linha. Devido a sua função de sincronização, é importante que o sinal de relógio possua uma boa qualidade. Oscilações indesejadas deste sinal podem ser interpretadas pelos registradores como bordas e ativarem os elementos seqüências do projeto.

O problema do *skew* pode ser minimizado tentando igualar as distâncias percorridas pelo relógio para chegar aos registradores. O problema de degradação de sinal é diminuído colocando *buffers* especiais, eles precisam possuir, na maioria das vezes, uma capacidade de fornecimento de corrente maior que os utilizados para lógica combinacional, para restaurar o sinal. No entanto, estes buffers possuem um atraso e, portanto, influenciam no *skew*. Além de ocuparem área e consumirem potência.

### 5.9.3 Extração de Parasitas

Até agora, ou considerava-se que os fios não possuíam nem capacitância nem resistência ou era utilizado um modelo que tentava prever o comportamento físico dos fios. Como consequência os atrasos estimados não são muito precisos. Nesta fase, analisa-se o circuito com a intenção de se conseguir valores reais para esses parasitas. Existem três métodos para fazer a extração. O primeiro assume que os valores são constantes ao longo dos fios. O segundo utiliza tabelas para uma melhor aproximação. O terceiro e mais preciso resolve as equações diferenciais que determinam o comportamento dos fios. Como esperado, os métodos mais precisos necessitam de um tempo maior para rodarem.

#### **5.9.4 Análise de temporização**

Com todas as informações de posicionamento, roteamento e parasitas do circuito, é possível estimar mais precisamente o comportamento temporal do circuito. Como a tendência é que o desempenho seja deteriorado conforme o projeto passa pelas diversas etapas já que os modelos utilizados nos cálculos são cada vez mais precisos conforme o projeto avança no fluxo. Esta nova análise pode ser um novo STA ou até mesmo a simulação do circuito em nível de transistores. Embora a simulação forneça resultados que são mais precisos, o tempo de execução é bem superior ao do STA.

#### **5.9.5 Análise de Ruído, de Queda da alimentação e Eletromagnetismo**

Linhas posicionadas próximas uma das outras podem ter problemas de ruído. Ou seja, parte do sinal pode ir de uma linha para outra através da capacitância existente entre eles. Esta interferência pode causar uma falha transiente, ou seja, que se extingue com o tempo, que pode comprometer o funcionamento do circuito.

Se o valor da alimentação diminuir durante o funcionamento do circuito, pode não ser possível fornecer a corrente elétrica necessária para o funcionamento do circuito. Novamente, existe uma falha transiente que o circuito deve suportar.

Variações de campos magnéticos, por sua vez, podem gerar uma corrente elétrica indesejada que leve o circuito a operar de maneira errada.

Todos estes problemas são aleatórios e, portanto, é necessário tentar estimar a força que eles terão e testar o circuito para os casos que são considerados os piores.

#### **5.9.6 Posicionamento dirigido à temporização**

Agora, existem informações sobre os parasitas do circuito. Pode-se então estimar o atraso proveniente dos fios das conexões entre as células. Por isso, é necessário pegar informações de parasitas, fazer a análise de temporização e, caso necessário, fazer modificações no posicionamento original e refazer as análises. Em grandes projetos, pode acontecer que este ciclo tenha que ser repetido diversas vezes até chegar-se a um momento em que novas melhorias acabam prejudicando outras otimizações feitas anteriormente.

O posicionamento para temporização tem como estratégia priorizar as células que fazem parte de caminhos críticos. Elas são posicionadas primeiro de forma que o comprimento dos fios que as conectam seja o menor possível e, conseqüentemente, o atraso entre as células também será reduzido.

#### **5.9.7 Análise de Consumo**

Pode-se optar por repetir a análise de consumo no fim do fluxo para aproveitar as informações sobre parasitas que agora estão disponíveis. Esta etapa ocorre da mesma forma que a primeira. Porém, como se tem informações mais detalhadas do circuito, o resultado é mais preciso.

#### **5.9.8 Correção das Regras de Desenho (DRC)**

Nesta fase o circuito deveria estar pronto para ser produzido. No entanto, devido às imperfeições nas ferramentas, é necessária, na maioria dos casos, uma correção no layout para que ele respeite as regras de desenho.



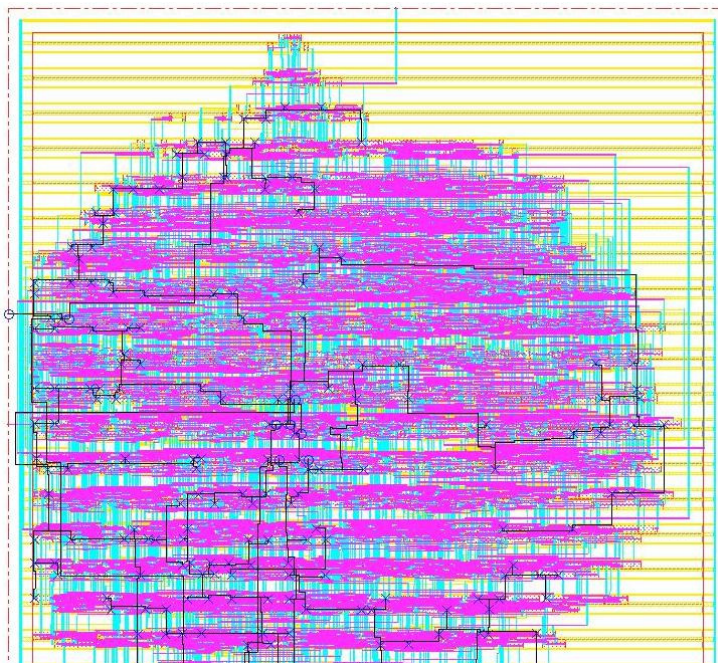


Figura 5.16: Roteamento das células do processador digital

Depois de feito o processo de *place and route*, é possível importar o arquivo gerado pela ferramenta e se chegar ao leiaute do circuito. Este circuito deve, então, ser ligado aos demais blocos que compõe o CHIP CALOR, apresentados no capítulo 2, o que é o objetivo maior do projeto.

Terminamos, assim, a implementação do processador digital que comporá o CHIP CALOR. Este módulo foi concebido no fluxo de projeto utilizando *standard cells*, visto que o objetivo final é a obtenção de um circuito integrado misto reunindo vários IPs. O circuito, como um todo, e conseqüentemente o módulo digital devem estar de acordo com a norma ISO/IEC 18000-2, que rege a comunicação de circuitos RFID em baixas frequências.

Para as próximas versões, dependendo dos testes e resultados em termos de comportamento do circuito inteiro, outras otimizações podem ser pensadas, como consumo do circuito e sua área total.

Para que o circuito possa ser fabricado, é preciso fazer com que todos os módulos do diagrama de blocos, desenvolvidos separadamente possam se comunicar entre si. Uma vez atingido este objetivo, será feito o que chamamos de *chip assembly*, processo pelo qual o leiaute de todo o circuito é desenhado. Com esse leiaute será possível enviar os desenhos das máscaras à *foundry* para a produção do primeiro protótipo.

## 6 CONCLUSÃO

Atualmente o mercado dos *transponders* RFID está em ascensão. As aplicações que empregam esta técnica são cada vez mais variadas e as suas vantagens perante os sistemas mais tradicionais de identificação são cada vez mais evidentes. O projeto CHIP CALOR se insere perfeitamente neste contexto e se apresenta como uma solução de alta versatilidade.

A vasta gama de aplicações que pode ser prevista para um transponder RFID com sensor de temperatura, que é a motivação final deste projeto, é um diferencial de suma importância. O monitoramento da cadeia do frio é tomado como uma aplicação de destaque dentro desta gama, que pode incluir o monitoramento da temperatura interna de circuitos eletrônicos ou peças mecânicas, ou ainda o monitoramento da secagem do concreto de uma viga, ou laje, através da combinação dos dados de temperatura e humidade, etc.

A proposta deste trabalho foi o desenvolvimento e validação da parte digital deste sistema, incluindo o módulo digital e os circuitos de interface com a memória. O conjunto destes blocos mostrou um bom funcionamento e uma boa adequação à norma.

Uma arquitetura para o módulo digital, dentro do diagrama de blocos do projeto CHIP CALOR, foi proposto. Sua arquitetura conta com 5 blocos funcionais e este processador responde a quatro comandos regulamentados pela norma ISO/IEC 18000-2. São os comandos obrigatórios “*Inventory*” e “*Stay Quiet*” além dos comandos opcionais “*Read Single Block*” e “*Read Multiple Blocks*”. Com a arquitetura dos blocos proposta, uma versão comportamental descrita em VHDL foi desenvolvida, seguida de um *testbench* para a sua validação.

A seqüência lógica do projeto foi a síntese do processador digital e o mapeamento tecnológico utilizando uma biblioteca de células para a tecnologia 0.35 $\mu$ . Como esperado, o circuito é capaz de respeitar a única restrição do projeto que é funcionar na frequência de 125kHz. O processador também foi sintetizado para alguns FPGAs disponíveis no mercado, com o objetivo de mostrar as performances que o módulo pode atingir neste tipo de circuito. Como o CHIP CALOR será fabricado em ASIC, devido ao objetivo de colocar vários IPs dentro de um circuito, o resto do desenvolvimento seguiu o fluxo de *standard cell*. Após o mapeamento tecnológico, foi feito o *place and route* do circuito.

O circuito final gerado possui em torno de 2000 células. O maior bloco, como esperado, é o seqüenciador do módulo, que deve interpretar as informações recebidas e formar as respostas esperadas, além de comandar a máquina de estados proposta pelo padrão adotado. É o seqüenciador, também, que sincroniza a lógica do módulo digital do chip, invocando todos os outros blocos do módulo para a execução das tarefas.

Fazendo a reutilização de alguns blocos funcionais de um chip já testado e em fase de fabricação e outros blocos já desenvolvidos anteriormente, faltará pouco a ser feito para que o circuito possa ser testado inteiramente. A conseqüente evolução do « CHIP CALOR » é a modificação, se necessária, de alguns blocos com o objetivo de permitir a comunicação entre eles, o desenho do leiaute do todo, as simulações pós-leiaute e, finalmente, o envio para fabricação e testes.

## REFERÊNCIAS

- [1] ISO/IEC 18000-2, First Edition, 2004-09-15
- [2] K. Finkenzeller, “RFID Handbook”, 2<sup>nd</sup> Edition. John Willey & Sons Ltd, 2003
- [3] Mattos, Diego Rossoni, Conception d'un TAG RFID avec Capteur de Température, 2008.
- [4] Pfeffer, Rodolfo Ambonati, Conception Etude et conception du circuit logique d'un chip RFID avec capteur de température, 2008
- [5] UTIYAMA, Lucas, Conception d'un TAG RFID en Technologie CMOS avec Capteur de Température Intégré, 2009
- [6] WARD, Matt ; Kranenburg, Rob van, RFID : Frequency, standards, adoption and innovation, 2006
- [7] Brunvand, Eric, “Digital VLSI Design with Cadence and Synopsys Tools”, 1st edition , Addison Wesley; 2009
- [8] Virtex-4 FPGA User Guide, UG070 (v2.6), 2008
- [9] Cyclone 4 Device Handbook, v 1.2, 2010
- [10] [http://www.normalizacao.cni.org.br/normas\\_tecnicas\\_internacionais.htm](http://www.normalizacao.cni.org.br/normas_tecnicas_internacionais.htm), acessado em 20/6/2010
- [11] RFID Standards White Paper, RFID Journal, 2005
- [12] Marcon, César; Bezerra, Eduardo, et. AL. “A 915 MHz Low Power RFID Tag”, SBCCI '07Proceedings of the 20th annual conference on Integrated circuits and systems design, 2007, pp. 276-281
- [13] WESTE, Neil H. E., et al; CMOS VLSI Design ; 3<sup>rd</sup> Edition, Pearson Education, 2006

- [14] Brock, D. 2001. The Physical Markup Language. MIT Auto-ID Center: Cambridge, MA. Available online at: <http://xml.coverpages.org/PML-MIT-AUTOID-WH->
- [15] THOMPSON, D. 2006. RFID technical tutorial. The Journal of Computing Sciences in Colleges, vol. 21, no. 5, pp. 8–9. Consortium for Computing Sciences in Colleges: USA.
- [16] Schürmann, Josef (1994) Einführung in die Hochfrequenz Identifikations Technologie, GME Technical Report, No. 13, Identifikationssysteme und kontaktlose Chipkarte, vde-Verlag, Berlin
- [17] Hassan, T. and Chatterjee, S. “A Taxonomy for RFID”. Hawaii International Conference on Systems Science. Jan. 2006
- [18] Leenaerts, D. “Low Power RF IC Design for Wireless Communication.” International Symposium on Low Power Electronics and Design. Pp. 428-433, August 2003
- [19] Porret, A-S. et al. “Tradeoffs and Design of an Ultra Low Power UHF Transceiver Integrated in a Standard Digital CMOS Process.” IEEE Journal of Solid-State Circuits, v. 35, n.3, March 2000
- [20] Giuseppe De Vita, Giuseppe Iannaccone ; “Ultra Low Power RF Section of a Passive Microwave RFID Transponder in 0.35  $\mu\text{m}$  BiCMOS” IEEE MTT-S Int. Microwave Symp. Circuits Systems (ISCAS 2005), Kobe, Japan, May 2005, pp. 5075-5078.
- [21] Sandeep S. Kumar and Christof Paar , “Are standards compliant Elliptic Curve Cryptosystems feasible on RFID?” , Workshop on RFID Security, 2006
- [22] L. Batina<sup>1</sup>, J. Guajardo<sup>2</sup>, T. Kerins<sup>2</sup>, N. Mentens<sup>1</sup>, P. Tuyls<sup>2</sup>, and I. Verbauwhede<sup>1</sup>; “An Elliptic Curve Processor Suitable For RFID-Tags”, Benelux Workshop Information and System Security (WISSec 06), 2006
- [23] Feldhofer Martin, et al.; “Strong Authentication for RFID Systems Using the AES Algorithm”, Cryptographic Hardware and Embedded Systems - CHES 2004



## ANEXO <RESULTADOS DO TESTBENCH>

### A.1: Parte 1: « STAY QUIET »

1.1 Os bits « SEL flag » e « ADR flag » em 00 ou em 11

Sof	Flags	Stay Quiet	SUID	CRC	EoF
	00000	000001			

Sof	Flags	Stay Quiet	SUID	CRC	EoF
	11000	000001			

Caso	SEL=0 e ADR=0
Resposta Esperada	O TAG não responde e continua no seu estado atual
Resposta do TAG	O TAG não responde e continua no seu estado atual Estado Inicial = « Ready » ; Estado Final = « Ready » ;
Comentários	
Resultado	Aprovado

Caso	SEL=1 e ADR=1
Resposta Esperada	O TAG não responde e continua no seu estado atual
Resposta do TAG	O TAG não responde e continua no seu estado atual Estado Inicial = « Ready » ; Estado Final = « Ready » ;
Comentários	
Resultado	Aprovado

« SEL flag » = 0 e « ADR flag » = 1

Sof	Flags	Stay Quiet	SUID	<del>CRC</del>	EoF
	10000	000001			

Caso	SUID correto
Resposta Esperada	O TAG não responde, passando ao estado «QUIET»
Resposta do TAG	O TAG não responde, passando ao estado «QUIET» Estado Inicial = « Ready » ; Estado Final = « Quiet » ;
Comentários	
Resultado	Aprovado

Caso	SUID incorreto
Resposta Esperada	O TAG não responde e continua no seu estado atual
Resposta do TAG	O TAG não responde e continua no seu estado atual Estado Inicial = « Ready » ; Estado Final = « Ready » ;
Comentários	
Resultado	Aprovado

« SEL flag » = 0; « ADR flag » = 1; « CRCT flag » = 1 e SUID correto

Sof	Flags	Stay Quiet	SUID	CRC	EoF
	10100	000001			

Caso	CRC correto
Resposta Esperada	O TAG não responde, passando ao estado «QUIET»
Resposta do TAG	O TAG não responde, passando ao estado «QUIET» Estado Inicial = « Ready » ; Estado Final = « Quiet » ;
Comentários	

Resultado	Aprovado
Caso	CRC incorreto
Resposta Esperada	O TAG não responde e continua no seu estado atual
Resposta do TAG	O TAG não responde e continua no seu estado atual Estado Inicial = « Ready » ; Estado Final = « Ready » ;
Comentários	
Resultado	Aprovado

« SEL flag » = 0; « ADR flag » = 1; « CRCT flag » = 1 e SUID incorreto

Sof	Flags	Stay Quiet	SUID	CRC	EoF
	10100	000001			

Caso	CRC correto
Resposta Esperada	O TAG não responde e continua no seu estado atual
Resposta do TAG	O TAG não responde e continua no seu estado atual Estado Inicial = « Ready » ; Estado Final = « Ready » ;
Comentários	Cálculo do CRC : ok Comparação do SUID : ok
Resultado	Aprovado

Caso	CRC incorreto
Resposta Esperada	O TAG não responde e continua no seu estado atual
Resposta do TAG	O TAG não responde e continua no seu estado atual Estado Inicial = « Ready » ; Estado Final = « Ready » ;
Comentários	Cálculo do CRC : ok

	Comparação do SUID : ok
Resultado	Aprovado

## Comentários da parte 1 :

Todos os casos desta parte do testbench foram aprovados

**A.2 : Parte 2 : « Inventory » com « Inventory flag » = 0**

## 2.1 : « CRCT flag » = 1

Sof	Flags	Inventory	AFI	CRC	EoF
	10X00	000000			

Caso	2.1.1 : CRC correto
Resposta Esperada	O TAG responde ao comando enviando seu SUID
Resposta do TAG	O TAG responde ao comando enviando seu SUID
Comentários	$T_{ap1} = 209 T_{ac}$ Envio do SUID : ok Cálculo e envio do CRC : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado

Caso	2.1.2 : CRC incorreto
Resposta Esperada	O TAG fica em silêncio
Resposta do TAG	O TAG fica em silêncio
Comentários	Cálculo do CRC : ok
Resultado	Aprovado

## 2.2 : « CRCT flag » = 0

Caso	2.2.1: « CRCT flag » = 0
Resposta Esperada	O TAG envia seu SUID

Resposta do TAG	O TAG envia seu SUID
Comentários	$T_{ap1} = 209 T_{ac}$ Envio do SUID : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado

Comentários da parte 2 :

Todos os casos desta parte do testbench foram aprovados

**A.3 : Parte 3: « Inventory » com « Inventory flag » = 1**

3.1 : « NOS flag » = 0 e « Mask length » = 4

Caso	3.1.1 : « Mask Value » = LSB do SUID
Resposta Esperada	O TAG envia os 40 bits de peso forte do seu SUID no slot 3
Resposta do TAG	O TAG envia os 40 bits de peso forte do seu SUID no slot 3
Comentários	$T_{ap1} = 209 T_{ac}$ Envio do SUID : ok Codificação e velocidade : <i>DPE</i> a 2 kbps
Resultado	Aprovado

Caso	3.1.2 : « Mask Value » $\neq$ LSB do SUID
Resposta Esperada	O TAG executa a seqüência anticolisão mas fica em silêncio pois seu SUID não é igual à máscara
Resposta do TAG	O TAG executa a seqüência anticolisão mas fica em silêncio pois seu SUID não é igual à máscara
Comentários	
Resultado	Aprovado

3.2 : « NOS flag » = 0 e « Mask length » = 8

Caso	3.1.1 : « Mask Value » = LSB do SUID
Resposta	O TAG envia os 36 bits de peso forte do seu SUID no slot 3

Esperada	
Resposta do TAG	O TAG envia os 36 bits de peso forte do seu SUID no slot 3
Comentários	$T_{ap1} = 209 T_{ac}$ Envio do SUID : ok Codificação e velocidade : <i>DPE</i> a 2 kbps
Resultado	Aprovado

3.3 : « NOS flag » = 1 e « Mask length » = 8

Caso	3.3.1 : « Mask Value » = LSB do SUID
Resposta Esperada	O TAG envia os 40 bits de peso forte do seu SUID no slot 3
Resposta do TAG	O TAG envia os 40 bits de peso forte do seu SUID no slot 3
Comentários	$T_{ap1} = 209 T_{ac}$ Envio do SUID : ok Codificação e velocidade : <i>DPE</i> a 2 kbps
Resultado	Aprovado

Comentários da parte 3 :

Todos os casos desta parte do testbench foram aprovados

#### A.4: Parte 4: « Read Single Block »

4.1 : « Inventory flag » = 0 ; « SEL flag » = 0 ; « ADR flag » = 0

Sof	Flags	RSB	<del>SUID</del>	Block Address	CRC	EoF
	00X00	010000	<del></del>	XXXXXXXXXX		

4.1.1 : « CRCT flag » = 1

4.1.1.1: CRC correto

Caso	4.1.1.1.1 : « Block Address » = 0
Resposta Esperada	O TAG envia o bloco de memória número 0
Resposta do TAG	O TAG envia o bloco de memória número 0
Comentários	$T_{ap1} = 209 T_{ac}$

	Envio do bloco ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado
Caso	4.1.1.1.2 : « Block Address » = 31
Resposta Esperada	O TAG envia o bloco de memória número 31
Resposta do TAG	O TAG envia o bloco de memória número 31
Comentários	$T_{ap1} = 209 T_{ac}$ Envio do bloco ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado
Caso	4.1.1.1.3 : « Block Address » = 255
Resposta Esperada	O TAG envia o código de erro 3 : « The specified block is not available »
Resposta do TAG	O TAG envia o código de erro 3 : « The specified block is not available »
Comentários	$T_{ap1} = 209 T_{ac}$ Envio do bloco ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado
4.1.1.2: CRC incorreto	
Caso	4.1.1.2.1 : « Block Address » = 15
Resposta Esperada	O TAG ignora o comando e fica em silêncio
Resposta do TAG	O TAG ignora o comando e fica em silêncio
Comentários	
Resultado	Aprovado
4.1.2 : « CRCT flag » = 0	

Caso	4.1.2.1 : « Block Address » = 0
Resposta Esperada	O TAG envia o bloco de memória número 0
Resposta do TAG	O TAG envia o bloco de memória número 0
Comentários	$T_{ap1} = 209 T_{ac}$ Envio do bloco ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado

Caso	4.1.2.2 : « Block Address » = 31
Resposta Esperada	O TAG envia o bloco de memória número 31
Resposta do TAG	O TAG envia o bloco de memória número 31
Comentários	$T_{ap1} = 209 T_{ac}$ Envio do bloco ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado

Caso	4.1.2.3 : « Block Address » = 255
Resposta Esperada	O TAG envia o código de erro 3 : « The specified block is not available »
Resposta do TAG	O TAG envia o código de erro 3 : « The specified block is not available »
Comentários	$T_{ap1} = 209 T_{ac}$ Envio do bloco ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado

4.2 : « Inventory flag » = 0 ; « SEL flag » = 0 ; « ADR flag » = 1

Sof	Flags	RSB	SUID	Block Address	CRC	EOF
	00X00	010000		XXXXXXXX		

4.2.1: SUID correto



Caso	4.2.1.1 : « CRCT flag » = 0 e « Block Address » = 255
Resposta Esperada	O TAG envia o código de erro 3 : « The specified block is not available »
Resposta do TAG	O TAG envia o código de erro 3 : « The specified block is not available »
Comentários	$T_{ap1} = 209 T_{ac}$ Envio do bloco ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado

Caso	4.2.1.2 : « CRCT flag » = 0 e « Block Address » = 0
Resposta Esperada	O TAG envia o bloco de memória número 0
Resposta do TAG	O TAG envia o bloco de memória número 0
Comentários	$T_{ap1} = 209 T_{ac}$ Envio do bloco ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado

Caso	4.2.1.3 : « CRCT flag » = 1; « Block Address » = 15 e CRC correto
Resposta Esperada	O TAG envia o bloco de memória número 15
Resposta do TAG	O TAG envia o bloco de memória número 15
Comentários	$T_{ap1} = 209 T_{ac}$ Envio do bloco ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado

Caso	4.2.1.4 : « CRCT flag » = 1; « Block Address » = 15 e CRC incorreto
Resposta Esperada	O TAG ignora o comando e fica em silêncio
Resposta do TAG	O TAG ignora o comando e fica em silêncio

Comentários	
Resultado	Aprovado

---

4.2.2: SUID incorreto

---

Caso	4.2.2.1 : « CRCT flag » = 0 e « Block Address » = 15
Resposta Esperada	O TAG ignora o comando e fica em silêncio
Resposta do TAG	O TAG ignora o comando e fica em silêncio
Comentários	
Resultado	Aprovado

Caso	4.2.2.2 : « CRCT flag » = 1; CRC correto e « Block Address » = 15
Resposta Esperada	O TAG ignora o comando e fica em silêncio
Resposta do TAG	O TAG ignora o comando e fica em silêncio
Comentários	Cálculo do CRC : ok
Resultado	Aprovado

Comentários da parte 4 :

Todos os casos desta parte do testbench foram aprovados

---

### A.5: Parte 5: « Read Multiple Blocks »

5.1 : « Inventory flag » = 0 ; « SEL flag » = 0 ; « ADR flag » = 0; « CRCT flag » = 0

Sof	Flags	RMB	<del>SUID</del>	First Block	Number Of Blocks	<del>CRC</del>	EoF
	00000	010010					

Caso	5.1.1 : « First Block » = 0 ; « Number of Blocks » = 0
Resposta Esperada	O TAG envia o bloco de memória número 0
Resposta do TAG	O TAG envia o bloco de memória número 0
Comentários	$T_{ap1} = 209 T_{ac}$ Envio dos blocos ou código de erro : ok

	Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado
Caso	5.1.2 : « First Block » = 0 ; « Number of Blocks » = 31
Resposta Esperada	O TAG envia os blocos de memória do número 0 ao 31
Resposta do TAG	O TAG envia os blocos de memória do número 0 a0 31
Comentários	$T_{ap1} = 209 T_{ac}$ Envio dos blocos ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado
Caso	5.1.3 : « First Block » = 0 ; « Number of Blocks » = 255
Resposta Esperada	O TAG envia o código de erro 3 : « The specified block is not available »
Resposta do TAG	O TAG envia o código de erro 3 : « The specified block is not available »
Comentários	$T_{ap1} = 209 T_{ac}$ Envio dos blocos ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado
Caso	5.1.4 : « First Block » = 31 ; « Number of Blocks » = 0
Resposta Esperada	O TAG envia o bloco de memória número 31
Resposta do TAG	O TAG envia o bloco de memória número 31
Comentários	$T_{ap1} = 209 T_{ac}$ Envio dos blocos ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado
Caso	5.1.5 : « First Block » = 31 ; « Number of Blocks » = 255
Resposta Esperada	O TAG envia o código de erro 3 : « The specified block is not available »

Resposta do TAG	O TAG envia o código de erro 3 : « The specified block is not available »
Comentários	$T_{ap1} = 209 T_{ac}$ Envio dos blocos ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado

Caso	5.1.6 : « First Block » = 255 ; « Number of Blocks » = X
Resposta Esperada	O TAG envia o código de erro 3 : « The specified block is not available »
Resposta do TAG	O TAG envia o código de erro 3 : « The specified block is not available »
Comentários	$T_{ap1} = 209 T_{ac}$ Envio dos blocos ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado

5.2 : « Inventory flag » = 0 ; « SEL flag » = 0 ; « ADR flag » = 1 ; « CRCT flag » = 0

Sof	Flags	RMB	SUID	First Block	Number Of Blocks	<del>CRCT</del>	EoF
	00000	010010					

#### 5.2.1: SUID correto

Caso	5.2.1.1 : « First Block » = 0 ; « Number of Blocks » = 0
Resposta Esperada	O TAG envia o bloco de memória número 0
Resposta do TAG	O TAG envia o bloco de memória número 0
Comentários	$T_{ap1} = 209 T_{ac}$ Envio dos blocos ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado

Caso	5.2.1.2 : « First Block » = 255 ; « Number of Blocks » = 255
Resposta Esperada	O TAG envia o código de erro 3 : « The specified block is not available »
Resposta do TAG	O TAG envia o código de erro 3 : « The specified block is not

TAG	available »
Comentários	$T_{ap1} = 209 T_{ac}$ Envio dos blocos ou código de erro : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado

## 5.2.2: SUID incorreto

Caso	5.2.2.1 : « First Block » = 0 ; « Number of Blocks » = 0
Resposta Esperada	O TAG ignora o comando e fica em silêncio
Resposta do TAG	O TAG ignora o comando e fica em silêncio
Comentários	
Resultado	Aprovado

5.3 : « Inventory flag » = 0 ; « SEL flag » = 0 ; « ADR flag » = 0; « CRCT flag » = 1

Sof	Flags	RMB	<del>SUID</del>	First Block	Number Of Blocks	CRC	EoF
	00000	010010					

Caso	5.3.1 : CRC correto; « First Block » = 0 ; « Number of Blocks » = 1
Resposta Esperada	O TAG envia o bloco de memória número 0 e 1
Resposta do TAG	O TAG envia o bloco de memória número 0 e 1
Comentários	$T_{ap1} = 209 T_{ac}$ Envio dos blocos ou código de erro : ok Cálculo do CRC : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado

Caso	5.3.2 : CRC incorreto; « First Block » = 0 ; « Number of Blocks » = 0
Resposta Esperada	O TAG ignora o comando e fica em silêncio
Resposta do	O TAG ignora o comando e fica em silêncio

TAG	
Comentários	
Resultado	Aprovado

5.4 : « Inventory flag » = 0 ; « SEL flag » = 0 ; « ADR flag » = 1 ; « CRCT flag » = 1

Sof	Flags	RMB	SUID	First Block	Number Of Blocks	CRC	EoF
	10100	010010					

#### 5.4.1: CRC incorreto

Caso	5.4.1.1 : SUID correto; « First Block » = 0 ; « Number of Blocks » = 0
Resposta Esperada	O TAG ignora o comando e fica em silêncio
Resposta do TAG	O TAG ignora o comando e fica em silêncio
Comentários	
Resultado	Aprovado

#### 5.4.2 : CRC correto

Caso	5.4.2.1 : SUID correto; « First Block » = 0 ; « Number of Blocks » = 1
Resposta Esperada	O TAG envia o bloco de memória número 0 e 1
Resposta do TAG	O TAG envia o bloco de memória número 0 ET 1
Comentários	$T_{ap1} = 209 T_{ac}$ Envio dos blocos ou código de erro : ok Cálculo do CRC : ok Codificação e velocidade : <i>Manchester</i> a 4 kbps
Resultado	Aprovado

Caso	5.4.2.2 : SUID incorreto; « First Block » = 0 ; « Number of Blocks » = 1
Resposta Esperada	O TAG ignora o comando e fica em silêncio
Resposta do TAG	O TAG ignora o comando e fica em silêncio

Comentários	
Resultado	Aprovado

Comentários da parte 5 :

Todos os casos desta parte do testbench obtiveram sucesso.

#### A.6: Parte 6 : Testes de seqüências de envio

Em todos os casos desta parte, uma seqüência de bits pseudo-aleatórios será enviado ao TAG entre seqüências de comandos conhecidos. Em todos os casos, o TAG deve ignorar os bits aleatórios e responder aos comandos enviados.

Caso	6.1 : Aleatório + « STAY QUIET » + Aleatório
Resposta Esperada	O TAG responde ao comando « STAY QUIET »
Resposta do TAG	O TAG responde ao comando « STAY QUIET »
Comentários	
Resultado	Aprovado

Caso	6.2 : Aleatório + « READ SINGLE BLOCK » + Aleatório
Resposta Esperada	O TAG responde ao comando « READ SINGLE BLOCK »
Resposta do TAG	O TAG responde ao comando « READ SINGLE BLOCK »
Comentários	
Resultado	Aprovado

Caso	6.3 : Aleatório + « STAY QUIET » + « READ SINGLE BLOCK » + Aleatório
Resposta Esperada	O TAG responde aos comandos « STAY QUIET » + « READ SINGLE BLOCK »
Resposta do TAG	O TAG responde aos comandos « STAY QUIET » + « READ SINGLE BLOCK »
Comentários	
Resultado	Aprovado

Caso	6.4 : Aleatório + « READ SINGLE BLOCK » + Aleatório + « STAY QUIET » + Aleatório
Resposta	O TAG responde aos comandos « READ SINGLE BLOCK » +

Esperada	« STAY QUIET »
Resposta do TAG	O TAG responde aos comandos « READ SINGLE BLOCK » + « STAY QUIET »
Comentários	
Resultado	Aprovado

#### A.7 : Parte 7 : O excesso, a falta e a rotação de bits

Nesta parte, os bits serão misturados a alguns bits indesejados. Em todos os casos o TAG deve permanecer no seu estado e ignorar o quadro recebido.

Caso	7.1 : EoF + bits do « STAY QUIET » + SoF + 0 + 0 + EoF
Resposta Esperada	TAG ignora o quadro
Resposta do TAG	TAG ignora o quadro
Comentários	
Resultado	Aprovado

Caso	7.2 : « STAY QUIET » sem EoF
Resposta Esperada	TAG ignora o quadro
Resposta do TAG	TAG responde o « STAY QUIET » pois ele reconhece o EoF quando o leitor fica em silêncio.
Comentários	Essa situação não acontece em uso
Resultado	<b>Aprovado</b>

Caso	7.3 : « STAY QUIET » sem SoF
Resposta Esperada	TAG ignora o quadro
Resposta do TAG	TAG ignora o quadro
Comentários	
Resultado	Aprovado

Caso	7.4 : « STAY QUIET » sem SoF + SoF
------	------------------------------------



Resposta Esperada	TAG ignora o quadro
Resposta do TAG	TAG ignora o quadro, mas ele entre no estado « flags » da máquina “LEITURA”. Isto não apresenta grandes problemas pois, a cada recepção de um SoF a máquina de estados é reinicializada.
Comentários	
Resultado	Aprovado

#### **A.8 : Parte 8 : Erros em flags e comandos não implementados**

Caso	8.1 : Código de um comando não implementado
Resposta Esperada	TAG envia o código de erro «001»: «The command is not supported, i.e., the request code is not recognized”
Resposta do TAG	TAG ignora o quadro
Comentários	
Resultado	Aprovado

Caso	8.2 : Código de um comando <i>custom</i>
Resposta Esperada	TAG envia o código de erro «001»: «The command is not supported, i.e., the request code is not recognized”
Resposta do TAG	TAG ignora o quadro
Comentários	
Resultado	Aprovado

Caso	8.3 : Código de um comando <i>proprietary</i>
Resposta Esperada	TAG envia o código de erro «001»: «The command is not supported, i.e., the request code is not recognized”
Resposta do TAG	TAG ignora o quadro
Comentários	
Resultado	Aprovado

Caso	8.4 : Código de um comando <i>opcional</i> não implementado
Resposta	TAG envia o código de erro «001»: «The command is not supported,

Esperada	i.e., the request code is not recognized»
Resposta do TAG	TAG envia o código de erro «001»: «The command is not supported, i.e., the request code is not recognized»
Comentários	
Resultado	Aprovado

Caso	8.5 : SoF + bit + comando + EoF
Resposta Esperada	TAG envia o código de erro «001»: «The command is not supported, i.e., the request code is not recognized »
Resposta do TAG	TAG ignora o quadro
Comentários	
Resultado	Aprovado

Caso	8.6 : SoF - bit + comando + EoF
Resposta Esperada	TAG envia o código de erro «001»: «The command is not supported, i.e., the request code is not recognized »
Resposta do TAG	TAG envia o código de erro «001»: «The command is not supported, i.e., the request code is not recognized »
Comentários	
Resultado	Aprovado

Caso	8.6 : Mudança dos flags
Resposta Esperada	TAG envia o código de erro «001»: «The command is not supported, i.e., the request code is not recognized »
Resposta do TAG	TAG envia o código de erro «001»: «The command is not supported, i.e., the request code is not recognized »
Comentários	
Resultado	Aprovado

## A.9 : Parte 9 : Os estados do TAG

9.1 : « STAY QUIET » + 2ms + « READ SINGLE BLOCK »	
Caso	9.1.1 : « ADR flag » = 0
Resposta	O TAG fica no estado «QUIET» e ignora o comando

Esperada	
Resposta do TAG	O TAG fica no estado «QUIET» e ignora o comando
Comentários	
Resultado	Aprovado

Caso	9.1.2 : « ADR flag » = 1
Resposta Esperada	O TAG fica no estado «QUIET» e responde adequadamente ao comando
Resposta do TAG	O TAG fica no estado «QUIET» e responde adequadamente ao comando
Comentários	
Resultado	Aprovado

---

9.2 : « STAY QUIET » + 2ms + « READ MULTIPLE BLOCKS »

---

Caso	9.2.1 : « ADR flag » = 0
Resposta Esperada	O TAG fica no estado «QUIET» e ignora o comando
Resposta do TAG	O TAG fica no estado «QUIET» e ignora o comando
Comentários	
Resultado	Aprovado

Caso	9.2.2 : « ADR flag » = 1
Resposta Esperada	O TAG fica no estado «QUIET» e responde adequadamente ao comando
Resposta do TAG	O TAG fica no estado «QUIET» e responde adequadamente ao comando
Comentários	
Resultado	Aprovado

---

9.3 : « STAY QUIET » + 2ms + « INVENTORY »

---

Caso	9.3.1 : « INVENTORY flag » = 0
Resposta	O TAG fica no estado «QUIET» e ignora o comando

Esperada	
Resposta do TAG	O TAG fica no estado «QUIET» e ignora o comando
Comentários	O comando « INVENTORY » não é valido no estado « QUIET »
Resultado	Aprovado

Caso	9.2.2 : « INVENTORY flag » = 1
Resposta Esperada	O TAG fica no estado «QUIET» e ignora o comando
Resposta do TAG	O TAG fica no estado «QUIET» e ignora o comando
Comentários	O comando « INVENTORY » não é valido no estado « QUIET »
Resultado	Aprovado