

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

**IMPLEMENTAÇÃO EM FPGA DE UMA
REDE NEURAL DE HOPFIELD**

Camila Cocolo

São Carlos
2015

Camila Cocolo

IMPLEMENTAÇÃO EM FPGA DE UMA REDE NEURAL DE HOPFIELD

Trabalho de Conclusão de Curso apresentado
à Escola de Engenharia de São Carlos, da
Universidade de São Paulo.

Curso de Engenharia Elétrica com Ênfase em
Eletrônica.

ORIENTADORA: Luiza Maria Romeiro Codá

São Carlos

2015

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

C667i Cocolo, Camila
Implementação em FPGA de uma Rede Neural de
Hopfield / Camila Cocolo; orientadora Luiza Maria
Romeiro Codá. São Carlos, 2015.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Eletrônica) -- Escola de Engenharia de São
Carlos da Universidade de São Paulo, 2015.

1. Redes Neurais Artificiais. 2. Hopfield. 3. FPGA.
4. VHDL. I. Título.

FOLHA DE APROVAÇÃO

Nome: Camila Cocolo

Título: "Implementação em FPGA de uma rede neural de Hopfield"

Trabalho de Conclusão de Curso defendido e aprovado
em 23 / 11 / 2019

com NOTA 8,0 (oito, zero), pela Comissão Julgadora:

*Profa Assistente Luiza Maria Romeiro Codá - (Orientadora -
SEL/EESC/USP)*

Prof. Dr. Maximilian Luppe - (SEL/EESC/USP)

Prof. Associado Rogério Andrade Flauzino - (SEL/EESC/USP)

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Dr. José Carlos de Melo Vieira Júnior

Dedicatória

Dedico este trabalho aos meus pais Marci Aparecida F. Cocolo e Claudedir Aparecido Cocolo que sempre apoiaram minhas escolhas e deram todo o incentivo e suporte para que eu evoluísse nos estudos, e à minha irmã Juliana Cocolo que é minha companheira de todos os momentos e me ajuda incondicionalmente.

Agradecimento

Agradeço à minha família pelo apoio emocional, compreensão, incentivo e pelo aprendizado diário que me transmitiram através dos exemplos de honestidade, esforço e fé.

À minha orientadora Luiza Maria Romeiro Codá por ser sempre prestativa e atenciosa.

Aos professores Maximilian Luppe e Rogério Andrade Flauzino pelas valiosas sugestões para o desenvolvimento deste trabalho.

Resumo

COCOLO, C. Implementação em FPGA de uma Rede Neural de Hopfield. Trabalho de Conclusão de Curso – Departamento de Engenharia Elétrica da Escola de Engenharia de São Carlos, São Carlos: Universidade de São Paulo, 2015.

As Redes neurais artificiais (RNAs) são sistemas de inteligência artificial cujo funcionamento é inspirado na estrutura neural de organismos inteligentes que adquirem conhecimento através da experiência e aplicam o conhecimento adquirido para novos problemas ou situações. As RNAs possuem aplicações em diversas áreas da ciência e tecnologia, sendo usadas principalmente na resolução de problemas que envolvem reconhecimento de padrões e generalização. Um exemplo é a rede de Hopfield, que quando aplicada como uma memória associativa é capaz de recuperar um padrão previamente armazenado em sua estrutura a partir de um elemento ruidoso ou deformado. Enquanto os computadores funcionam sequencialmente, uma característica intrínseca às redes neurais biológicas é a execução de operações em paralelo, o que faz com que sua velocidade de processamento seja muito maior. Neste sentido as FPGAs (*Field Programmable Gate Array*) são apresentadas como excelente alternativa a implementação de redes neurais artificiais, pois possuem elevada quantidade de elementos lógicos e permitem a execução de operações paralelas em nível de *hardware*. Este trabalho consiste na descrição e síntese de uma rede de Hopfield composta por nove neurônios, que apresenta dinâmica síncrona e é aplicada como uma memória associativa. Primeiramente a rede foi descrita no MATLAB (*Matrix Laboratory*) e seus resultados serviram como referência para a rede parametrizada em VHDL (*Very High Speed Integrated Circuits Hardware Description Language*) e sintetizada na FPGA Cyclone IV EP4CE115F29, da Altera.

Palavras-chave: redes neurais artificiais, Hopfield, VHDL, FPGA.

Abstract

COCOLO, C. FPGA Implementation of a Hopfield Neural Network. Final Paper - Department of Electrical Engineering of the School of Engineering of São Carlos, São Carlos: Universidade de São Paulo, 2015.

Artificial neural networks (ANNs) are artificial intelligence systems which performance is inspired by the neural structure of intelligent organisms that acquire knowledge through experience and apply the knowledge acquired to new problems or concerns. The ANNs have applications in several areas of science and technology and is mainly used to solve problems involving pattern recognition and generalization. An example is the Hopfield network, which when applied as an associative memory is capable of retrieving in a previously stored pattern in its structure from a noisy or distorted element. While computers work sequentially, an intrinsic characteristic of the biological neural network is performing operations in parallel, which means that its processing speed is much higher. In this regard the FPGAs (Field Programmable Gate Array) are presented as an excellent alternative to implementing artificial neural networks because they have high amount of logic elements and allow you to perform parallel operations in hardware. This work is the description and synthesis of a Hopfield network consisted by nine neurons, presenting synchronous dynamic and is applied as an associative memory. First the network was described in MATLAB (*Matrix Laboratory*) and its results served as reference for the parameterized network in VHDL (Very High Speed Integrated Circuits Hardware Description Language) and synthesized in the FPGA Cyclone IV EP4CE115F29, Altera.

Keywords: artificial neural networks, Hopfield, VHDL, FPGA

Lista de Figuras

Figura 1 - Representação em diagrama em blocos do sistema nervoso	5
Figura 2 - Exemplos de neurônios biológicos.....	6
Figura 3 - Transmissão de impulso nervoso através da sinapse.....	7
Figura 4 - Etapas de variação do potencial de ação do neurônio.....	8
Figura 5 - Neurônio Artificial.....	9
Figura 6 - Função degrau	10
Figura 7 - Função degrau bipolar	10
Figura 8 - Função rampa simétrica	11
Figura 9 - Função logística.....	11
Figura 10 - Função tangente hiperbólica.....	11
Figura 11 - Função gaussiana	12
Figura 12 - Função linear	12
Figura 13 - Monocamada com direção unidimensional	13
Figura 14 - Multicamada com alimentação unidimensional	13
Figura 15 - Exemplo de rede recorrente.....	14
Figura 16 - Função de energia e seus pontos de equilíbrio.....	15
Figura 17 - Rede de Hopfield	15
Figura 18 - Derivadas parciais de primeira ordem das funções logística e tangente hiperbólica	18
Figura 19 - Pontos de equilíbrio e seus campos de atração para um sistema de dois estados estáveis e dois neurônios	18
Figura 20 - Padrão reconhecido pela rede, versão com ruído e sua recuperação.....	19
Figura 21 - Espaço contendo 3 padrões e suas regiões de atração.....	21
Figura 22 - Gráficos da capacidade de armazenamento da rede de Hopfield em função da quantidade de elementos dos padrões	23
Figura 23 - Evolução assíncrona	24
Figura 24 - Evolução síncrona	25
Figura 25 - Padrões armazenados na memória associativa.....	27
Figura 26 - Algoritmo da rede Hopfield aplicada como memória associativa.....	28
Figura 27 - Representação em ponto fixo binário.....	29
Figura 28 - Rede de Hopfield descrita no MATLAB.....	30
Figura 29 - Diagrama de blocos da rede neural descrita em VHDL.....	32
Figura 30 - Bloco do Multiplexador.....	33
Figura 31 - Descrição do Multiplexador em VHDL	33

Figura 32 - Bloco do Neurônio N1	34
Figura 33 - Descrição do Neurônio em VHDL	34
Figura 34 - Bloco do Flip Flop descrito em VHDL	35
Figura 35 - Descrição do Flip Flop em VHDL	35
Figura 36 - Bloco do Comparador descrito em VHDL	36
Figura 37 - Descrição do Comparador em VHDL	36
Figura 38 - Bloco do Contador descrito em VHDL	37
Figura 39 - Descrição do Contador em VHDL	37
Figura 40 - Bloco da Máquina de estados descrita em VHDL	38
Figura 41 - Descrição da Máquina de estados em VHDL	38
Figura 42 - Evolução dos estados da máquina de estados	39
Figura 43 - Bloco do Conversor Binário - Decimal descrito em VHDL	40
Figura 44 - Descrição do Display de 7 segmentos em VHDL	40
Figura 45 - Arquitetura principal	41
Figura 46 - Placa do módulo DE2-115	43
Figura 47 - Interface do software Quartus II exibindo o número de componentes internos do FPGA utilizados no projeto	44
Figura 48 - Conexões não nulas entre os neurônios	49
Figura 49 - Simulação no Quartus II para uma saída convergente	50
Figura 50 - Circuito da rede neural implantada no Kit DE2-115 apresentando saída convergente	50
Figura 51 - Simulação no Quartus para uma saída oscilatória	51
Figura 52 - Circuito da rede neural implementada no Kit DE2-115 apresentando saída oscilatória	51
Figura 53 - Circuito RTL do <i>Multiplexador</i>	56
Figura 54 - Circuito RTL do <i>Neurônio N1</i>	56
Figura 55 - Circuito RTL do <i>Flip Flop</i> tipo D	56
Figura 56 - Circuito RTL do <i>Comparador</i>	57
Figura 57 - Circuito RTL do <i>Contador</i>	57
Figura 58 - Circuito RTL da <i>Máquina de estados</i>	57
Figura 59 - Circuito RTL do <i>Conversor Binário - Decimal</i>	58

Lista de Tabelas

Tabela 1 - Evolução dos estados considerando atualização assíncrona.....	24
Tabela 2 - Evolução dos estados considerando atualização síncrona	25
Tabela 3 - Representação binária dos componentes da matriz de pesos	29
Tabela 4 - Erro relativo gerado pela aproximação numérica	30
Tabela 5 - Relação das Chaves e LEDs com as entradas (e) e saídas (s) da rede.....	44
Tabela 6 - Amostra dos resultados obtidos a partir da rede descrita no MATLAB.....	46
Tabela 7 - Número de ocorrência para cada estado final da rede	47
Tabela 8 - Ocorrência dos ciclos de dois estados	48
Tabela 9 - Capacidade de armazenamento da rede de Hopfield de dimensão $n = 9$	52
Tabela 10 - Resultados obtidos a partir da rede descrita no MATLAB	59

Lista de Abreviaturas

ANN	<i>Artificial Neural Network</i>
FPGA	<i>Field Programmable Gate Arrays</i>
HDL	<i>Hardware Description Language</i>
MATLAB	<i>Matrix Laboratory</i>
RNA	Rede Neural Artificial
RTL	<i>Register Transfer Level</i>
VHDL	<i>VHSIC Hardware Description Language</i>

Sumário

Dedicatória	i
Agradecimento	iii
Resumo	v
<i>Abstract</i>	vii
Lista de Figuras	ix
Lista de Tabelas	xi
Lista de Abreviaturas	xiii
1 Introdução	1
1.1 Contextualização e Motivações	1
1.2 Objetivos	2
1.3 Organização do documento	3
2 Revisão Bibliográfica	4
2.1 Caracterização das redes neurais	4
2.2 O Neurônio Biológico	5
2.3 O Neurônio Artificial	8
2.4 Arquiteturas de redes neurais artificiais	12
2.5 O Modelo de Hopfield	14
2.5.1 Estabilidade da rede de Hopfield	16
2.5.2 O Modelo de Hopfield Discreto como uma Memória Associativa	19
2.5.3 Armazenamento de padrões na rede	20
2.5.4 Atratores Espúrios	21
2.5.5 Capacidade de armazenamento das memórias associativas	22
2.5.6 Atualização síncrona e assíncrona	23
3 Metodologia	26
3.1 Ferramentas utilizadas	26
3.2 Considerações de projeto	27
3.3 Adequação da rede	29
3.4 Desenvolvimento da rede no MATLAB	30
3.5 Descrição da rede em VHDL	32
3.6 Módulo de desenvolvimento	43
4 Resultados e Análises	46

4.1	Resultados obtidos no MATLAB	46
4.2	Síntese no FPGA	49
4.3	Discussão dos Resultados.....	52
5	Conclusão	54
6	Referências.....	55
	Apêndice I - Circuitos RTL gerados pelo software Quartus II	56
	Apêndice II – Resultados obtidos no MATLAB	59

1 Introdução

1.1 Contextualização e Motivações

As redes neurais artificiais (RNAs) são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes, e que adquirem conhecimento através da experiência. São compostas por diversas unidades computacionais, os chamados neurônios artificiais, que operam em paralelo e são total ou parcialmente interconectados. Cada neurônio artificial executa operações simples e transmite seus resultados aos neurônios vizinhos, com os quais possui conexão (PERELMUTER et al., 1995).

O método de processamento por arquiteturas neurais tem comprovada eficiência em reconhecimento de padrões, nesse sentido a arquitetura de Hopfield, através do método de memórias associativas, se destaca especialmente na recuperação de padrões defeituosos ou com ruídos. O objetivo é apresentar à rede um conjunto de padrões conhecidos de modo que, através de um processo de aprendizado, ela seja capaz de reconhecer as características dos padrões armazenados em um novo elemento qualquer que a ela seja apresentado, ou seja, os estados estáveis da rede neural corresponderão aos padrões armazenados *a priori*. Entretanto, a rede pode apresentar um comportamento indesejado onde nem todos os estados estáveis estão associados aos padrões fundamentais, são os chamados estados espúrios, estes estados devem ser reduzidos para aumentar a assertividade da memória associativa.

Fazendo um comparativo entre o cérebro humano e um computador, sabe-se que a velocidade de processamento dos neurônios biológicos é da ordem de milissegundos, enquanto os eventos nas portas lógicas ocorrem na ordem de nano segundos. No entanto, o cérebro compensa sua lentidão pela enorme quantidade de neurônios existentes, com conexões maciças entre si. Estima-se que haja cerca de 10 bilhões de neurônios no córtex humano e 60 trilhões de sinapses ou conexões (SHEPHERD e KOCH, 1990). Outra diferença significativa é que o cérebro executa seus processos paralelamente, enquanto os computadores executam operações de forma sequencial. Nesse sentido, a implementação de redes neurais em FPGAs, consiste na melhor maneira de aproveitar seu funcionamento, pois permite execuções paralelas em nível de *hardware*.

O FPGA é um circuito integrado que contém milhares de unidades lógicas idênticas, tais unidades são componentes padrões que podem ser configurados independentemente e interconectados a partir de uma matriz de trilhas condutoras e *switches* programáveis. Um arquivo binário é gerado para configuração da FPGA a partir de ferramentas de *software* seguindo um determinado fluxo de projeto. Este arquivo binário contém as informações necessárias para especificar a função de cada unidade lógica e para seletivamente fechar os *switches* da matriz de interconexão.

Linguagens de Descrição de *Hardware* apresentam grandes semelhanças com as linguagens de programação, pois permitem que o circuito eletrônico seja descrito em sentenças, com a possibilidade de simulação e síntese, mas são especificamente orientadas à descrição da estrutura e do comportamento do *hardware*. As linguagens VHDL (*VHSIC Hardware Description Language*) e Verilog são as HDLs mais empregadas na indústria e no meio acadêmico (GODSE; GODSE, 2009). A linguagem VHDL suporta projetos com múltiplos níveis de hierarquia, e sua descrição pode consistir na interligação de outras descrições menores. É composta pela declaração de uma entidade e arquitetura, e com exceção de regiões específicas do código, todos os comandos são executados concorrentemente.

Dessa forma, a maior aproximação do funcionamento das redes neurais artificiais com o cérebro humano, através da implementação em *hardware*, foi a principal motivação para o desenvolvimento deste trabalho. Além do fato de a rede neural de Hopfield configurada como memória associativa ser uma das mais utilizadas RNAs, apresentando uma vasta gama de aplicações em áreas como o acesso ao conteúdo de memória, reconhecimento de padrões e controle inteligente.

1.2 Objetivos

Este trabalho tem por objetivo a implementação em FPGA, de uma rede neural de Hopfield composta por nove neurônios, que apresenta dinâmica síncrona e é aplicada como uma memória associativa. E a criação de uma rede neural de referência gerada no MATLAB com todas as entradas simuladas, a fim de obter as saídas de referência para validação do correto funcionamento da rede sintetizada em *hardware*.

Através dos resultados procura-se analisar a capacidade de armazenamento de padrões das memórias associativas, o percentual de assertividade da rede e a ocorrência de estados espúrios.

1.3 Organização do documento

O presente Trabalho de Conclusão de Curso está organizado em cinco capítulos que descrevem os conceitos teóricos das redes neurais, o processo de desenvolvimento da rede neural de Hopfield e os resultados finais obtidos.

No Capítulo 1 é apresentado o contexto em que está inserido este trabalho, bem como sua relevância, objetivos e justificativas para seu desenvolvimento.

No Capítulo 2 é apresentado o estudo de redes neurais artificiais, descrevendo o neurônio biológico, o neurônio artificial, as principais arquiteturas de redes neurais e o enfoque no modelo de Hopfield, que foi a arquitetura de rede escolhida para o desenvolvimento deste trabalho.

No Capítulo 3 são expostas as ferramentas utilizadas e os detalhes da rede de Hopfield estudada, bem como os procedimentos adotados para sua síntese na FPGA, que são a elaboração da rede de referência no MATLAB, e a descrição do circuito em VHDL.

No Capítulo 4 são discutidos os resultados obtidos com a síntese da rede de Hopfield em *hardware* e é analisado o seu desempenho.

No Capítulo 5 é apresentada a conclusão a respeito dos resultados obtidos e são feitas propostas para a continuação deste trabalho.

2 Revisão Bibliográfica

2.1 Caracterização das redes neurais

As redes neurais artificiais são estruturas computacionais desenvolvidas a partir de modelos conhecidos de sistemas nervosos biológicos. Possuem a capacidade de adquirir conhecimento e podem ser definidas como um conjunto de unidades de processamento, caracterizadas por neurônios artificiais que tentam reproduzir o funcionamento de neurônios biológicos baseando-se no modo como impulsos elétricos são transmitidos pelas membranas celulares dos neurônios (Silva, Spatti & Flauzino, 2010).

Uma rede neural se assemelha ao cérebro nos seguintes aspectos:

1. O conhecimento é adquirido pela rede a partir de um processo de aprendizagem em seu ambiente.
2. Forças de conexões entre neurônios, denominadas pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

O procedimento de aprendizagem da rede é chamado de *algoritmo de aprendizagem* ou *treinamento*, e possui a função de modificar os pesos sinápticos de forma ordenada a fim de alcançar um objetivo de projeto desejado (Haykin, 2009).

As principais características das redes neurais artificiais são (Silva, Spatti & Flauzino, 2010):

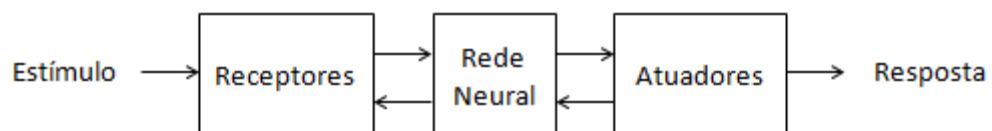
- **Adaptação por experiência:** o conhecimento é obtido por experimentação, pois os pesos sinápticos da rede são ajustados a partir da apresentação sucessiva de exemplos associados ao comportamento do processo.
- **Capacidade de aprendizado:** através do treinamento a rede é capaz de se adaptar ao processo e relacionar as variáveis que compõem a aplicação.
- **Habilidade de generalização:** após o treinamento a rede é capaz de generalizar o conhecimento adquirido e assim apresentar soluções para um conjunto de amostras que eram até então desconhecidas.
- **Organização de dados:** a partir das amostras de entrada entregue, a rede é capaz de organizar-se internamente a fim de agrupar padrões que possuem particularidades em comum.

- **Tolerância a falhas:** a rede neural é capaz de operar mesmo quando parte de sua estrutura interna foi corrompida, isso ocorre devido ao elevado grau de interconexão entre os neurônios artificiais.
- **Armazenamento distribuído:** o conhecimento da rede neural é distribuído entre as diversas sinapses dos neurônios artificiais, o que confere a rede robustez frente a neurônios que se tornaram inoperantes.
- **Facilidade de prototipagem:** após o treinamento da rede seus resultados são obtidos por operações matemáticas elementares, podendo ser facilmente prototipada em hardware ou software.

2.2 O Neurônio Biológico

O sistema nervoso humano pode ser representado como um sistema de três estágios, como mostrado no diagrama de blocos da Figura 1. O cérebro é o centro do sistema, representado pela *rede neural*, ele recebe continuamente informações e toma as decisões apropriadas. No diagrama, as setas que apontam da esquerda para a direita indicam a transmissão para frente do sinal portador de informação, e as que apontam da direita para a esquerda indicam a realimentação do sistema. Os *receptores* são capazes de converter estímulos do corpo humano ou do ambiente externo em impulsos elétricos que transmitem informação compreensível para a rede neural, enquanto os *atuadores* convertem os impulsos elétricos processados pela rede neural em respostas discerníveis como saídas do sistema (HAYKIN, 2009).

Figura 1 - Representação em diagrama em blocos do sistema nervoso



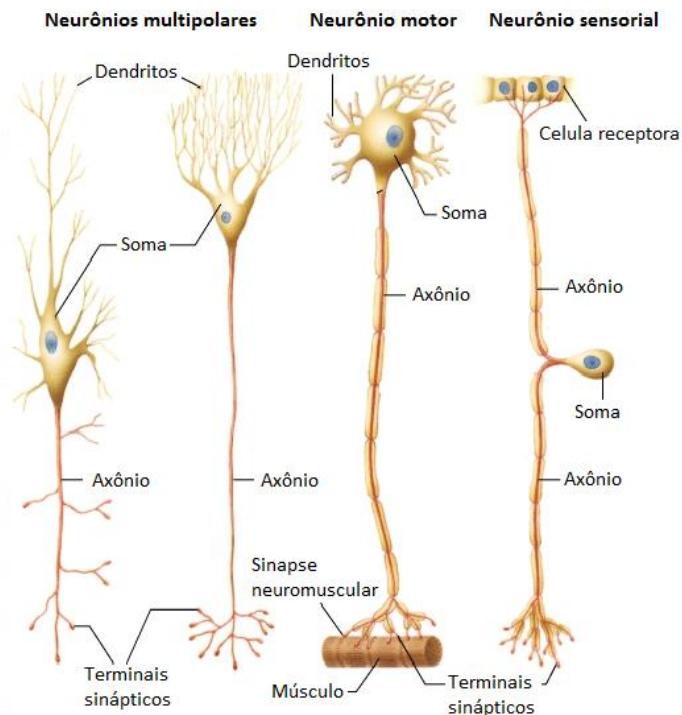
Fonte: HAYKIN, 2009, página 32

O trabalho pioneiro de Ramon & Cajál, em 1911, ajudou a compreender o funcionamento do cérebro ao introduzir a ideia de que os neurônios são os constituintes estruturais do cérebro. Tipicamente, os neurônios são de 5 a 6 ordens de grandeza mais lentos que as portas lógicas em silício, entretanto, a imensa

quantidade de neurônios (células nervosas) presentes no cérebro humano e as conexões maciças entre si compensam sua relativa lentidão.

Os neurônios aparecem em grande variedade de formas e tamanhos em diferentes partes do cérebro, mas todos podem ser divididos em dendritos, soma (ou corpo celular) e axônio, como ilustrado na Figura 2.

Figura 2 - Exemplos de neurônios biológicos



Fonte: Fundamentada na imagem publicada em:
<https://biochemstyles.wordpress.com/2013/09/25/relationship-between-structure-and-function-of-a-motor-neurons/>

Os dendritos apresentam superfície irregular e diversas ramificações, são responsáveis por captar os estímulos vindos de outros neurônios ou do meio externo onde estão inseridos, como é o caso dos neurônios sensoriais. A soma, ou corpo celular, tem a função de processar as informações recebidas pelos dendritos e produzir um potencial de ativação que classifica se o neurônio poderá disparar um impulso elétrico ao longo de seu axônio (Silva, Spatti & Flauzino, 2010).

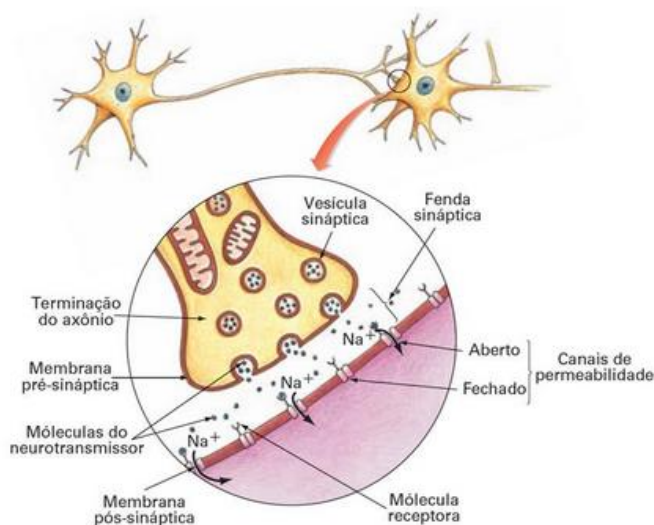
O axônio é uma estrutura muito longa e fina, caracterizada por alta resistência elétrica e grande capacitância, tais características fazem com que a tensão elétrica aplicada em uma extremidade do axônio decaia exponencialmente ao longo de seu comprimento, resultando em uma tensão insignificante no momento em que chega à

outra extremidade. É por isso que os sinais no corpo celular são emitidos por uma série de pulsos breves de tensão, dessa forma os sinais conseguem se propagar através dos neurônios a uma velocidade e amplitude constantes (Haykin, 2009). O axônio conduz os impulsos elétricos para outros neurônios ou ao tecido muscular e sua terminação é constituída de ramificações denominadas terminais sinápticos.

Sinapse é o processo pelo qual estímulos elétricos e químicos passam de um neurônio para outro. O processo sináptico pode ser visto na Figura 3, e é descrito da seguinte forma: as terminações dos axônios possuem uma membrana denominada pré-sináptica que converte os impulsos elétricos provenientes do corpo celular em um sinal químico ao emitir neurotransmissores armazenados em vesículas. Os neurotransmissores ligam-se aos receptores da membrana pós-sináptica dos dendritos, provocando a absorção de íons de sódio através dos canais abertos.

Dependendo dos neurotransmissores envolvidos, esta ação pode causar um pulso elétrico no segundo neurônio, caso de uma conexão excitatória, ou não transmitir impulso elétrico, caso de uma conexão inibitória. Ambos os casos são possíveis porque a emissão de um pulso elétrico no axônio do segundo neurônio só acontece quando os neurotransmissores do primeiro conseguem despolarizar a membrana pós-sináptica em um determinado valor, conhecido como *limiar de ativação*. Assim, a transmissão de impulsos pelos neurônios está relacionada à intensidade das ligações sinápticas estabelecidas (KOVÁCKS, 2006).

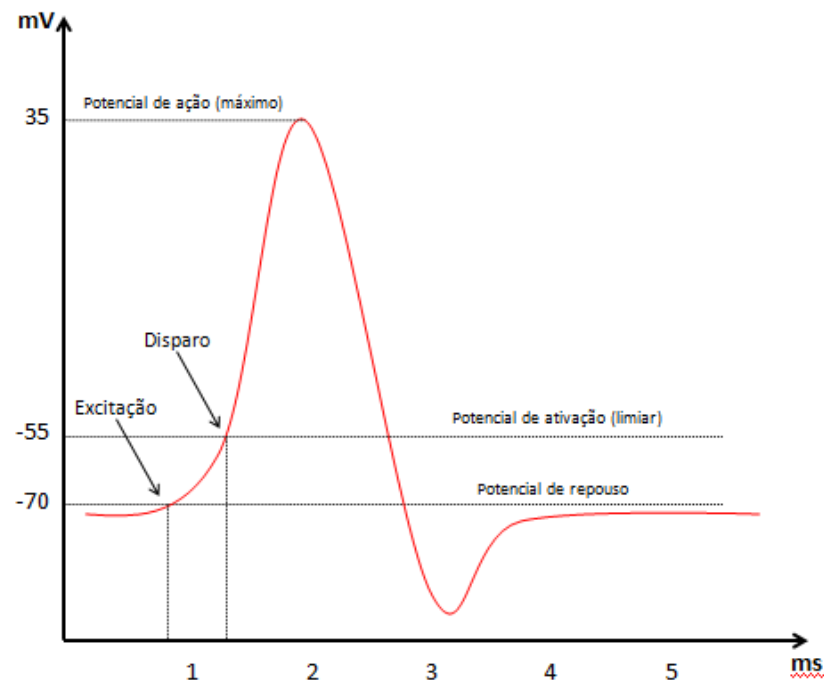
Figura 3 - Transmissão de impulso nervoso através da sinapse



Fonte: Fundamentada na imagem publicada em: <http://www.culturamix.com/cultura/sinapse-artificial>

As variações de potencial de ação do neurônio, antes, durante e após o processo sináptico pode ser visto na Figura 4.

Figura 4 - Etapas de variação do potencial de ação do neurônio



Fonte: Silva, Spatti & Flauzino, 2010, página 32

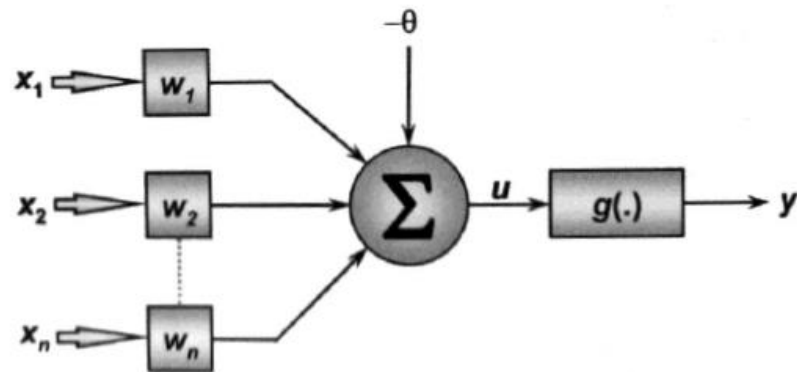
O potencial de ação da membrana pós-sináptica, quando em repouso (polarizada), é de -70 mV, ou seja, há concentração maior de íons negativos em seu interior em relação ao meio exterior. O limiar de ativação da membrana é de -55 mV, e é obtido através do aumento de íons positivos no interior do neurônio. Quando é atingido o limiar de ativação, o impulso elétrico é transmitido pelo axônio, atingindo o valor máximo de 35 mV, independentemente do tipo de neurônio. Quando a membrana é repolarizada, o potencial de ação da membrana retorna ao seu potencial de repouso (Silva, Spatti & Flauzino, 2010).

2.3 O Neurônio Artificial

A modelagem matemática de funcionamento de um neurônio artificial mais utilizada nas diferentes arquiteturas de redes neurais é a proposta por McCulloch & Pitts em 1943, neste modelo o neurônio recebe vários estímulos de entrada e sua

saída é obtida através de uma função somatório das entradas ponderadas, como ilustrado na Figura 5.

Figura 5 - Neurônio Artificial



Fonte: Silva, Spatti & Flauzino, 2010, página 34

O Neurônio artificial é constituído dos seguintes elementos básicos (Silva, Spatti & Flauzino, 2010):

Sinais de entrada $\{x_1, x_2, \dots, x_n\}$ - representam os impulsos elétricos externos captados pelos dendritos no neurônio biológico.

Pesos sinápticos $\{w_1, w_2, \dots, w_3\}$ - são as ponderações exercidas pelas junções sinápticas do modelo biológico.

Combinador linear $\{\Sigma\}$ - o somatório das entradas ponderadas representa a integração espacial do neurônio biológico.

Limiar de ativação $\{\theta\}$ - representa o limiar de ativação do neurônio biológico.

Potencial de ativação $\{u\}$ - representa o pulso de saída do neurônio biológico, se for positivo, o neurônio produz um potencial excitatório, se negativo, o potencial é inibitório.

Função de ativação $\{g\}$ - limita a saída do neurônio dentro de um intervalo de valores convenientes para a imagem funcional da rede neural.

Sina de saída $\{y\}$ - representa o valor final produzido pelo neurônio a partir dos sinais de entrada.

As equações (1) e (2) representam o modelo de neurônio artificial da Figura 5, onde a saída y é obtida através da soma ponderada das entradas, seguida pela comparação com o limiar de ativação e função de ativação.

$$u = \sum_{i=1}^n w_i * x_i - \theta \quad (1)$$

$$y = g(u) \quad (2)$$

A variação dos parâmetros w e θ permite a adequação dos neurônios às características requeridas por diferentes projetos, e a união de vários neurônios, que é a própria rede neural artificial, possibilita a execução de funções complexas.

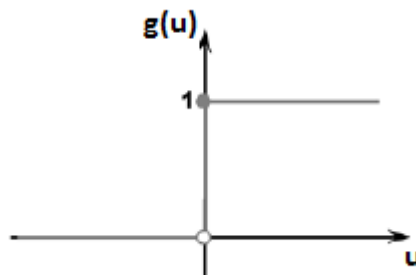
As funções de ativação são classificadas em relação à sua diferenciabilidade, sendo divididas em dois grupos:

1) Funções parcialmente diferenciáveis: aquelas que possuem pontos cuja derivada de primeira ordem são inexistentes. Exemplo: função degrau, função degrau bipolar e função rampa simétrica.

a) Função degrau

$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ 0, & \text{se } u < 0 \end{cases}$$

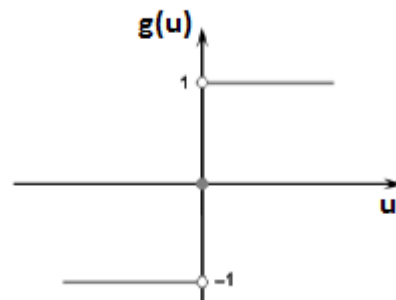
Figura 6 - Função degrau



b) Função degrau bipolar

$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ 0, & \text{se } u = 0 \\ -1, & \text{se } u < 0 \end{cases}$$

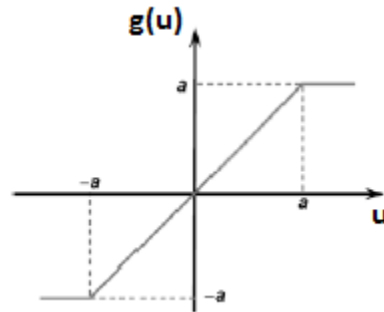
Figura 7 - Função degrau bipolar



c) Função rampa simétrica

Figura 8 - Função rampa simétrica

$$g(u) = \begin{cases} a, & \text{se } u > a \\ u, & \text{se } -a \leq u \leq a \\ -a, & \text{se } u < -a \end{cases}$$

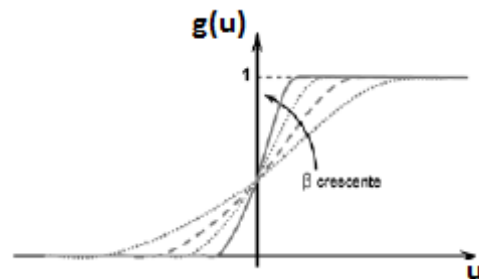


2) Funções totalmente diferenciáveis: aquelas cujas derivadas de primeira ordem existem em todos os pontos de seu domínio. Exemplo: função logística, tangente hiperbólica, gaussiana e função linear.

a) Função logística

Figura 9 - Função logística

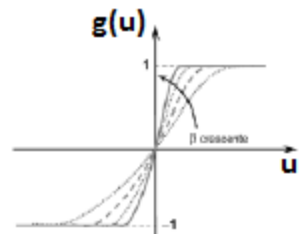
$$g(u) = \frac{1}{1 + e^{-\beta \cdot u}}$$



b) Função tangente hiperbólica

Figura 10 - Função tangente hiperbólica

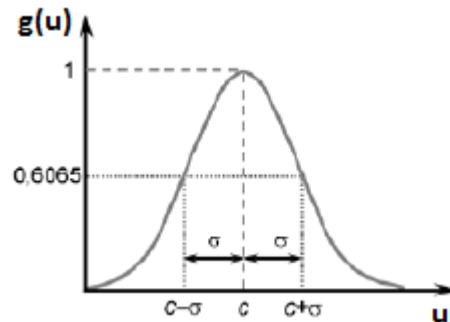
$$g(u) = \frac{1 - e^{-\beta \cdot u}}{1 + e^{-\beta \cdot u}}$$



c) Função gaussiana

$$g(u) = e^{\frac{-(u-c)^2}{2\sigma^2}}$$

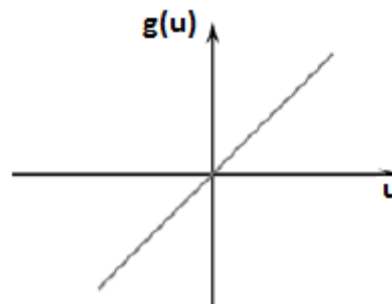
Figura 11 - Função gaussiana



d) Função linear

$$g(u) = u$$

Figura 12 - Função linear



2.4 Arquiteturas de redes neurais artificiais

As diferentes arquiteturas de redes neurais, considerando a disposição de seus neurônios, estão associadas ao problema que se deseja solucionar, e corresponde a um fator importante na definição do algoritmo de aprendizagem a ser utilizado.

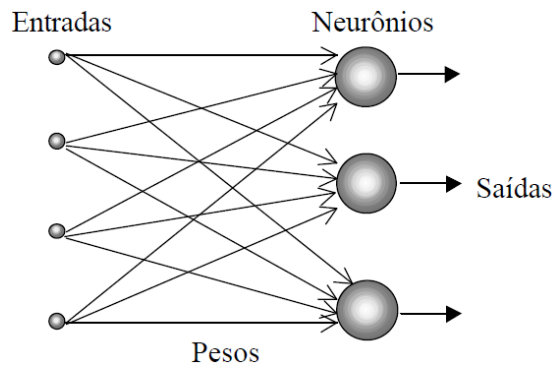
As RNAs podem ser vistas como grafos direcionados, onde os neurônios são os vértices e os arcos são as conexões entre eles. Cada arco está associado a um peso, que é correspondente ao peso sináptico (SILVA, 2003).

As principais arquiteturas de redes neurais são divididas em dois grupos:

1) Redes *feedforward*: são aquelas onde o grafo não possui laços. Elas são estáticas, pois produzem um único conjunto de valor de saída. Também pode-se dizer que não possuem memória, já que sua resposta a uma entrada independe do estado prévio da rede. Dentro desta arquitetura pode-se destacar:

- Monocamada com alimentação unidirecional, Figura 13. Nesse caso, existe apenas uma camada de neurônios ligados diretamente aos vértices de entrada, e eles próprios fornecem os dados de saída.

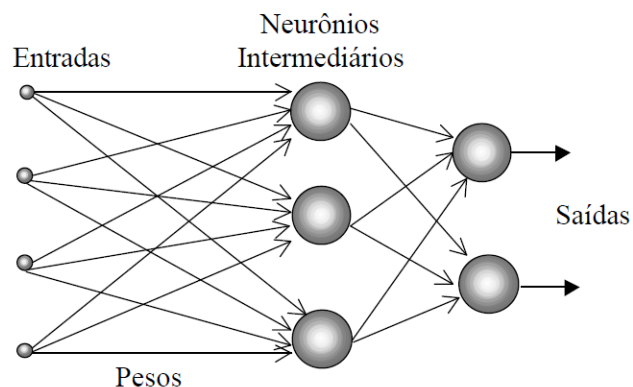
Figura 13 - Monocamada com direção unidimensional



Fonte: SILVA, 2003, página 11

- Multicamada com alimentação unidirecional, Figura 14. Esse tipo de arquitetura possui camadas de neurônios escondidos, cuja função é processar os sinais de entrada antes de enviá-los aos neurônios de saída. Essa arquitetura possibilita um melhor treinamento, pois há maior interação entre os neurônios.

Figura 14 - Multicamada com alimentação unidirecional

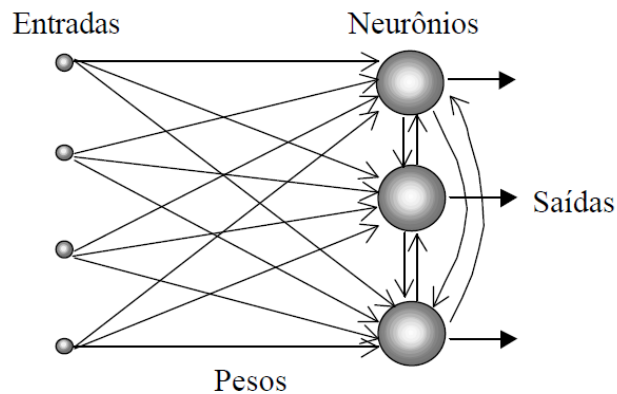


Fonte: SILVA, 2003, página 12

2) Redes Recorrentes: por possuírem realimentação, as redes recorrentes respondem a estímulos dinamicamente, ou seja, após aplicar uma entrada, a saída é

calculada e realimentada para ser a nova entrada, por isso essas redes são ditas “com memória”. Podem ser monocamada ou multicamada, como ilustrado na Figura 15.

Figura 15 - Exemplo de rede recorrente



Fonte: SILVA, 2003, página 12

A definição da arquitetura utilizada depende das características do sistema a ser desenvolvido (MENEZES, C. C., 2001).

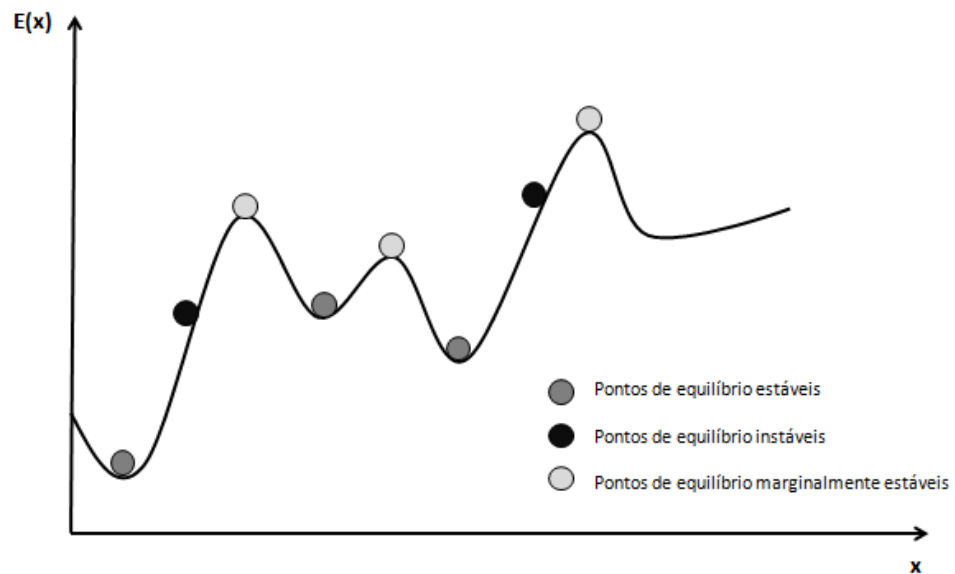
2.5 O Modelo de Hopfield

Conforme visto no item 2.4, as redes recorrentes são aquelas cuja resposta é realimentada à entrada da rede para o próximo ciclo de processamento, dando ao sistema um comportamento dinâmico e não linear.

Havia pouco interesse no estudo das redes neurais recorrentes, pois estas apresentavam difícil treinamento e análise. Até que em 1982 o físico J. J. Hopfield apresentou um modelo recorrente que faz analogia a um sistema físico real: uma função de energia. Hopfield mostrou que cada estado da rede pode ser associado a um valor de energia que decresce monotonicamente à medida que uma trajetória é descrita no espaço de estados em direção a pontos de equilíbrio estáveis, estes pontos são obtidos durante o treinamento e correspondem à solução desejada de um sistema específico (Silva, Spatti & Flauzino, 2010).

Observa-se na Figura 16 que dependendo das características da função de energia pode haver diversos pontos de equilíbrio estáveis.

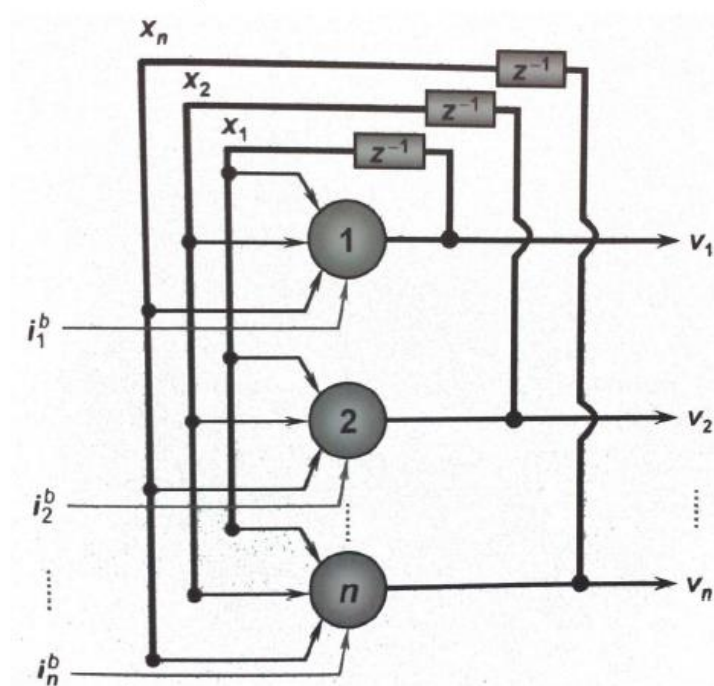
Figura 16 - Função de energia e seus pontos de equilíbrio



Fonte: Silva, Spatti & Flauzino, 2010, página 200

A rede de Hopfield original consiste em um conjunto de neurônios e seus correspondentes atrasos unitários, formando um sistema realimentado de múltiplos laços, como visto na Figura 17.

Figura 17 - Rede de Hopfield



Fonte: Silva, Spatti & Flauzino, 2010, página 201

O comportamento da rede de Hopfield pode ser descrito pelas equações (3) e (4) (Silva, Spatti & Flauzino, 2010).

$$\dot{u}_j(t) = -\eta * u_j(t) + \sum_{i=1}^n W_{ij} * v_i(t) + i_j^b, \text{ onde } j = 1, \dots, n \quad (3)$$

$$v_j(t) = g(u_j(t)) \quad (4)$$

Onde:

$\dot{u}_j(t)$ é o estado interno do j-ésimo neurônio, com $\dot{u}_j(t) = du/dt$;

$v_j(t)$ é a saída do j-ésimo neurônio;

W_{ij} é o valor do peso sináptico conectando o j-ésimo neurônio ao i-ésimo neurônio;

i_j^b é o limiar (bias) aplicado ao j-ésimo neurônio;

$g(.)$ é uma função de ativação, monótona crescente, que limita a saída de cada neurônio em um intervalo predefinido;

$\eta * u_j(t)$ é um termo de decaimento passivo;

$x_j(t)$ é a entrada do j-ésimo neurônio.

Dessa forma, o comportamento dinâmico da rede de Hopfield é caracterizado pela aplicação de uma entrada inicial e geração de uma saída, a seguir a rede é realimentada e seu estado é atualizado a cada iteração. Este processo se repete até que a rede evolua para um estado estável, ou seja, produza uma saída que seja igual à saída anterior.

A estabilidade da rede esta associada à configuração de seus parâmetros livres, podendo funcionar como um sistema estável, um oscilador ou um sistema caótico.

2.5.1 Estabilidade da rede de Hopfield

Para analisar a estabilidade da rede de Hopfield usa-se uma função de energia, ou função de Lyapunov, associada à dinâmica da rede. Tal função é construída para provar que o sistema dissipa energia ao longo do tempo ao ser submetido a certas condições.

Uma função de Lyapunov para a rede de Hopfield, considerando que os neurônios são alterados de forma assíncrona, é definida por (Hopfield, 1984):

$$E(t) = -\frac{1}{2} \mathbf{v}(t)^T \mathbf{W} \mathbf{v}(t) - \mathbf{v}(t)^T \mathbf{i}^b \quad (5)$$

Para garantir que o sistema é estável, o segundo método de Lyapunov estabelece que as derivadas temporais de sua função devem ser menores ou iguais a zero.

Assim, obtém-se a expressão para as derivadas temporais de (5):

$$\dot{E}(t) = \frac{dE(t)}{dt} = (\nabla_v E(t))^T \dot{\mathbf{v}}(t) \quad (6)$$

Onde ∇_v é o operador gradiente em relação ao vetor \mathbf{v} .

Impondo a condição de a matriz de pesos ser simétrica ($\mathbf{W} = \mathbf{W}^T$), obtém-se a seguinte relação entre (5) e (6):

$$\nabla_v E(t) = -\mathbf{W}^* \mathbf{v}(t) - \mathbf{i}^b \quad (7)$$

Assumindo que o termo de decaimento passivo seja nulo ($\mathbf{i}^b = 0$) na expressão (3), tem-se o seguinte relacionamento com a equação (7):

$$\nabla_v E(t) = -\dot{\mathbf{u}}(t) \quad (8)$$

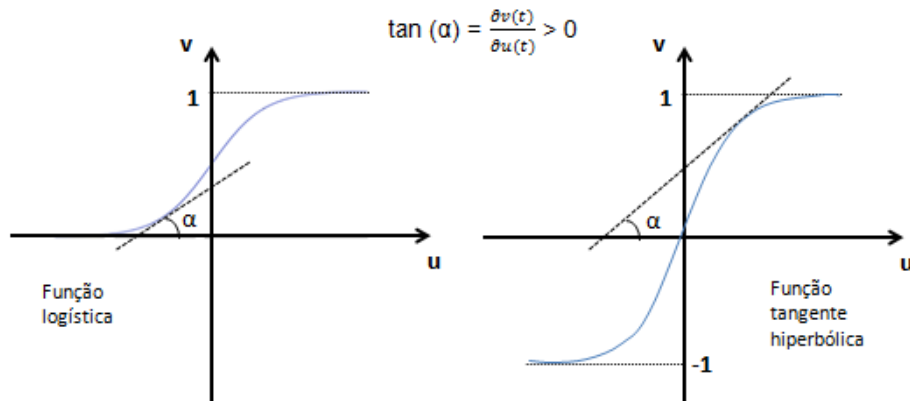
Substituindo (8) em (6):

$$\begin{aligned} \dot{E}(t) &= -\dot{\mathbf{u}}(t)^T \dot{\mathbf{v}}(t) \\ &= -\sum_{j=1}^n \dot{u}_j(t) \dot{v}_j(t) = -\sum_{j=1}^n \dot{u}_j(t) \frac{\partial v_j}{\partial u_i} \frac{\partial u_i}{\partial t} \\ &= -\sum_{j=1}^n \underbrace{(\dot{u}_j(t))^2}_{\text{parcela (i)}} * \underbrace{\frac{\partial v_j(t)}{\partial u_i(t)}}_{\text{parcela (ii)}} \end{aligned} \quad (9)$$

Para concluir a demonstração, basta mostrar que as derivadas temporais da expressão (9) são menores ou iguais a zero, ou seja, mostrar que as parcelas (i) e (ii) produzirão sempre valores positivos, já que a expressão possui sinal negativo.

A parcela (i) sempre fornecerá um valor positivo, pois está elevada ao quadrado. E a parcela (ii) terá resultado positivo ao se utilizar funções de ativação monótona crescente, como a função logística e a tangente hiperbólica (Silva, Spatti & Flauzino, 2010), ilustradas na Figura 18.

Figura 18 - Derivadas parciais de primeira ordem das funções logística e tangente hiperbólica



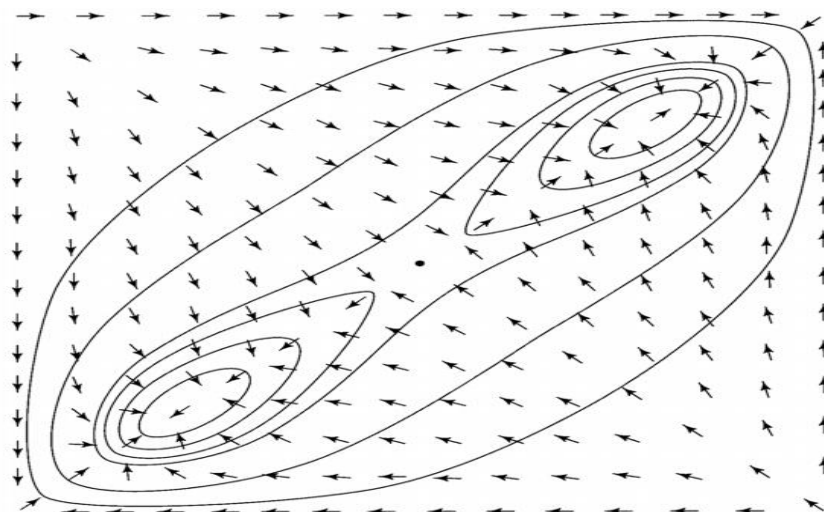
Fonte: Silva, Spatti & Flauzino, 2010, página 205

Assim, as condições para que a rede de Hopfield seja estável, ou seja, tenha saída convergente para pontos de equilíbrio, são:

- (i) A matriz de pesos deve ser simétrica.
- (ii) A função de ativação deve ser monótona crescente.

A Figura 19 representa o fluxo para os pontos de equilíbrio produzido por uma rede de Hopfield de dois neurônios. Os pontos de equilíbrio estão localizados próximos dos vértices inferior esquerdo e superior direito, e são interpretados como a solução para a minimização da função de energia da rede (Haykin, 2009).

Figura 19 - Pontos de equilíbrio e seus campos de atração para um sistema de dois estados estáveis e dois neurônios



Fonte: Haykin, 2009, página 738

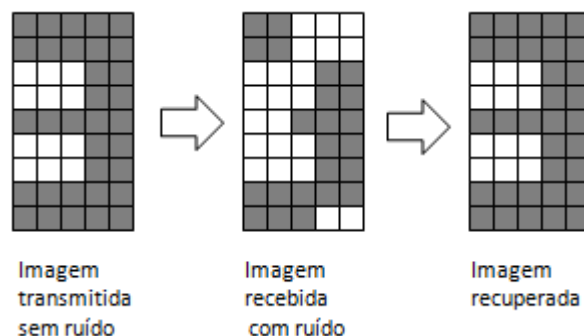
Dessa forma, se as condições iniciais estiverem dentro de um campo de atração, a rede convergirá para o referido ponto, sendo a quantidade de pontos de equilíbrio e suas localizações determinadas pelos parâmetros livres da rede (Silva, Spatti & Flauzino, 2010).

2.5.2 O Modelo de Hopfield Discreto como uma Memória Associativa

A rede de Hopfield é muito utilizada como uma memória associativa, também conhecida como memória endereçável por conteúdo. Nesta aplicação os pontos de equilíbrio correspondem aos padrões armazenados na estrutura da rede, e a função da memória associativa é recuperar este padrão a partir de uma amostra parcial ou ruidosa do mesmo.

A Figura 20 ilustra a versão original de um padrão, sua versão com ruído e o padrão recuperado pela rede.

Figura 20 - Padrão reconhecido pela rede, versão com ruído e sua recuperação



Fonte: Silva, Spatti & Flauzino, 2010, página 220

Quando é apresentada uma entrada para a rede de Hopfield caracterizada como uma memória associativa, a rede responde produzindo o padrão armazenado que seja mais parecido com a entrada inicial, ou seja, realiza o menor número de substituições necessárias para transformar a entrada em um padrão conhecido. Este cálculo é conhecido como distância de Hamming, que calcula a menor distância entre duas sequências através da quantificação do número de bits que uma difere da outra.

2.5.3 Armazenamento de padrões na rede

Como visto anteriormente, os padrões a serem armazenados na memória associativa consistem nos pontos fixos da rede *a priori*, assim o que se deve determinar são os pesos sinápticos que produzem os pontos fixos desejados. Para esta tarefa foi utilizado o método do produto externo, que é a forma mais simples para determinação da matriz de pesos \mathbf{W} e do vetor de limiares \mathbf{i}^b (Silva, Spatti & Flauzino, 2010).

Segundo Hopfield (1982), sendo p a quantidade de padrões $\{\mathbf{z}\}$ a serem inseridos na estrutura da memória, constituídos por n elementos cada um, pode-se definir os parâmetros livres da rede de Hopfield por:

$$\mathbf{W} = \frac{1}{n} \sum_{k=1}^p \mathbf{z}^{(k)} * (\mathbf{z}^{(k)})^T \quad (10)$$

$$\mathbf{i}^b = 0 \quad (11)$$

No caso das memórias associativas não deve haver auto-alimentação, dessa forma a diagonal da matriz de pesos deve ser zerada. Modificando a expressão (10) para este fim, tem-se:

$$\mathbf{W} = \frac{1}{n} \sum_{k=1}^p \mathbf{z}^{(k)} * (\mathbf{z}^{(k)})^T - \frac{p}{n} * \mathbf{I} \quad (12)$$

onde $\mathbf{I} \in \mathbb{R}^{n \times n}$ é uma matriz identidade.

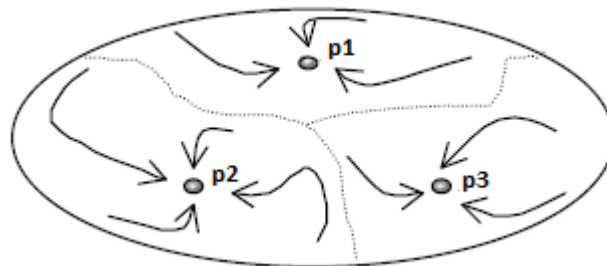
A equação (10) realiza o produto externo entre os elementos de cada um dos padrões e a parcela adicional incluída na equação (12) neutraliza os elementos da diagonal principal.

Se dois neurônios de um determinado padrão tiverem o mesmo valor (0,0) ou (1, 1) o peso sináptico entre eles, dado pelo produto externo, deverá reforçar esta relação. Caso os dois neurônios tenham valores opostos (1, 0) ou (0, 1), então um peso negativo entre eles deverá reforçar a relação inversa. Para representar mais eficientemente a ideia de reforço, uma notação alternativa é adotada para os números binários e ao invés de utilizar 0 e 1, utiliza-se -1 (correspondendo a 0) e 1, esta notação é chamada polarizada ou representação em spin. Assim, caso os dois neurônios

tenham valores iguais na maioria dos padrões a soma terá mais parcelas do tipo 1×1 e -1×-1 , tendo um resultado positivo. Caso os neurônios tenham valores opostos as parcelas serão predominantemente do tipo 1×-1 e 1×-1 e o valor total negativo. Logo, a notação polarizada é uma boa maneira de fixar os pesos da rede para que ela armazene os padrões.

Cada padrão armazenado está associado a um estado estável (um mínimo local) e é chamado atrator. A rede quando inicializada dentro de um ponto da base de atração é influenciada por um atrator e tende a se estabilizar, dinamicamente, neste atrator, como ilustra a Figura 21.

Figura 21 - Espaço contendo 3 padrões e suas regiões de atração



Fonte: SILVA, 2003, página 23

2.5.4 Atratores Espúrios

Quando a rede neural de Hopfield armazena seus padrões, são gerados estados estáveis além dos estados associados aos padrões armazenados, denominados atratores espúrios ou estados espúrios.

Os atratores espúrios são ocasionados pelos seguintes fatores:

1. A função de energia E é simétrica, no sentido de que os estados correspondentes ao reverso dos padrões armazenados também são estados estáveis;
2. Toda combinação linear de um número ímpar de estados estáveis também é um estado estável (AMIT, 1989). O sistema não escolhe um número par de padrões, pois na somatória um termo pode anular o outro, e assim o estado não será estável.

3. Para um grande número de padrões armazenados, a função de energia vai produzir pontos de equilíbrio que não estão correlacionados com nenhum dos padrões armazenados na rede (VON ZUBEN, ATTUX).

2.5.5 Capacidade de armazenamento das memórias associativas

A rede de Hopfield possui limitações relacionadas à quantidade de padrões que podem ser inseridos em sua estrutura visando a recuperação destes a partir de uma amostra parcial ou distorcida.

Baseado em experimentos computacionais, Hopfield concluiu que a capacidade de armazenamento de padrões que resultará na recuperação de padrões com relativamente poucos erros é dada por:

$$C^{\text{Hopf}} = 0,15 * n \quad (13)$$

Onde n , é a quantidade de elementos de cada padrão.

A capacidade de armazenamento quase sem erros é definida em Haykin (2006) por meio de análises de probabilidade, sendo sua expressão dada por:

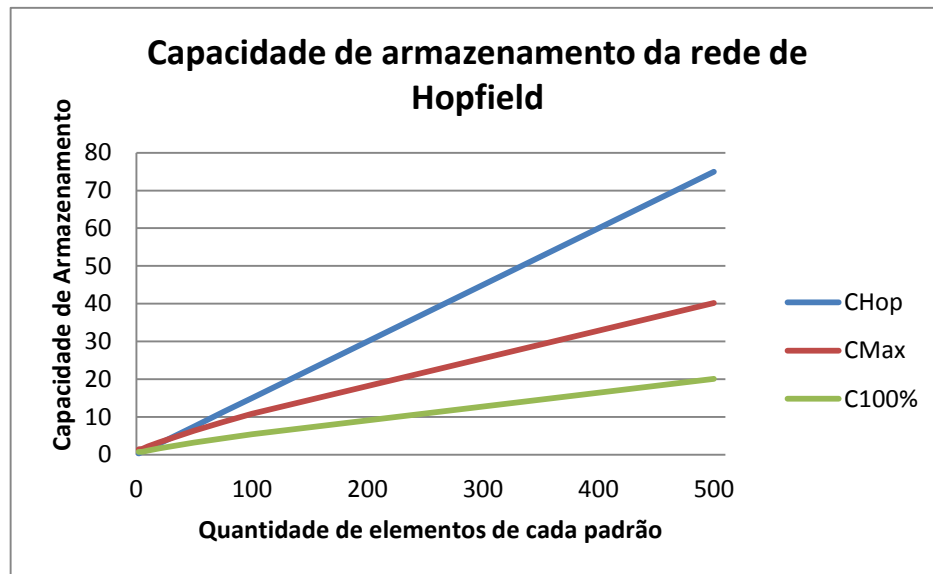
$$C^{\text{Max}} = \frac{n}{2 * \ln(n)} \quad (14)$$

A estimativa para uma recuperação perfeita dos padrões é dado pela seguinte expressão:

$$C^{100\%} = \frac{n}{4 * \ln(n)} \quad (15)$$

A Figura 22 mostra as capacidades de armazenamento da rede de Hopfield para diversos valores de n considerando as expressões (13), (14) e (15).

Figura 22 - Gráficos da capacidade de armazenamento da rede de Hopfield em função da quantidade de elementos dos padrões



A análise da Figura 22 mostra que a capacidade de armazenamento da rede deve ser mantida pequena quando se deseja que a recuperação seja a mais exata possível. Entretanto, se padrões excessivos às quantidades recomendadas pelas equações (10) e (11) forem inseridos, haverá o aparecimento dos estados espúrios (Silva, Spatti & Flauzino, 2010), citados na seção 2.5.4.

2.5.6 Atualização síncrona e assíncrona

Para entender o funcionamento da rede de Hopfield, além das condições de estabilidade, deve ser definido o modo como os neurônios serão atualizados, que pode ser de maneira síncrona ou assíncrona.

A atualização é síncrona quando todos os neurônios são atualizados simultaneamente, e é assíncrona quando os neurônios são atualizados sequencialmente, mas de forma independente, de acordo com uma constante de probabilidade.

Para demonstrar os efeitos gerados pelo modo como as atualizações são feitas, é dado o exemplo da rede de Hopfield com três neurônios que armazena dois padrões: $[-1 \ +1 \ +1]$ e $[+1 \ +1 \ -1]$.

Utilizando a equação (12) e considerando $p=2$ e $n=3$, foi obtida a matriz de pesos para os padrões:

$$W = \begin{matrix} & 0 & 0 & -0,667 \\ & 0 & 0 & 0 \\ -0,667 & 0 & 0 & 0 \end{matrix}$$

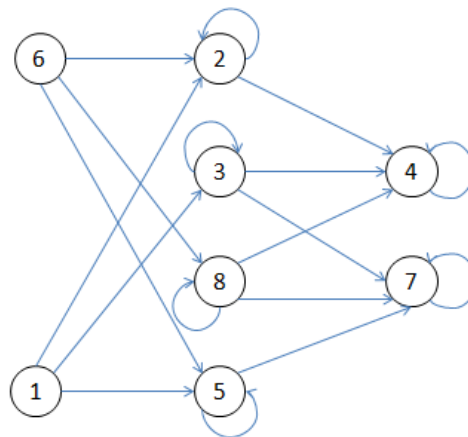
A Tabela 1 apresenta a evolução dos estados da rede considerando atualização assíncrona.

Tabela 1 - Evolução dos estados considerando atualização assíncrona

	Entrada	Estados		
		1	2	3
1	-1 -1 -1	+1 -1 -1	-1 +1 -1	-1 -1 +1
2	-1 -1 +1	-1 -1 +1	-1 +1 +1	-1 -1 +1
3	-1 +1 -1	+1 +1 -1	-1 +1 -1	-1 +1 +1
4	-1 +1 +1	-1 +1 +1	-1 +1 +1	-1 +1 +1
5	+1 -1 -1	+1 -1 -1	+1 +1 -1	+1 -1 -1
6	+1 -1 +1	-1 -1 +1	+1 +1 +1	+1 -1 -1
7	+1 +1 -1	+1 +1 -1	+1 +1 -1	+1 +1 -1
8	+1 +1 +1	-1 +1 +1	+1 +1 +1	+1 +1 -1

Pela análise da Tabela 1 percebe-se que a evolução dos estados da rede ocorre como ilustrado na Figura 23, na qual as entradas 4 e 7, que correspondem aos padrões armazenados, são de mínima energia, e portanto estáveis.

Figura 23 - Evolução assíncrona



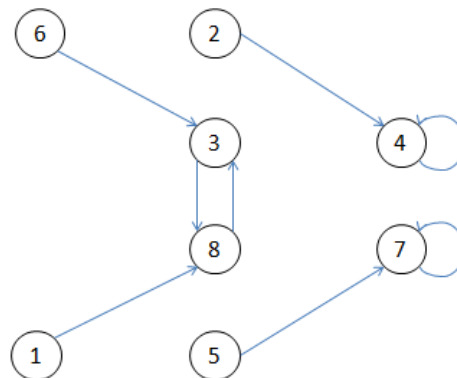
Fonte: BITTENCOURT, página 18

A Tabela 2 apresenta a evolução dos estados da rede considerando atualização síncrona.

Tabela 2 - Evolução dos estados considerando atualização síncrona

	Entrada	Estados		
		1	2	3
1	-1 -1 -1	+1 +1 +1	-1 +1 -1	+1 +1 +1
2	-1 -1 +1	-1 +1 +1	-1 +1 +1	-1 +1 +1
3	-1 +1 -1	+1 +1 +1	-1 +1 -1	+1 +1 +1
4	-1 +1 +1	-1 +1 +1	-1 +1 +1	-1 +1 +1
5	+1 -1 -1	+1 +1 -1	+1 +1 -1	+1 +1 -1
6	+1 -1 +1	-1 +1 -1	+1 +1 +1	-1 +1 -1
7	+1 +1 -1	+1 +1 -1	+1 +1 -1	+1 +1 -1
8	+1 +1 +1	-1 +1 -1	+1 +1 +1	-1 +1 -1

No caso da atualização síncrona a dinâmica da rede é determinística e cada estado tem apenas um sucessor, como ilustrado na Figura 24. Além dos estados estáveis 4 e 7, este tipo de dinâmica permite a formação de ciclos com dois estados, os chamados ciclos limites, como o ciclo $3 \leftrightarrow 8$.

Figura 24 - Evolução síncrona

Fonte: BITTENCOURT, página 19

Os diferentes comportamentos apontados para cada tipo de atualização pode ser explicado pela teoria de Hebb (1949). Ele observou uma característica dos neurônios que consiste no reforço ou enfraquecimento do sinal sináptico de acordo com certos pré-requisitos. Se dois neurônios, um em cada lado de uma sinapse, são ativados assincronamente então a força daquela sinapse é seletivamente enfraquecida ou eliminada. Entretanto, se dois neurônios, um em cada lado de uma sinapse são ativados simultaneamente, então a força daquela sinapse é seletivamente aumentada. A diminuição da energia nas atualizações assíncronas faz com que todos os estados estáveis gerados pela rede sejam pontos fixos, e o aumento da energia nas atualizações síncronas permite a formação de ciclos limites.

3 Metodologia

Neste capítulo são descritas as ferramentas utilizadas, considerações sobre o projeto e as implementações no MATLAB e em VHDL.

3.1 Ferramentas utilizadas

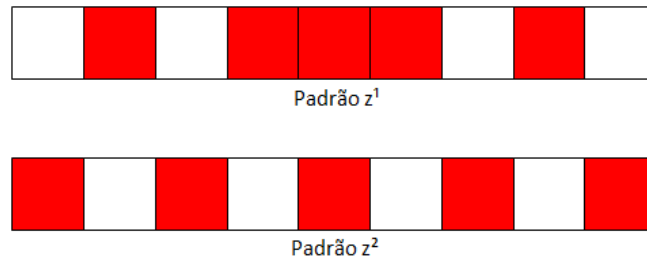
A seguir são descritas as ferramentas utilizadas no desenvolvimento do projeto:

- O MATLAB (MATrix LABoratory) é um "*software*" de alto desempenho destinado a fazer cálculos com matrizes podendo funcionar como uma calculadora ou como uma linguagem de programação científica, em que os comandos são mais próximos da forma como são escritas expressões algébricas, tornando o seu uso mais simples. Atualmente, o MATLAB é definido como um sistema interativo e uma linguagem de programação para computação técnica e científica em geral, integrando a capacidade de fazer cálculos, visualização gráfica e programação (TONINI e COUTO, 1999). Foi utilizada a versão 2015a do MATLAB, disponível por licença institucional aos alunos da USP.
- O Quartus II é um *software* de *design* de dispositivo lógico programável produzido pela Altera, que permite a análise e síntese de projetos HDL, possibilitando ao desenvolvedor compilar seus projetos, realizar análises temporais, examinar diagramas RTL (*Register Transfer Level*), simular a reação de um projeto a diferentes estímulos, e configurar o dispositivo de destino. Foi utilizada a versão 12.0 *Web Edition* do Quartus II para a síntese da rede neural na FPGA, e a versão 4.0 foi utilizada para as simulações, ambas as versões são gratuitas e podem ser adquiridas no site da Altera.

3.2 Considerações de projeto

A rede de Hopfield utilizada é uma adaptação do exemplo apresentado por Silva, Spatti & Flauzino, 2010 (página 209), e consiste em uma memória associativa capaz de reconhecer os padrões ilustrados na Figura 25.

Figura 25 - Padrões armazenados na memória associativa



Fonte: Fundamentada na imagem publicada em Silva, Spatti & Flauzino, 2010, pagina 209

Os padrões a serem armazenados na rede estão sobre um vetor de dimensão 9×1 , onde as quadrículas vermelhas são representadas pelo valor $+1$, e as quadrículas brancas pelo valor -1 . Assim, são formados os vetores \mathbf{z}^1 e \mathbf{z}^2 :

$$\mathbf{z}^1 = [-1 \quad +1 \quad -1 \quad +1 \quad +1 \quad +1 \quad -1 \quad +1 \quad -1]^T$$

$$\mathbf{z}^2 = [+1 \quad -1 \quad +1 \quad -1 \quad +1 \quad -1 \quad +1 \quad -1 \quad +1]^T$$

Substituindo a quantidade de padrões, $p = 2$, constituídos por n elementos, $n = 9$, na expressão (12), obtém-se a matriz de pesos \mathbf{W} para os padrões descritos acima:

$$\mathbf{W} = \begin{vmatrix} 0 & -0,22 & 0,22 & -0,22 & 0 & -0,22 & 0,22 & -0,22 & 0,22 \\ -0,22 & 0 & -0,22 & 0,22 & 0 & 0,22 & -0,22 & 0,22 & -0,22 \\ 0,22 & -0,22 & 0 & -0,22 & 0 & -0,22 & 0,22 & -0,22 & 0,22 \\ -0,22 & 0,22 & -0,22 & 0 & 0 & 0,22 & -0,22 & 0,22 & -0,22 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0,22 & 0,22 & -0,22 & 0,22 & 0 & 0 & -0,22 & 0,22 & -0,22 \\ 0,22 & -0,22 & 0,22 & -0,22 & 0 & -0,22 & 0 & -0,22 & 0,22 \\ -0,22 & 0,22 & -0,22 & 0,22 & 0 & 0,22 & -0,22 & 0 & -0,22 \\ 0,22 & -0,22 & 0,22 & -0,22 & 0 & -0,22 & 0,22 & -0,22 & 0 \end{vmatrix}$$

Observa-se que a matriz gerada é simétrica, condição exigida para a rede ser estável, e os elementos de sua diagonal principal são nulos, caracterizando a ausência de auto realimentação das memórias associativas.

A função de ativação considerada para o neurônio artificial é a degrau bipolar, expressa na equação (16)

$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ -1, & \text{se } u < 0 \end{cases} \quad (16)$$

O comportamento dinâmico da rede foi implantado seguindo os passos do algoritmo ilustrado na Figura 26, que considera a atualização síncrona dos neurônios.

Figura 26 - Algoritmo da rede Hopfield aplicada como memória associativa

```

Início

  Declarar os padrões a serem armazenados  $\mathbf{z}^1$  e  $\mathbf{z}^2$ ;
  Definir a matriz de pesos  $\mathbf{W}$ ;
  Apresentar o vetor inicial de entrada  $\mathbf{x}$ ;
   $\mathbf{v\_anterior} \leftarrow \mathbf{x}$ ;
   $\mathbf{v\_atual} \leftarrow \mathbf{W} * \mathbf{v\_anterior}$ ;
   $\mathbf{v\_atual} \leftarrow g(\mathbf{v\_atual})$ ; --  $g$  é a função de ativação
  Repetir as instruções no máximo  $n$  vezes:
    Se  $\mathbf{v\_atual} = \mathbf{v\_anterior}$ ;
      Sair do loop;
    Senão
       $\mathbf{v\_anterior} \leftarrow \mathbf{v\_atual}$ ;
       $\mathbf{v\_atual} \leftarrow \mathbf{W} * \mathbf{v\_anterior}$ ;
       $\mathbf{v\_atual} \leftarrow g(\mathbf{v\_atual})$ ;
  Até que:  $\mathbf{v\_atual} = \mathbf{v\_anterior}$ ;
   $\mathbf{v\_final} \leftarrow \mathbf{v\_atual}$ ;

Fim

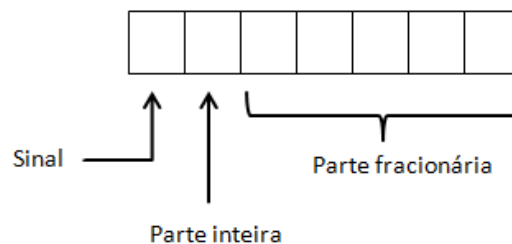
```

Dessa forma, quando for apresentada à rede uma versão incompleta ou ruidosa dos padrões \mathbf{z}^1 ou \mathbf{z}^2 , esta convergirá para um desses padrões de acordo com a localização da entrada no campo de atração dos estados estáveis.

3.3 Adequação da rede

Para a descrição da rede em VHDL os elementos da matriz de pesos **W** foram representados em notação binária de ponto fixo. O bit mais a esquerda do número foi utilizado para representar o sinal: 0 indica um valor positivo e 1 indica um valor negativo. A parte inteira é representada por um bit, e cinco bits são utilizados para a parte fracionária. Esta representação é ilustrada na Figura 27.

Figura 27 - Representação em ponto fixo binário



Utilizou-se a equação (17) para obter os pesos binários da Tabela 3, onde d_b representa o valor decimal, b_i corresponde aos bits que formam o número binário e b_7 é o bit mais significativo.

$$d_b = \sum_{i=0}^5 b_i \cdot 2^{i-5} \quad (17)$$

Tabela 3 - Representação binária dos componentes da matriz de pesos

Peso (Decimal)	Peso (Binário)
0,22	00,00111
-0,22	10,00111

O separador decimal é utilizado para facilitar a relação entre os dados, não sendo implementado na prática.

O erro relativo de aproximação $e\%$, ocasionado pela representação da parte fracionária, foi calculado utilizando-se a equação (18), onde d é o valor original da entrada e d_b é o valor decimal convertido a partir do valor binário, de acordo com (17).

$$e\% = \frac{(d - d_b)}{d} \cdot 100 \quad (18)$$

A Tabela 4 exibe os resultados da comparação, e nota-se que o erro relativo é menor que 0,6%, sendo tolerável para a aplicação na rede neural.

Tabela 4 - Erro relativo gerado pela aproximação numérica

Peso (d)	Peso (d _b)	e%
0,22	0,21875	0,57
-0,22	-0,21875	0,57

3.4 Desenvolvimento da rede no MATLAB

Com o objetivo de ter uma rede neural de referencia com todos os valores de entrada e saída simulados, utilizou-se o MATLAB para descrição da rede de Hopfield apresentada na seção 3.2.

A Figura 28 apresenta o código comentado da rede no MATLAB.

Figura 28 - Rede de Hopfield descrita no MATLAB

```

1      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Determinação da matriz de pesos W %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2
3      % z1 e z2 são os padrões a serem armazenados
4      z1 = [-1 1 -1 1 1 1 -1 1 -1]';
5      z2 = [1 -1 1 -1 1 -1 1 -1 1]';
6
7      p = 2; % p representa o número de padrões
8      m = 9; % m representa o número de elementos de cada padrão
9
10     % A expressão abaixo realiza o produto externo entre os elementos de cada
11     % um dos padrões, e garante que a diagonal principal da matriz W seja nula
12     W = (1/m).*((z1*z1')+(z2*z2')) - (p/m).*eye(9);
13
14
15     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Comportamento da rede recorrente de Hopfield %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16
17     % x corresponde ao vetor de entradas inserido na rede
18     x = [1 -1 1 -1 1 -1 -1 -1 -1]';
19     v_anterior = x;
20
21     % A saída atual (v_atual) é obtida pela soma ponderada da saída anterior
22     % (v_anterior), sendo a ponderação dada pela matriz de pesos W
23     v_atual = W*v_anterior;

```

```

24
25 % A saída tem seus valores limitados pela função de ativação degrau bipolar
26 - for i=1:1:9
27 -     if v_atual(i,1)>= 0
28 -         v_atual(i,1) = 1;
29 -     else
30 -         v_atual(i,1) = -1;
31 -     end
32 - end
33
34 % Neste passo a rede testa se a saída anterior é diferente da saída atual,
35 % se sim a rede é realimentada e o teste é refeito, este processo se repete
36 % até um limite máximo de 1000 iterações; se a saída anterior for igual a
37 % saída atual, a rede atingiu um estado estável.
38 - for n = 2:1:1000
39 -     if v_anterior == v_atual
40 -         n = n-1;
41 -         break
42 -     else
43 -         v_anterior = v_atual;
44 -         v_atual = W*v_anterior;
45 -         for j=1:1:9
46 -             if v_atual(j,1)>= 0
47 -                 v_atual(j,1) = 1;
48 -             else
49 -                 v_atual(j,1) = -1;
50 -             end
51 -         end
52 -     end
53 - end
54
55 % As saídas de interesse da rede são:
56 - n ;
57 - v_anterior ;
58 - v_atual ;

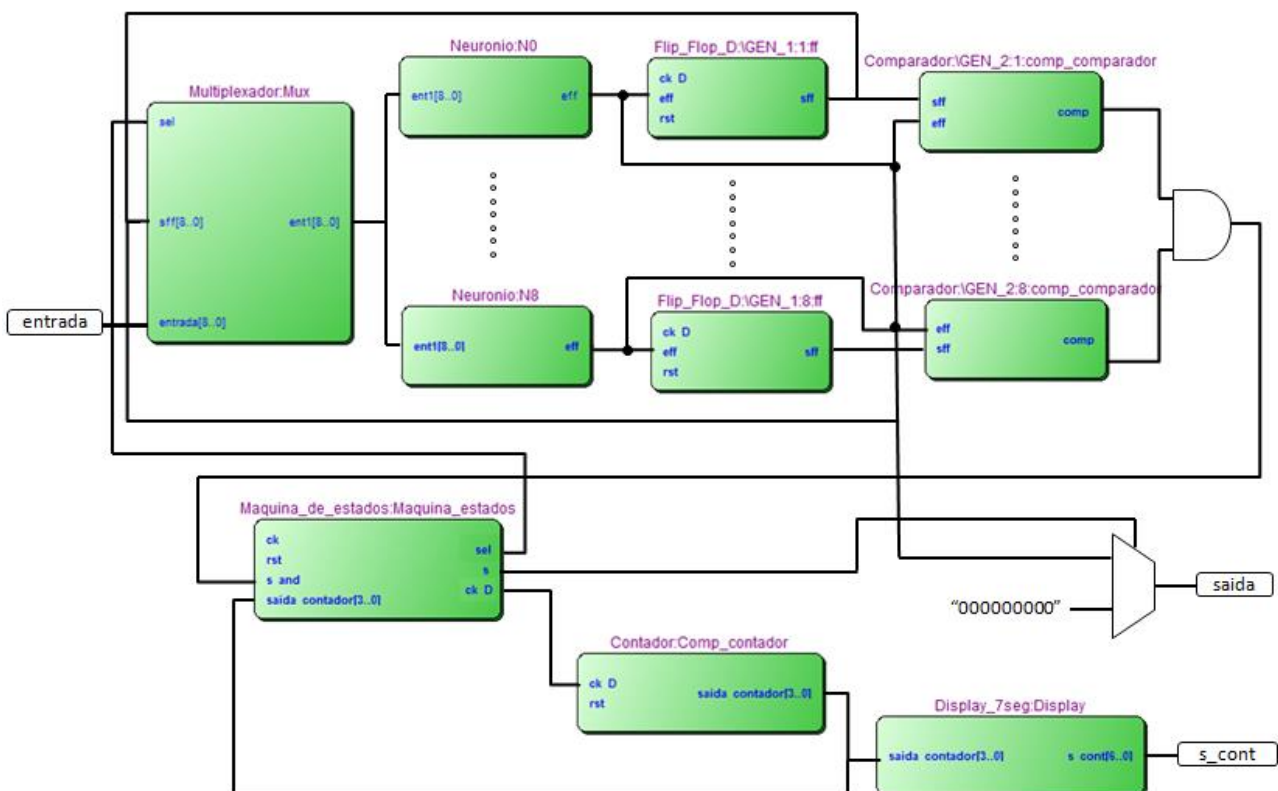
```

Análise importante para definição dos parâmetros do *hardware* foi que apesar de ter sido utilizado o número máximo de 1000 iterações na simulação em MATLAB, nos casos de estabilidade em pontos fixos a rede necessitou de apenas duas iterações, e em casos de ciclos limites, quatro iterações foram suficientes para identificar os ciclos. Os detalhes sobre a convergência da rede serão explorados no capítulo 4 Resultados e Análises.

3.5 Descrição da rede em VHDL

Para facilitar o desenvolvimento e organização do programa em VHDL, a estrutura da rede neural foi dividida em sete componentes: *Multiplexador*, *Neuronio*, *Flip Flop*, *Comparador*, *Maquina de estados*, *Contador* e *Display de 7 segmentos*. A relação entre estes componentes é apresentada no diagrama de blocos da Figura 29.

Figura 29 - Diagrama de blocos da rede neural descrita em VHDL



As funções de cada componente são especificadas a seguir:

Multiplexador

O *Multiplexador* tem a função de habilitar as entradas no *Neuronio* segundo um critério de seleção. As Figuras 30 e 31 mostram seu bloco e descrição em VHDL, respectivamente, cujos sinais de entrada e saída são:

- *sel*: sinal de seleção;
- *entrada*: sinal de entrada inicial da rede inserido por *hardware*;

- *sff*: sinal de realimentação da saída na entrada;
- *ent1*: sinal de saída da entrada habilitada.

Figura 30 - Bloco do Multiplexador

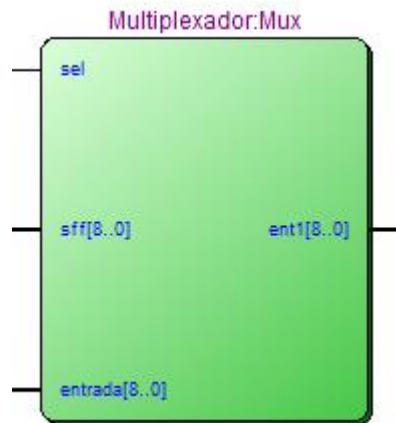


Figura 31 - Descrição do Multiplexador em VHDL

```

1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.ALL;
3  USE IEEE.std_logic_signed.ALL;
4  USE IEEE.std_logic_arith.ALL;
5  USE IEEE.numeric_bit.ALL;
6
7  --Declaracao do componente multiplexador
8  ENTITY Multiplexador IS
9  PORT (sff, entrada : IN STD_LOGIC_VECTOR(8 DOWNTO 0);
10       sel          : IN STD_LOGIC;
11       ent1         : OUT STD_LOGIC_VECTOR(8 DOWNTO 0));
12  END Multiplexador;
13
14  ARCHITECTURE behavior OF Multiplexador IS
15  BEGIN
16      WITH sel SELECT
17          ent1 <= sff      WHEN '0',
18                  entrada  WHEN OTHERS;
19  END behavior;

```

Se *sel* = '1' o *Neuronio* recebe a *entrada* inserida por *hardware*; se *sel* = '0' ocorre a realimentação, ou seja, a saída da última iteração é inserida no *Neuronio*.

Neuronio

Cada *Neuronio* apresenta um vetor de pesos da matriz **W**, totalizando 9 componentes. A Figura 32 e 33 mostram seu bloco e descrição em VHDL, respectivamente, cujos sinais de entrada e saída são:

- *ent1*: sinal de entrada no *Neuronio*;

- *eff*: sinal de saída do *Neuronio*.

Figura 32 - Bloco do Neuronio N1

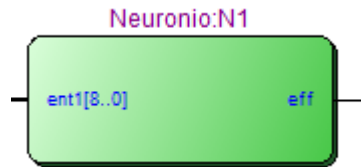


Figura 33 - Descrição do Neuronio em VHDL

```

1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.ALL;
3  USE IEEE.std_logic_signed.ALL;
4  USE IEEE.std_logic_arith.ALL;
5  USE IEEE.numeric_bit.ALL;
6
7  -- Declaracao do componente Neuronio
8  ENTITY Neuronio IS
9      GENERIC (E8_N, E7_N, E6_N, E5_N, E4_N, E3_N, E2_N, E1_N, E0_N : STD_LOGIC_VECTOR (6 DOWNTO 0));
10     PORT (ent1 : IN STD_LOGIC_VECTOR (8 DOWNTO 0);
11           eff : OUT STD_LOGIC);
12 END Neuronio;
13
14 ARCHITECTURE arch OF Neuronio IS
15     SIGNAL multi_N0, multi_N1, multi_N2, multi_N3, multi_N4, multi_N5, multi_N6, multi_N7, multi_N8 : STD_LOGIC_VECTOR (8 DOWNTO 0);
16     SIGNAL somaN : STD_LOGIC_VECTOR (8 DOWNTO 0);
17 BEGIN
18
19     PROCESS (ent1)
20     BEGIN
21         multi_N8 <= "00" & E8_N;
22         IF ent1(8) = '0' AND E8_N /= "0000000" THEN
23             multi_N8(6) <= NOT E8_N(6);
24         END IF;
25         multi_N7 <= "00" & E7_N;
26         IF ent1(7) = '0' AND E7_N /= "0000000" THEN
27             multi_N7(6) <= NOT E7_N(6);
28         END IF;
29         multi_N6 <= "00" & E6_N;
30         IF ent1(6) = '0' AND E6_N /= "0000000" THEN
31             multi_N6(6) <= NOT E6_N(6);
32         END IF;
33         multi_N5 <= "00" & E5_N;
34         IF ent1(5) = '0' AND E5_N /= "0000000" THEN
35             multi_N5(6) <= NOT E5_N(6);
36         END IF;
37         multi_N4 <= "00" & E4_N;
38         IF ent1(4) = '0' AND E4_N /= "0000000" THEN
39             multi_N4(6) <= NOT E4_N(6);
40         END IF;
41         multi_N3 <= "00" & E3_N;
42         IF ent1(3) = '0' AND E3_N /= "0000000" THEN
43             multi_N3(6) <= NOT E3_N(6);
44         END IF;
45         multi_N2 <= "00" & E2_N;
46         IF ent1(2) = '0' AND E2_N /= "0000000" THEN
47             multi_N2(6) <= NOT E2_N(6);
48         END IF;
49         multi_N1 <= "00" & E1_N;
50         IF ent1(1) = '0' AND E1_N /= "0000000" THEN
51             multi_N1(6) <= NOT E1_N(6);
52         END IF;
53         multi_N0 <= "00" & E0_N;
54         IF ent1(0) = '0' AND E0_N /= "0000000" THEN
55             multi_N0(6) <= NOT E0_N(6);
56         END IF;
57     END PROCESS;
58
59     somaN <= multi_N8 + multi_N7 + multi_N6 + multi_N5 + multi_N4 + multi_N3 + multi_N2 + multi_N1 + multi_N0;
60
61     eff <= '0' WHEN somaN(8) = '1' ELSE
62           '1' WHEN somaN(8) = '0';
63 END arch;
64

```

Os componentes *Neuronio* executam operações correspondentes à multiplicação matricial entre o vetor de entrada e o vetor de pesos, e limitam o resultado desta operação utilizando a função de ativação degrau bipolar. Estes processos resultam na obtenção do estado atual do neurônio, a saída *eff*.

Flip Flop

O componente *Flip Flop* é utilizado como uma linha de atraso para armazenar o estado da penúltima iteração (*sff*). As Figuras 34 e 35 mostram seu bloco e descrição em VHDL, respectivamente, cujos sinais de entrada e saída são:

- *rst*: sinal de controle reset, ativo em nível alto;
- *ck_D*: sinal de sincronismo gerado pela *Maquina de estados*;
- *eff*: sinal de entrada do *Flip Flop*;
- *sff*: sinal de saída do *Flip Flop*.

Figura 34 - Bloco do Flip Flop descrito em VHDL



Figura 35 - Descrição do Flip Flop em VHDL

```

1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.ALL;
3  USE IEEE.std_logic_signed.ALL;
4  USE IEEE.std_logic_arith.ALL;
5  USE IEEE.numeric_bit.ALL;
6
7  --Declaracao do componente flip flop
8  ENTITY Flip_Flop_D IS
9  PORT (eff, ck_D, rst : IN STD_LOGIC;
10       sff : OUT STD_LOGIC);
11  END Flip_Flop_D;
12
13  ARCHITECTURE behavior OF Flip_Flop_D IS
14  BEGIN
15  PROCESS (eff, ck_D, rst)
16  BEGIN
17      if rst = '1' then
18          sff <= '0';
19      elsif (ck_D = '1' AND ck_D'EVENT) THEN
20          sff <= eff;
21      end if;
22  END PROCESS;
23  END behavior;

```

O estado da penúltima iteração de cada *Neuronio* é colocada na saída do *Flip Flop* um ciclo depois de ter chegado à entrada, sendo o sinal de sincronismo (*ck_D*) controlado pela *Maquina de estados*.

Comparador

O *Comparador* atua na sequência do *Flip Flop*, sendo responsável pela comparação entre o estado atual e o da penúltima iteração. As Figuras 36 e 37 mostram seu bloco e descrição em VHDL, respectivamente, cujos sinais de entrada e saída são:

- *eff*: sinal que representa o estado da última iteração do *Neuronio*;
- *sff*: sinal que representa o estado da penúltima iteração do *Neuronio*;
- *comp*: sinal de saída resultado da comparação.

Figura 36 - Bloco do Comparador descrito em VHDL

Comparador:\GEN_2:1:comp_comparador



Figura 37 - Descrição do Comparador em VHDL

```

1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.ALL;
3  USE IEEE.std_logic_signed.ALL;
4  USE IEEE.std_logic_arith.ALL;
5  USE IEEE.numeric_bit.ALL;
6
7  --Declaracao do componente comparador
8  ENTITY Comparador IS
9      PORT (eff, sff : IN STD_LOGIC;
10           comp      : OUT STD_LOGIC);
11  END Comparador;
12
13  ARCHITECTURE behavior OF Comparador IS
14  BEGIN
15      comp <= eff xnor sff;
16  END behavior;

```

O *Comparador* corresponde à porta lógica XNOR, também conhecida como função coincidência, ele compara os dois últimos estados do *Neuronio* e apresenta saída igual a um quando estes valores forem iguais.

Contador

Este componente conta o número de iterações que a rede realizou, as Figuras 38 e 39 mostram seu bloco e descrição em VHDL, respectivamente, cujos sinais de entrada e saída são:

- *rst*: sinal de controle reset, ativo em nível alto;
- *ck_D*: sinal de entrada de sincronismo gerado pela *Maquina de estados*;
- *saída_contador*: sinal que indica a quantidade de iterações da rede.

Figura 38 - Bloco do Contador descrito em VHDL

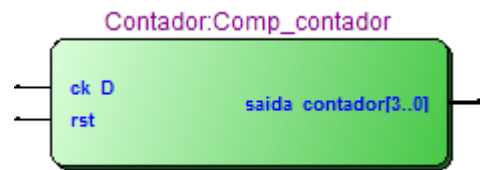


Figura 39 - Descrição do Contador em VHDL

```

1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.ALL;
3  USE IEEE.std_logic_signed.ALL;
4  USE IEEE.std_logic_arith.ALL;
5  USE IEEE.numeric_bit.ALL;
6
7  --Declaracao do componente contador
8  ENTITY Contador IS
9  PORT(ck_D, rst      : IN   STD_LOGIC;
10      saida_contador : OUT  STD_LOGIC_VECTOR (3 DOWNTO 0));
11  END Contador;
12
13  ARCHITECTURE behavior OF Contador IS
14  BEGIN
15  PROCESS (ck_D, rst)
16  VARIABLE contagem: INTEGER RANGE 0 TO 15;
17  BEGIN
18      IF rst = '1' THEN
19          contagem := 1;
20      ELSIF (ck_D = '1' AND ck_D'EVENT) THEN
21          contagem := contagem + 1;
22      END IF;
23      saida_contador <= conv_std_logic_vector(contagem, 4);
24  END PROCESS;
25  END behavior;

```

Um Contador de quatro bits foi suficiente para esta aplicação, pois o limite máximo estipulado é de 4 iterações, sendo sua contagem incrementada a cada evento do sinal *ck_D*.

Maquina de estados

A *Maquina de estados* é responsável pela sincronia dos componentes *Neuronio* e atualização da saída da rede. As Figuras 40 e 41 mostram seu bloco e descrição em VHDL, respectivamente, cujos sinais de entrada e saída são:

- *rst*: sinal de controle reset, ativo em nível alto;
- *ck*: sinal de entrada de sincronismo;
- *s_and*: sinal que verifica se todos os comparadores são coincidentes;
- *saida_contador*: sinal de controle que limita a quantidade de iterações a no máximo 4;

- *sel*: sinal de controle da *Maquina de estados* que seleciona as entradas do *Multiplexador*;
- *ck_D*: sinal de controle gerado pela *Maquina de estados* para ser sinal de sincronismo nos flip flops e no contador;
- *s*: sinal de controle que habilita a exibição na saída da rede neural.

Figura 40 - Bloco da Maquina de estados descrita em VHDL

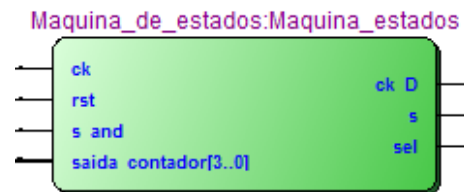


Figura 41 - Descrição da Maquina de estados em VHDL

```

1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.ALL;
3  USE IEEE.std_logic_signed.ALL;
4  USE IEEE.std_logic_arith.ALL;
5  USE IEEE.numeric_bit.ALL;
6
7  --Declaracao do componente maquina de estados
8  ENTITY Maquina_de_estados IS
9  PORT(s_and, ck, rst : IN STD_LOGIC;
10      saida_contador : IN STD_LOGIC_VECTOR (3 DOWNTO 0);
11      sel, ck_D, s : OUT STD_LOGIC);
12  END Maquina_de_estados;
13
14  ARCHITECTURE behavior OF Maquina_de_estados IS
15
16      TYPE st IS (caso_start, caso_comp0, caso_comp1, caso_final);
17      SIGNAL estado : st;
18
19  BEGIN
20  abc: PROCESS (ck, rst, saida_contador, s_and)
21  BEGIN
22
23      IF rst = '1' THEN
24          sel <= '1';
25          estado <= caso_start;
26      ELSIF (ck = '1' AND ck'EVENT) THEN
27          CASE estado IS
28              WHEN caso_start =>
29                  sel <= '1';
30                  estado <= caso_comp0;
31              WHEN caso_comp0 =>
32                  sel <= '0';
33                  IF saida_contador < "0100" THEN estado <= caso_comp1;
34                  ELSE estado <= caso_final;
35                  END IF;
36              WHEN caso_comp1 =>
37                  sel <= '0';
38                  IF s_and = '0' THEN estado <= caso_comp0;
39                  ELSE estado <= caso_final;
40                  END IF;
41              WHEN caso_final =>
42                  sel <= '0';
43          END CASE;
44      END IF;
45  END PROCESS abc;

```

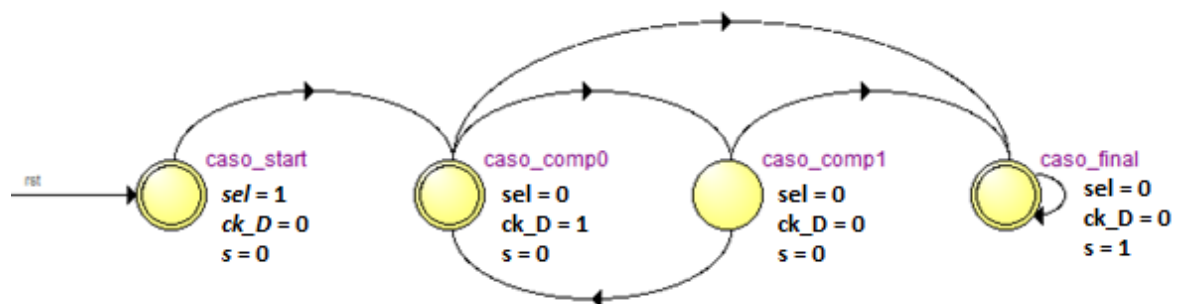
```

46
47 WITH estado SELECT
48     ck_D <= '0' WHEN caso_start,
49         '1' WHEN caso_comp0,
50         '0' WHEN caso_comp1,
51         '0' WHEN caso_final;
52
53 WITH estado SELECT
54     s <= '0' WHEN caso_start,
55         '0' WHEN caso_comp0,
56         '0' WHEN caso_comp1,
57         '1' WHEN caso_final;
58 END behavior;

```

O sinal de reset, *rst*, quando ativo coloca o sinal de seleção do *Multiplexador*, *sel*, em nível alto, permitindo que as entradas inseridas no circuito alimentem os componentes *Neuronio*, e leva a *Maquina de estados* ao seu primeiro estado, o *caso_start*. Em seguida, a cada evento do sinal de clock, *ck*, a *Maquina de estados* percorre os estados ilustrados na Figura 42, e chega ao estado *caso_final* quando uma das seguintes condições for satisfeita: sinal *s_and* = '1' indicando que o estado atual é igual ao estado da penúltima iteração, ou o sinal binário *saida_contador* = "0100" indicando o número máximo de iterações que a rede deve executar. Este número foi escolhido com base nos resultados apresentados pela rede descrita em MATLAB (ver Tabela 10 do Apêndice II) e visando economizar recursos de hardware.

Figura 42 - Evolução dos estados da máquina de estados



Display de 7 segmentos

Este componente realiza a decodificação do número binário em sua representação decimal em um display de 7 segmentos. O display é do tipo catodo comum, ou seja, os segmentos acendem em nível lógico alto e apagam em nível lógico

baixo. As Figuras 43 e 44 mostram seu bloco e descrição em VHDL, respectivamente, cujos sinais de entrada e saída são:

- *saida_contador*: sinal que apresenta o número de iterações da rede em representação binária;
- *s_cont*: sinal de saída exibido no display de 7 segmentos.

Figura 43 - Bloco do Conversor Binário - Decimal descrito em VHDL

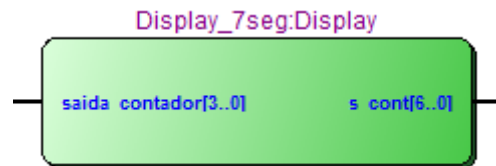


Figura 44 - Descrição do Display de 7 segmentos em VHDL

```

1  LIBRARY IEEE;
2  USE IEEE.std_logic_1164.ALL;
3  USE IEEE.std_logic_signed.ALL;
4  USE IEEE.std_logic_arith.ALL;
5  USE IEEE.numeric_bit.ALL;
6
7  --Declaracao do componente display de 7 segmentos
8
9  ENTITY Display_7seg IS
10     PORT(saida_contador : IN STD_LOGIC_VECTOR(3 DOWNTO 0);
11          s_cont          : OUT STD_LOGIC_VECTOR(6 DOWNTO 0));
12  END Display_7seg;
13
14  ARCHITECTURE behavior OF Display_7seg IS
15     SIGNAL s: STD_LOGIC_VECTOR(6 DOWNTO 0);
16  BEGIN
17     abc: PROCESS (saida_contador)
18     BEGIN
19         CASE saida_contador IS
20             WHEN "0000" =>
21                 s <= "1111110"; --abcdefg
22             WHEN "0001" =>
23                 s <= "0110000";
24             WHEN "0010" =>
25                 s <= "1101101";
26             WHEN "0011" =>
27                 s <= "1111001";
28             WHEN "0100" =>
29                 s <= "0110011";
30             WHEN "0101" =>
31                 s <= "1011011";
32             WHEN "0110" =>
33                 s <= "1011111";
34             WHEN "0111" =>
35                 s <= "1110000";
36             WHEN "1000" =>
37                 s <= "1111111";
38             WHEN "1001" =>
39                 s <= "1111011";
40             WHEN OTHERS =>
41                 s <= "1001111";
42         END CASE;
43         s_cont <= NOT s;
44     END PROCESS;
45  END behavior;
46

```

A entrada do decodificador recebe a saída do *Contador* e exibe no display o número iterações realizadas pela rede.

A descrição da arquitetura principal que relaciona todos os componentes é apresentada na Figura 45.

Figura 45 - Arquitetura principal

```

69  USE IEEE.std_logic_arith.ALL;
70  USE IEEE.numeric_bit.ALL;
71
72  ENTITY Hopfield IS
73  PORT(ck, rst      : IN      STD_LOGIC;
74        entrada     : IN      STD_LOGIC_VECTOR(8 DOWNTO 0);
75        s_cont       : OUT     STD_LOGIC_VECTOR(6 DOWNTO 0);
76        entrada_led  : OUT     STD_LOGIC_VECTOR(8 DOWNTO 0);
77        saida        : OUT     STD_LOGIC_VECTOR(8 DOWNTO 0));
78  END Hopfield;
79
80  ARCHITECTURE rede OF Hopfield IS
81
82    -- Lembranca dos componentes
83    COMPONENT Neuronio
84      GENERIC (E8_N, E7_N, E6_N, E5_N, E4_N, E3_N, E2_N, E1_N, E0_N : STD_LOGIC_VECTOR (6 DOWNTO 0));
85      PORT (ent1 : IN      STD_LOGIC_VECTOR(8 DOWNTO 0);
86            eff  : OUT     STD_LOGIC);
87    END COMPONENT;
88
89    COMPONENT Multiplexador IS
90      PORT (sff, entrada : IN      STD_LOGIC_VECTOR(8 DOWNTO 0);
91            sel          : IN      STD_LOGIC;
92            ent1         : OUT     STD_LOGIC_VECTOR(8 DOWNTO 0));
93    END COMPONENT;
94
95    COMPONENT Flip_Flop_D IS
96      PORT (eff, ck_D, rst : IN      STD_LOGIC;
97            sff           : OUT     STD_LOGIC);
98    END COMPONENT;
99
100   COMPONENT Comparador IS
101     PORT(eff, sff : IN      STD_LOGIC;
102           comp   : OUT     STD_LOGIC);
103   END COMPONENT;
104
105   COMPONENT Contador IS
106     PORT(ck_D, rst : IN      STD_LOGIC;
107           saida_contador : OUT STD_LOGIC_VECTOR (3 DOWNTO 0));
108   END COMPONENT;
109
110   COMPONENT Maquina_de_estados IS
111     PORT(s_and, ck, rst : IN      STD_LOGIC;
112           saida_contador : IN      STD_LOGIC_VECTOR (3 DOWNTO 0);
113           sel, ck_D, s   : OUT     STD_LOGIC);
114   END COMPONENT;
115
116   COMPONENT Display_7seg IS
117     PORT(saida_contador : IN      STD_LOGIC_VECTOR(3 DOWNTO 0);
118           s_cont        : OUT     STD_LOGIC_VECTOR(6 DOWNTO 0));
119   END COMPONENT;
120

```

```

121 -- Declaracao dos sinais
122 SIGNAL s, ck_D: STD_LOGIC := '0';
123 SIGNAL eff, sff, ent1, comp: STD_LOGIC_VECTOR (8 DOWNTO 0);
124 SIGNAL saida_contador: STD_LOGIC_VECTOR(3 DOWNTO 0);
125 SIGNAL sel: STD_LOGIC := '0';
126 SIGNAL s_and : STD_LOGIC := '1';
127
128 BEGIN
129
130     entrada_led <= entrada;
131
132     Mux : Multiplexador PORT MAP(sff, entrada, sel, ent1);
133
134     N8 : Neuronio
135     GENERIC MAP ("0000000", "1000111", "0000111", "1000111", "0000000", "1000111", "0000111", "1000111", "0000111")
136     PORT MAP (ent1, eff(8));
137
138     N7 : Neuronio
139     GENERIC MAP ("1000111", "0000000", "1000111", "0000111", "0000000", "0000111", "1000111", "0000111", "1000111")
140     PORT MAP (ent1, eff(7));
141
142     N6 : Neuronio
143     GENERIC MAP ("0000111", "1000111", "0000000", "1000111", "0000000", "1000111", "0000111", "1000111", "0000111")
144     PORT MAP (ent1, eff(6));
145
146     N5 : Neuronio
147     GENERIC MAP ("1000111", "0000111", "1000111", "0000000", "0000000", "0000111", "1000111", "0000111", "1000111")
148     PORT MAP (ent1, eff(5));
149
150     N4 : Neuronio
151     GENERIC MAP ("0000000", "0000000", "0000000", "0000000", "0000000", "0000000", "0000000", "0000000", "0000000")
152     PORT MAP (ent1, eff(4));
153
154     N3 : Neuronio
155     GENERIC MAP ("1000111", "0000111", "1000111", "0000111", "0000000", "0000000", "1000111", "0000111", "1000111")
156     PORT MAP (ent1, eff(3));
157
158     N2 : Neuronio
159     GENERIC MAP ("0000111", "1000111", "0000111", "1000111", "0000000", "1000111", "0000000", "1000111", "0000111")
160     PORT MAP (ent1, eff(2));
161
162     N1 : Neuronio
163     GENERIC MAP ("1000111", "0000111", "1000111", "0000111", "0000000", "0000111", "1000111", "0000000", "1000111")
164     PORT MAP (ent1, eff(1));
165
166     N0 : Neuronio
167     GENERIC MAP ("0000111", "1000111", "0000111", "1000111", "0000000", "1000111", "0000111", "1000111", "0000000")
168     PORT MAP (ent1, eff(0));
169
170     GEN_1: FOR i IN 8 DOWNTO 0 GENERATE
171     ff : Flip_Flop_D PORT MAP(eff(i), ck_D, rst, sff(i));
172     END GENERATE GEN_1;
173
174     GEN_2: FOR i IN 8 DOWNTO 0 GENERATE
175     comp_comparador : Comparador PORT MAP(eff(i), sff(i), comp(i));
176     END GENERATE GEN_2;
177
178     s_and <= comp(8) AND comp(7) AND comp(6) AND comp(5) AND comp(4) AND comp(3) AND comp(2) AND comp(1) AND comp(0);
179
180     Comp_contador : Contador PORT MAP(ck_D,rst,saida_contador);
181
182     Display : Display_7seg PORT MAP(saida_contador,s_cont);
183
184     Maquina_estados : Maquina_de_estados PORT MAP(s_and, ck, rst, saida_contador, sel, ck_D, s);
185
186     PROCESS (s)
187     BEGIN
188     CASE s IS
189     WHEN '1' => saida <= eff;
190     WHEN '0' => saida <= "11111111";
191     END CASE;
192     END PROCESS;
193
194     rede;

```

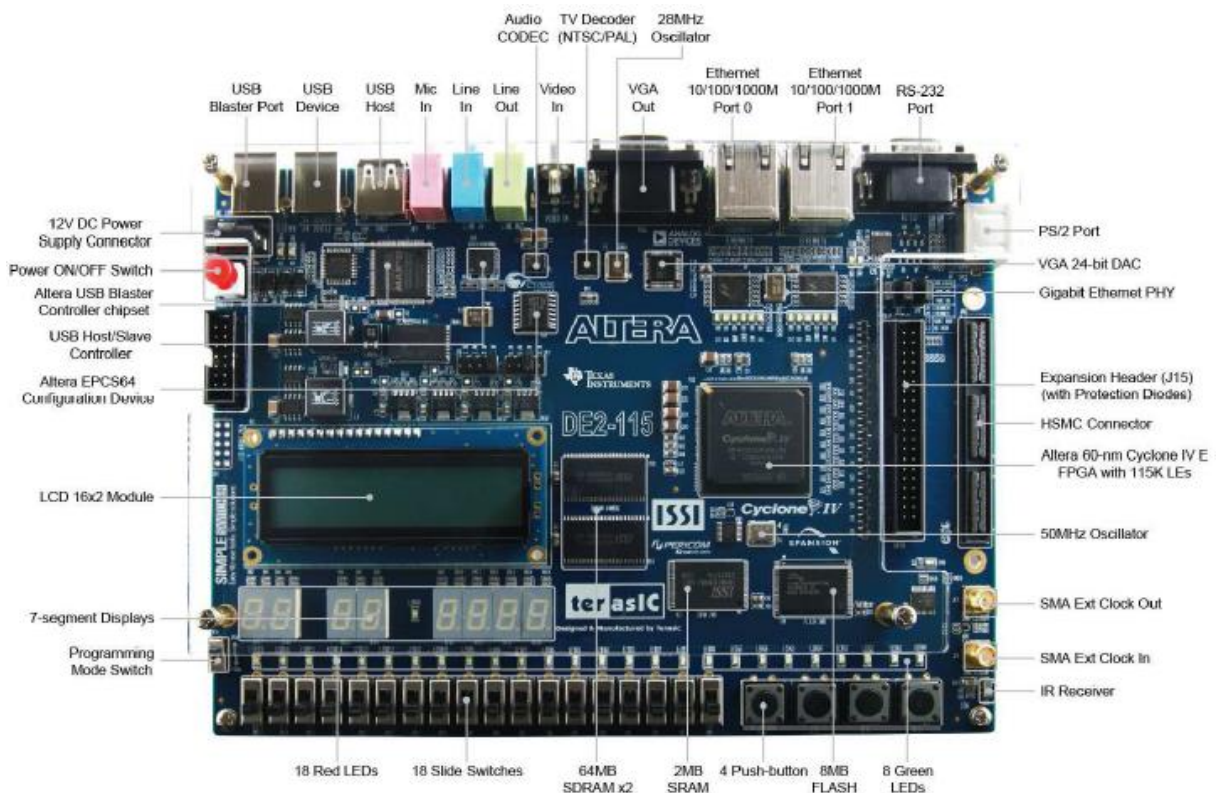
Os circuitos RTL gerados para cada componente são apresentados no Apêndice I.

3.6 Módulo de desenvolvimento

Para síntese em *hardware* da rede de Hopfield foi utilizado o módulo DE2-115 da Terasic, ilustrado na Figura 46.

Este módulo possui programador USB-Blaster embutido, cartão SD, conector para microfone, saída e entrada de vídeo, porta RS232, sensor infravermelho, porta para teclado ou mouse, ethernet, USB (tipos A e B), 8 displays de 7 segmentos, display de caracteres LCD 16x2, 18 Chaves Deslizantes, 18 LEDs vermelhos e 9 verdes, 4 botões (sem ruído), clock de 50 MHz e entrada e saída para clocks externos. Esta grande quantidade de interfaces permite que o módulo possa ser utilizado em diversas aplicações.

Figura 46 - Placa do modulo DE2-115



Fonte: TERICAS

Como ilustrado na Tabela 5, na implementação da rede neural de Hopfield serão utilizadas 9 das 18 Chaves Deslizantes (SW17 a SW0) para inserir as entradas, e os LEDs vermelhos (LEDR17 a LEDR0) serão divididos para a apresentação da entrada e do estado final da rede.

Tabela 5 - Relação das Chaves e LEDs com as entradas (e) e saídas (s) da rede

LED R17	LED R16	LED R15	LED R14	LED R13	LED R12	LED R11	LED R10	LED R9	LED R8	LED R7	LED R6	LED R5	LED R4	LED R3	LED R2	LED R1	LED R0
e(9)	e(8)	e(7)	e(6)	e(5)	e(4)	e(3)	e(2)	e(1)	s(9)	s(8)	s(7)	s(6)	s(5)	s(4)	s(3)	s(2)	s(1)
SW 17	SW 16	SW 15	SW 14	SW 13	SW 12	SW 11	SW 10	SW 9	SW 8	SW 7	SW 6	SW 5	SW 4	SW 3	SW 2	SW 1	SW 0
e(9)	e(8)	e(7)	e(6)	e(5)	e(4)	e(3)	e(2)	e(1)	NC	NC	NC	NC	NC	NC	NC	NC	NC

Em um display de 7 segmentos é apresentado o número de iterações realizadas pela rede, e um botão sem ruído é utilizado para o sinal de reset.

O principal componente do módulo é o FPGA Cyclone IV EP4CE115F29 da Altera, que possui 114.480 elementos lógicos, 3.88Kbits de memória interna e 266 multiplicadores de 18 x 18 bits. A Figura 47 exibe a descrição de *hardware* da rede neural implementada, evidenciando o número de componentes internos utilizados.

Figura 47 - Interface do software Quartus II exibindo o número de componentes internos do FPGA utilizados no projeto

Flow Summary	
Flow Status	Successful - Thu Nov 05 20:22:50 2015
Quartus II 32-bit Version	12.0 Build 263 08/02/2012 SP 2 SJ Web Edition
Revision Name	Hopfield
Top-level Entity Name	Hopfield
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	102 / 114,480 (< 1 %)
Total combinational functions	101 / 114,480 (< 1 %)
Dedicated logic registers	18 / 114,480 (< 1 %)
Total registers	18
Total pins	36 / 529 (7 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

Da Figura 47, observa-se que as características do FPGA estão muito acima das necessidades requeridas para o desenvolvimento deste trabalho, no entanto, a portabilidade oferecida pela linguagem VHDL permite que o circuito seja sintetizado em diversos dispositivos diferentes.

A configuração do FPGA é feita através da entrada Blaster do Kit conectada ao computador por um cabo USB, e o *software* utilizado para síntese do projeto é o Quartus II.

4 Resultados e Análises

Neste capítulo são apresentados e discutidos os resultados do trabalho.

4.1 Resultados obtidos no MATLAB

Utilizando o programa descrito em MATLAB, foi simulado o comportamento da rede para todas as entradas possíveis, totalizando 512 combinações. A Tabela 6 apresenta uma amostra com os testes de 1 a 15, e a tabela completa com todos os testes é encontrada no Apêndice II (Tabela 10).

Tabela 6 - Amostra dos resultados obtidos a partir da rede descrita no MATLAB

Nº do teste	Entrada		Saída em representação polarizada		It n	Distância de Hamming	
	Decimal	Polarizada	v_anterior (n ímpar)	v_atual (n par)		Dist z1	Dist z2
313	0	-1-1-1-1-1-1-1-1-1	+1+1+1+1+1+1+1+1	-1-1-1-1+1-1-1-1-1	1000	10	10
411	1	-1-1-1-1-1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
201	2	-1-1-1-1-1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
314	3	-1-1-1-1-1-1-1+1+1	+1+1+1+1+1+1+1-1-1	-1-1-1-1+1-1-1+1+1	1000	10	10
412	4	-1-1-1-1-1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
475	5	-1-1-1-1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
315	6	-1-1-1-1-1-1+1+1-1	+1+1+1+1+1+1-1-1+1	-1-1-1-1+1-1+1+1-1	1000	10	10
413	7	-1-1-1-1-1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
202	8	-1-1-1-1-1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
316	9	-1-1-1-1-1+1-1-1+1	+1+1+1+1+1-1+1+1-1	-1-1-1-1+1+1-1-1+1	1000	10	10
103	10	-1-1-1-1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
203	11	-1-1-1-1-1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
317	12	-1-1-1-1-1+1+1-1-1	+1+1+1+1+1-1-1+1+1	-1-1-1-1+1+1+1-1-1	1000	10	10
414	13	-1-1-1-1-1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
204	14	-1-1-1-1-1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12

Como descrito na seção 2.5.6. sobre os modos de atualização da rede de Hopfield, na atualização síncrona cada estado possui apenas um sucessor, e assim a rede pode atingir a estabilidade nos pontos fixos correspondentes aos padrões previamente armazenados, ou se estabilizar em ciclos limites. Para representar a ocorrência deste fenômeno são apresentados na Tabela 6 os dois últimos estados

gerados a partir da entrada de cada teste, sendo eles o **v_anterior**, que representa o estado da rede para iterações de número ímpar, e **v_atual**, que representa o estado para iterações de número par. Além disso, são apresentados os parâmetros $Dist\ z_1$ e $Dist\ z_2$ que correspondem às distâncias de Hamming entre o vetor de entrada e o padrão z^1 e entre o vetor de entrada e o padrão z^2 , respectivamente.

As entradas que levam a rede a convergir para o padrão z^1 são destacadas em vermelho, e as que levam a convergir para o padrão z^2 são destacadas em azul, como esperado, nestes dois casos os estados **v_anterior** e **v_atual** são iguais. As demais entradas fazem com que a rede permaneça em um modo oscilatório entre dois estados, os ciclos limites. Este comportamento é explicado pelo cálculo da distância da Hamming, que corresponde à menor distância entre dois padrões e é calculada pelo número de bits que um padrão difere do outro. Assim, a rede converge para o padrão z^1 quando a distância de Hamming for menor para o padrão z^1 do que para o padrão z^2 , do contrário a rede converge para o padrão z^2 . No caso da distância de Hamming ser igual para os dois padrões, ocorrem os ciclos limites.

A Tabela 6 também apresenta o número n de iterações realizadas pela rede neural para cada teste, nota-se que quando a rede se estabiliza em um ponto fixo ocorrem duas iterações, sendo que na primeira iteração a saída já apresenta o estado estável, mas é necessário realizar a segunda para confirmar que **v_anterior** = **v_atual**. Nos casos de estabilidade em ciclos limites a rede realiza 1000 iterações, que é o limite máximo estipulado, e nota-se que a saída da rede para todas as iterações de números ímpar são iguais entre si, do mesmo modo que as iterações de número par. Assim, quatro iterações são suficientes para determinar o comportamento oscilatório periódico da rede, sendo esta consideração utilizada para a implementação da rede neural em *hardware*.

A Tabela 7 considera as 512 entradas inseridas na rede neural e apresenta a quantidade de vezes que cada estado final foi obtido.

Tabela 7 - Número de ocorrência para cada estado final da rede

Estado final	Ocorrência	% de Ocorrência
Padrão z_1 "-1 +1 -1 +1 +1 +1 -1 +1 -1"	186	36,33
Padrão z_2 "+1 -1 +1 -1 +1 -1 +1 -1 +1"	186	36,33
Ciclo limite	140	27,34

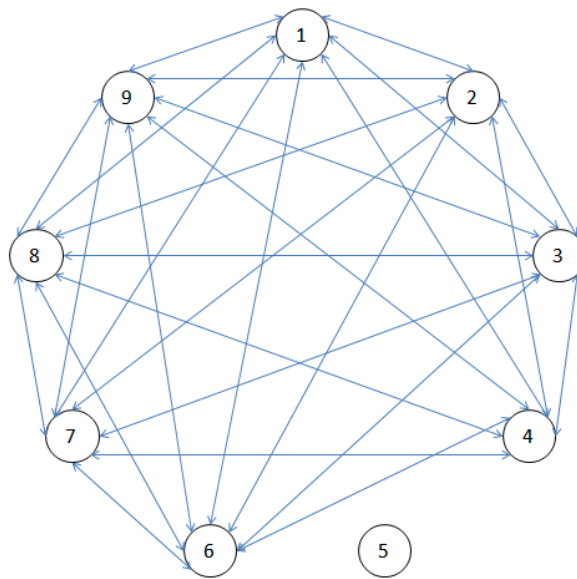
A rede simulada teve como saída 35 ciclos de dois estados, sendo que cada um deles ocorreu para 4 entradas distintas, totalizando 140 ciclos limite. A Tabela 8 apresenta estes 35 ciclos e aponta sua ocorrência nos testes.

Tabela 8 - Ocorrência dos ciclos de dois estados

v_anterior	v_atual	Ocorrência nas entradas decimais
+1 +1 +1 +1 +1 +1 +1 +1 +1	-1 -1 -1 -1 +1 -1 -1 -1 -1	0, 16, 495 e 511
+1 +1 +1 +1 +1 +1 +1 -1 -1	-1 -1 -1 -1 +1 -1 -1 +1 +1	3, 19, 492 e 508
+1 +1 +1 +1 +1 +1 -1 -1 +1	-1 -1 -1 -1 +1 -1 +1 +1 -1	6, 22, 489 e 505
+1 +1 +1 +1 +1 -1 +1 +1 -1	-1 -1 -1 -1 +1 +1 -1 -1 +1	9, 25, 486 e 502
+1 +1 +1 +1 +1 -1 -1 +1 +1	-1 -1 -1 -1 +1 +1 +1 -1 -1	12, 28, 483 e 499
+1 +1 +1 +1 +1 -1 -1 -1 -1	-1 -1 -1 -1 +1 +1 +1 +1 +1	15, 31, 480 e 496
+1 +1 +1 -1 +1 +1 +1 +1 -1	-1 -1 -1 +1 +1 -1 -1 -1 +1	33, 49, 462 e 478
+1 +1 +1 -1 +1 +1 -1 +1 +1	-1 -1 -1 +1 +1 -1 +1 -1 -1	36, 52, 459 e 475
+1 +1 +1 -1 +1 +1 -1 -1 -1	-1 -1 -1 +1 +1 -1 +1 +1 +1	39, 55, 456 e 472
+1 +1 +1 -1 +1 -1 -1 +1 -1	-1 -1 -1 +1 +1 +1 +1 -1 +1	45, 61, 450 e 466
+1 +1 -1 +1 +1 +1 +1 -1 +1	-1 -1 +1 -1 +1 -1 -1 +1 -1	66, 82, 429 e 445
+1 +1 -1 +1 +1 -1 +1 +1 +1	-1 -1 +1 -1 +1 +1 -1 -1 -1	72, 88, 423 e 439
+1 +1 -1 +1 +1 -1 +1 -1 -1	-1 -1 +1 -1 +1 +1 -1 +1 +1	75, 91, 420 e 436
+1 +1 -1 +1 +1 -1 -1 -1 +1	-1 -1 +1 -1 +1 +1 +1 +1 -1	78, 94, 417 e 433
+1 +1 -1 -1 +1 +1 +1 +1 +1	-1 -1 +1 +1 -1 -1 -1 -1 -1	96, 112, 399 e 415
+1 +1 -1 -1 +1 +1 +1 -1 -1	-1 -1 +1 +1 +1 -1 -1 +1 +1	99, 115, 396 e 412
+1 +1 -1 -1 +1 +1 -1 -1 +1	-1 -1 +1 +1 +1 -1 +1 +1 -1	102, 118, 393 e 409
+1 +1 -1 -1 +1 -1 +1 +1 -1	-1 -1 +1 +1 +1 +1 -1 -1 +1	105, 121, 390 e 406
+1 +1 -1 -1 +1 -1 -1 +1 +1	-1 -1 +1 +1 +1 +1 +1 -1 -1	108, 124, 387 e 403
+1 +1 -1 -1 +1 -1 -1 -1 -1	-1 -1 +1 +1 +1 +1 +1 +1 +1	111, 127, 384 e 400
+1 -1 +1 +1 +1 +1 +1 +1 -1	-1 +1 -1 -1 +1 -1 -1 -1 +1	129, 145, 375 e 382
+1 -1 +1 +1 +1 +1 -1 +1 +1	-1 +1 -1 -1 +1 -1 +1 -1 -1	132, 148, 363 e 379
+1 -1 +1 +1 +1 +1 -1 -1 -1	-1 +1 -1 -1 +1 -1 +1 +1 +1	135, 151, 360 e 376
+1 -1 +1 +1 +1 -1 -1 +1 -1	-1 +1 -1 -1 +1 +1 +1 -1 +1	141, 157, 354 e 370
+1 -1 +1 -1 +1 +1 -1 +1 -1	-1 +1 -1 +1 +1 -1 +1 -1 +1	165, 181, 330 e 346
+1 -1 -1 +1 +1 +1 +1 +1 +1	-1 +1 +1 -1 +1 -1 -1 -1 -1	192, 208, 303 e 319
+1 -1 -1 +1 +1 +1 +1 -1 -1	-1 +1 +1 -1 +1 -1 -1 +1 +1	195, 211, 300 e 316
+1 -1 -1 +1 +1 +1 -1 -1 +1	-1 +1 +1 -1 +1 -1 +1 +1 -1	198, 214, 297 e 313
+1 -1 -1 +1 +1 -1 +1 +1 -1	-1 +1 +1 -1 +1 +1 -1 -1 +1	201, 217, 294 e 310
+1 -1 -1 +1 +1 -1 -1 +1 +1	-1 +1 +1 -1 +1 +1 +1 -1 -1	204, 220, 291 e 307
+1 -1 -1 +1 +1 -1 -1 -1 -1	-1 +1 +1 -1 +1 +1 +1 +1 +1	207, 223, 288 e 304
+1 -1 -1 -1 +1 +1 +1 +1 -1	-1 +1 +1 +1 +1 -1 -1 -1 +1	225, 241, 270 e 286
+1 -1 -1 -1 +1 +1 -1 +1 +1	-1 +1 +1 +1 +1 -1 +1 -1 -1	228, 244, 267 e 283
+1 -1 -1 -1 +1 +1 -1 -1 -1	-1 +1 +1 +1 +1 -1 +1 +1 +1	231, 247, 264 e 280
+1 -1 -1 -1 +1 -1 -1 +1 -1	-1 +1 +1 +1 +1 +1 +1 -1 +1	237, 253, 258, 274

Observa-se da Tabela 8 que os ciclos limite são formados por estados de sinais opostos, ou seja, $v_{\text{anterior}} = -v_{\text{atual}}$, com exceção do quinto elemento, que é sempre igual a +1. Este comportamento pode ser explicado através da matriz \mathbf{W} , que apresenta peso nulo para a conexão do quinto neurônio com todos os outros. Para ilustrar este fato, a Figura 48 apresenta todas as conexões não nulas entre os neurônios.

Figura 48 - Conexões não nulas entre os neurônios

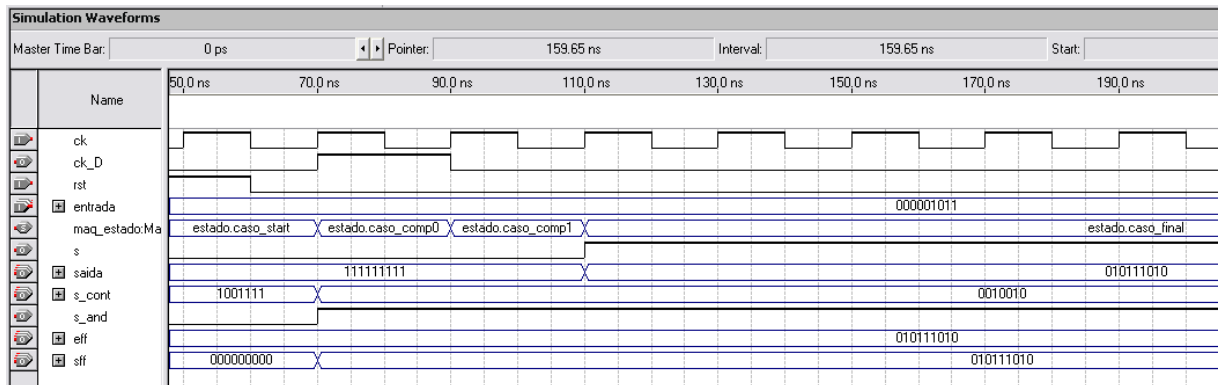


Como o quinto neurônio da rede é independente de todos os outros, seus estados são afetados apenas pela função de ativação adotada pela rede, que é a degrau bipolar.

4.2 Síntese no FPGA

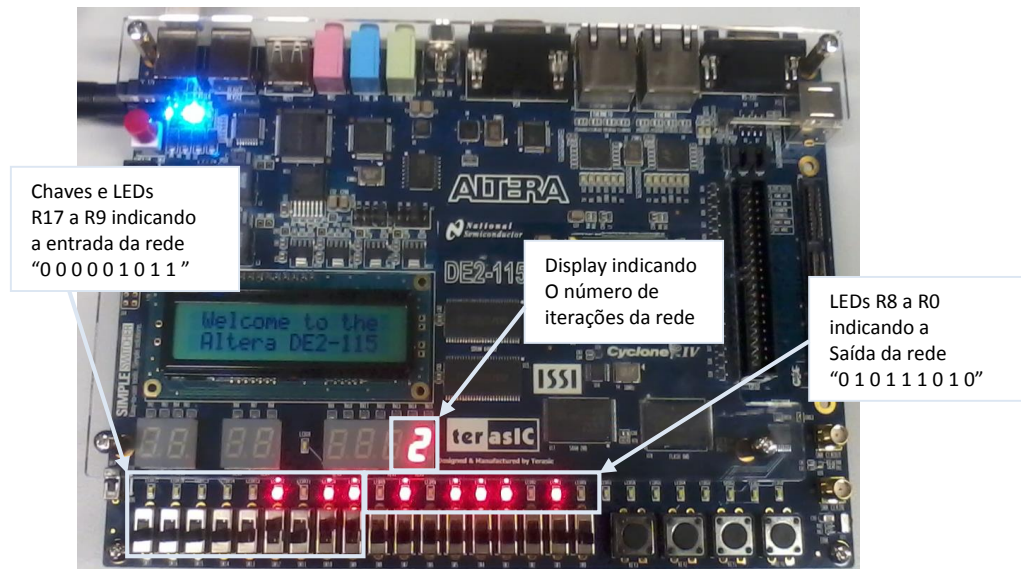
A rede neural descrita em VHDL foi simulada no *software* Quartus II, e a Figura 49 mostra uma simulação da rede para a mesma entrada do teste nº 203 da Tabela 6, que corresponde a um caso em que o estado final da rede converge para o padrão z^1 .

Figura 49 - Simulação no Quartus II para uma saída convergente



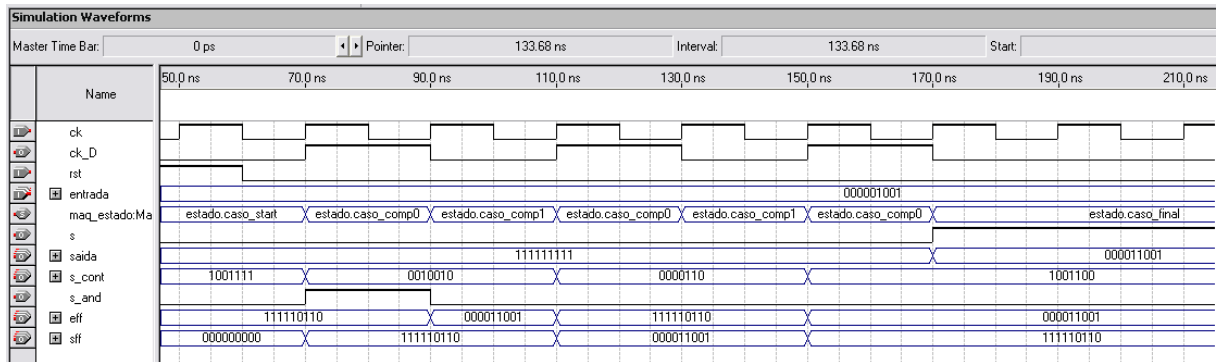
Após a primeira iteração a rede é realimentada, e como o novo estado obtido (*eff*) é igual ao estado anterior (*sff*) o sinal que compara as saídas (*s_and*) é levado a nível alto, sendo esta uma das condições que determinam o fim do processo de realimentação. A Figura 50 ilustra o funcionamento da rede de Hopfield sintetizada no FPGA para a mesma entrada de simulação da Figura 49.

Figura 50 - Circuito da rede neural implantada no Kit DE2-115 apresentando saída convergente



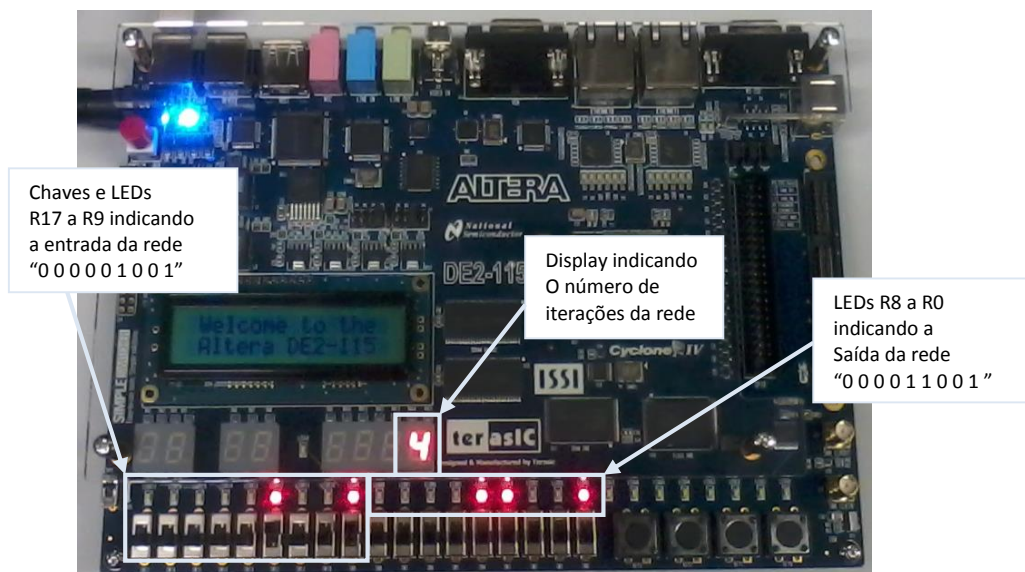
De modo análogo, as Figuras 51 e 52 mostram, respectivamente, a simulação e a implementação em *hardware* da rede para a mesma entrada do teste nº 316 da Tabela 6, que corresponde a um caso em que o estado final da rede é um ciclo limite.

Figura 51 - Simulação no Quartus para uma saída oscilatória



Na simulação da Figura 51 a condição que determina o fim das realimentações é o contador não ultrapassar 4 contagens.

Figura 52 - Circuito da rede neural implementada no Kit DE2-115 apresentando saída oscilatória



Como observado nas Figuras 50 e 52 as entradas da rede sintetizada no FPGA estão em notação binária, no entanto, a estrutura da rede descrita em VHDL realiza uma adaptação para efetuar seus processos considerando notação polarizada. Esta abordagem é feita no componente *Neuronio*, que para a etapa de multiplicação das entradas pelos pesos, executa mudança de sinal no caso de a entrada ser “0” e o peso ser não nulo, correspondendo à multiplicação por “-1”. Além de utilizar a função degrau

como função de ativação para obter na saída números binários. Essas adaptações são apenas periféricas e, portanto não prejudicam o desempenho da rede.

4.3 Discussão dos Resultados

A verificação do funcionamento da rede de Hopfield sintetizada em *hardware*, procedimento ilustrado nas Figuras 50 e 52, foi executada considerando como entradas 70% dos valores de teste da simulação em MATLAB (Tabela 10 do Apêndice II), e as saídas do *hardware* reproduziram com 100% de precisão as saídas da simulação do MATLAB. Isso comprova que o erro relativo de 0,57%, calculado na seção 3.3 (Tabela 4), relacionado à conversão do número decimal para binário na matriz de pesos W , não prejudicou o desempenho da rede sintetizada na FPGA.

Pela análise da Tabela 10 observa-se que, devido à utilização da notação polarizada, o número de elementos corrompidos em cada padrão corresponde à metade da distância de Hamming entre o vetor de entrada e os padrões, e que quando o número de elementos corrompidos é menor para o padrão \mathbf{z}^1 do que para o padrão \mathbf{z}^2 a rede converge para o padrão \mathbf{z}^1 , como exemplo o teste nº 1, caso contrário converge para o padrão \mathbf{z}^2 , exemplo do teste nº 257, e quando o número de elementos corrompidos é igual para ambos os padrões a rede converge para ciclos limites, exemplo o teste 131. Portanto, para a memória associativa desenvolvida neste trabalho foi evidenciado que conhecendo a distância de Hamming entre o vetor de entrada e cada padrão é possível prever a saída que a rede apresentará.

Segundo as diferentes abordagens para o cálculo da capacidade de armazenamento das memórias associativas, vistos na seção 2.5.5. e utilizando as expressões (13), (14) e (15), tem-se a quantidade de padrões indicados a serem armazenados em uma rede neural de dimensão $n=9$, apresentada na Tabela 9.

Tabela 9 - Capacidade de armazenamento da rede de Hopfield de dimensão $n = 9$

Dimensão	C^{Hopf}	C^{Max}	$C^{100\%}$
$n=9$	1,35	2,04	1,02

É possível verificar que para uma rede neural com dimensão $n=9$, é recomendado o armazenamento de um único padrão quando se espera 100% de

acertos na recuperação, como indica $C^{100\%} = 1,02$. Entretanto, para o caso correspondente ao armazenamento de dois padrões, como considerado neste trabalho, é esperada uma recuperação com poucos erros, já que $C^{\text{Hopf}} = 1,35$ e $C^{\text{Max}} = 2,04$. Esta análise está relacionada aos resultados apresentados na Tabela 7, relativos à quantidade de ocorrência de cada estado final, onde observou-se que a probabilidade de convergência da rede para cada um dos padrões é de aproximadamente 36,33% e a probabilidade de que a rede atinja ciclos limites, isto é, estados espúrios, é de 27,34%. Este resultado mostra que quando a precisão exigida para a recuperação não for um fator tão significativo, pode-se armazenar mais padrões na memória associativa, mas se a precisão na recuperação for o fator mais importante, então a quantidade de padrões armazenados deve ser mínima, sendo esta a grande fragilidade da rede de Hopfield aplicada como memória associativa.

A utilização de componentes para descrição da rede neural em VHDL tornou sua estrutura relativamente generalizada e de fácil adaptação para outras memórias associativas. Pois para gerar uma rede que reconhecerá diferentes padrões, basta modificar os vetores de peso inseridos nos componentes *Neuronio* e a quantidade de vezes que os componentes *Flip Flop* e *Comparador* serão executados, de acordo com a dimensão dos padrões a serem armazenados.

5 Conclusão

Neste trabalho foi descrita e sintetizada em *hardware*, utilizando FPGA, uma rede neural de Hopfield composta por nove neurônios, a qual apresenta dinâmica síncrona, e é aplicada como uma memória associativa, exemplo este apresentado por Silva, Spatti & Flauzino (2009). A rede neural sintetizada na FPGA reproduziu os resultados da rede simulada no MATLAB com 100% de precisão para os casos testados, esse resultado comprova que a representação binária ilustrada na Figura 27, com utilização de cinco bits para a parte fracionária, corresponde a uma boa representação numérica para a descrição da rede considerada, apesar do erro de 0,6% relacionado à conversão do número binário para decimal.

Foi verificado que é possível prever a convergência da memória associativa baseando-se na distância de Hamming entre o vetor de entrada e cada padrão, sendo que a saída da rede é direcionada para o padrão que possui menor distância. O aparecimento de estados espúrios, corresponde aos casos em que a distância de Hamming é igual para ambos os padrões. Este acontecimento também está relacionado à quantidade de padrões armazenados na rede, que foi superior ao limite recomendado, evidenciando a maior fragilidade da rede neural de Hopfield funcionando como memória associativa, que é a baixa capacidade de armazenamento de padrões.

A linguagem VHDL simplificou o processo de desenvolvimento da rede neural, pois sua descrição é relacionada ao comportamento que se deseja obter do circuito, possibilitando a utilização de diversos componentes que quando interligados executam as funções da RNA.

Sugestões para trabalhos futuros: otimização do código em VHDL com o objetivo de reduzir o número de elementos lógicos utilizados e generalização de sua estrutura para diferentes memórias associativas, descrição da mesma rede de Hopfield descrita neste trabalho mas com atualização assíncrona, implementação do exemplo proposto por Silva, Spatti e Flauzino (2009) acerca de um sistema de transmissão de imagens codificado por 45 bits (página 219), e síntese de outras topologias de redes neurais em FPGA.

6 Referências

- AMIT, D.J. **Modeling Brain Function: The World of Attractor Neural Networks**. Cambridge University Press, 1989.
- GODSE, A.;GODSE, D. **Fundamentals of HDL**. Technical Publications, 2009.
- HAYKIN, S. **Redes Neurais Princípios e Prática**, 2008.
- HEBB, D. O. **The Organization of Behavior**, 1949
- HOPFIELD, J.J. **Neurons with graded response have collective computational proprieties like those of two-state neurons**, 1984.
- JAIN A. K., MAO J. **Artificial Neural Networks: A tutorial**, 1966.
- KOVÁCS, L. Z. **Redes neurais Artificiais Fundamentos e Aplicações**, 2006
- MENEZES, C. C. **Redes Neurais**. Instituto de Informática, Universidade do Rio Grande do Sul, 2001.
- RAMON & CAJÁL. **Histologie du système nerveux de l' homme et des vertébrés**. Paris: Maloine, 1911
- SILVA, I. N., SPATTI, D. H., FLAUZINO, R. A. **Redes Neurais Artificiais para Engenharia e Ciências Aplicadas**, 2010.
- SILVA, N. M. J. **Redes Neurais Artificiais: Rede Hopfield e Redes Estocásticas**, 2002.
- SHEPERD, G. M., C. Koch. **Introduction to synaptic circuits**, 1990
- TERASIC. **Altera DE2-115 Development and Education Board**, 2013.
Disponível em: <http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=502&PartNo=4>
Acesso em: 05/10/2015
- TONINI A. M., COUTO B. R. G. M. **Ensinando Geometria Analítica com uso do MATLAB**. Departamento de Ciências Exatas e Tecnologia do Centro Universitário de Belo Horizonte / DECET - UniBH.
- VON ZUBEN, F. J., ATTUX, R. R. F. **Neurocomputação, Dinâmica Não-Linear e Redes Neurais de Hopfield**. Unicamp
- PERELMUTER, G.,CARRERA, E. V., VELLASCO, M., Pacheco, A. **Reconhecimento de Imagens Bidimensionais utilizando Redes Neurais Artificiais**. Anais do VIII SIBGRAPI, pp. 197-203, 1995

Apêndice I - Circuitos RTL gerados pelo software Quartus II

Figura 53 - Circuito RTL do *Multiplexador*

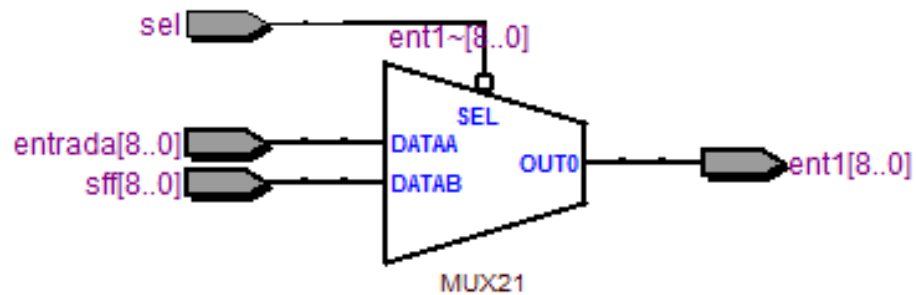


Figura 54 - Circuito RTL do *Neurônio N1*

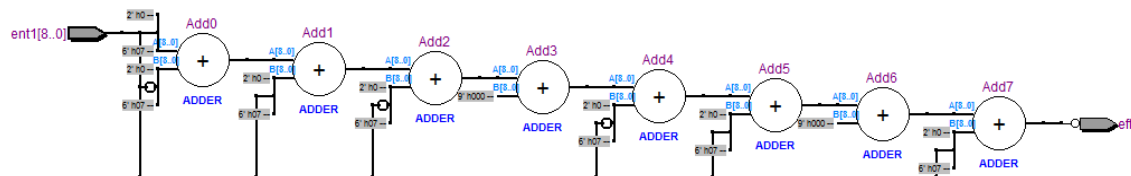


Figura 55 - Circuito RTL do *Flip Flop* tipo D

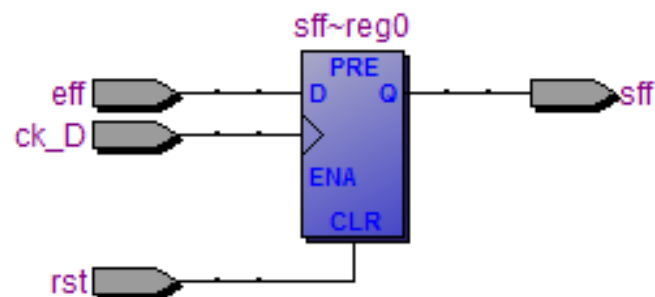


Figura 56 - Circuito RTL do *Comparador*



Figura 57 - Circuito RTL do *Contador*

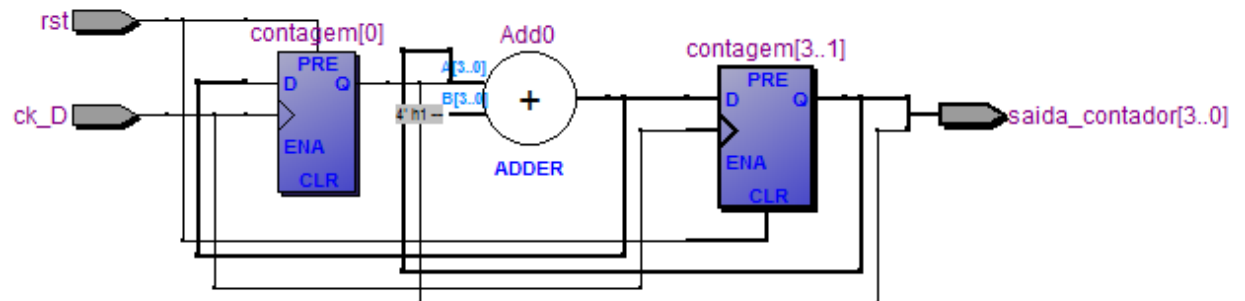


Figura 58 - Circuito RTL da *Maquina de estados*

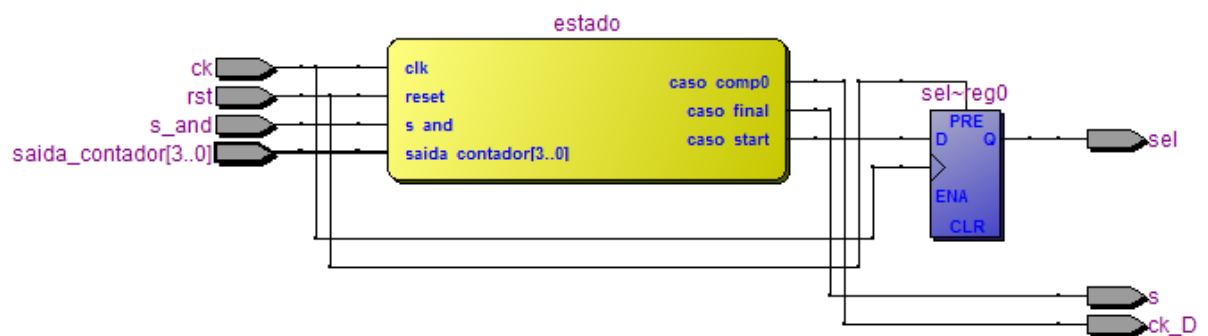
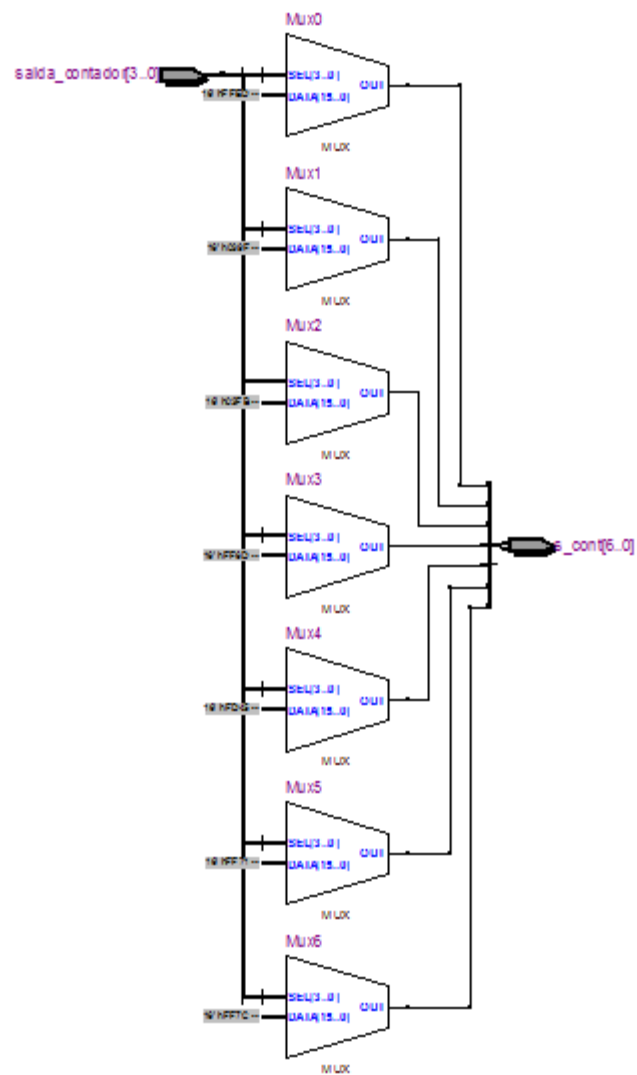


Figura 59 - Circuito RTL do *Conversor Binário - Decimal*



Apêndice II – Resultados obtidos no MATLAB

A Tabela 10 exibe a resposta gerada pela rede para todos os vetores de entrada existentes, apresentando os estados gerados nas iterações com n ímpar (v_{anterior}) e par (v_{atual}), bem como a distancia de Hamming entre a entrada e o padrão z^1 (Dist z1) e a entrada e o padrão z^2 (Dist z2), e o número n de iterações.

Tabela 10 - Resultados obtidos a partir da rede descrita no MATLAB

Nº do teste	Entrada		Saída em representação polarizada		Iterações	Distância de Hamming	
	Decimal	Polarizada	v_{anterior} (n ímpar)	v_{atual} (n par)	It n	Dist z1	Dist z2
Elementos corrompidos: 0 do padrão z1 e 8 do padrão z2							
1	186	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	1	0	16
Elementos corrompidos: 1 do padrão z1 e 7 do padrão z2							
2	58	-1-1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	2	14
3	154	-1+1-1-1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	2	14
4	178	-1+1-1+1+1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	2	14
5	184	-1+1-1+1+1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	2	14
6	187	-1+1-1+1+1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	2	14
7	190	-1+1-1+1+1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	2	14
8	250	-1+1+1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	2	14
9	442	+1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	2	14
Elementos corrompidos: 1 do padrão z1 e 9 do padrão z2							
10	170	-1+1-1+1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	2	18
Elementos corrompidos: 2 do padrão z1 e 6 do padrão z2							
11	26	-1-1-1-1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
12	50	-1-1-1+1+1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
13	56	-1-1-1+1+1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
14	59	-1-1-1+1+1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
15	62	-1-1-1+1+1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
16	122	-1-1+1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
17	146	-1+1-1-1+1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
18	152	-1+1-1-1+1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
19	155	-1+1-1-1+1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
20	158	-1+1-1-1+1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
21	176	-1+1-1+1+1-1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
22	179	-1+1-1+1+1-1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
23	182	-1+1-1+1+1-1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12

24	185	-1+1-1+1+1+1-1-1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
25	188	-1+1-1+1+1+1+1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
26	191	-1+1-1+1+1+1+1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
27	218	-1+1+1-1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
28	242	-1+1+1+1+1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
29	248	-1+1+1+1+1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
30	251	-1+1+1+1+1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
31	254	-1+1+1+1+1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
32	314	+1-1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
33	410	+1+1-1-1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
34	434	+1+1-1+1+1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
35	440	+1+1-1+1+1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
36	443	+1+1-1+1+1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
37	446	+1+1-1+1+1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
38	506	+1+1+1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	12
Elementos corrompidos: 2 do padrão z1 e 8 do padrão z2							
39	42	-1-1-1+1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	16
40	138	-1+1-1-1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	16
41	162	-1+1-1+1-1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	16
42	168	-1+1-1+1-1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	16
43	171	-1+1-1+1-1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	16
44	174	-1+1-1+1-1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	16
45	234	-1+1+1+1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	16
46	426	+1+1-1+1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	4	16
Elementos corrompidos: 3 do padrão z1 e 5 do padrão z2							
47	18	-1-1-1-1+1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
48	24	-1-1-1-1+1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
49	27	-1-1-1-1+1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
50	30	-1-1-1-1+1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
51	48	-1-1-1+1+1-1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
52	51	-1-1-1+1+1-1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
53	54	-1-1-1+1+1-1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
54	57	-1-1-1+1+1+1-1-1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
55	60	-1-1-1+1+1+1+1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
56	63	-1-1-1+1+1+1+1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
57	90	-1-1+1-1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
58	114	-1-1+1+1+1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
59	120	-1-1+1+1+1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
60	123	-1-1+1+1+1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
61	126	-1-1+1+1+1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
62	144	-1+1-1-1+1-1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
63	147	-1+1-1-1+1-1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
64	150	-1+1-1-1+1-1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
65	153	-1+1-1-1+1+1-1-1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
66	156	-1+1-1-1+1+1+1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10

67	159	-1+1-1-1+1+1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
68	177	-1+1-1+1+1-1-1-1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
69	180	-1+1-1+1+1-1+1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
70	183	-1+1-1+1+1-1+1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
71	189	-1+1-1+1+1+1+1-1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
72	210	-1+1+1-1+1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
73	216	-1+1+1-1+1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
74	219	-1+1+1-1+1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
75	222	-1+1+1-1+1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
76	240	-1+1+1+1+1-1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
77	243	-1+1+1+1+1-1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
78	246	-1+1+1+1+1-1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
79	249	-1+1+1+1+1+1-1-1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
80	252	-1+1+1+1+1+1+1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
81	255	-1+1+1+1+1+1+1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
82	282	+1-1-1-1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
83	306	+1-1-1+1+1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
84	312	+1-1-1+1+1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
85	315	+1-1-1+1+1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
86	318	+1-1-1+1+1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
87	378	+1-1+1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
88	402	+1+1-1-1+1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
89	408	+1+1-1-1+1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
90	411	+1+1-1-1+1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
91	414	+1+1-1-1+1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
92	432	+1+1-1+1+1-1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
93	435	+1+1-1+1+1-1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
94	438	+1+1-1+1+1-1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
95	441	+1+1-1+1+1+1-1-1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
96	444	+1+1-1+1+1+1+1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
97	447	+1+1-1+1+1+1+1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
98	474	+1+1+1-1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
99	498	+1+1+1+1+1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
100	504	+1+1+1+1+1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
101	507	+1+1+1+1+1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
102	510	+1+1+1+1+1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	10
Elementos corrompidos: 3 do padrão z1 e 7 do padrão z2							
103	10	-1-1-1-1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
104	34	-1-1-1+1-1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
105	40	-1-1-1+1-1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
106	43	-1-1-1+1-1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
107	46	-1-1-1+1-1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
108	106	-1-1+1+1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
109	130	-1+1-1-1-1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
110	136	-1+1-1-1-1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14

111	139	-1+1-1-1-1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
112	142	-1+1-1-1-1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
113	160	-1+1-1+1-1-1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
114	163	-1+1-1+1-1-1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
115	166	-1+1-1+1-1-1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
116	169	-1+1-1+1-1+1-1-1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
117	172	-1+1-1+1-1+1+1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
118	175	-1+1-1+1-1+1+1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
119	202	-1+1+1-1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
120	226	-1+1+1+1-1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
121	232	-1+1+1+1-1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
122	235	-1+1+1+1-1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
123	238	-1+1+1+1-1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
124	298	+1-1-1+1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
125	394	+1+1-1-1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
126	418	+1+1-1+1-1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
127	424	+1+1-1+1-1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
128	427	+1+1-1+1-1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
129	430	+1+1-1+1-1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
130	490	+1+1+1+1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	6	14
Elementos corrompidos: 4 do padrão z1 e 4 do padrão z2							
131	16	-1-1-1-1+1-1-1-1-1	+1+1+1+1+1+1+1+1+1	-1-1-1-1+1-1-1-1-1	1000	8	8
132	19	-1-1-1-1+1-1-1+1+1	+1+1+1+1+1+1+1-1-1	-1-1-1-1+1-1-1+1+1	1000	8	8
133	22	-1-1-1-1+1-1+1+1-1	+1+1+1+1+1+1-1-1+1	-1-1-1-1+1-1+1+1-1	1000	8	8
134	25	-1-1-1-1+1+1-1-1+1	+1+1+1+1+1-1+1+1-1	-1-1-1-1+1+1-1-1+1	1000	8	8
135	28	-1-1-1-1+1+1+1-1-1	+1+1+1+1+1-1-1+1+1	-1-1-1-1+1+1+1-1-1	1000	8	8
136	31	-1-1-1-1+1+1+1+1+1	+1+1+1+1+1-1-1-1-1	-1-1-1-1+1+1+1+1+1	1000	8	8
137	49	-1-1-1+1+1-1-1-1+1	+1+1+1-1+1+1+1+1-1	-1-1-1+1+1-1-1-1+1	1000	8	8
138	52	-1-1-1+1+1-1+1-1-1	+1+1+1-1+1+1-1+1+1	-1-1-1+1+1-1+1-1-1	1000	8	8
139	55	-1-1-1+1+1-1+1+1+1	+1+1+1-1+1+1-1-1-1	-1-1-1+1+1-1+1+1+1	1000	8	8
140	61	-1-1-1+1+1+1+1-1+1	+1+1+1-1+1-1-1+1-1	-1-1-1+1+1+1+1-1+1	1000	8	8
141	82	-1-1+1-1+1-1-1+1-1	+1+1-1+1+1+1+1-1+1	-1-1+1-1+1-1-1+1-1	1000	8	8
142	88	-1-1+1-1+1+1-1-1-1	+1+1-1+1+1-1+1+1+1	-1-1+1-1+1+1-1-1-1	1000	8	8
143	91	-1-1+1-1+1+1-1+1+1	+1+1-1+1+1-1+1-1-1	-1-1+1-1+1+1-1+1+1	1000	8	8
144	94	-1-1+1-1+1+1+1+1-1	+1+1-1+1+1-1-1-1+1	-1-1+1-1+1+1+1+1-1	1000	8	8
145	112	-1-1+1+1+1-1-1-1-1	+1+1-1-1+1+1+1+1+1	-1-1+1+1-1-1-1-1-1	1000	8	8
146	115	-1-1+1+1+1-1-1+1+1	+1+1-1-1+1+1+1-1-1	-1-1+1+1+1-1-1+1+1	1000	8	8
147	118	-1-1+1+1+1-1+1+1-1	+1+1-1-1+1+1-1-1+1	-1-1+1+1+1-1+1+1-1	1000	8	8
148	121	-1-1+1+1+1+1-1-1+1	+1+1-1-1+1-1+1+1-1	-1-1+1+1+1+1-1-1+1	1000	8	8
149	124	-1-1+1+1+1+1+1-1-1	+1+1-1-1+1-1-1+1+1	-1-1+1+1+1+1+1-1-1	1000	8	8
150	127	-1-1+1+1+1+1+1+1+1	+1+1-1-1+1-1-1-1-1	-1-1+1+1+1+1+1+1+1	1000	8	8
151	145	-1+1-1-1+1-1-1-1+1	+1-1+1+1+1+1+1+1-1	-1+1-1-1+1-1-1-1+1	1000	8	8
152	148	-1+1-1-1+1-1+1-1-1	+1-1+1+1+1+1-1+1+1	-1+1-1-1+1-1+1-1-1	1000	8	8
153	151	-1+1-1-1+1-1+1+1+1	+1-1+1+1+1+1-1-1-1	-1+1-1-1+1-1+1+1+1	1000	8	8
154	157	-1+1-1-1+1+1+1-1+1	+1-1+1+1+1+1-1+1-1	-1+1-1-1+1+1+1-1+1	1000	8	8

155	181	-1+1-1+1+1-1+1-1+1	+1-1+1-1+1+1-1+1-1	-1+1-1+1+1-1+1-1+1	1000	8	8
156	208	-1+1+1-1+1-1-1-1-1	+1-1-1+1+1+1+1+1+1	-1+1+1-1+1-1-1-1-1	1000	8	8
157	211	-1+1+1-1+1-1-1+1+1	+1-1-1+1+1+1+1-1-1	-1+1+1-1+1-1-1+1+1	1000	8	8
158	214	-1+1+1-1+1-1+1+1-1	+1-1-1+1+1+1-1-1+1	-1+1+1-1+1-1+1+1-1	1000	8	8
159	217	-1+1+1-1+1+1-1-1+1	+1-1-1+1+1-1+1+1-1	-1+1+1-1+1+1-1-1+1	1000	8	8
160	220	-1+1+1-1+1+1+1-1-1	+1-1-1+1+1-1-1+1+1	-1+1+1-1+1+1+1-1-1	1000	8	8
161	223	-1+1+1-1+1+1+1+1+1	+1-1-1+1+1-1-1-1-1	-1+1+1-1+1+1+1+1+1	1000	8	8
162	241	-1+1+1+1+1-1-1-1+1	+1-1-1-1+1+1+1+1-1	-1+1+1+1+1-1-1-1+1	1000	8	8
163	244	-1+1+1+1+1-1+1-1-1	+1-1-1-1+1+1-1+1+1	-1+1+1+1+1-1+1-1-1	1000	8	8
164	247	-1+1+1+1+1-1+1+1+1	+1-1-1-1+1+1-1-1-1	-1+1+1+1+1-1+1+1+1	1000	8	8
165	253	-1+1+1+1+1+1+1-1+1	+1-1-1-1+1-1-1+1-1	-1+1+1+1+1+1+1-1+1	1000	8	8
166	274	+1-1-1-1+1-1-1+1-1	-1+1+1+1+1+1+1-1+1	+1-1-1-1+1-1-1+1-1	1000	8	8
167	280	+1-1-1-1+1+1-1-1-1	-1+1+1+1+1-1+1+1+1	+1-1-1-1+1+1-1-1-1	1000	8	8
168	283	+1-1-1-1+1+1-1+1+1	-1+1+1+1+1-1+1-1-1	+1-1-1-1+1+1-1+1+1	1000	8	8
169	286	+1-1-1-1+1+1+1+1-1	-1+1+1+1+1-1-1-1+1	+1-1-1-1+1+1+1+1-1	1000	8	8
170	304	+1-1-1+1+1-1-1-1-1	-1+1+1-1+1+1+1+1+1	+1-1-1+1+1-1-1-1-1	1000	8	8
171	307	+1-1-1+1+1-1-1+1+1	-1+1+1-1+1+1+1-1-1	+1-1-1+1+1-1-1+1+1	1000	8	8
172	310	+1-1-1+1+1-1+1+1-1	-1+1+1-1+1+1-1-1+1	+1-1-1+1+1-1+1+1-1	1000	8	8
173	313	+1-1-1+1+1+1-1-1+1	-1+1+1-1+1-1+1+1-1	+1-1-1+1+1+1-1-1+1	1000	8	8
174	316	+1-1-1+1+1+1+1-1-1	-1+1+1-1+1-1-1+1+1	+1-1-1+1+1+1+1-1-1	1000	8	8
175	319	+1-1-1+1+1+1+1+1+1	-1+1+1-1+1-1-1-1-1	+1-1-1+1+1+1+1+1+1	1000	8	8
176	346	+1-1+1-1+1+1-1+1-1	-1+1-1+1+1-1+1-1+1	+1-1+1-1+1+1-1+1-1	1000	8	8
177	370	+1-1+1+1+1-1-1+1-1	-1+1-1-1+1+1+1-1+1	+1-1+1+1+1-1-1+1-1	1000	8	8
178	376	+1-1+1+1+1+1-1-1-1	-1+1-1-1+1-1+1+1+1	+1-1+1+1+1+1-1-1-1	1000	8	8
179	379	+1-1+1+1+1+1-1+1+1	-1+1-1-1+1-1+1-1-1	+1-1+1+1+1+1-1+1+1	1000	8	8
180	382	+1-1+1+1+1+1+1+1-1	-1+1-1-1+1-1-1-1+1	+1-1+1+1+1+1+1+1-1	1000	8	8
181	400	+1+1-1-1+1-1-1-1-1	-1-1+1+1+1+1+1+1+1	+1+1-1-1+1-1-1-1-1	1000	8	8
182	403	+1+1-1-1+1-1-1+1+1	-1-1+1+1+1+1+1-1-1	+1+1-1-1+1-1-1+1+1	1000	8	8
183	406	+1+1-1-1+1-1+1+1-1	-1-1+1+1+1+1-1-1+1	+1+1-1-1+1-1+1+1-1	1000	8	8
184	409	+1+1-1-1+1+1-1-1+1	-1-1+1+1+1-1+1+1-1	+1+1-1-1+1+1-1-1+1	1000	8	8
185	412	+1+1-1-1+1+1+1-1-1	-1-1+1+1+1-1-1+1+1	+1+1-1-1+1+1+1-1-1	1000	8	8
186	415	+1+1-1-1+1+1+1+1+1	-1-1+1+1+1-1-1-1-1	+1+1-1-1+1+1+1+1+1	1000	8	8
187	433	+1+1-1+1+1-1-1-1+1	-1-1+1-1+1+1+1+1-1	+1+1-1+1+1-1-1-1+1	1000	8	8
188	436	+1+1-1+1+1-1+1-1-1	-1-1+1-1+1+1-1+1+1	+1+1-1+1+1-1+1-1-1	1000	8	8
189	439	+1+1-1+1+1-1+1+1+1	-1-1+1-1+1+1-1-1-1	+1+1-1+1+1-1+1+1+1	1000	8	8
190	445	+1+1-1+1+1+1+1-1+1	-1-1+1-1+1-1-1+1+1	+1+1-1+1+1+1+1-1+1	1000	8	8
191	466	+1+1+1-1+1-1-1+1-1	-1-1-1+1+1+1+1-1+1	+1+1+1-1+1-1-1+1-1	1000	8	8
192	472	+1+1+1-1+1+1-1-1-1	-1-1-1+1+1-1+1+1+1	+1+1+1-1+1+1-1-1-1	1000	8	8
193	475	+1+1+1-1+1+1-1+1+1	-1-1-1+1+1-1+1-1-1	+1+1+1-1+1+1-1+1+1	1000	8	8
194	478	+1+1+1-1+1+1+1+1-1	-1-1-1+1+1-1-1-1+1	+1+1+1-1+1+1+1+1-1	1000	8	8
195	496	+1+1+1+1+1-1-1-1-1	-1-1-1-1+1+1+1+1+1	+1+1+1+1+1-1-1-1-1	1000	8	8
196	499	+1+1+1+1+1-1-1+1+1	-1-1-1-1+1+1+1-1-1	+1+1+1+1+1-1-1+1+1	1000	8	8
197	502	+1+1+1+1+1-1+1+1-1	-1-1-1-1+1+1-1-1+1	+1+1+1+1+1-1+1+1-1	1000	8	8
198	505	+1+1+1+1+1+1-1-1+1	-1-1-1-1+1-1+1+1-1	+1+1+1+1+1+1-1-1+1	1000	8	8
199	508	+1+1+1+1+1+1+1-1-1	-1-1-1-1+1-1-1+1+1	+1+1+1+1+1+1+1-1-1	1000	8	8

200	511	+1+1+1+1+1+1+1+1+1	-1-1-1-1+1-1-1-1-1	+1+1+1+1+1+1+1+1+1	1000	8	8
Elementos corrompidos: 4 do padrão z1 e 6 do padrão z2							
201	2	-1-1-1-1-1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
202	8	-1-1-1-1-1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
203	11	-1-1-1-1-1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
204	14	-1-1-1-1-1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
205	32	-1-1-1+1-1-1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
206	35	-1-1-1+1-1-1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
207	38	-1-1-1+1-1-1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
208	41	-1-1-1+1-1+1-1-1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
209	44	-1-1-1+1-1+1+1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
210	47	-1-1-1+1-1+1+1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
211	74	-1-1+1-1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
212	98	-1-1+1+1-1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
213	104	-1-1+1+1-1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
214	107	-1-1+1+1-1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
215	110	-1-1+1+1-1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
216	128	-1+1-1-1-1-1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
217	131	-1+1-1-1-1-1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
218	134	-1+1-1-1-1-1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
219	137	-1+1-1-1-1+1-1-1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
220	140	-1+1-1-1-1+1+1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
221	143	-1+1-1-1-1+1+1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
222	161	-1+1-1+1-1-1-1-1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
223	164	-1+1-1+1-1-1+1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
224	167	-1+1-1+1-1-1+1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
225	173	-1+1-1+1-1+1+1-1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
226	194	-1+1+1-1-1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
227	200	-1+1+1-1-1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
228	203	-1+1+1-1-1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
229	206	-1+1+1-1-1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
230	224	-1+1+1+1-1-1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
231	227	-1+1+1+1-1-1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
232	230	-1+1+1+1-1-1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
233	233	-1+1+1+1-1+1-1-1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
234	236	-1+1+1+1-1+1+1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
235	239	-1+1+1+1-1+1+1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
236	266	+1-1-1-1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
237	290	+1-1-1+1-1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
238	296	+1-1-1+1-1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
239	299	+1-1-1+1-1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
240	302	+1-1-1+1-1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
241	362	+1-1+1+1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
242	386	+1+1-1-1-1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
243	392	+1+1-1-1-1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12

244	395	+1+1-1-1-1+1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
245	398	+1+1-1-1-1+1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
246	416	+1+1-1+1-1-1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
247	419	+1+1-1+1-1-1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
248	422	+1+1-1+1-1-1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
249	425	+1+1-1+1-1+1-1-1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
250	428	+1+1-1+1-1+1+1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
251	431	+1+1-1+1-1+1+1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
252	458	+1+1+1-1-1+1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
253	482	+1+1+1+1-1-1-1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
254	488	+1+1+1+1-1+1-1-1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
255	491	+1+1+1+1-1+1-1+1+1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
256	494	+1+1+1+1-1+1+1+1-1	-1+1-1+1+1+1-1+1-1	-1+1-1+1+1+1-1+1-1	2	8	12
Elementos corrompidos: 5 do padrão z1 e 3 do padrão z2							
257	17	-1-1-1-1+1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
258	20	-1-1-1-1+1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
259	23	-1-1-1-1+1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
260	29	-1-1-1-1+1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
261	53	-1-1-1+1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
262	80	-1-1+1-1+1-1-1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
263	83	-1-1+1-1+1-1-1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
264	86	-1-1+1-1+1-1+1+1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
265	89	-1-1+1-1+1+1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
266	92	-1-1+1-1+1+1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
267	95	-1-1+1-1+1+1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
268	113	-1-1+1+1+1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
269	116	-1-1+1+1+1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
270	119	-1-1+1+1+1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
271	125	-1-1+1+1+1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
272	149	-1+1-1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
273	209	-1+1+1-1+1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
274	212	-1+1+1-1+1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
275	215	-1+1+1-1+1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
276	221	-1+1+1-1+1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
277	245	-1+1+1+1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
278	272	+1-1-1-1+1-1-1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
279	275	+1-1-1-1+1-1-1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
280	278	+1-1-1-1+1-1+1+1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
281	281	+1-1-1-1+1+1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
282	284	+1-1-1-1+1+1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
283	287	+1-1-1-1+1+1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
284	305	+1-1-1+1+1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
285	308	+1-1-1+1+1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
286	311	+1-1-1+1+1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
287	317	+1-1-1+1+1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6

288	338	+1-1+1-1+1-1-1+1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
289	344	+1-1+1-1+1+1-1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
290	347	+1-1+1-1+1+1-1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
291	350	+1-1+1-1+1+1+1+1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
292	368	+1-1+1+1+1-1-1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
293	371	+1-1+1+1+1-1-1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
294	374	+1-1+1+1+1-1+1+1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
295	377	+1-1+1+1+1+1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
296	380	+1-1+1+1+1+1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
297	383	+1-1+1+1+1+1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
298	401	+1+1-1-1+1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
299	404	+1+1-1-1+1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
300	407	+1+1-1-1+1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
301	413	+1+1-1-1+1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
302	437	+1+1-1+1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
303	464	+1+1+1-1+1-1-1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
304	467	+1+1+1-1+1-1-1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
305	470	+1+1+1-1+1-1+1+1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
306	473	+1+1+1-1+1+1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
307	476	+1+1+1-1+1+1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
308	479	+1+1+1-1+1+1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
309	497	+1+1+1+1+1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
310	500	+1+1+1+1+1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
311	503	+1+1+1+1+1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
312	509	+1+1+1+1+1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	10	6
Elementos corrompidos: 5 do padrão z1 e 5 do padrão z2							
313	0	-1-1-1-1-1-1-1-1-1	+1+1+1+1+1+1+1+1+1	-1-1-1-1+1-1-1-1-1	1000	10	10
314	3	-1-1-1-1-1-1-1+1+1	+1+1+1+1+1+1+1-1-1	-1-1-1-1+1-1-1+1+1	1000	10	10
315	6	-1-1-1-1-1-1+1+1-1	+1+1+1+1+1+1-1-1+1	-1-1-1-1+1-1+1+1-1	1000	10	10
316	9	-1-1-1-1-1+1-1-1+1	+1+1+1+1+1-1+1+1-1	-1-1-1-1+1+1-1-1+1	1000	10	10
317	12	-1-1-1-1-1+1+1-1-1	+1+1+1+1+1-1-1+1+1	-1-1-1-1+1+1+1-1-1	1000	10	10
318	15	-1-1-1-1-1+1+1+1+1	+1+1+1+1+1-1-1-1-1	-1-1-1-1+1+1+1+1+1	1000	10	10
319	33	-1-1-1+1-1-1-1-1+1	+1+1+1-1+1+1+1+1-1	-1-1-1+1+1-1-1-1+1	1000	10	10
320	36	-1-1-1+1-1-1+1-1-1	+1+1+1-1+1+1-1+1+1	-1-1-1+1+1-1+1-1-1	1000	10	10
321	39	-1-1-1+1-1-1+1+1+1	+1+1+1-1+1+1-1-1-1	-1-1-1+1+1-1+1+1+1	1000	10	10
322	45	-1-1-1+1-1+1+1-1+1	+1+1+1-1+1-1-1+1-1	-1-1-1+1+1+1+1-1+1	1000	10	10
323	66	-1-1+1-1-1-1-1+1-1	+1+1-1+1+1+1+1-1+1	-1-1+1-1+1-1-1+1-1	1000	10	10
324	72	-1-1+1-1-1+1-1-1-1	+1+1-1+1+1-1+1+1+1	-1-1+1-1+1+1-1-1-1	1000	10	10
325	75	-1-1+1-1-1+1-1+1+1	+1+1-1+1+1-1+1-1-1	-1-1+1-1+1+1-1+1+1	1000	10	10
326	78	-1-1+1-1-1+1+1+1-1	+1+1-1+1+1-1-1-1+1	-1-1+1-1+1+1+1+1-1	1000	10	10
327	96	-1-1+1+1-1-1-1-1-1	+1+1-1-1+1+1+1+1+1	-1-1+1+1-1-1-1-1-1	1000	10	10
328	99	-1-1+1+1-1-1-1+1+1	+1+1-1-1+1+1+1-1-1	-1-1+1+1+1-1-1+1+1	1000	10	10
329	102	-1-1+1+1-1-1+1+1-1	+1+1-1-1+1+1-1-1+1	-1-1+1+1+1-1+1+1-1	1000	10	10
330	105	-1-1+1+1-1+1-1-1+1	+1+1-1-1+1-1+1+1-1	-1-1+1+1+1+1-1-1+1	1000	10	10
331	108	-1-1+1+1-1+1+1-1-1	+1+1-1-1+1-1-1+1+1	-1-1+1+1+1+1+1-1-1	1000	10	10

332	111	-1-1+1+1-1+1+1+1	+1+1-1-1+1-1-1-1	-1-1+1+1+1+1+1+1	1000	10	10
333	129	-1+1-1-1-1-1-1+1	+1-1+1+1+1+1+1-1	-1+1-1-1+1-1-1+1	1000	10	10
334	132	-1+1-1-1-1-1+1-1	+1-1+1+1+1+1-1+1	-1+1-1-1+1-1+1-1	1000	10	10
335	135	-1+1-1-1-1-1+1+1	+1-1+1+1+1+1-1-1	-1+1-1-1+1-1+1+1	1000	10	10
336	141	-1+1-1-1-1+1+1-1	+1-1+1+1+1-1+1-1	-1+1-1-1+1+1+1-1	1000	10	10
337	165	-1+1-1+1-1-1+1-1	+1-1+1-1+1+1-1+1	-1+1-1+1+1-1+1-1	1000	10	10
338	192	-1+1+1-1-1-1-1-1	+1-1-1+1+1+1+1+1	-1+1+1-1+1-1-1-1	1000	10	10
339	195	-1+1+1-1-1-1+1+1	+1-1-1+1+1+1-1-1	-1+1+1-1+1-1+1+1	1000	10	10
340	198	-1+1+1-1-1-1+1-1	+1-1-1+1+1+1-1+1	-1+1+1-1+1-1+1-1	1000	10	10
341	201	-1+1+1-1-1+1-1-1	+1-1-1+1+1-1+1+1	-1+1+1-1+1+1-1+1	1000	10	10
342	204	-1+1+1-1-1+1+1-1	+1-1-1+1+1-1-1+1	-1+1+1-1+1+1+1-1	1000	10	10
343	207	-1+1+1-1-1+1+1+1	+1-1-1+1+1-1-1-1	-1+1+1-1+1+1+1+1	1000	10	10
344	225	-1+1+1+1-1-1-1+1	+1-1-1-1+1+1+1-1	-1+1+1+1-1-1-1+1	1000	10	10
345	228	-1+1+1+1-1-1+1-1	+1-1-1-1+1+1-1+1	-1+1+1+1-1+1-1-1	1000	10	10
346	231	-1+1+1+1-1-1+1+1	+1-1-1-1+1+1-1-1	-1+1+1+1-1+1+1+1	1000	10	10
347	237	-1+1+1+1-1+1-1+1	+1-1-1-1+1-1-1+1	-1+1+1+1+1+1+1-1	1000	10	10
348	258	+1-1-1-1-1-1+1-1	-1+1+1+1+1+1-1+1	+1-1-1-1+1-1+1-1	1000	10	10
349	264	+1-1-1-1-1+1-1-1	-1+1+1+1+1-1+1+1	+1-1-1-1+1+1-1-1	1000	10	10
350	267	+1-1-1-1-1+1-1+1	-1+1+1+1+1-1+1-1	+1-1-1-1+1+1-1+1	1000	10	10
351	270	+1-1-1-1-1+1+1-1	-1+1+1+1+1-1-1+1	+1-1-1-1+1+1+1-1	1000	10	10
352	288	+1-1-1+1-1-1-1-1	-1+1+1-1+1+1+1+1	+1-1-1+1+1-1-1-1	1000	10	10
353	291	+1-1-1+1-1-1+1+1	-1+1+1-1+1+1-1-1	+1-1-1+1+1-1-1+1	1000	10	10
354	294	+1-1-1+1-1-1+1-1	-1+1+1-1+1+1-1+1	+1-1-1+1+1-1+1-1	1000	10	10
355	297	+1-1-1+1-1+1-1-1	-1+1+1-1+1+1+1-1	+1-1-1+1+1+1-1+1	1000	10	10
356	300	+1-1-1+1-1+1+1-1	-1+1+1-1+1-1-1+1	+1-1-1+1+1+1+1-1	1000	10	10
357	303	+1-1-1+1-1+1+1+1	-1+1+1-1+1-1-1-1	+1-1-1+1+1+1+1+1	1000	10	10
358	330	+1-1+1-1-1+1+1-1	-1+1-1+1+1-1+1+1	+1-1+1-1+1+1-1+1	1000	10	10
359	354	+1-1+1+1-1-1+1-1	-1+1-1-1+1+1+1+1	+1-1+1+1+1-1+1-1	1000	10	10
360	360	+1-1+1+1-1+1-1-1	-1+1-1-1+1-1+1+1	+1-1+1+1+1+1-1-1	1000	10	10
361	363	+1-1+1+1-1+1-1+1	-1+1-1-1+1-1+1-1	+1-1+1+1+1+1-1+1	1000	10	10
362	366	+1-1+1+1-1+1+1-1	-1+1-1-1+1-1-1+1	+1-1+1+1+1+1+1-1	1000	10	10
363	384	+1+1-1-1-1-1-1-1	-1-1+1+1+1+1+1+1	+1+1-1-1+1-1-1-1	1000	10	10
364	387	+1+1-1-1-1-1+1+1	-1-1+1+1+1+1-1-1	+1+1-1-1+1-1-1+1	1000	10	10
365	390	+1+1-1-1-1-1+1-1	-1-1+1+1+1+1-1+1	+1+1-1-1+1-1+1-1	1000	10	10
366	393	+1+1-1-1-1+1-1+1	-1-1+1+1+1-1+1+1	+1+1-1-1+1+1-1+1	1000	10	10
367	396	+1+1-1-1-1+1+1-1	-1-1+1+1+1-1+1+1	+1+1-1-1+1+1+1-1	1000	10	10
368	399	+1+1-1-1-1+1+1+1	-1-1+1+1+1-1-1-1	+1+1-1-1+1+1+1+1	1000	10	10
369	417	+1+1-1+1-1-1-1+1	-1-1+1-1+1+1+1-1	+1+1-1+1+1-1-1+1	1000	10	10
370	420	+1+1-1+1-1-1+1-1	-1-1+1-1+1+1-1+1	+1+1-1+1+1-1+1-1	1000	10	10
371	423	+1+1-1+1-1-1+1+1	-1-1+1-1+1+1-1-1	+1+1-1+1+1-1+1+1	1000	10	10
372	429	+1+1-1+1-1+1+1-1	-1-1+1-1+1-1-1+1	+1+1-1+1+1+1+1-1	1000	10	10
373	450	+1+1+1-1-1-1+1-1	-1-1-1+1+1+1+1+1	+1+1+1-1+1-1-1+1	1000	10	10
374	456	+1+1+1-1-1+1-1-1	-1-1-1+1+1-1+1+1	+1+1+1-1+1+1-1-1	1000	10	10
375	459	+1+1+1-1-1+1+1+1	-1-1-1+1+1-1+1-1	+1+1+1-1+1+1-1+1	1000	10	10
376	462	+1+1+1-1-1+1+1-1	-1-1-1+1+1-1-1+1	+1+1+1-1+1+1+1-1	1000	10	10

377	480	+1+1+1+1-1-1-1-1-1	-1-1-1-1+1+1+1+1+1	+1+1+1+1+1-1-1-1-1	1000	10	10
378	483	+1+1+1+1-1-1-1+1+1	-1-1-1-1+1+1+1-1-1	+1+1+1+1+1-1-1+1+1	1000	10	10
379	486	+1+1+1+1-1-1+1+1-1	-1-1-1-1+1+1-1-1+1	+1+1+1+1+1-1+1+1-1	1000	10	10
380	489	+1+1+1+1-1+1-1-1+1	-1-1-1-1+1-1+1+1-1	+1+1+1+1+1+1-1-1+1	1000	10	10
381	492	+1+1+1+1-1+1+1-1-1	-1-1-1-1+1-1-1+1+1	+1+1+1+1+1+1+1-1-1	1000	10	10
382	495	+1+1+1+1-1+1+1+1+1	-1-1-1-1+1-1-1-1-1	+1+1+1+1+1+1+1+1+1	1000	10	10
Elementos corrompidos: 6 do padrão z1 e 2 do padrão z2							
383	21	-1-1-1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
384	81	-1-1+1-1+1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
385	84	-1-1+1-1+1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
386	87	-1-1+1-1+1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
387	93	-1-1+1-1+1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
388	117	-1-1+1+1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
389	213	-1+1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
390	273	+1-1-1-1+1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
391	276	+1-1-1-1+1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
392	279	+1-1-1-1+1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
393	285	+1-1-1-1+1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
394	309	+1-1-1+1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
395	336	+1-1+1-1+1-1-1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
396	339	+1-1+1-1+1-1-1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
397	342	+1-1+1-1+1-1+1+1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
398	345	+1-1+1-1+1+1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
399	348	+1-1+1-1+1+1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
400	351	+1-1+1-1+1+1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
401	369	+1-1+1+1+1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
402	372	+1-1+1+1+1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
403	375	+1-1+1+1+1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
404	381	+1-1+1+1+1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
405	405	+1+1-1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
406	465	+1+1+1-1+1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
407	468	+1+1+1-1+1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
408	471	+1+1+1-1+1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
409	477	+1+1+1-1+1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
410	501	+1+1+1+1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	4
Elementos corrompidos: 6 do padrão z1 e 4 do padrão z2							
411	1	-1-1-1-1-1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
412	4	-1-1-1-1-1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
413	7	-1-1-1-1-1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
414	13	-1-1-1-1-1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
415	37	-1-1-1+1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
416	64	-1-1+1-1-1-1-1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
417	67	-1-1+1-1-1-1-1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
418	70	-1-1+1-1-1-1+1+1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
419	73	-1-1+1-1-1+1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8

420	76	-1-1+1-1-1+1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
421	79	-1-1+1-1-1+1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
422	97	-1-1+1+1-1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
423	100	-1-1+1+1-1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
424	103	-1-1+1+1-1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
425	109	-1-1+1+1-1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
426	133	-1+1-1-1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
427	193	-1+1+1-1-1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
428	196	-1+1+1-1-1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
429	199	-1+1+1-1-1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
430	205	-1+1+1-1-1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
431	229	-1+1+1+1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
432	256	+1-1-1-1-1-1-1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
433	259	+1-1-1-1-1-1-1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
434	262	+1-1-1-1-1-1+1+1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
435	265	+1-1-1-1-1+1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
436	268	+1-1-1-1-1+1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
437	271	+1-1-1-1-1+1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
438	289	+1-1-1+1-1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
439	292	+1-1-1+1-1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
440	295	+1-1-1+1-1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
441	301	+1-1-1+1-1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
442	322	+1-1+1-1-1-1-1+1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
443	328	+1-1+1-1-1+1-1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
444	331	+1-1+1-1-1+1-1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
445	334	+1-1+1-1-1+1+1+1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
446	352	+1-1+1+1-1-1-1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
447	355	+1-1+1+1-1-1-1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
448	358	+1-1+1+1-1-1+1+1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
449	361	+1-1+1+1-1+1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
450	364	+1-1+1+1-1+1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
451	367	+1-1+1+1-1+1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
452	385	+1+1-1-1-1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
453	388	+1+1-1-1-1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
454	391	+1+1-1-1-1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
455	397	+1+1-1-1-1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
456	421	+1+1-1+1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
457	448	+1+1+1-1-1-1-1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
458	451	+1+1+1-1-1-1-1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
459	454	+1+1+1-1-1-1+1+1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
460	457	+1+1+1-1-1+1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
461	460	+1+1+1-1-1+1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
462	463	+1+1+1-1-1+1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
463	481	+1+1+1+1-1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
464	484	+1+1+1+1-1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8

465	487	+1+1+1+1-1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
466	493	+1+1+1+1-1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	12	8
Elementos corrompidos: 7 do padrão z1 e 1 do padrão z2							
467	85	-1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	2
468	277	+1-1-1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	2
469	337	+1-1+1-1+1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	2
470	340	+1-1+1-1+1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	2
471	343	+1-1+1-1+1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	2
472	349	+1-1+1-1+1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	2
473	373	+1-1+1+1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	2
474	469	+1+1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	2
Elementos corrompidos: 7 do padrão z1 e 3 do padrão z2							
475	5	-1-1-1-1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
476	65	-1-1+1-1-1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
477	68	-1-1+1-1-1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
478	71	-1-1+1-1-1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
479	77	-1-1+1-1-1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
480	101	-1-1+1+1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
481	197	-1+1+1-1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
482	257	+1-1-1-1-1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
483	260	+1-1-1-1-1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
484	263	+1-1-1-1-1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
485	269	+1-1-1-1-1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
486	293	+1-1-1+1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
487	320	+1-1+1-1-1-1-1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
488	323	+1-1+1-1-1-1-1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
489	326	+1-1+1-1-1-1+1+1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
490	329	+1-1+1-1-1+1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
491	332	+1-1+1-1-1+1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
492	335	+1-1+1-1-1+1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
493	353	+1-1+1+1-1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
494	356	+1-1+1+1-1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
495	359	+1-1+1+1-1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
496	365	+1-1+1+1-1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
497	389	+1+1-1-1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
498	449	+1+1+1-1-1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
499	452	+1+1+1-1-1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
500	455	+1+1+1-1-1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
501	461	+1+1+1-1-1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
502	485	+1+1+1+1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	14	6
Elementos corrompidos: 8 do padrão z1 e 2 do padrão z2							
503	69	-1-1+1-1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	16	4
504	261	+1-1-1-1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	16	4
505	321	+1-1+1-1-1-1-1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	16	4
506	324	+1-1+1-1-1-1+1-1-1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	16	4

507	327	+1-1+1-1-1+1+1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	16	4
508	333	+1-1+1-1-1+1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	16	4
509	357	+1-1+1+1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	16	4
510	453	+1+1+1-1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	16	4
Elementos corrompidos: 8 do padrão z1 e 0 do padrão z2							
511	341	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	1	16	0
Elementos corrompidos: 9 do padrão z1 e 1 do padrão z2							
512	325	+1-1+1-1-1-1+1-1+1	+1-1+1-1+1-1+1-1+1	+1-1+1-1+1-1+1-1+1	2	18	2