

**Hamilton Luiz de Souza**

**GRAF CET COMO FERRAMENTA NO  
DESENVOLVIMENTO DE TECNOLOGIA  
ASSISTIVA**

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para a obtenção do Título de Mestre em Engenharia Elétrica.

DEDALUS - Acervo - EESC



31100050058

**Orientadora: Profa Dra Annie France Frère Slaets**



São Carlos  
2004

Curr. 5807  
Tncwi 1 CD ROM  
Tombo T283/04  
Sysno 1411261

Ficha catalográfica preparada pela Seção de Tratamento  
da Informação do Serviço de Biblioteca - EESC/USP

s729g

Souza, Hamilton Luiz de  
GRAF CET como ferramenta no desenvolvimento de  
tecnologia assistiva / Hamilton Luiz de Souza. -- São  
Carlos, 2004.

Dissertação (Mestrado) -- Escola de Engenharia de São  
Carlos-Universidade de São Paulo, 2004.  
Área: Engenharia Elétrica.  
Orientador: Profa. Dra. Annie France Frère Slaets.

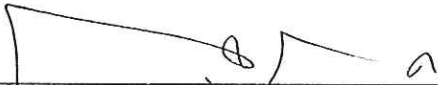
1. Tecnologia Assistiva. 2. GRAFCET. 3. FPGA.  
I. Título.


**FOLHA DE JULGAMENTO**

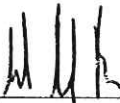
Candidato: Engenheiro HAMILTON LUIZ DE SOUZA


Dissertação defendida e julgada em 01-10-2004 perante a Comissão Julgadora:

  
\_\_\_\_\_  
Prof. Titular **ANNIE FRANCE FRÈRE SLAETS (Orientadora)**  
(Escola de Engenharia de São Carlos/USP) aprovado

  
\_\_\_\_\_  
Prof. Titular **MOACYR MARTUCCI JUNIOR**  
(Escola Politécnica/USP) Aprovado

  
\_\_\_\_\_  
Prof. Titular **ALBERTO CLIQUET JÚNIOR**  
(Escola de Engenharia de São Carlos/USP) Aprovado

  
\_\_\_\_\_  
Prof. Associado **MURILO ARAUJO ROMERO**  
Coordenador do Programa de Pós-Graduação  
em Engenharia Elétrica

  
\_\_\_\_\_  
Prof. Associado **MARCIO ROBERTO SILVA CORRÊA**  
Vice-Presidente da Comissão de Pós-Graduação, em Exercício

Dedico este trabalho aos meus pais, meu alicerce, e a minha companheira e cúmplice Vera Lucia, pilar central da nossa edificação chamada FAMÍLIA.

### **A excelência da união fraternal**

*(Cântico de romagem de David)*

**1** Oh! Como é bom e agradável viverem unidos os irmãos! **2** É como o óleo precioso sobre a cabeça, o qual desce para a barba, a barba de Arão, e desce para a gola de suas vestes. **3** É como o orvalho do Hermom, que desce sobre os montes de Sião. Ali, ordena o SENHOR a sua bênção e a vida para sempre.

Salmos 133 1-3

## Agradecimentos

Os meus mais sinceros agradecimentos à Dra. Annie France, minha orientadora até o final deste trabalho, pela sua postura profissional de educadora, e pela acolhida quando precisei.

A minha mulher, que mais do que ficar ao lado esteve muitas vezes me empurrando, com sua juventude, perseverança e carinho, para que este trabalho não esmaecesse.

Aos meus filhos, Victor José e Thiago Luiz, aqueles a quem procuro ser exemplo de conduta, exemplo este que tive de meus anteriores.

Ao meu amigo Breno Ortega, a vida ainda tem muito a te ensinar, obrigado por permitir te acompanhar.

Especiais agradecimentos devem ser dados aos amigos Dib Karan, Aziz Kalaf, pois destes retirei exemplos que aplico em minha vida acadêmica.

Luiz Adriano, obrigado, quanta coisa passamos nestes últimos tempos, quantos quilômetros juntos (quatro voltas completas ao redor do planeta Terra), quantos problemas, alegrias e lágrimas compartilhadas, é isso aí meu amigo.

Amigo Julio Tanomaru agradeço pelo exemplo de coragem, abraços aos ninjinhas.

Obrigado ao meu padrinho e irmão Milton Leo, aquele que me permitiu ver a Luz.

D<sup>a</sup> Otília, obrigado por me compreender apenas com o olhar.

A meu pai, pelo caráter ilibado e pela luta em educar e dar educação a seus filhos.

Vó Nair, Sr. Jorge, Tio Valdir e Jessé, saudades.

Mãe, sua benção.

## RESUMO

Souza, H. L. (2004). GRAFCET como ferramenta no desenvolvimento de tecnologia assistiva. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2004.

A escassez de equipamentos e soluções à disposição das pessoas portadoras de necessidades especiais é evidente e, na maioria dos casos, não está entre as prioridades primárias dos governos, tanto nos países desenvolvidos como, e principalmente, nos subdesenvolvidos. O desenvolvimento de dispositivos a preços acessíveis é, desta forma, uma carência real. O alto custo de fabricação de dispositivos assistivos é oriundo da necessidade de se construí-los de forma “personalizada” o que geralmente utiliza o estado da arte de determinada tecnologia. Aglutinando então, tecnologias já difundidas com novas abordagens e ferramentas, foi implementado um dispositivo assistivo com um índice de “personalização” extremamente baixo e com custo acessível. O método GRAFCET como ferramenta de desenvolvimento, aliado a tecnologia FPGA, nas fases de prototipação e produção, mostram-se eficazes e de fácil aplicabilidade. Para essa finalidade foi desenvolvida uma ferramenta de conversão GRAFCET → Circuito Digital que facilita e torna possível não só na elaboração do esquema GRAFCET, mas principalmente possibilita antever semelhança com outras aplicações que não são facilmente visualizadas num primeiro momento, permitindo a sua simplificação. O dispositivo concebido, mostrou que, com pequenas modificações, é possível desenvolvê-los de forma que atenda não apenas um, mas vários portadores de necessidades especiais, viabilizando assim a diminuição dos custos de projeto, desenvolvimento e construção dos equipamentos que venham a atender estas pessoas, permitindo assim melhora significativa em sua qualidade de vida.

Palavras chave: tecnologia assistiva, grafcet, fpga

## ABSTRACT

Souza, H. L. (2004). GRAFCET as a tool to develop assistive technology. Dissertation (Master) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos – SP, Brazil, 2004

In all countries, particularly in underdeveloped ones, there are few equipment and solutions available to people with physical disabilities, whose needs hardly rank among the government's top priorities. Therefore, there is a real need to develop low cost equipment to people with disabilities. The high cost of production of assistive devices results from the necessity of building them in a customized way, generally employing some state-of-the-art technology. This work proposes using well known technologies as tools, in such a way to allow the design of assistive devices with a low degree of customization and, consequently, at low cost. Using GRAFCET as a development tool and FPGA technology have shown to be an effective and easily applicable approach during the phases of prototyping and production. We developed a tool that converts GRAFCET designs into a digital circuit. The tool makes it easier to develop GRAFCET schemes and highlights the similarities among different applications, which otherwise could pass unnoticed, resulting in simplified designs. With small modifications, the proposed device can meet the needs of several people with disabilities, resulting in lower design, development and building costs of equipment that can significantly improve their quality of life.

Keywords: assistive technology, GRAFCET, FPGA



## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>1</b>
1.1- CONSIDERAÇÕES GERAIS .....	1
1.2- OBJETIVOS .....	5
1.3- ORGANIZAÇÃO DA DISSERTAÇÃO .....	6
<b>2. FUNDAMENTOS DA TECNOLOGIA ASSISTIVA .....</b>	<b>8</b>
2.1- INTRODUÇÃO .....	8
2.2- DISPOSITIVOS PARA AUDIÇÃO ASSISTIVA.....	14
2.3- DISPOSITIVOS PARA VISÃO ASSISTIVA .....	17
2.4- DISPOSITIVOS PARA INABILIDADES MOTORAS.....	18
2.5- OS DESAFIOS DA ENGENHARIA DE REABILITAÇÃO .....	20
<b>3. DISPOSITIVOS PARA AUXÍLIO A VIDA DIÁRIA.....</b>	<b>22</b>
3.1 – INTRODUÇÃO.....	22
3.2 - ESTAÇÃO DE TRABALHO “MASTER” .....	22
<b>4. TEORIA ASSOCIADA AO PROJETO.....</b>	<b>28</b>
4.1- MÉTODO GRAFCET .....	28
4.1.1- Etapas .....	29
4.1.2- Ações Associadas às Etapas .....	31
4.1.3 – Transições.....	33
4.1.4 – Ligações Orientadas .....	34
4.1.5- Regras de Evolução.....	35
4.1.5.1- Situação Inicial .....	35
4.1.5.2- Transposição de uma transição.....	35
4.1.5.3- Evolução das etapas ativas .....	35
4.1.5.4- Estruturas de uso freqüente .....	35
4.1.5.5- Evoluções simultâneas .....	37
4.1.6- Principais Estruturas .....	38
4.2- DISPOSITIVO LÓGICO PROGRAMÁVEL FPGA.....	40
4.2.1- Introdução .....	40

4.2.2- FPGA .....	43
4.2.2.1- Definição de Blocos Lógicos.....	43
4.2.2.2 – Definição de Roteamento .....	44
4.2.2.3- Definição de Bloco de E/S .....	45
4.2.3 – Vantagens da Utilização do FPGA.....	45
<b>4.3- LINGUAGEM DE DESCRIÇÃO DE HARDWARE (VHDL) .....</b>	<b>47</b>
4.3.1- Vantagens e Desvantagens do VHDL.....	48
<b>5. METODOLOGIA.....</b>	<b>49</b>
5.1- INTRODUÇÃO .....	49
5.2- CONCEPÇÃO DO DISPOSITIVO ASSISTIVO .....	50
5.2.1- Conversor GRAFCET → Circuito Digital .....	51
<b>6. RESULTADOS .....</b>	<b>58</b>
<b>7. CONCLUSÕES .....</b>	<b>75</b>
<b>Referências .....</b>	<b>77</b>
<b>Apêndice 1 – Listagem dos programas do Sistema GRAFCET → Circuito Digital .....</b>	<b>80</b>
<b>Apêndice 2 – Listagem do programa escrito em VHDL para gravação do FPGA.....</b>	<b>122</b>

## ÍNDICE DE FIGURAS

Figura 1.1 – Disposição Tradicional _____	4
Figura 1.2 – Dispositivo com Autômato Genérico para Acionamento _____	5
Figura 2.1 – Anatomia do Ouvido com o Diapositivo Implantado (ANONYMOUS, 1988) _____	16
Figura 2.2 – Exemplo de Dispositivo para Implantação em Pacientes com Surdez Profunda (ANONYMOUS, 1998) _____	16
Figura 2.3 – Visão do Eletrodo que será Implantado (ANONYMOUS, 1988) _____	17
Figura 2.4 – Cadeira de Rodas Tradicional (WWW.FREEDOM.IND.BR) _____	19
Figura 2.5 – Cadeira de Rodas Motorizada, Controle na Alavanca do Braço (WWW.FREEDOM.IND.BR) _____	20
Figura 3.1 – Visão Geral da Estação de Trabalho “MASTER RAID” (BUSNEL, 1998) _____	24
Figura 3.2 – Janela da Interface do Computador (BUSNEL, 1998) _____	25
Figura 4.1 – Representação de uma Etapa _____	30
Figura 4.2 - Etapa Ativa _____	30
Figura 4.3 – Representação da Etapa Inicial _____	31
Figura 4.4 – Ação Associada à Etapa _____	31
Figura 4.5 – Representação de várias Ações Associadas a uma Etapa _____	32
Figura 4.6 – Definição da ação associada à etapa _____	32
Figura 4.7 – Representação de condições associadas às transições _____	33
Figura 4.8 – Sentido das ligações orientadas _____	34
Figura 4.9 – Divergência em OU _____	36
Figura 4.10 – Convergência em OU _____	36
Figura 4.11 – Divergência em Y _____	37
Figura 4.12 – Convergência em E _____	37
Figura 4.13 – Seqüências paralelas (simultâneas) _____	38
Figura 4.14 – Seqüência Alternativa (Seletiva) _____	40
Figura 4.15 – Reutilização de uma seqüência _____	40
Figura 4.16 – Estrutura de uma FPGA (TERROSO, 1998) _____	43
Figura 4.17 – Roteamento de uma FPGA (TERROSO, 1998) _____	45
Figura 4.18 – Representação de um Bloco de E/s da família XC3000 da XILINX (TERROSO, 1998) _____	46
Figura 5.1 – Diagrama de blocos de entrada e saída de um dispositivo assistivo _____	50
Figura 5.2 – Tela de Abertura Sistema GRAFCET → Circuito Digital _____	52
Figura 5.3 – Tela de especificação de “AÇÃO” _____	53
Figura 5.4 – Estado especificado _____	53
Figura 5.5 – Tela de especificação de TRANSIÇÃO _____	54
Figura 5.6 – Tela de especificação da “TRANSIÇÃO” _____	55
Figura 5.7 – Transição especificada e posicionada. Componentes disponíveis. _____	55
Figura 5.8 – Blocos básicos do Sistema Grafcet→Circuito Digital: (a)- Memória; (b)- Porta E; (c)- Porta OU _____	56
Figura 6.1- Esquema GRAFCET desenhado em ferramenta gráfica _____	59
Figura 6.2- Esquema GRAFCET da Máquina Fotográfica com controle paralelo _____	60

Figura 6.3- Esquema Digital equivalente ao Esquema GRAFCET (Máquina Fotográfica com controle paralelo)	61
Figura 6.4- Circuito Digital simulado no Eletronics Workbench Pro	62
Figura 6.5- Circuito elétrico desenhado com a ferramenta Xilinx ECS	64
Figura 6.6- Dispositivo simulador	65
Figura 6.7- Placas AEE 1201 e AEE 1301, fonte controladora e placa de prototipação, respectivamente	66
Figura 6.8- Chave "START" e LED "1" ativos indicando início do processo	66
Figura 6.9- Chaves "A" e "B" ativas e LED "2" acesso, indicando que o processo "abrir lente" foi ativado	67
Fig. 6.10- Seqüência indicando que a lente está aberta	68
Figura 6.11- Estado indicando estado de espera do dispositivo	69
Figura 6.12- Chaves "D" e "X" ativadas	69
Figura 6.13- Estado indicando movimentos simultâneos	70
Figura 6.14- Chave "G" não acionada retorna ao estado de espera de movimento	70
Fig. 6.15- Chave "G" acionada, disparo do obturador e armazenagem da imagem	71
Figura. 6.16- Impressão da foto	71
Figura 6.17 – Estado inicial do processo	72
Figura 6.18- GRAFCET par controle de Cadeira de Rodas Motorizada	74

## 1. INTRODUÇÃO

### 1.1- CONSIDERAÇÕES GERAIS

As leis governamentais, da grande maioria dos países, estipulam punições a qualquer tipo de discriminação que um cidadão possa sofrer. Por outro lado, esses mesmos governos nem sempre provêem de forma eficaz o acesso do indivíduo portador de deficiência física, às mais básicas formas de interação com a sociedade. Acrescente-se a isso, o número de pessoas portadoras de restrições que em todo o mundo cresce. As melhorias nos tratamentos médicos, e as novas tecnologias aplicadas principalmente nos automóveis, permitem salvar mais vidas humanas que no passado, mesmo que a essas custe o ônus de possivelmente ter que conviver com uma restrição física (BUSNEL 1999).

Conforme apresentado por Yamamoto 1995, na história da humanidade as pessoas deficientes eram “castigadas” pelos membros comuns da sociedade – “homens comuns” - por motivos sócio-culturais. Nas civilizações mais antigas, esses indivíduos eram eliminados ou eram aceitos com algum tipo de restrição. Entretanto após a Segunda Grande Guerra os combatentes voltaram mutilados e muitos permaneceram deficientes em definitivo. A sociedade, nesta época, passou a sentir a necessidade de melhor tratar e atender essas pessoas, consideradas como “bons rapazes”, com o propósito de recuperá-las à convivência familiar e à comunidade de origem.

Através da política de concepção democrática, o Estado tornou-se responsável por garantir o direito de equidade de oportunidade a pessoa portadora de restrição em todos os aspectos, a fim de promover a integração e a participação no meio social comum. Mas o grau de dependência e autonomia dos portadores de deficiência varia de país para país, principalmente graças à maior difusão dos equipamentos de apoio e à utilização de tecnologias assistivas de ponta incorporadas a esses equipamentos.

Na área da reabilitação física, é usual a fabricação de instrumentos ou a utilização de ferramentas e maquinários para compensar ou substituir a função humana quando as técnicas reabilitadoras não são capazes de restabelecer a função em sua totalidade (MELLO, 1999). O restabelecimento das funções humanas pelo uso de instrumentos é um assunto de preocupação não só dos profissionais de reabilitação (médicos, fisioterapeutas e terapeutas ocupacionais), como também de engenheiros, arquitetos, biomédicos, desenhistas profissionais e profissionais de informática (Trombly), mas da comunidade como um todo.

De acordo com (ROCHA, 1997), um dos princípios da tecnologia assistiva é de acomodar o maior número possível de diferenças individuais, não construindo produtos ou serviços específicos para grupos de pessoas e pensar em produtos e serviços como sistemas que possam ter peças intercambiáveis.

Apesar do conhecimento e desenvolvimento dos poucos profissionais atuantes na área serem grandes, e de se ter alguns centros de pesquisa bem instalados no Brasil, a tecnologia assistiva é pouco desenvolvida no país, principalmente se comparado ao potencial que o mercado oferece, com deficientes e pacientes em quantidades significativas (MORAES, 2000).

Aliado a estes fatores observa-se que a crescente complexidade dos automatismos que os equipamentos vêm adquirindo, é traduzida numa maior dificuldade em definir, de maneira clara e sem ambigüidades, as especificações funcionais a que estes devem responder. Esta dificuldade é agravada pela crescente quantidade de informações de entrada e saída a que os equipamentos estão submetidos. Trabalhos realizados nos últimos anos têm

empregado representações gráficas das funcionalidades, que são totalmente independentes das tecnologias com que são implementadas.

A tecnologia assistiva incorporada às pesquisas, via de regra, faz uso do computador como meio de transformação e adaptação de dispositivos. Equipamentos com o que se tem de mais moderno estão disponíveis no mercado para atender principalmente dificuldades de locomoção (cadeiras de rodas), ou de comunicação (sintetizadores e reconhedores de voz). Pouco ou nada se encontra que atenda pessoas com outras deficiências que não estas. Outro fato importante é que estes equipamentos não atendem os preâmbulos da tecnologia assistiva que é de possuir peças intercambiáveis tornando o equipamento pronto para ser reutilizável.

Para atender a esse requisito idealizamos uma ferramenta que permite conceber dispositivos assistivos que possam ser utilizados por vários portadores de deficiências para acionar vários equipamentos. Este autômato atende a necessidade de produzir equipamentos cujo custo de produção seja relativamente baixo, com amplas possibilidades de adaptação, aliados a redução dos tempos de desenvolvimento e prototipação.

Os dispositivos atualmente disponíveis no mercado e ao alcance do consumidor, normalmente, atendem no máximo um pequeno grupo de pessoas portadoras de restrições. As adaptações inerentes à aceitação são promovidas caso a caso, o que leva o equipamento a ser único e com poucas possibilidades de reutilização por outro deficiente. A figura 1.1 mostra a configuração normalmente disponível, onde, para cada pessoa portadora de restrição, é desenvolvido um equipamento específico para assisti-lo na execução de uma, também específica, tarefa. O atuador acionado pelo deficiente age sobre um equipamento em particular, e não pode ser utilizado por outro deficiente ou para outro equipamento.

No modelo convencional, cada adaptador age sobre um equipamento, desta forma uma pessoa portadora de restrição visual pode fazer uso do computador através de software para reconhecimento de fala e sintetizadores de voz que possuem a capacidade de reproduzir o que foi enviado ao terminal de saída. Mas um tetraplégico que deseja fazer uso de um eletrodoméstico, a partir de sua cadeira de rodas, motorizada ou não, poderia realizar a tarefa

através de acionamento remoto. Finalmente, um deficiente motor, impossibilitado de fazer uso das mãos, que deseja ligar um aparelho de televisão, o fará apenas através de um capacete com apontador ótico.

Idealizamos um autômato genérico, que possibilite o acionamento de vários equipamentos a partir do desejo de um ou mais deficientes. Este desenvolvimento vem ao encontro dos anseios de profissionais que, devido a acidentes traumáticos a restrições desde o nascimento, venham desenvolver ou recuperar suas atividades profissionais, possibilitando desta forma maior independência e aceitação pela sociedade.

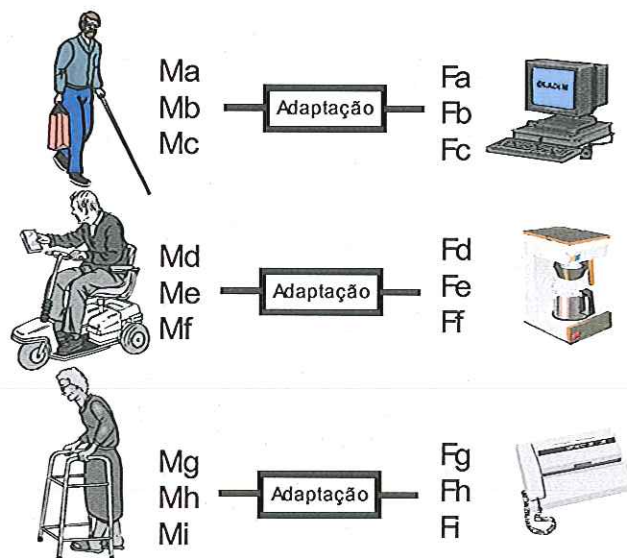


Figura 1.1 – Disposição Tradicional

Na figura 1.2 temos a representação de como o autômato permite atuar sobre os vários equipamentos. Um mesmo adaptador recebe os comandos dos vários sensores acionados por portadores de restrições, as quais refletem suas intenções de ação. O autômato é capaz de reconhecer as intenções do deficiente visual em utilizar o computador para, por exemplo, se comunicar via WEB com outras pessoas, acionar um eletrodoméstico, ligar a televisão e selecionar canais e volume ou realizar qualquer outra tarefa programada a qual



o autômato está habilitado a operar. Se o portador com restrições motoras deseja fazer uso de algum equipamento, o mesmo autômato deve poder acionar o computador, o telefone ou qualquer outro equipamento.

Neste estudo preliminar observamos que para grande parte das atividades diárias das pessoas, sejam elas portadoras ou não de algum tipo de restrição, as ações e conseqüentes reações são na sua maioria semelhantes, análogas ou próximas. Desta forma é possível o desenvolvimento de um autômato que reconhece os comandos enviados por pessoas diferentes e aciona equipamentos dos mais variados.

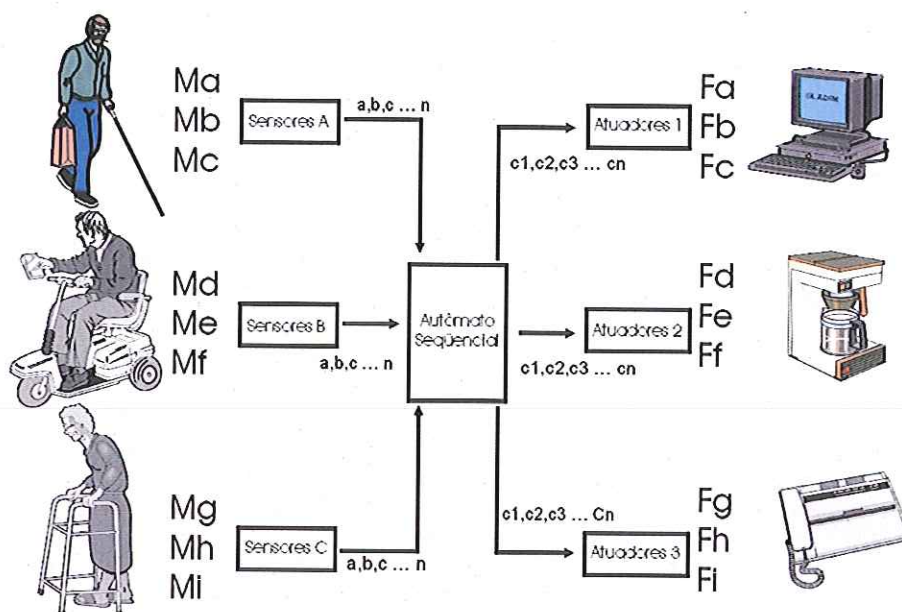


Figura 1.2 – Dispositivo com Autômato Genérico para Acionamento

## 1.2- OBJETIVOS

Neste trabalho foi desenvolvido um dispositivo, independente do uso de computador, que permita não só intercambiar, mas interconectar partes, atender o portador de restrição na melhor forma possível, a custos relativamente baixos. Para tanto utilizaremos o GRAFCET, para conceber e validar o dispositivo. Como ferramenta de modelagem, o GRAFCET permite

visualizar o funcionamento do dispositivo , além de possibilitar a detecção de possíveis falhas e evitar redundâncias de projeto.

### **1.3- ORGANIZAÇÃO DA DISSERTAÇÃO**

O presente trabalho busca demonstrar que um mesmo circuito ou componente eletrônico pode e deve ser utilizado em mais de um dispositivo de auxílio a pessoas portadoras de necessidades especiais.

No capítulo 2 – Fundamentos da Tecnologia Assistiva, procuramos posicionar o leitor frente a esta tecnologia que antes de tudo busca atender às pessoas portadoras de restrições sejam elas físicas, mentais ou outras. Mostramos como os governos de forma geral tratam o assunto, e mais do que isso, como nós mesmos, na condição de sociedade devemos tratar.

O capítulo 3 – Dispositivos para auxílio a vida diária, mostramos que os equipamentos assistentes existentes no mercado e que encontram-se em estudo são únicos e de difícil reutilização, posicionando o trabalho frente ao cenário atual.

No capítulo 4 – Teoria associada ao projeto, apresentamos o GRAFCET, esquema de representação de estados, a teoria associada ao componente FPGA e a linguagem de programação utilização para preparação deste componente (VHDL).

A concepção do dispositivo proposto ocorre no capítulo 5 – Metodologia, onde apresentamos uma breve introdução com as intenções do projeto e em seguida as etapas para a elaboração e concepção do dispositivo assistivo proposto.

Os resultados e testes são apresentados no capítulo 6 – Resultados, onde demonstramos a eficiência do dispositivo, sua robustez de utilização e as possíveis aplicações.

O capítulo 7 – Conclusões, fazemos uma avaliação da tecnologia GRAFCET associada à utilização de componentes como o FPGA, donde sugerimos pontos que devem ser explorados em trabalhos futuros.

## 2. FUNDAMENTOS DA TECNOLOGIA ASSISTIVA

### 2.1- INTRODUÇÃO

Muitas pessoas portadoras de restrições físicas dependem da tecnologia adaptável (também conhecida como tecnologia assistiva ou AT), e dos serviços por ela disponibilizados para participar da vida diária e da sociedade. Porém poucas são aquelas que podem ser beneficiadas devido aos custos envolvidos na aquisição e no treinamento destas tecnologias.

Em 1983 a Associação Americana de Terapia Ocupacional formulou conceito para adaptação terapêutica – “o *design* e a reestruturação de um ambiente físico para auxiliar no desempenho de atividades de autocuidado, trabalho e lazer, isso inclui selecionar, adquirir, ajustar e fabricar equipamento e orientar o paciente, a família e a equipe terapêutica no uso apropriado desse equipamento” (ROCHA 1997) (MELLO 1999). Tecnologia adaptável ou assistiva foi definida nos Estados Unidos no ano de 1988, por uma lei pública (Technology-Related Assistance for Individuals with Disabilities Act Public 100-147) como “qualquer item, peça de equipamento ou sistema de produtos, que adquiridos no mercado, modificados ou feitos sob medida, que aumentam ou melhoram as habilidades funcionais do indivíduo com limitações” (ROCHA 1997) (MELLO 1999).

A lei pública de 1988, além de definir o que seria tecnologia assistiva aplicada a equipamentos, define também o que vem a ser serviços de tecnologia assistiva como – “qualquer serviço que direta ou indiretamente venha a ajudar um portador de restrição física na seleção, aquisição, ou uso de um dispositivo de tecnologia assistiva” (RESNA 1996). Tal definição inclui:

- a) a avaliação da tecnologia necessária ao portador de restrição levando-se em conta sua inaptidão, incluindo-se aí a avaliação funcional promovida pelo uso da tecnologia apropriada;
- b) serviços que consistem em comprar, arrendar ou prover a aquisição de dispositivos de tecnologia assistiva por indivíduos com inaptidões;
- c) serviços de selecionar, projetar, ajustar, personalizar, adaptar, manter, consertar ou substituir dispositivos com tecnologia assistiva;
- d) coordenar o uso de terapias necessárias, intervenções ou serviços com dispositivos de tecnologia assistiva ligados à educação e a programas de reabilitação planejada;
- e) treinar ou prestar ajuda técnica ao indivíduo com restrição física, ou pessoa responsável pelo acompanhamento do paciente; e
- f) treinar ou prestar ajuda técnica a profissionais , empregadores ou outros indivíduos para que se provenha serviços, empregos ou envolvimento do portador de restrição junto ao seu ambiente de convívio (USC 2001).

Como definido por Steinfeld apud Rocha 1997, são princípios da tecnologia assistiva:

- a) acomodar o maior número possível de diferenças individuais, não construindo produtos ou serviços específicos para grupos de pessoas, no sentido de assegurar que todos possam utilizar os componentes do ambiente, os produtos e os serviços;
- b) reduzir a quantidade de energia necessária para a utilização dos produtos, serviços e meio ambiente;
- c) tornar o ambiente, serviços e produtos mais abrangentes; e

- d) pensar e conceber produtos, serviços e ambientes como sistemas, que talvez tenham peças intercambiáveis ou a possibilidade de acrescentar características para as pessoas com necessidades especiais.

Tecnologia assistiva pode ser classificada como geral, quando é aplicada à maioria das atividades que o usuário desenvolve, ou específica, quando aplicada em uma ou outra atividade visando o potencial máximo, como aparelhos para alimentação. Pode também ser individualizada, particularizada, produzida para atender um caso específico. Muitas são as vezes que é necessária à modificação dos dispositivos de tecnologia assistiva existentes no comércio para atender às necessidades e características individuais do usuário (OHIO 1999).

Chama-se de instrumentos os dispositivos que requerem habilidades específicas e/ou treinamento do usuário para que possam ser utilizados, como por exemplo, o manejo de cadeiras de rodas. Os equipamentos são definidos como dispositivos que não necessitam de habilidades especiais do usuário para o seu uso, como por exemplo, sistemas de assento. Instrumentos e equipamentos podem ser simples ou complexos de acordo com o tipo de material e tecnologia empregados (MELLO 1999).

Uma recente pesquisa bibliográfica define que (Rede On-Line 2000) (Click Website 2000), a tecnologia assistiva pode ser caracterizada de acordo com a área de aplicabilidade:

- Adaptações para a vida diária – materiais e produtos que auxiliam no desempenho de tarefas cotidianas como o banho, o autocuidado, a preparação de alimentos, alimentar-se, vestir-se e a manutenção do lar;
- Sistemas de comunicação alternativa – recursos eletrônicos, computadorizados ou não, que permitem a comunicação expressiva e receptiva das pessoas sem ou com limitações de fala. Estes recursos variam de acordo com o tipo, severidade e progressão da incapacidade;

- Dispositivos para utilização de computadores – são recursos para a recepção e emissão de mensagens, acessos alternativos, teclados e mouses adaptados, que permitem a pessoas com restrição física operar computadores. São exemplos destes recursos: ponteiras de cabeça, de luz, sintetizadores de voz, método Braille, reconhecimento de voz, etc.;
- Sistemas de controle ambiental – são unidades computadorizadas que permitem o controle de equipamentos eletrodomésticos, sistemas de segurança, de comunicação, de iluminação, em casa ou em outros ambientes familiares ao usuário;
- Adaptações estruturais – dispositivos que reduzem ou eliminam barreiras arquitetônicas facilitando a locomoção do portador de deficiência, como por exemplo: rampas e elevadores entre outros;
- Adequação da postura sentada – acomodações para cadeira de rodas ou outro sistema de sentar que visam à redução da pressão na superfície da pele (almofadas especiais, assentos e encostos anatômicos), bem como posicionadores que propiciam maior estabilidade e postura adequada do corpo através do suporte e posicionamento de tronco/cabeça/membros;
- Adaptações para deficientes visuais – auxílios que incluem lupas e lentes, método Braille, para equipamentos com síntese de voz, etc.;
- Adaptações para deficientes auditivos – auxílios que incluem equipamentos e aparelhos para surdez, telefone com visor e teclado para recebimento e envio da mensagem, sistemas com alerta tátil-visual, entre outros;
- Equipamentos para a mobilidade – cadeiras de rodas, andadores, bengalas, muletas e acessórios. Devem ser indicados observando a necessidade funcional do usuário, avaliando-se a força, equilíbrio, coordenação, capacidades cognitivas, medidas antropométricas e postura funcional; e

- Adaptações em veículos – incluem a modificação de veículos para a direção segura, sistemas de acesso e saída do veículo como elevadores de plataforma ou dobráveis, plataformas rotativas, correias e barras.

O uso de instrumentos e equipamentos deve ser contextualizado de acordo com o processo de reabilitação e as necessidades do paciente. É necessário que se obtenha uma adequação entre o equipamento ou instrumento indicado e as demandas do paciente. As necessidades relativas às deficiências físicas podem ser várias, assim como o tipo de limitação funcional, as conseqüências dessa limitação, a condição social do paciente, sua história de vida (VIEIRA 1996), seu estado psicológico, o acesso aos recursos da comunidade, etc. Estes aspectos influenciam a aceitação ou não do recurso tecnológico.

Para uma maior efetividade do uso do dispositivo tecnológico é imprescindível o parecer do paciente, respeitando seus valores, suas reais necessidades e expectativas. Dessa forma, evita-se as chances de abandono do recurso (ROCHA 1997) (MELLO 1999). Tanto os profissionais de reabilitação, quanto os pacientes devem ter acesso às informações recentes sobre os recursos tecnológicos existentes no mercado, facilitando a escolha do dispositivo a ser utilizado ou confeccionado. “A falta destas informações poderá resultar na aquisição de um dispositivo menos eficaz, cujos aspectos negativos representados para um determinado usuário, tais como aparência, complexidade de manuseio ou assistência técnica de difícil acesso, favorecerão o abandono” (ROCHA 1997) (LUPTON 2000).

Um dos principais fatores na dificuldade de aquisição dos dispositivos está na falta de recursos financeiros. O Sistema Único de Saúde (SUS) e a Previdência Social recebem verbas para o fornecimento de alguns produtos como cadeiras de rodas, próteses e andadores, mas além de não serem suficientes, eles não contemplam outros inúmeros recursos tão necessários à vida dos portadores de restrições físicas como, por exemplo, uma almofada especial para evitar úlceras de pressão. Da mesma forma, os serviços de reabilitação disponíveis no país são precários, recebem pouco investimento e atenção e, ao contrário do que ocorre em países desenvolvidos, as empresas



seguradoras de saúde não prevêm o financiamento dos recursos tecnológicos para seus clientes, não sendo anormal que o próprio paciente venha a arcar com os custos financeiros para a obtenção dos dispositivos ou equipamentos assistivos (ROCHA 1997) (MELLO 1999).

Aliado aos fatores apontados acima, existe, ainda, um desconhecimento dos profissionais da reabilitação e dos pacientes quanto a estes tipos de recursos tecnológicos (VINCENT 1999) (MORIN 1999). O mercado nacional é incipiente no que se refere à fabricação de equipamentos e dispositivos envolvendo tecnologia assistiva, desta forma na sua grande maioria, são importados e apresentados em catálogos, impossibilitando a experimentação do produto pelo paciente antes da aquisição. Em países onde os direitos do cidadão deficiente são respeitados, o uso desses recursos é rotineiro e facilitado. Pacientes e profissionais podem obter informações atualizadas em softwares, publicações, boletins eletrônicos, agências especializadas, livrarias entre outras fontes. Não existe uma sistematização de informações sobre tecnologia assistiva no mercado brasileiro, poucos são os locais onde se podem obter informações sobre como e onde obter esses produtos ou recursos, quais são os fabricantes e fornecedores, sobre a possibilidade de importação, quais os custos envolvidos e qual o tempo para entrega. Essa falta de informação agrava o problema relacionado à prescrição destes produtos, dificultando uma melhor seleção e desestimulando o uso.

Um grave problema a ser levado em consideração, refere-se ao abandono do equipamento. De acordo com Phillips e Zhao (1993) (ROCHA 1997), de 227 adultos consultados nos Estados Unidos, portadores de várias deficiências, 29,3% abandonaram os recursos tecnológicos, sendo esta ocorrência verificada principalmente entre pacientes incapacitados recentemente. Em pesquisa desenvolvida no Hospital de Reabilitação Nacional (1991), em Washington, concluiu-se que as principais causas de abandono são:

- A exclusão do paciente durante o processo de escolha do recurso tecnológico;

- O fácil acesso a estes dispositivos, predispõe a falta de acompanhamento de um profissional capacitado na orientação e escolha do produto;
- O desempenho do recurso tecnológico aquém das expectativas do paciente; e
- Alterações nas demandas e prioridades do paciente.

A indicação dos recursos tecnológicos deve ser de acordo com o “ambiente físico e psicossocial, traços de personalidade e temperamento do sujeito que utilizará a tecnologia e das características próprias do produto” (MELLO 1999) A eficácia da tecnologia assistiva irá depender também do significado que esta apresenta para o paciente. Se representar um atributo que expõe sua deficiência, que revela sua dificuldade e impotência, dificilmente ela irá promover autonomia e independência. O uso dos recursos tecnológicos deve fazer parte do processo de reabilitação (PHILOT 1997) Devem ser prescritos e selecionados por profissionais da área (MELLO 1999), levando em consideração a participação e o contexto de vida do paciente e respeitando sua decisão quanto à escolha de seu modo de vida (ROCHA 1997).

## **2.2- DISPOSITIVOS PARA AUDIÇÃO ASSISTIVA**

Os dispositivos automáticos de entrada (DAE) para audição foram patenteados primeiramente em 1890, mas os primeiros trabalhos utilizando excitação elétrica só ocorreram em 1955 e 1956 com Djourno e Ninhos (SIMMONS 1985). Em 1961 foram escolhidos os primeiros pacientes para receber o implante do dispositivo, mas devido aos agentes tóxicos contidos nos eletrodos surgiram reações alérgicas o que forçou a retirada dos implantes. Mesmo permanecendo por pouco tempo com o equipamento implantado, cerca de 3 semanas, os resultados obtidos foram satisfatórios mostrando que os pacientes implantados se beneficiaram do dispositivo (SIMMONS 1985).

A mesma estrutura básica do DAE de audição que foi desenvolvido há um século, ainda é usada hoje. Esta estrutura contém diversos componentes: microfone, amplificador, fones e fonte de alimentação. O som é captado pelo microfone através da reação do diafragma as ondas que nele agem. O amplificador aplica o sinal originário do microfone no alto-falante após ampliação. Para reduzir as distorções de saída, o sinal passa por três ou quatro estágios de ampliação. Os alto-falantes fazem a transformação do sinal elétrico em formas mecânicas de ondas sonoras. A fonte de alimentação é uma bateria que permite ao dispositivo ser portátil.

A classificação do DAE de audição é geralmente feita por sua posição no corpo. São quatro os tipos básicos: 1) atrás da orelha; 2) na forma de óculos; 3) interno ao corpo; e 4) dentro do ouvido. O dispositivo colocado atrás da orelha é formado por um pacote que contém os três componentes principais e a bateria. O sistema acoplado aos óculos é mais raro, sendo pouco usado. É indicado para clientes que possuem uma perda severa de audição. O DAE interno ao corpo é o mais avançado sendo que na primeira versão os três componentes eram peças separadas e conectadas por fios, sendo hoje, mesmo que experimentalmente, construído em uma única peça. Por último os colocados internamente ao ouvido são os mais utilizados e populares devido ao seu tamanho reduzido. Visão anatômica do ouvido interno, onde será aplicado o implante e do dispositivo podem ser observadas nas figuras 2.1 e 2.2 respectivamente com detalhe para o eletrodo na figura 2.3.

Estão sendo desenvolvidos estudos e protótipos de dispositivos DAE para audição na forma de próteses para implante intra-auricular para pessoas surdas, embora o som não seja tão natural quanto alguém com audição normal (WEBSTER *et al*, 1985).

A tecnologia, os procedimentos de avaliação e a experiência dos médicos qualificados vêm crescendo rapidamente. Como a tecnologia e a experiência avançaram, mudanças notáveis ocorreram. Foram desenvolvidos desde multi-canais para implantes até sistemas de reconhecimento de fala com propriedades excepcionais, muito embora em pequena, mas crescentes, porcentagens de implantes (ANONYMOUS 1988).

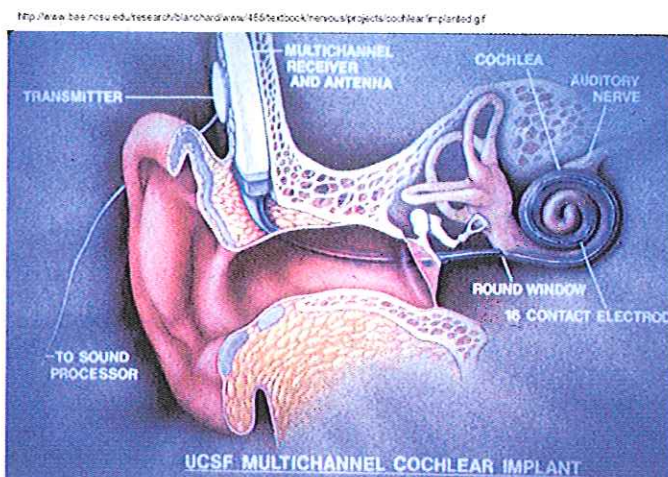


Figura 2.1 – Anatomia do Ouvido com o Dispositivo Implantado (ANONYMOUS, 1988)

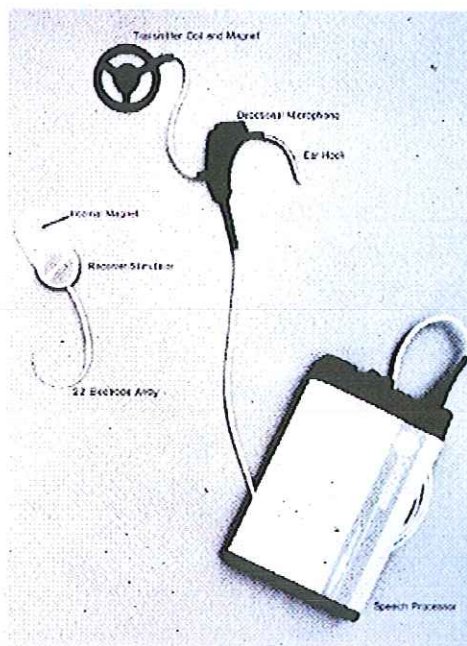


Figura 2.2 – Exemplo de Dispositivo para Implantação em Pacientes com Surdez Profunda (ANONYMOUS, 1998)

<http://www.bas.nyu.edu/research/blechard/blew/445/led2006/hatyoua/projecta/occhina/occhinaelectrode.gif>



Figura 2.3 – Visão do Eletrodo que será Implantado (ANONYMOUS, 1988)

### 2.3- DISPOSITIVOS PARA VISÃO ASSISTIVA

As pessoas imaginam cegos como pessoas com longos bastões ou bengalas, ou ainda como pessoas que seguem seus cães guias. Supõe-se que os cegos são todos iguais. Os graus de prejuízo na visão tornam estas pessoas com deficiência visual diferentes entre si. No dia a dia a comunicação entre as pessoas, depende muito da habilidade em ler e escrever, o que para o cego pode ser um problema. Muitos dispositivos foram desenvolvidos para ajudar a

pessoa portadora de restrição visual, sendo que estes podem ser divididos em: 1) DAE para baixa visão; 2) DAE visuais táteis; 3) DAE soletrado; e 4) estimulação diretamente no córtex visual.

Um exemplo de DAE visual tátil é o Braille. O Braille consiste de uma matriz de seis pontos em duas colunas que pode representar uma letra ou um conjunto de letras. Mesmo sendo um método difícil de aprender, os cegos conseguem velocidade muito boa para leitura. O problema é que o material Braille é caro de ser produzido o que explica porque apenas 10% da comunidade cega tem acesso a ele.

O DAE soletrado consiste em diversos tipos de dispositivos com saída de voz. A saída falada é obtida através de um sistema que faz a varredura da página a ser lida convertendo os caracteres em sinais sonoros. A estimulação do córtex visual pode causar uma percepção visual real, ao contrário dos DAE apresentados anteriormente. A imagem pode ser transmitida ao cérebro por sinais elétricos e a estimulação do córtex visual pode produzir percepções visuais. Este método é o mais complexo de todos e para que o implante possa ser aceito pelo corpo devem ser fabricados de tal forma que não sofram corrosão ao longo do tempo (WEBSTER et al., 1985).

Outros avanços têm sido obtidos nos últimos tempos, sendo um muito interessante o chamado projeto IRIS que transmite a informação codificada aos usuários equipados com um receptor apropriado, informação esta convertida em um discurso obtido artificialmente na língua do usuário (MILNER, 1993).

## **2.4- DISPOSITIVOS PARA INABILIDADES MOTORAS**

Estima-se que 10% da população tenha algum tipo de inabilidade motora que limita a atividade normal do indivíduo. Os DAE para mobilidade fornecem um aumento importante na independência das pessoas com inabilidades severas. O equipamento mais popular é a cadeira de rodas usada por uma grande população de deficientes motores e pode ser de dois tipos: aquela

movida pela força humana (Figura 2.4), através da ação nas rodas laterais, e a movida à bateria de automóvel (Figura 2.5).

Atualmente são muitas as variações no projeto e construção das cadeiras de rodas podendo ter um efeito significativo no desempenho, nas exigências de energia e na durabilidade sob várias circunstâncias ambientais de uso. Com uma grande variedade de usuários, é de se esperar que haja não uma, mas várias opções de modelos para escolha. Vários aspectos devem ser levados em consideração na escolha do equipamento, tais como: exame físico do cliente, o uso pretendido e os recursos disponíveis. O conhecimento do desempenho de cada tipo e parte componente pode ajudar na escolha lógica. Também se faz necessário avaliar as potencialidades do usuário junto ao ambiente e uso pretendido (MCLAURIN & BRUBAKER, 1991).

O uso de equipamentos automotivos modificados é extremamente importante para o portador de inabilidade motora. Com estes equipamentos ele pode aprender a dirigir, com segurança e independentemente, um veículo motorizado. Algumas exigências devem ser observadas: 1) todos os controles essenciais devem estar dentro da área de alcance do usuário, e identificados apropriadamente; 2) os equipamentos da adaptação não podem diminuir ou modificar a eficácia de proteção interna existente; e 3) todos os dispositivos de controle adaptável devem permitir o uso convencional dos controles automotivo padrão (WEBSTER *et al*, 1985).



Figura 2.4 – Cadeira de Rodas Tradicional (WWW.FREEDOM.IND.BR)



Figura 2.5 – Cadeira de Rodas Motorizada, Controle na Alavanca do Braço  
(WWW.FREEDOM.IND.BR)

## 2.5- OS DESAFIOS DA ENGENHARIA DE REABILITAÇÃO

Nas últimas duas décadas, houve um aumento significativo no número e nas especializações de médicos, assistentes, e tecnologias envolvidos na assistência e ajuda às pessoas portadoras de deficiências. Dispositivos de tecnologia assistiva tendem a possuir uma alta taxa de uso quando forem projetados para ser de peso reduzido e portáteis, fáceis de usar e montar, passíveis de se obter e manter, e na medida do possível, apresentarem as mesmas características daqueles usados por pessoas não deficientes. A flexibilidade e o grau com que eles podem ser ajustados ou ligados a outros dispositivos também podem ser importantes. Dispositivos deixam de ser usados quando não valem o esforço, não têm aplicação prática, criam desconforto ou inconveniência (SCHERER, 1993). O envolvimento do consumidor na concepção e desenvolvimento de equipamentos com tecnologia assistiva assegurariam maior confiabilidade e ganho funcional, além das características próprias a atender o grupo de pessoas a que se propõe. A



melhor aproximação é a minimalista, onde os dispositivos de utilização fácil, custo baixo de fabricação e aplicabilidade garantida são postos no mercado e oferecidos aos usuários.

O propósito da tecnologia assistiva é promover um aumento da funcionalidade da pessoa portadora de restrição, recuperar sua auto-estima e qualidade de vida. Se o dispositivo não cumpre estas características, não será usado. Assim sendo, é de se esperar que o profissional da área observe as características pessoais do paciente, pois somente assim assegurar-se-á o sucesso da reabilitação (SCHERER, 1993).

### **3. DISPOSITIVOS PARA AUXÍLIO A VIDA DIÁRIA**

#### **3.1 – INTRODUÇÃO**

Na base de dados da ABLEDATA (ABLEDATA, 2002), constam dispositivos assistenciais disponíveis nos Estados Unidos para venda a portadores de deficiências. Detectamos nesta ampla base que, no que concerne dispositivos para vida diária, existe um número muito grande de opções, muito embora basicamente o que se encontra são orteses, cadeiras de rodas, equipamentos para surdez e visão, ou então dispositivos com aplicações mais amplas, mas neste caso conectados a computadores, dedicados ou não.

Entre os dispositivos para auxílio diário disponível no mercado, o que se destaca, muito embora ainda em fase de pesquisa e desenvolvimento, é um equipamento conhecido como “MASTER”. O “MASTER” é uma estação de trabalho que busca proporcionar alguma independência ao deficiente, seja para execução de tarefas tanto do cotidiano como vocacionais.

#### **3.2 - ESTAÇÃO DE TRABALHO “MASTER”**

No ano de 1975, a Comissão de Energia Atômica francesa estudou a possibilidade de transferir sua experiência no campo da robótica nuclear para

problemas específicos na ajuda a pessoas portadoras de inaptidões. Ao mesmo tempo nos Estados Unidos estudava-se a concepção de uma estação de trabalho robotizada. Muitos trabalhos foram publicados no que se refere à concepção de equipamentos de ajuda técnica, e as evoluções no atendimento em situações reais: SPARTACUS (GUITTET, 1979) (KWEE, 1983); HANDY1 (TOPPING, 1998) e MASTER (DÉTRICHÉ, 1989) (BUSNEL, 1998).

Em 1992, após avanços obtidos com o comando de controle do "MASTER" e os resultados das avaliações francesas, o "MASTER" foi reprojetoado por OXIM para a European Communit TIDE (Technology for the socio-economic Integration of the Disabled and Elderly people), iniciativa que recebeu o nome de RAID (Robot for Assisting the Integration of the Disabled).

A idéia do grupo era após uma análise ergonômica desenvolver a técnica necessária para ajudar uma pessoa tetraplégica nas suas atividades funcionais diárias. Informações gerais tais como a que horas do dia um robô poderia substituir a ajuda humana, e quando tal ajuda atenderia suas necessidades, eram necessárias. As conclusões da análise foram que se deve procurar satisfazer as necessidades de tarefas repetitivas (por exemplo, pegar o fone, carregar um disquete no computador, dar uma bebida e etc), mas com dispositivos adaptados para operar dentro do ambiente do usuário.

O equipamento desenvolvido pode ser operado de vários modos: automático, manual, remoto e com controle ambiental (ECS). No modo automático as tarefas são executadas com maior velocidade, sem a intervenção do usuário, mas neste caso os objetos que serão manipulados devem permanecer sempre na mesma posição, enquanto no modo manual os objetos podem ser movimentados, muito embora, dependendo das limitações do paciente este tipo de operação poderia levar tempos muito grandes para ser completado. O equipamento deveria ainda oferecer a possibilidade ser carregado com programas para escritório permitindo, desta forma, que o usuário exerça uma atividade sem a ajuda de um auxiliar.

O protótipo construído por OXIM inclui um eixo vertical e outro horizontal que permite ao braço movimentação no espaço. O movimento é realizado na frente de uma estrutura metálica formada por estantes onde os objetos que podem ser manipulados estão disponíveis. O dispositivo é munido de uma

garra e de uma ferramenta pneumática especializada em manipular livros e folhas de papel. A ferramenta especializada pode ser trocada por outra de tal forma que a tarefa a ser executada seja realizada.

Vista geral do sistema é apresentada na figura 3.1 onde se pode observar a complexidade do dispositivo.

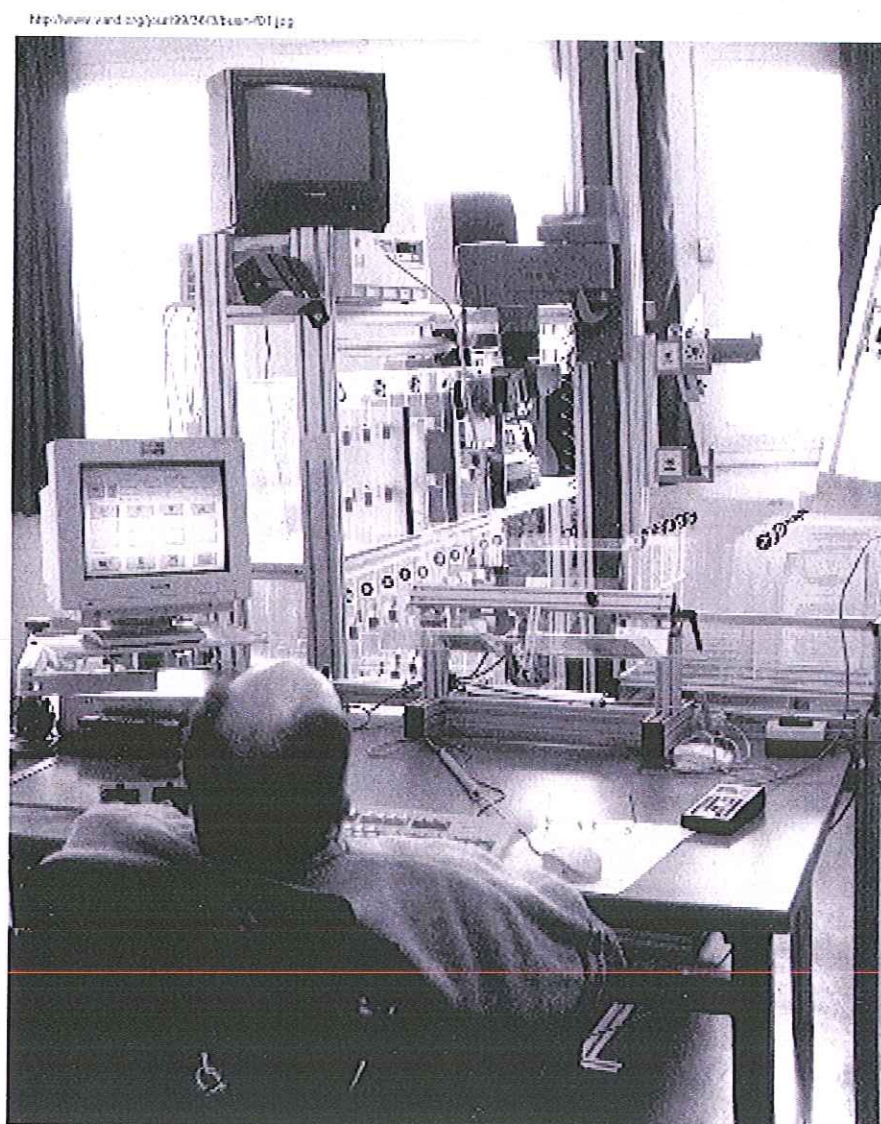


Figura 3.1 – Visão Geral da Estação de Trabalho "MASTER RAID" (BUSNEL, 1998)

O modo de operação ECS permite controlar eletrodomésticos remotamente (infravermelho). Uma interface telefônica está preparada para

realizar chamadas a partir de um banco de dados ou discado digito a digito pelo usuário. Da mesma forma todas estas ações podem ser controladas diretamente pelo usuário ou programadas.

Estes modos de operação disponíveis no robô são necessários para assegurar um bom desempenho, em situações reais, onde a cooperação homem máquina possa tornar a atividade produtiva e adaptável. Uma interface de comunicação homem-máquina mantém o usuário informado do estado do sistema e permite a seleção de seu modo corrente, iniciando ou cancelando rotinas automáticas, ou mudando o modo de operação se necessário. Esta comunicação deve ser executada em tempo real e depende da motricidade residual do usuário para informar o estado do sistema exibida em uma janela do computador como a apresentada na figura 3.2. Métodos diferentes estão disponíveis para seleção da ação: mouse, joystick, rolo, reconhecimento de fala, interruptores que podem ser acionados por sopro ou sucção.

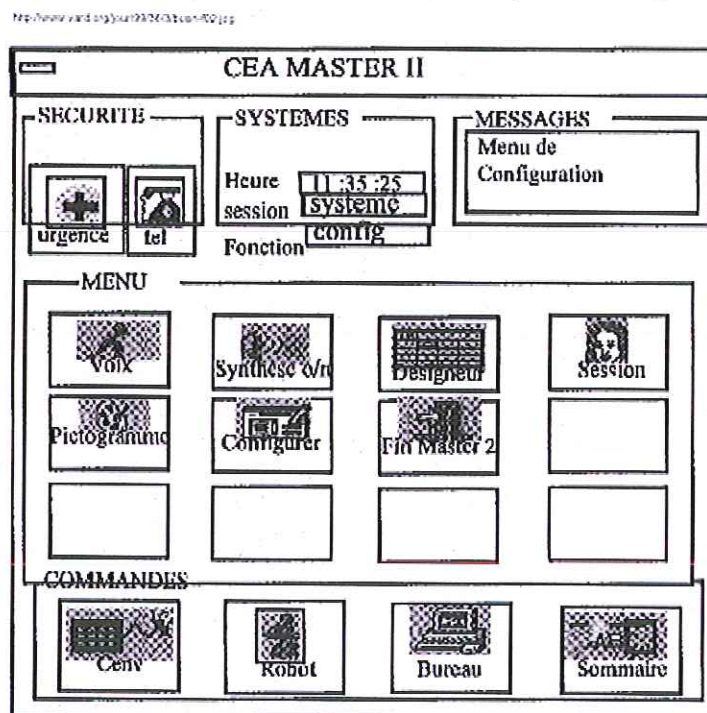


Figura 3.2 – Janela da Interface do Computador (BUSNEL, 1998)

O movimento do robô no modo manual é provocado usando-se interruptores ou sensores proporcionais (joystick). Cada movimento envolve três movimentos primários e um máximo de três sensores são necessários (por exemplo, um joystick com dois graus de liberdade e um rolo para prover um

terceiro grau de liberdade). De qualquer forma, o sistema pode ser operado com apenas um ou dois graus de motricidade se for configurado adequadamente. Neste caso, uma seqüência de procedimentos é automaticamente executada.

O processo de configuração do sistema se mostrou fundamental para efetiva utilização pelas pessoas com inaptidões, pois para cada usuário a interface de comunicação deve ser adaptada à natureza das habilidades do indivíduo. Nesta tarefa são levados em consideração os seguintes parâmetros: motricidade (valores de 1 a 3), velocidade de rolagem, se será utilizado o reconhecimento de fala. Se a pessoa for usar a seleção de um conjunto pré-definido de ações através da fala este é mais eficaz do que rolar um menu, se um sintetizador de voz para informar as instruções operacionais (útil para pessoas com baixa visão), o tempo de espera entre dois pulsos que vêm do interruptor de interface do usuário (útil para pessoas com movimento sem coordenação), a zona morta para cada sensor proporcional. O estudo apresentado acima permite definir parâmetros consistentes para cada comando com respeito à posição do sensor e a máxima velocidade de cada movimento.

A programação do "MASTER" usa uma interface gráfica sendo que determinação das tarefas é de responsabilidade de um terapeuta profissional com treinamento especial e é levada a cabo com a colaboração do usuário final

Rotinas de segurança foram incluídas para possibilitar a mudança rápida de um modo de operação para outro. Às vezes tal mudança pode apresentar algum risco, por exemplo, tendo aberto uma torneira e estando no tempo de espera se o modo automático for desabilitado poderá haver um transbordamento. Por este motivo foram incluídas rotinas de segurança que no caso do exemplo fecharia a torneira imediatamente após a mudança do controle da ação.

É possível modificar coordenadas de pontos durante a execução da tarefa. O programa permite a execução de ordens para deslocamento do braço (vá, circule, linha reta, direita, etc.). Às vezes, em situações reais, é necessário mudar alguns pontos devido a situações não previstas como, por exemplo, no caso do usuário desejar beber algum líquido, o posicionamento preciso da cadeira de rodas, na frente da estação de trabalho, é bastante delicado pois posicionar a boca do usuário, de tal forma que seja possível beber um líquido,

é dependente do tempo necessário para o posicionamento. O modo manual oferece ao usuário a oportunidade para fazer os ajustes necessários. Além disso, o sistema pode memorizar os pontos onde os objetos estão localizados para executar as tarefas automaticamente.

O interruptor que funciona como parada de emergência quando o braço está em movimento, pode ser situado em qualquer lugar no raio de ação do usuário (pés, queixo, cotovelo, ombro). O terapeuta especializado determina o tipo de sensor e o local ideal da instalação. O procedimento de carga do sistema é automaticamente acionado quando o dispositivo é ligado de forma que este pode ser reiniciado facilmente, independente de ter havido algum problema no tempo de execução. O sintetizador de voz pode ser utilizado para lembrar o usuário das instruções. A zona de funcionamento próxima ao usuário esta situada no extremo do alcance do braço do robô para minimizar as possibilidades e conseqüências de uma colisão acidental. Quando o braço entra nesta zona, a velocidade do robô é automaticamente reduzida, rotinas de segurança estão automaticamente carregadas e executadas quando uma parada de emergência é acionada.

Ao equipamento foram aplicadas avaliações técnica e funcional sendo que aos usuários uma avaliação psicossocial foi aplicada. Estas avaliações aconteceram em 1995 e foram aplicadas a 91 pessoas (65 homens e 26 mulheres) com idade média de 32,7 anos (de 15 a 66 anos). Na avaliação técnica, engenheiros e terapeutas trabalharam juntos durante duas semanas nas quais foram analisadas as dificuldades de acesso ao computador, necessidades e possibilidades dos tipos de interfaces, os modos de operação e as falhas ocorridas no processo. Um grupo voluntário de pessoas deficientes trabalhando junto com os terapeutas definiu uma lista de perguntas que deveriam ser respondidas pelos usuários durante a fase de avaliação. A avaliação das tarefas foi baseada em três domínios: vida diária, lazer e aplicações vocacionais.

As avaliações mostraram que o aparelho é útil à pessoa portadora de restrições. O fato de não ficar inteiramente dependente de outra pessoa, faz com que ela recupere parte da liberdade tornando-se apta a realizar tarefas cotidianas além das vocacionais.

## 4. TEORIA ASSOCIADA AO PROJETO

### 4.1- MÉTODO GRAFCET

A crescente complexidade dos automatismos industriais se traduz numa maior dificuldade em se definir de uma maneira clara e sem ambigüidades as especificações funcionais a que os autômatos devem responder. Esta dificuldade é agravada pela utilização de um número relativamente grande de informações de entradas e saídas (HAGBEBELL, 1999).

Várias são as formas de se representar e apresentar um processo, e estes podem ser classificados em grupos distintos:

- Uma descrição unicamente literal poderia ser longa, incomoda e às vezes imprecisa ou incompleta;
- Uma descrição lógica poderia quase exclusivamente focar uma determinada tecnologia e não ser apropriada para definição de programas;
- Uma apresentação por organograma, mais geral, se adapta bem às realizações mediante programas, porém é muito pobre no caso das seqüências e não mostra os possíveis funcionamentos simultâneos; e
- Uma representação gráfica das especificações funcionais que são totalmente independentes das tecnologias envolvidas na implementação, podendo ser modelada por componentes pré-



definidos (módulos pneumáticos, relés eletromecânicos, módulos eletrônicos) ou programada (PLC, microprocessador, etc.) ;

Esta nova forma de representação baseia-se no conceito de etapas e transições, que simplificam muito a síntese dos automatismos seqüenciais, ao se considerar que em um trecho onde um grande número de informações estão disponíveis, poucas têm um significado em um determinado momento.

A partir destas idéias, os trabalhos efetuados pelas comissões do AFCET (Association Française pour la Cybernétique et Technique) e da ADEPA (Agence Nationale pour le Development de la Production Automatisée) definem uma forma de representação disposta em um diagrama funcional: o GRAFCET (Graphe de Comandes Étape/Transition) (BOSSY, 1979). Este diagrama permite descrever os comportamentos do automatismo em relação às informações que recebe, impondo um funcionamento rigoroso, evitando desta forma incoerências, bloqueios e conflitos no funcionamento, por outro lado é possível, a partir do diagrama, se obter um circuito digital equivalente facilitando sua implementação. Em cada nível de descrição, este diagrama pode ser modificado ou corrigido, sem a necessidade de se voltar a estudar as etapas anteriores.

O GRAFCET é composto por um conjunto de:

- Etapas ou Estados a que vêm associadas ações;
- Transições as quais são associadas receptividades; e
- Uniões orientadas que unem etapas às transições e as transições às etapas (DAVID, 1995).

#### **4.1.1- Etapas**

Uma etapa se caracteriza por um comportamento invariável em uma parte ou na totalidade de uma ação.

Em um determinado momento, e conforme ocorre à evolução do sistema:

- Uma etapa pode estar ativa ou inativa;
- O conjunto das etapas ativo define a simulação da parte comandada.

As etapas são representadas por um quadrado com um número em sua parte superior que funciona como indicação. A entrada e a saída de uma etapa aparecem na parte superior e inferior, respectivamente de cada símbolo. O conjunto formado por um quadrado e as extensões de entrada e saída constitui o símbolo completo da etapa (Figura 4.1).

Quando é necessário indicar ou determinar a situação do GRAFCET em um determinado momento, é cômodo identificar todas as etapas ativas neste momento, mediante um ponto na parte inferior dos símbolos destas etapas (Figura 4.2).

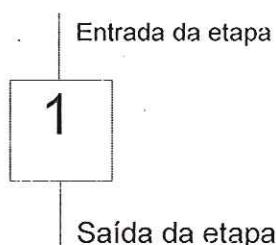


Figura 4.1 – Representação de uma Etapa



Figura 4.2 - Etapa Ativa

A eficiência e precisão de um GRAFCET estão diretamente relacionadas à quantidade de etapas utilizadas para descrever determinado processo ou sistema. Desta forma, quanto maior for o número de etapas em que se pode dividi-lo, maior será sua eficiência de descrição de cada etapa e maior será a precisão do diagrama como um todo (GEORGINI, 1971).

A etapa inicial é ativada incondicionalmente no início do controle de um sistema e indica sua situação inicial. Podem existir tantas etapas iniciais quantas se fizerem necessárias, sendo que todas são ativadas

simultaneamente no início do controle do sistema. A representação da etapa inicial é ligeiramente diferente das outras, como pode ser observado na figura 4.3.

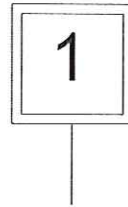


Figura 4.3 – Representação da Etapa Inicial

#### 4.1.2- Ações Associadas às Etapas

As ações associadas às etapas somente serão executadas quando e se a etapa a qual está associada estiver ativa, caso contrário são ignoradas. Quando a etapa, cuja ação está associada, for desativada essa ação pode ser continuada ou finalizada, conforme a definição utilizada.

A ação descrita é inserida em um retângulo, para sua representação, conectado ao lado direito da etapa correspondente (Figura 4.4). Para se representar mais de uma ação associada à mesma etapa, utiliza-se uma das representações apresentadas na Figura 4.5 e, embora estejam representadas em retângulos individuais, as simbologias utilizadas não especificam qualquer seqüência entre as ações associadas à etapa.

Um ponto chave na definição da ação associada à etapa é que ela seja feita de forma clara e sem ambigüidades. Além de definir o comportamento do sistema em determinado momento, deve definir se a ação é continuada ou se será finalizada após a desativação da etapa.

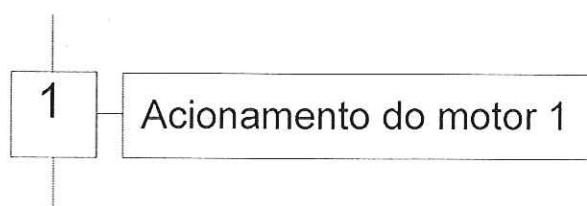


Figura 4.4 – Ação Associada à Etapa

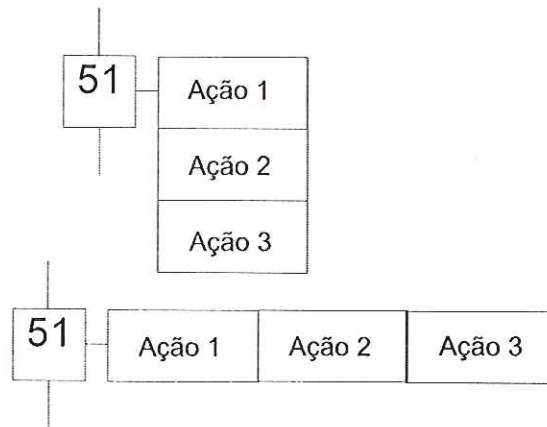


Figura 4.5 – Representação de várias Ações Associadas a uma Etapa

A figura 4.6 apresenta dois exemplos de definição da ação associada à etapa:

- em (a) a ação é iniciada ao ser ativada a etapa correspondente e assim mantida enquanto ela estiver ativa, sendo finalizada pela sua desativação;
- em (b) a ação é iniciada quando a etapa 7 é ativada e assim permanece até a etapa 17, quando é finalizada.

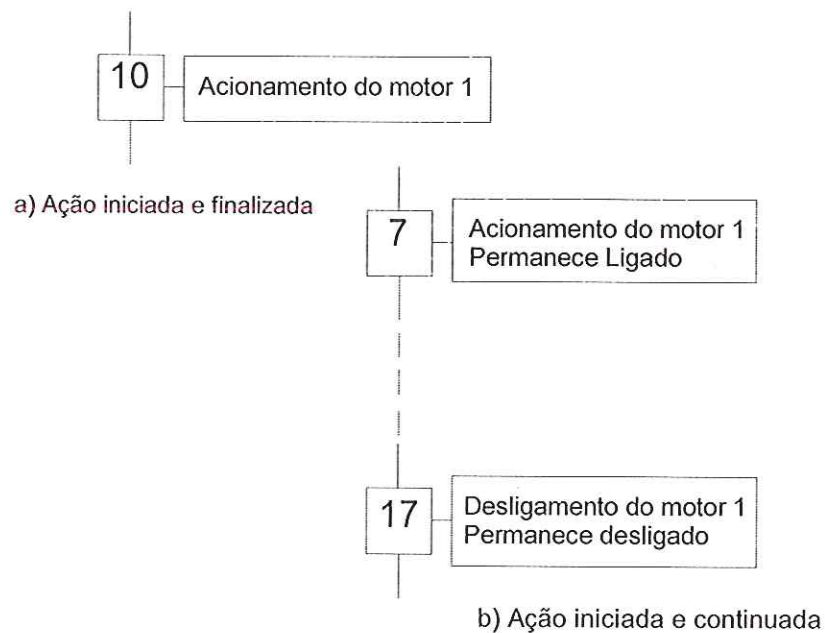


Figura 4.6 – Definição da ação associada à etapa

### 4.1.3 – Transições

Uma transição indica a possibilidade de evolução entre etapas. Esta evolução se consuma ao se transpor à transição. Esta transposição provoca a passagem de uma situação no diagrama para outra. A transição é representada por um pequeno traço entre duas etapas e estão associadas a condições que determinam a transposição de uma etapa para outra (Figura 4.7). Para que uma transição esteja habilitada (com possibilidade de ser transposta), é necessário que todas as etapas imediatamente precedentes a ela estejam ativadas, e neste instante, se a condição de transposição ou transição for satisfeita ela será transposta.

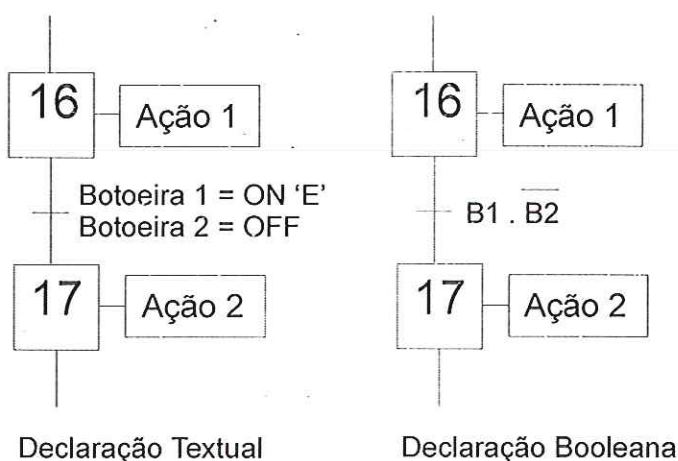


Figura 4.7 – Representação de condições associadas às transições

A transposição de uma transição ocasiona a ativação de todas as etapas imediatamente seguintes e a desativação de todas as etapas imediatamente precedentes, simultaneamente.

As condições associadas às transições são proposições lógicas, as quais podem ser verdadeiras ou não em determinado instante. Expressam condições, internas ou externas, que devem ser satisfeitas para que a transição ocorra e se tenha a transposição de uma etapa a outra. Podem ser

representadas por declarações textuais, expressões booleanas. As condições associadas às transições podem apresentar detalhes que as relacionem ao tempo ou o estado de uma variável.

As entradas externas provêm da parte operacional, fornecida por operadores ou de outros sistemas.

#### 4.1.4 – Ligações Orientadas

As ligações orientadas unem as etapas às transições e estas a outras etapas. Assinalam o caminho de evolução de um processo.

As ligações orientadas são representadas, preferencialmente, por linhas horizontais ou verticais. Convencionalmente, o sentido de evolução do GRAFCET é de cima para baixo. Para facilitar o entendimento ou para indicar o sentido inverso, em certas ocasiões, utilizam-se setas (Figura 4.8).

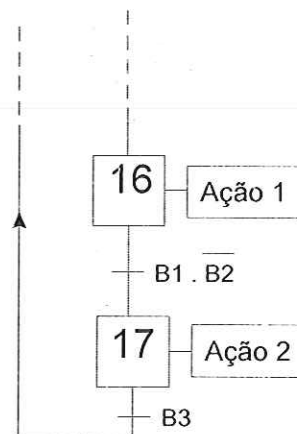


Figura 4.8 – Sentido das ligações orientadas

Existem duas regras simples para conexões etapa-transição e transição-etapa por intermédio de ligações orientadas:

- duas etapas nunca podem ser conectadas diretamente, portanto, devem ser separadas por uma única transição;
- duas transições nunca podem ser conectadas diretamente, portanto, devem ser separadas por uma única etapa.

#### 4.1.5- Regras de Evolução

##### 4.1.5.1- Situação Inicial

A situação inicial do GRAFCET caracteriza o comportamento inicial da parte mandante em relação à parte operacional e corresponde às etapas ativas no começo do funcionamento. Se esta situação é sempre a mesma, como no caso dos processos cíclicos, estará caracterizado o estado inicial e este corresponde a um comportamento de repouso. Obrigatoriamente deve existir pelo menos um estado inicial em cada GRAFCET.

##### 4.1.5.2- Transposição de uma transição

Uma transição somente será transposta quando:

- ela estiver habilitada, ou seja, todas as etapas imediatamente anteriores estiverem ativadas; e
- a condição associada à transposição é verdadeira.

Quando as duas condições são atendidas, a transição é obrigatória.

##### 4.1.5.3- Evolução das etapas ativas

A transposição de uma transição provoca a ativação das etapas imediatamente seguintes e a desativação daquelas imediatamente anteriores, simultaneamente.

##### 4.1.5.4- Estruturas de uso freqüente

- Divergência em OU é representado pelo esquema da figura 4.9:

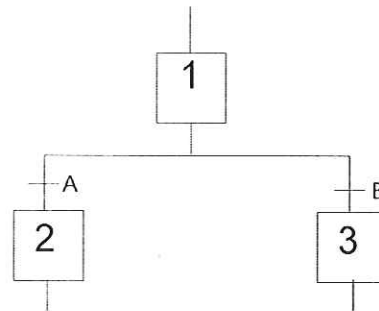


Figura 4.9 – Divergência em OU

Quando a etapa 1 está ativa, dependendo de qual transição for ativada (A ou a B), a próxima etapa poderá ser a 2 ou a 3.

- Convergência em OU, pode ser observada na figura 4.10:

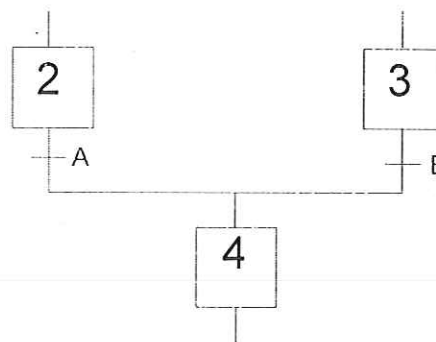


Figura 4.10 – Convergência em OU

Neste caso, se a etapa 2 for a que está ativada, deve-se cumprir as exigências da transição A, isto ocorrendo haverá a transposição para a etapa 4, por outro lado, se a etapa ativa for a 3, a transição a ser ativada é a B para que ocorra a mesma transposição para a etapa 4.

- Divergência em E. A divergência em E se diferencia da anterior uma vez que quando a transição é ativada, as etapas seguintes são, simultaneamente, ativadas (Figura 4.11).

Estando a etapa 1 ativa, e se cumprindo as exigências da transição C, passam a estar ativas as etapas 2 e 3.



- Convergência em E. Neste caso, para que a etapa 4 seja ativada, as etapas 2 e 3 devem estar ativas e a transição D deve ser satisfeita (Figura 4.12).

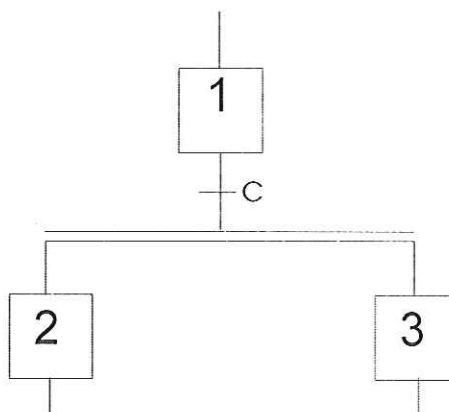


Figura 4.11 – Divergência em Y

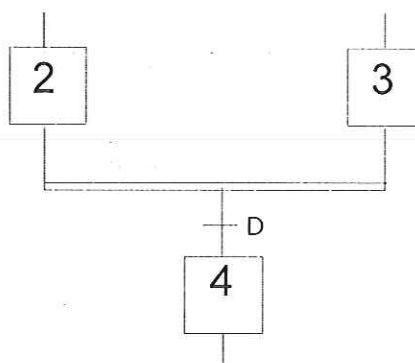


Figura 4.12 – Convergência em E

#### 4.1.5.5- Evoluções simultâneas

Esta regra de evolução permite decompor o GRAFCET em vários diagramas, especificando claramente suas interconexões. Neste caso, é indispensável à referência das etapas envolvidas. Um asterisco ao lado das transições é obrigatório para indicar a simultaneidade.

#### 4.1.6- Principais Estruturas

Além da seqüência simples, como exibido anteriormente, é possível representar estruturas com seqüências seletivas (ou alternativas) e seqüências simultâneas (ou paralelas).

Quando a ativação de uma transição condiz à ativação de várias seqüências de etapas ao mesmo tempo, se diz que são seqüências simultâneas. Depois da ativação de suas seqüências, as evoluções das etapas ativas em cada uma das seqüências são independentes. Para assegurar o sincronismo de desativação de várias seqüências ao mesmo tempo, geralmente se colocam etapas de espera recíprocas. O final de uma seqüência simultânea é sempre convergente, sendo que todas as etapas imediatamente precedentes à transição final devem estar ativadas para que esta seja habilitada (Figura 4.13).

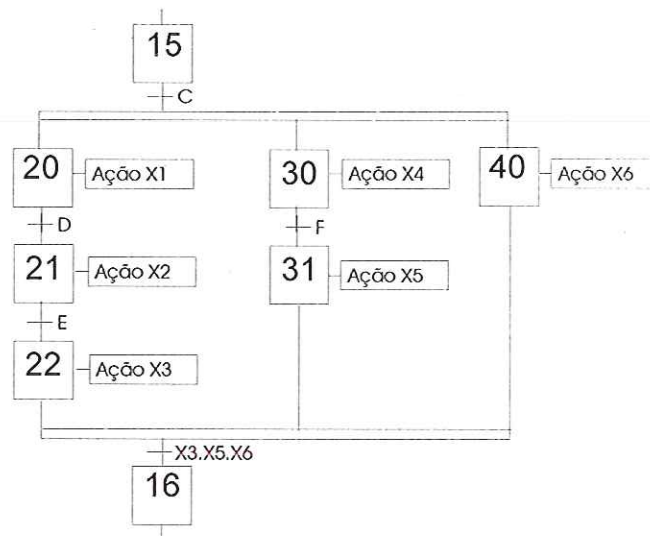


Figura 4.13 – Seqüências paralelas (simultâneas)

Outra representação muito utilizada é a de seqüências seletivas ou alternativas. Este tipo de seqüência é utilizado quando se faz necessário representar uma decisão, na qual é definida apenas uma seqüência a ser seguida.

As condições associadas às transições no início de cada seqüência devem ser exclusivas, ou seja, apenas uma condição deve ser satisfeita em determinado instante (Figura 4.14).

O início de uma seqüência seletiva é sempre divergente e o final convergente, no entanto não é necessário que todas as seqüências que compõem uma seqüência seletiva tenha fim, ou início, no mesmo ponto.

Uma seqüência de etapas (simples, seletiva ou simultânea) que apareça mais de uma vez no Diagrama Funcional e em situações distintas, pode ter a representação reutilizada (Figura 4.15).

Uma vez completado o GRAFCET do processo que se deseja controlar, o passo seguinte é a obtenção das condições de ativação das etapas, assim como as ações associadas às mesmas. Para isso se utiliza um processo de normalização com a qual, e partindo do GRAFCET realizado, obtém-se as condições de ativação para cada uma das etapas e ações associadas. A obtenção destas condições booleanas se baseará na utilização do sinal proveniente da etapa anterior:

- uma etapa se ativará quando, estando ativa a etapa imediatamente anterior a ela, evolue devido à validação existente entre estas duas etapas, desativando a etapa anterior e ativando a nova etapa; e
- uma ação deve ser executada se a etapa a que ela está associada se tornar ativa.

Uma vez obtidas as condições de validação, o passo seguinte é a implementação em uma linguagem apropriada para o controlador escolhido como unidade de controle.

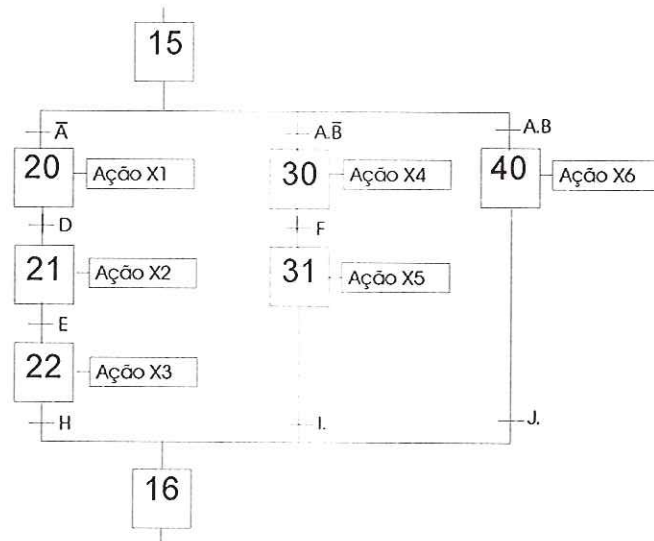


Figura 4.14 – Seqüência Alternativa (Seletiva)

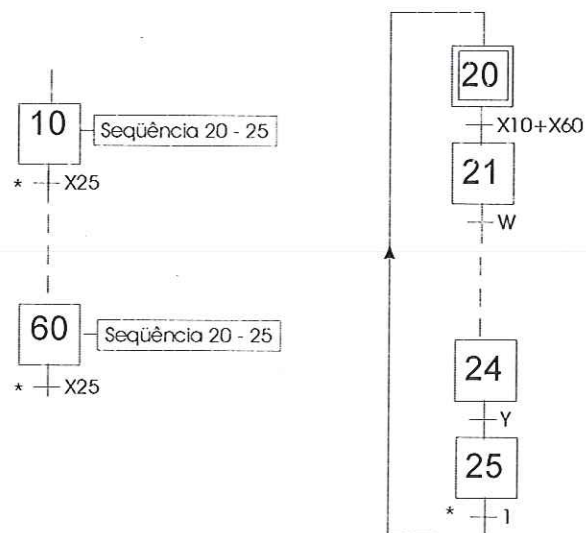


Figura 4.15 – Reutilização de uma seqüência

## 4.2- DISPOSITIVO LÓGICO PROGRAMÁVEL FPGA

### 4.2.1- Introdução

A demanda cada vez maior por poder de processamento e o surgimento de novas aplicações promovem uma constante busca por alternativas e

arquiteturas que visem melhorar a performance dos computadores, especialmente em aplicações de tempo real (TAYLOR e VARGA).

Uma alternativa que vem sendo cada vez mais empregada para se conseguir taxas de processamento em tempo real é o processamento paralelo e uma alternativa é o processamento em hardware. Entre outros fatores, o que possibilita ganho de performance em hardware é o fato de que este pode tirar vantagem do paralelismo da aplicação enquanto elimina o *overhead* gerado pelas operações de busca, decodificação e execução de instruções, bem como desvios condicionais ou não (URL [www.vcc.com](http://www.vcc.com)) (URL [www.reconfig.com](http://www.reconfig.com)). O hardware implementa algoritmos sob a forma de máquinas de estado, funções lógicas e *pipelines* diretamente (TAYLOR, 1996). Um outro fator que aumenta a performance do sistema é que quando uma etapa de processamento é feita em hardware, a CPU fica livre para realizar outras tarefas, como a aquisição de dados ou atualização da interface gráfica.

Os projetistas de circuitos integrados freqüentemente se deparam com o problema do correto balanceamento entre performance e generalidade dos chips (VILLASENO, 1997). Os microprocessadores, por natureza, são elementos destinados a executar processamento genérico. Sua arquitetura interna é totalmente dedicada a implementar a funcionalidade de busca, decodificação e execução de instruções, armazenadas em algum tipo de memória. A possibilidade de reprogramação é sua característica principal, pois o seu comportamento é governado pelas instruções que executa. Devido à sua generalidade consegue baixa performance se comparado a um hardware dedicado para o mesmo processamento.

Até alguns anos atrás, a alternativa existente para se conseguir alto índice de performance era a construção de circuitos integrados dedicados para aplicações específicas, os chamados ASICs (*Application Specific Integrated Circuits*). Como exemplo pode-se citar: os chips aceleradores gráficos, que são capazes de executar operações como desenhar linhas ou mover blocos de memória de vídeo com velocidades 10 a 100 vezes maiores que aquelas conseguidas com microprocessadores; co-processadores matemáticos; co-processadores de rede; entre outros. Nestes casos, a arquitetura interna dos dispositivos é otimizada e voltada para a aplicação em questão, conseguindo-se, então, uma performance superior à que seria conseguida com a arquitetura

de um microprocessador tradicional. A característica marcante destes dispositivos é a sua imutabilidade, ou seja, as funções que executam estão amarradas ao próprio hardware e, uma vez construídos, não podem ser reprogramados.

Com o desenvolvimento dos circuitos integrados, surgiu uma terceira opção: os dispositivos lógicos programáveis. Com eles, consegue-se combinar a característica de alta performance do hardware com a generalidade e reprogramabilidade dos microprocessadores. Dispositivos mais modernos como as FPGA's (*Field Programmable Gate Arrays*) (XILINX) ( BROWN, 1996), são capazes de serem re-programadas no próprio circuito, em frações de segundo, o que gerou um novo paradigma de computação: a Computação Reconfigurável (VILLASENO, 1997) (URL [www.reconfig.com](http://www.reconfig.com)). Desta forma, a própria arquitetura interna do chip pode ser alterada, dinamicamente, para adaptar-se a uma dada aplicação.

Existem diversos tipos de dispositivos programáveis capazes de implementar funções lógicas (URL [www.vcc.com](http://www.vcc.com)): EPROM (*Erasable Programmable Read Only Memory*), PLA (*Programmable Logic Array*), PAL e etc. Devido à necessidade de funções mais complexas, surgiram os dispositivos conhecidos como CPLD's (*Complex Programmable Logic Devices*). Alguns membros dessa família de dispositivos são a MPGA (*Mask Programmable Gate Array*) e FPGA (*Field Programmable Gate Array*)(Xilinx). Nas MPGA's, a especificação das funções lógicas deve ser feita antes do processo de fabricação do circuito integrado, e portanto introduz um alto custo, que só é compensado por produção em alta escala, e um longo ciclo desenvolvimento→fabricação (CHAN, 1994) (URL [www.reconfig.com](http://www.reconfig.com)). As FPGA's são dispositivos reprogramáveis em campo, ou seja, podem ter sua configuração alterada sem que ela seja retirada do circuito. A capacidade destes dispositivos hoje chega à casa das dezenas de milhares de portas lógicas. Programas poderosos de CAD (*Computer Aided Design*) auxiliam o projetista na criação de aplicações com FPGA's, assim como outros dispositivos lógicos programáveis.

### 4.2.2- FPGA

FPGA (Field Programmable Gate Array) são circuitos programáveis compostos por um conjunto de células lógicas ou blocos lógicos alocados em forma de uma matriz. Em geral, a funcionalidade destes blocos, assim como seu roteamento é configurável por *software*. A palavra *field* indica que a configuração do circuito pode ser feita pelo usuário final sem a necessidade da utilização de recursos de fabricação. Na figura 4.16 podemos observar a estrutura de uma FPGA.

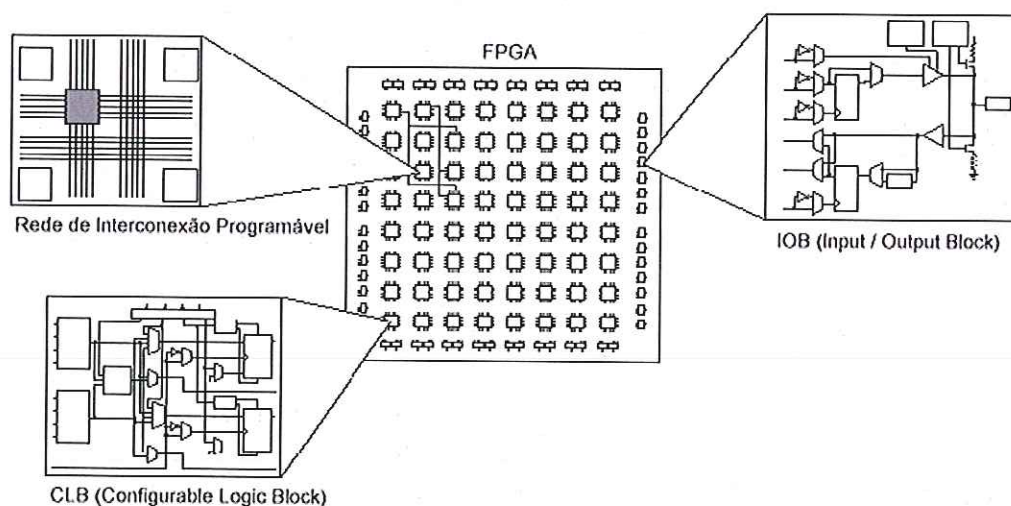


Figura 4.16 – Estrutura de uma FPGA (TERROSO, 1998)

#### 4.2.2.1- Definição de Blocos Lógicos

As funções lógicas são implementadas no interior dos Blocos Lógicos. Em algumas arquiteturas os Blocos Lógicos possuem recursos seqüenciais tais como flip-flop ou registradores. O fabricante Xilins chama seu Bloco Lógico de CLB (Configurable Logic Block), enquanto que a Actel usa o termo LM (Logic Modules), e a Altera o LE (Logic Element) para as séries 8000 e 10000.

#### 4.2.2.2 – Definição de Roteamento

A interconexão entre os blocos é feita através de uma rede de duas camadas de metal. As conexões físicas entre os fios são feitas ora com transistores de passagem controlados por bits de memória (PIP) ora com chaves de interconexão (Matriz de Chaves – Switch Matrix).

Os recursos de roteamento da série XC3000 da Xilinx possuem:

- Conexões Globais que formam uma rede de interconexão em linha e colunas de cinco fios de metal cada, ligados através de chaves. Esta rede circunda os blocos lógicos e os blocos de E/S (Figura 4.17);
- Matrizes de Conexão que são chaves que permitem o roteamento entre os blocos lógicos através das conexões globais programável pelo *software* do fabricante ou manualmente com uma ferramenta chamada EDINET;
- Conexões Diretas entre CLB's vizinhos que permitem interligar blocos com menor atraso, pois não utilizam os recursos globais de roteamento (Figura 4.17);
- Linhas Longas atravessam todo o circuito sem passar pelas matrizes de chaves e são utilizadas para conectar sinais longos entre múltiplos destinos (Figura 4.17).



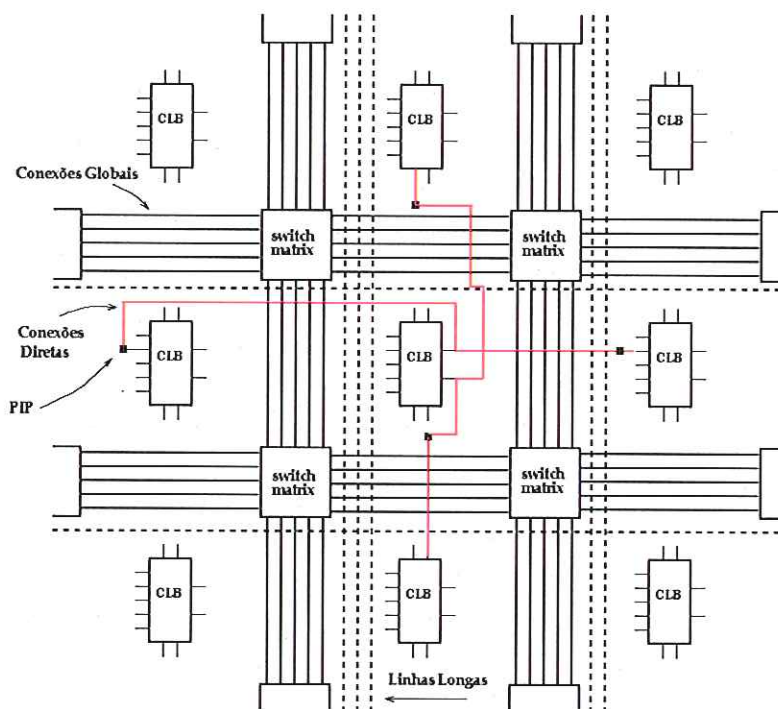


Figura 4.17 – Roteamento de uma FPGA (TERROSO, 1998)

#### 4.2.2.3- Definição de Bloco de E/S

Os blocos de E/S possuem um amplificador e um flip-flop para os sinais de saída. O sinal de saída pode ser invertido por programa, assim como o sinal de controle do *tristate* (Figura 4.18).

### 4.2.3 – Vantagens da Utilização do FPGA

Devido às suas características, as FPGA's possuem dois grandes campos de utilização: o primeiro é no rápido desenvolvimento de protótipos de circuitos (*Rapid Prototyping*) (CHAN, 1994). A possibilidade de reprogramação no próprio circuito acarreta uma diminuição significativa no tempo de desenvolvimento devido à agilidade no processo de simulação → teste →

depuração → alteração do projeto. As ferramentas de CAD permitem, por exemplo, que o projetista

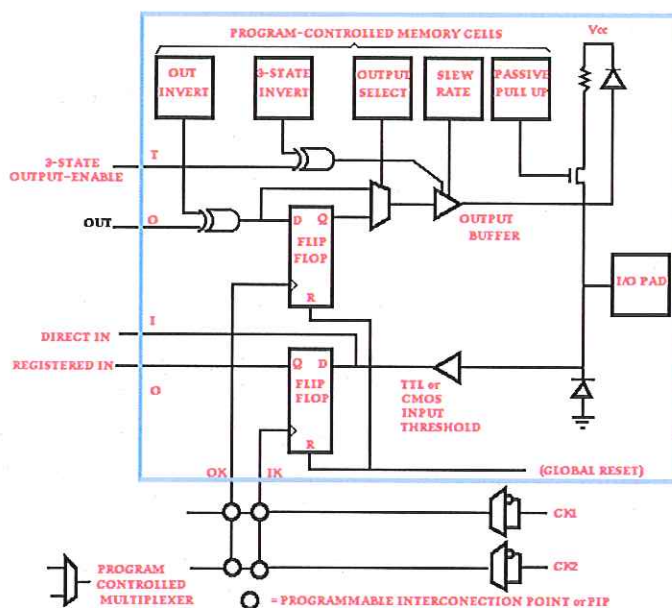


Figura 4.18 – Representação de um Bloco de E/s da família XC3000 da XILINX (TERROSO, 1998)

especifique o comportamento da FPGA através de diagramas elétricos e/ou linguagens de descrição de *hardware* (HDL - *Hardware Description Language*), como ABEL ou VHDL. O compilador produz então um arquivo de configuração que, quando carregado na FPGA, faz com que ela se comporte como o circuito projetado. O projetista tem também a possibilidade de verificar o funcionamento do circuito através de simulação funcional e temporal, já considerando os tempos de atraso gerados pela lógica resultante do processo de compilação.

O segundo campo de utilização, que vem crescendo cada vez mais, é a Computação Reconfigurável (*Reconfigurable Computing Systems*) (URL [www.vcc.com](http://www.vcc.com)) e Objetos de Hardware (*Hardware Objects*)(URK [www.vcc.com](http://www.vcc.com)). A computação reconfigurável representa sistemas em que a arquitetura interna pode ser modificada por software em tempo real, para se adaptar a uma aplicação específica (BUELL, 1996). Ela é conseguida na prática através de placas que se conectam a computadores (PC's, Estações de trabalhos e etc.) e

que possuem FPGA's atuando como co-processadores re-programáveis (RPU - *Reprogrammable Processing Unit*). Com isso, o programa do usuário pode dar um *download* de um projeto eletrônico diretamente para a RPU de forma que ele tenha um hardware específico para a sua aplicação. Este processo também pode ser feito dinamicamente, à medida que o usuário necessitar de diferentes características funcionais.

A implementação de algoritmos em hardware é particularmente eficiente para aplicações orientadas a bit (*bit oriented*), ou seja, que trabalham com manipulação direta de bits (Xilinx). Como exemplos de tais aplicações podemos citar compressão de dados, reconhecimento de padrões, criptografia, tratamento de imagens, processamento de sinais, aritmética de inteiros (valor absoluto, máximo, mínimo e comparação), entre outros.

#### 4.3- LINGUAGEM DE DESCRIÇÃO DE HARDWARE (VHDL)

VHDL é uma forma de descrever, através de um programa, o comportamento de um circuito ou componente digital.

**Very High Speed Integrated Circuit**

**Hardware**

**Description**

**Language**

Esta linguagem descreve o que um sistema faz e como. Esta descrição é um modelo do sistema hardware que será executado por um *software* denominado simulador e implementado em um dispositivo programável (FPGA), permitindo assim o uso em campo do sistema, tendo a grande vantagem da alteração do código a qualquer momento pelo usuário sem a necessidade de dispositivos ou equipamentos especiais ligados à fabricação do dispositivo.

### 4.3.1- Vantagens e Desvantagens do VHDL

A Descrição de um sistema em VHDL apresenta inúmeras vantagens, tais como:

- Intercâmbio de projetos entre grupos de pesquisa sem a necessidade de alteração;
- Permite ao projetista considerar no seu projeto os atrasos comuns aos circuitos digitais;
- A linguagem independe da tecnologia atual, ou seja, pode-se desenvolver um sistema hoje e implementá-lo depois;
- Os projetos são fáceis de serem modificados;
- O custo de produção de um circuito dedicado é elevado, enquanto que usando VHDL e Dispositivos Programáveis isto passa a ser muito menor; e
- Reduz consideravelmente o tempo de projeto e implementação.

Quanto às desvantagens, a mais relevante é que o VHDL não gera hardware otimizado.

## **5. METODOLOGIA**

### **5.1- INTRODUÇÃO**

Optamos por automatizar um dispositivo posicionador para câmara fotográfica que será acoplada a uma cadeira de rodas, uma vez que este dispositivo atende às expectativas do presente trabalho.

A ferramenta GRAFCET nos permite visualizar outras possibilidades de utilização do mesmo dispositivo, com pequenas modificações, a fim de se atender a outras necessidades, visto que o princípio de funcionamento e acionamento são semelhantes, como por exemplo a automatização para controle de uma cadeira de rodas.

O FPGA pela sua versatilidade e facilidades na gravação de novos processos se mostra como o principal componente a ser empregado nesta automatização, sem este componente a concepção seria baseada em circuitos lógicos ou micro-processadores.

A fim de podermos comprovar as possíveis utilizações e verificar a robustez do dispositivo, fabricamos um painel composto por chaves e LEDs que simulam sensores, dispositivos de entradas e dispositivos de saídas.

## 5.2- CONCEPÇÃO DO DISPOSITIVO ASSISTIVO

A altíssima capacidade de integração e grande quantidade de conexões para entrada e saída, aliado ao seu baixo custo de aquisição e as características apresentadas anteriormente, elegem o FPGA como componente central para a concepção de nosso dispositivo assistivo.

Interfaces de entrada e saída entre o mundo real e o FPGA devem ser implementadas, a fim de adequar as tensões e correntes fornecidas e drenadas pelos sensores e circuitos envolvidos, às características elétricas suportadas pelo FPGA.

Todo o tratamento dos sinais de entrada ou saída, quais sejam: conversão A/D D/A, tratamento matemático, medida de largura de pulso, etc. poderão ser implementados através da programação do FPGA, tornando desta forma os circuitos de entrada ou saída extremamente simples.

A figura 5.1 apresenta o diagrama de blocos de entrada e saída do autômato que desenvolvido para o dispositivo assistivo, a implementação é apresentada de forma destacada, sendo as outras opções as outras possibilidades.

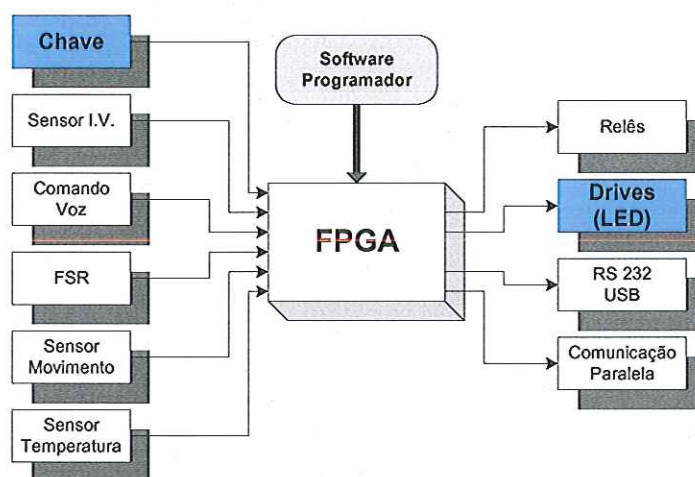


Figura 5.1 – Diagrama de blocos de entrada e saída de um dispositivo assistivo

### 5.2.1- Conversor GRAFCET → Circuito Digital

As facilidades oferecidas pelo GRAFCET quanto à representação de sistemas complexos automatizados se contrapõem à necessidade de transformar esta representação em um circuito digital equivalente, quando aplicado a automações eletrônicas.

O processo repetitivo sugere a possibilidade de implementação de um sistema informatizado que possibilite desenvolver tal tarefa, buscando desta forma, não só agilidade na implementação, como também segurança quanto a falhas motivadas por erros de tradução e/ou interpretação.

Desenvolvemos o Sistema Grafcet-Circuito Digital (SGC), uma ferramenta computacional, que permite transpor o diagrama grafcet em circuito digital utilizando análise essencial e linguagem de programação Delphi.

Na tela de abertura (Figura 5.2) observamos as várias possibilidades de componentes disponíveis para a construção do esquema GRAFCET, com as opções iniciais da paleta "Arquivo".

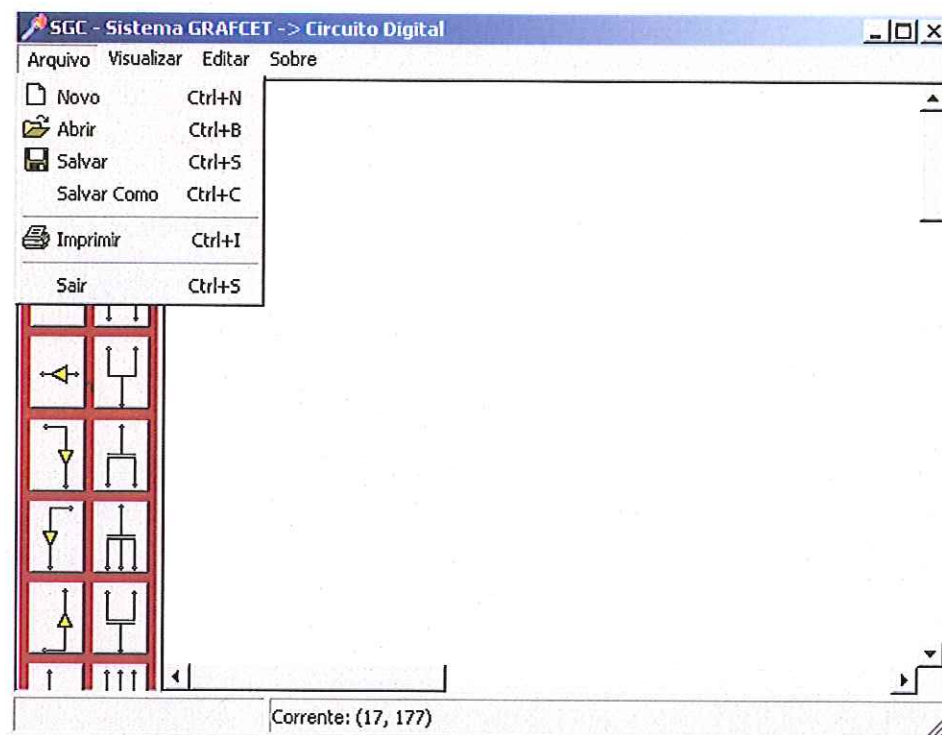


Figura 5.2 – Tela de Abertura Sistema GRAFCET → Circuito Digital

Uma vez que cada componente, passível de implementação no esquema GRAFCET, possui características próprias devemos de alguma forma contemplar seus parâmetros através de informações disponibilizadas pelo usuário, a fim de que quando da passagem para o circuito digital tenhamos todas as informações pertinentes a cada elemento. Para o componente “AÇÃO” temos a solicitação obrigatória de uma descrição, sendo esta uma regra definida pelo próprio método GRAFCET. Caso o componente “ESTADO” seja o primeiro do diagrama o sistema automaticamente irá desenhá-lo com características diferenciadas, possibilitando a qualquer instante que se obtenha uma localização rápida do local onde se inicia o processo (Figura 5.3 e 5.4).



**Descrição**

**O objeto incluído exige descrição, faça isto agora.**

Tipo: **Ação**                      Identificação: **1**

Descrição:

Figura 5.3 – Tela de especificação de “AÇÃO”

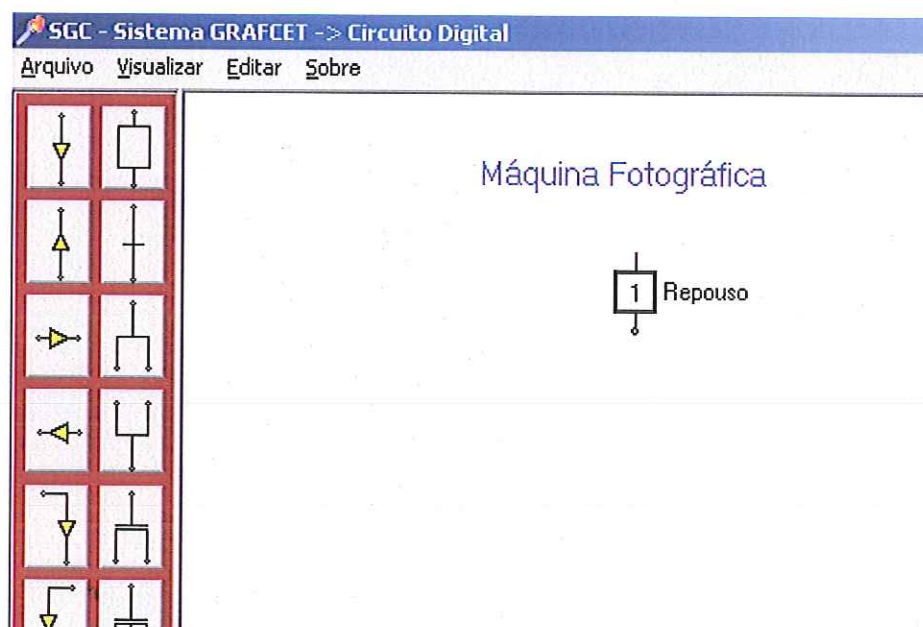


Figura 5.4 – Estado especificado

Uma vez que o estado está devidamente identificado, ele é posicionado na página do GRAFCET para conexão com o próximo componente. A colocação de componentes no diagrama GRAFCET segue uma ordem ao mesmo tempo rígida e cronológica, de tal forma que após um componente “ESTADO” obrigatoriamente virá um de “TRANSIÇÃO” (Figura 5.5).

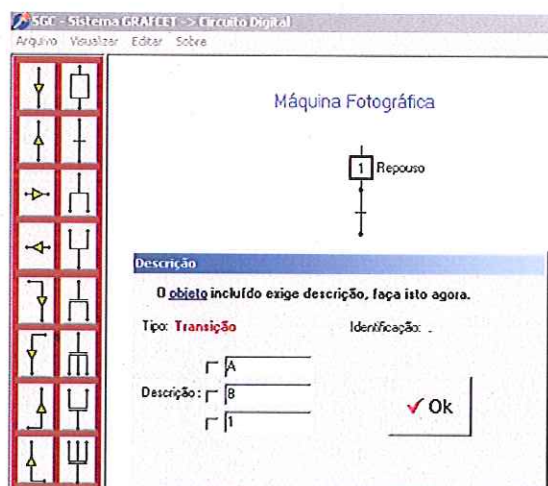


Figura 5.5 – Tela de especificação de TRANSIÇÃO

Para o componente “TRANSIÇÃO”, as informações solicitadas definem que condições devem ser atendidas para promover o acionamento ou liberação da próxima “AÇÃO”. Quando houver necessidade de negação de alguma variável do componente “TRANSIÇÃO”, a mesma deverá ser feita através do apontamento do quadro específico ao lado da referida variável. O sistema está preparado para permitir até três sinais ou variáveis, que combinados liberam a transição, sendo que os sinais nas entradas não utilizados devem ser anulados através de sua ativação lógica (valor lógico 1) (Figura 5.6).

O processo de inclusão de novas AÇÕES e TRANSIÇÕES é simples e pouco exige do usuário, sendo possível ainda, quando houver necessidade, a inclusão de processos que serão executados paralelamente ou

processos exclusivos que dependem da combinação de ações. Para esta última opção são oferecidos dois tipos de blocos, com duas ou três entradas, havendo ainda a possibilidade de combinações entre eles (Figura 5.7).

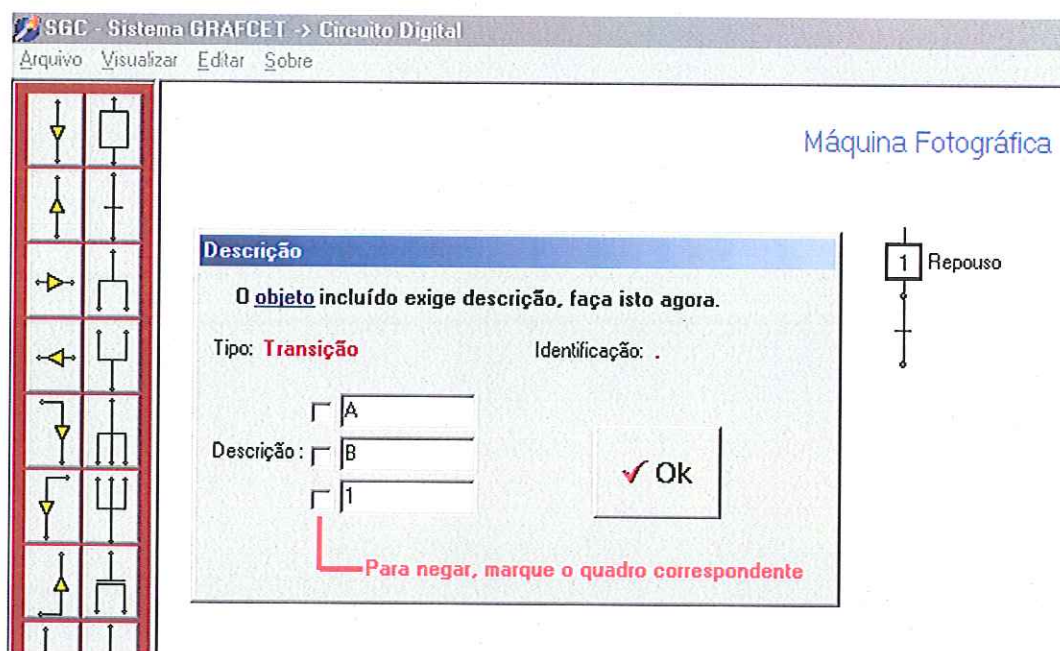


Figura 5.6 – Tela de especificação da “TRANSIÇÃO”

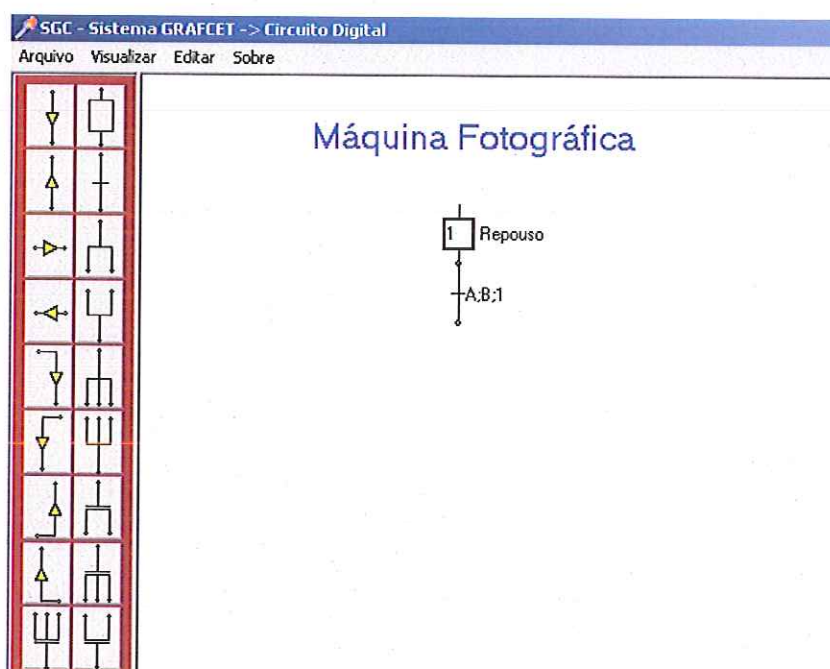


Figura 5.7 – Transição especificada e posicionada. Componentes disponíveis.

Uma vez que o GRAFCET esteja completo, o sistema disponibiliza, a qualquer tempo o circuito digital equivalente.

A passagem do esquema desenvolvido em GRAFCET para seu equivalente digital é simples e automática. Durante a concepção do esquema GRAFCET, as informações, dos dois possíveis esquemas, são armazenadas em arquivos separados para eventual utilização futura.

Basicamente o esquema digital possui três blocos fundamentais e distintos (Fig. 5.8):

- 1- Memória, cujo funcionamento se iguala ao de um Flip-Flop;
- 2- Porta lógica "E"; e
- 3- Porta lógica "OU".

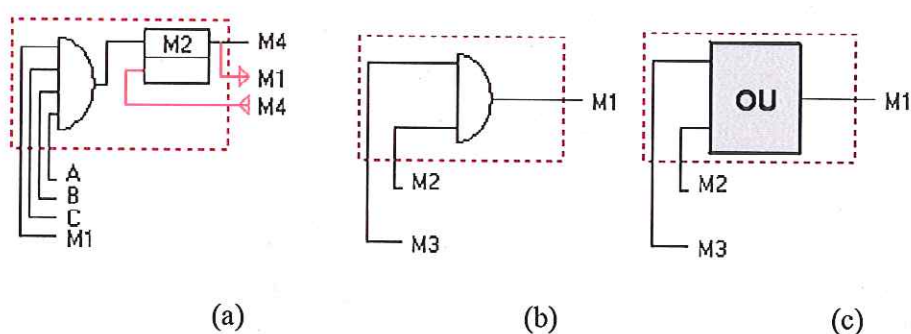


Figura 5.8 – Blocos básicos do Sistema Grafcet→Circuito Digital: (a)- Memória; (b)- Porta E; (c)- Porta OU

#### 5.2.1.6- Software de edição VHDL

Para a gravação do FPGA utilizamos o software **ISE WebPACK™ 6.2i** da XILINX, por ser esta a ferramenta que acompanha o componente, sendo que esta é uma versão gratuita e disponível na página da empresa, porém apenas alguns produtos da família XILINX são suportados por esta versão.

Esta ferramenta possui uma IDE (Ambiente de Desenvolvimento Integrado), completo, incluindo as funções de edição, compilação e depuração do código VHDL, aliados a um poderoso editor e conversor de esquemas digitais. O **ISE WebPACK™ 6.2i** permite configurar um FPGA de duas formas:

- 1- Através do fornecimento da descrição de hardware do circuito com a utilização da linguagem VHDL propriamente dita; e
- 2- Através da edição do esquema digital do circuito.

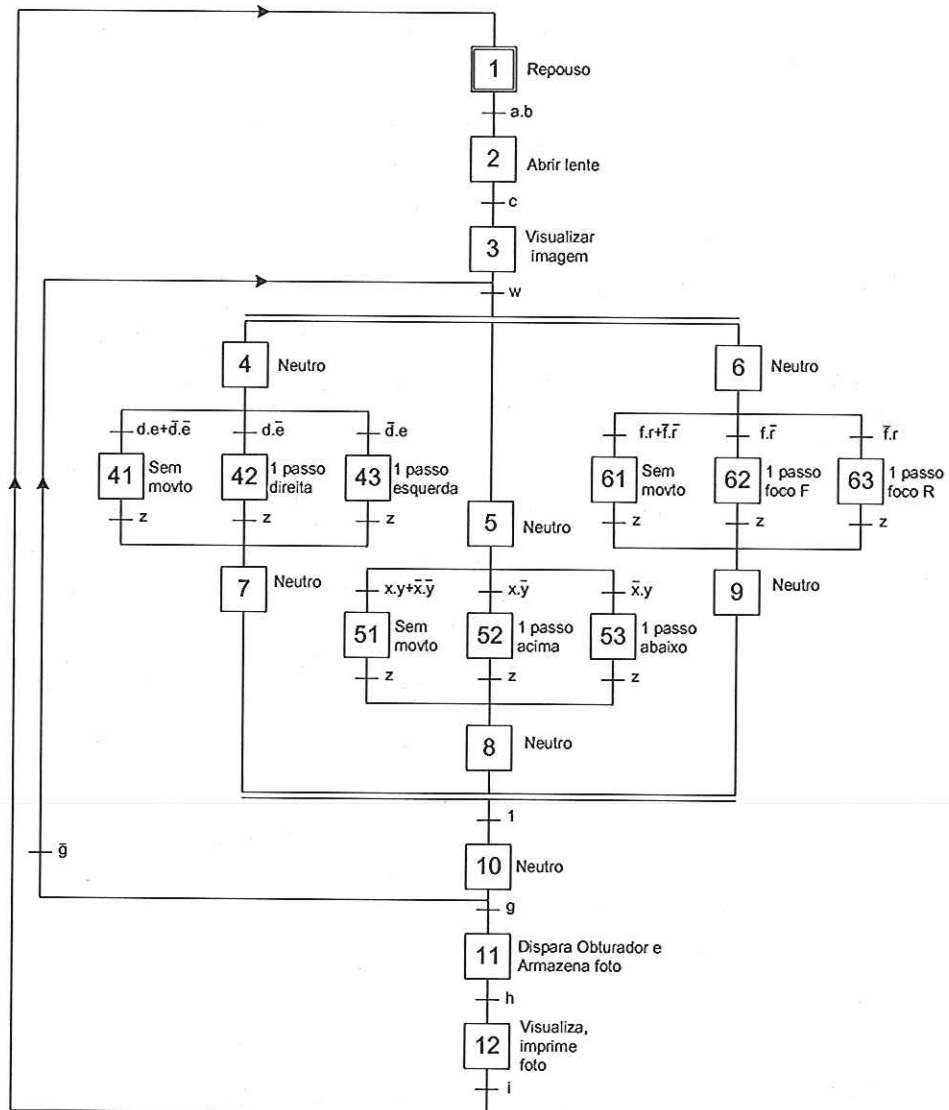
O ISE *WebPACK*™ 6.2i possui ainda as rotinas de gravação de dispositivo, permitindo que o FPGA tenha sua concepção realizada via interface JTAG/Porta Paralela para gravação e depuração.

## 6. RESULTADOS

Automatizamos o dispositivo controlador do equipamento fotográfico que atenderá um deficiente motor que sofre de tetraplegia. O dispositivo permite o posicionamento correto da câmera fotográfica, através de movimentos para frente e para trás, controle de foco e zoom aproximando ou distanciando o ponto do objeto focado, e finalmente promover o disparo do obturador.

Como passo inicial preparamos o esquema GRAFCET em na ferramenta gráfica Visio © 2000, que oferece facilidades que em outra ferramenta de desenho não encontraríamos (Fig. 6.1), tais como blocos predefinidos, facilidades de correções e movimentações de blocos, sendo este esquema, em seguida, lançado este esquema no Sistema Conversor GRAFCET → Circuito Digital (SGD) que foi desenvolvido para este trabalho e possibilita rápida transformação do esquema GRAFCET (Fig. 6.2) em esquema digital.

## GRAF CET Máquina Fotográfica Controle Paralelo



### Sensores posicionados nos atuadores

- a- Máquina acoplada
- c- Lente aberta
- z- Movimento realizado
- h- Foto armazenada
- i- Visualização/Impressão OK

### Sensores acionados pelo usuário

- b- Inicia processo
- d- } Controle de posicionamento para a direita e para a esquerda
- e- }
- x- } Controle de posicionamento para cima e para baixo
- y- }
- f- } Controle de posicionamento foco a frente e foco a ré
- r- }
- g- Disparo do obturador

Figura 6.1- Esquema GRAFCET desenhado em ferramenta gráfica

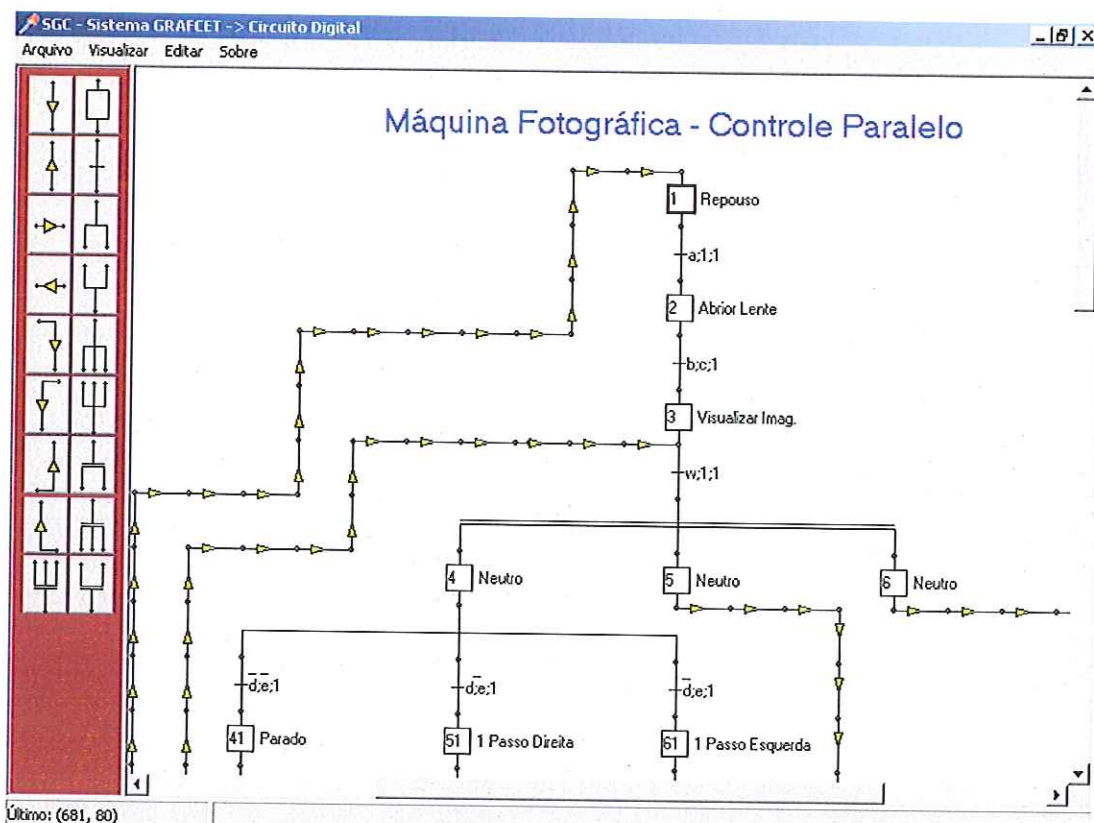


Figura 6.2- Esquema GRAFCET da Máquina Fotográfica com controle paralelo

Uma vez que o esquema GRAFCET tenha sido preparado e lançado no SGD, obtivemos o circuito digital mostrado na figura 6.3 e verificamos a existência ou não de possíveis erros utilizando o Electronics Workbench Pro (Fig. 6.4). As várias simulações realizadas com esta ferramenta possibilitaram confirmar a eficiência do sistema SGD no que se refere à fidelidade de transformação de esquema GRAFCET em esquema digital.

A próxima etapa consistiu em desenhar o circuito digital, validado anteriormente, preparando para a gravação do FPGA. Para esta etapa lançamos mão da ferramenta Xilinx ECS que é um editor de esquema elétrico (Fig. 6.5). Em seguida, ainda com ferramenta da Xilinx, o circuito elétrico foi compilado gerando um programa VHDL, que então foi gravado no FPGA.



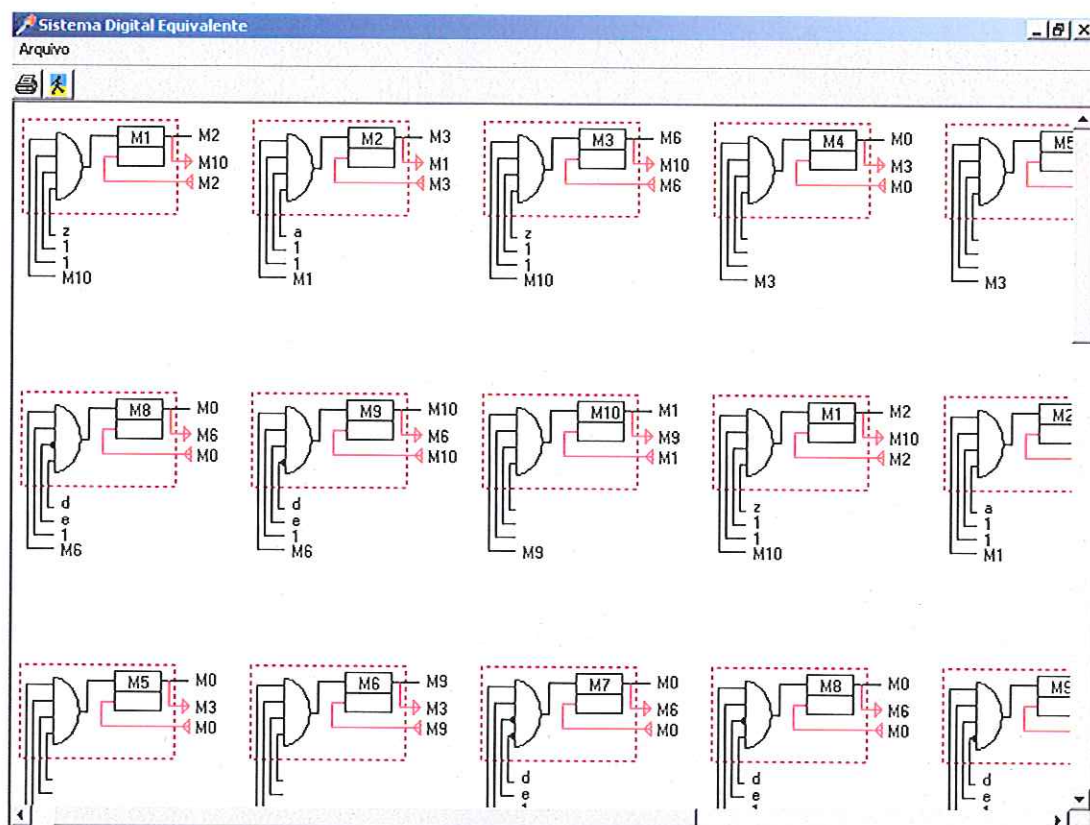


Figura 6.3- Esquema Digital equivalente ao Esquema GRAFCET (Máquina Fotográfica com controle paralelo)

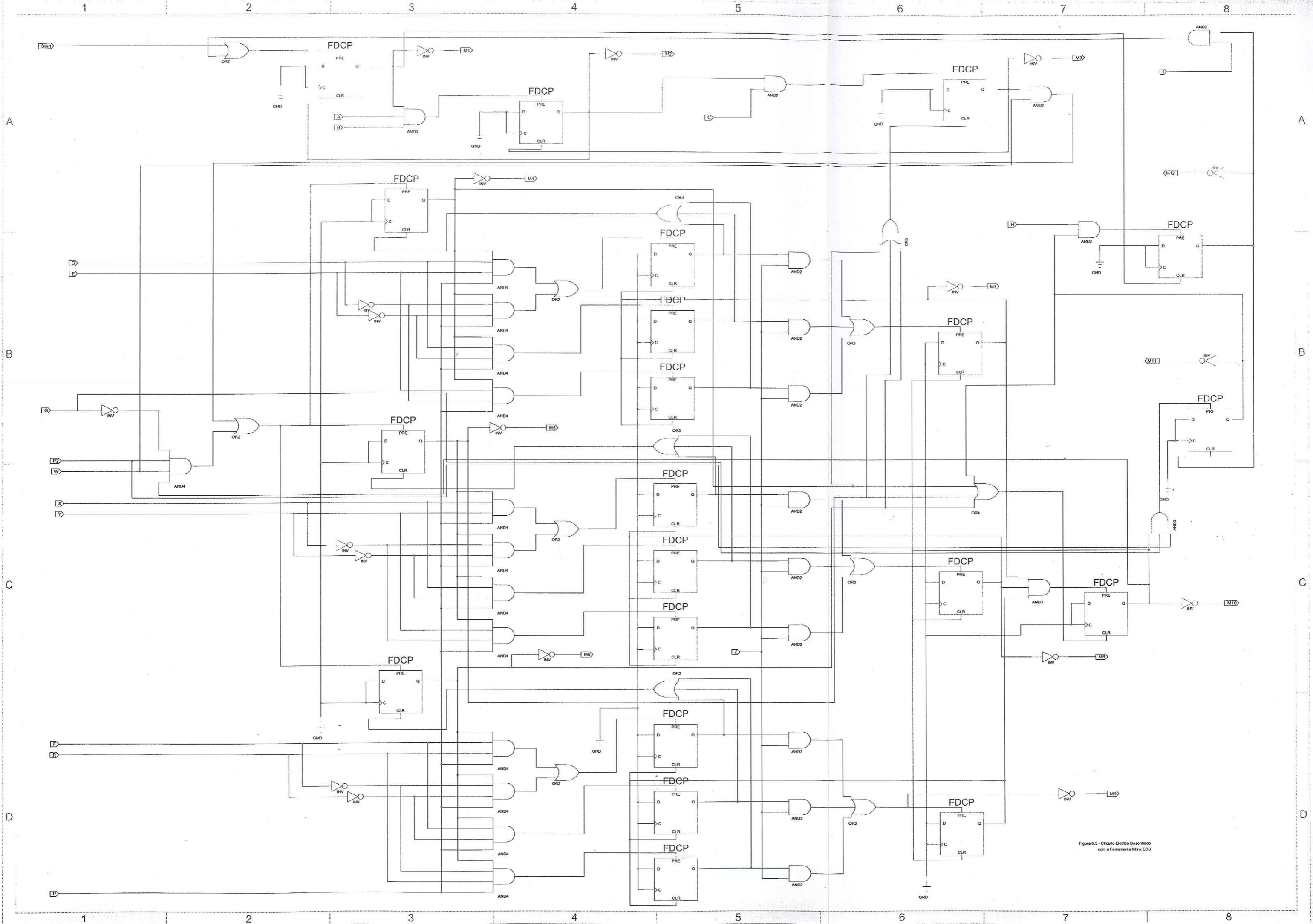


Figura 6.5 - Circuito Elétrico Desenhado com a Ferramenta Xilinx ECS

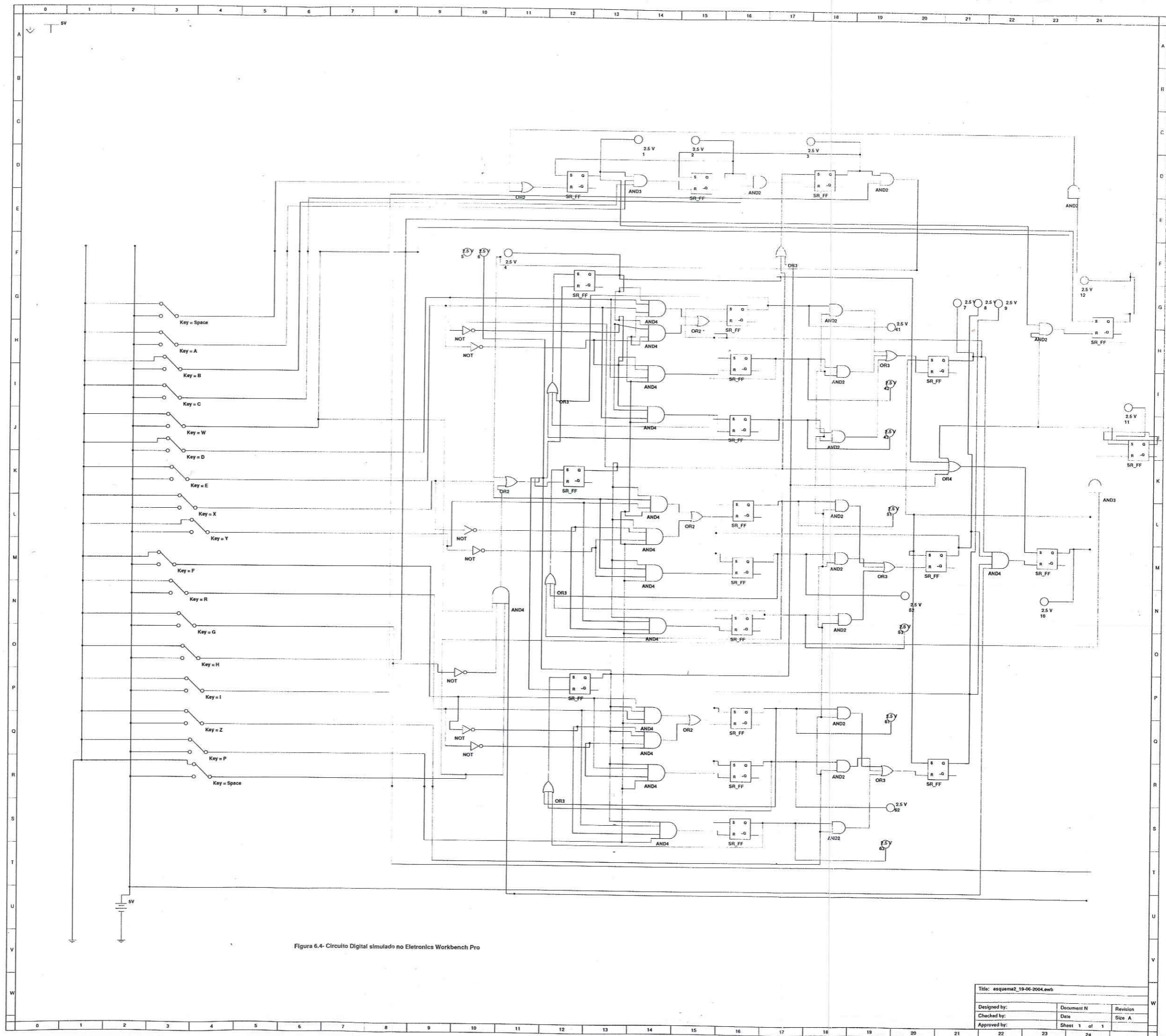


Figura 6.4- Circuito Digital simulado no Electronics Workbench Pro

Title: esquema2_19-06-2004.awb		
Designed by:	Document N:	Revision
Checked by:	Date	Size A
Approved by:	Sheet 1 of 1	

Para implementação utilizamos um kit de desenvolvimento modelo AEE-IFC de fabricação da AEE Engenharia Eletrônica, que conta com um FPGA modelo XC9572, montado sobre CPLD Prototype Board AEE 1301 R1.0 e AEE 1201 R1.0.

A placa AEE 1201 funciona exatamente como o cabo de configuração da Xilinx e possui interface JTAG e Slave Serial. Além disso, possui um regulador de tensão ajustável para 5V ou 3,3V que serve para alimentar as demais placas do kit ou mesmo para uso geral na bancada.

A placa AEE 1301 é para prototipagem com CPLDs. Possui dois conectores de 20 pinos que permitem acesso a todos os sinais de I/O do CPLD e das alimentações. As alimentações podem ser escolhidas por estrapes originando do conector de alimentação/Programação ou de um conector extra. Existe um clip oscilador, além de uma tecla e um LED para testes do usuário.

Para o teste final do dispositivo assistivo confeccionamos uma interface composta de chaves, para simulação de entradas dos sinais digitais (0 ou 1), e LED's para indicar o estado ativo no esquema GRAFCET (Fig. 6.6), que é controlada pelo FPGA disposto na placa AEE1301, alimentada pela placa AEE1201 (Fig. 6.7).

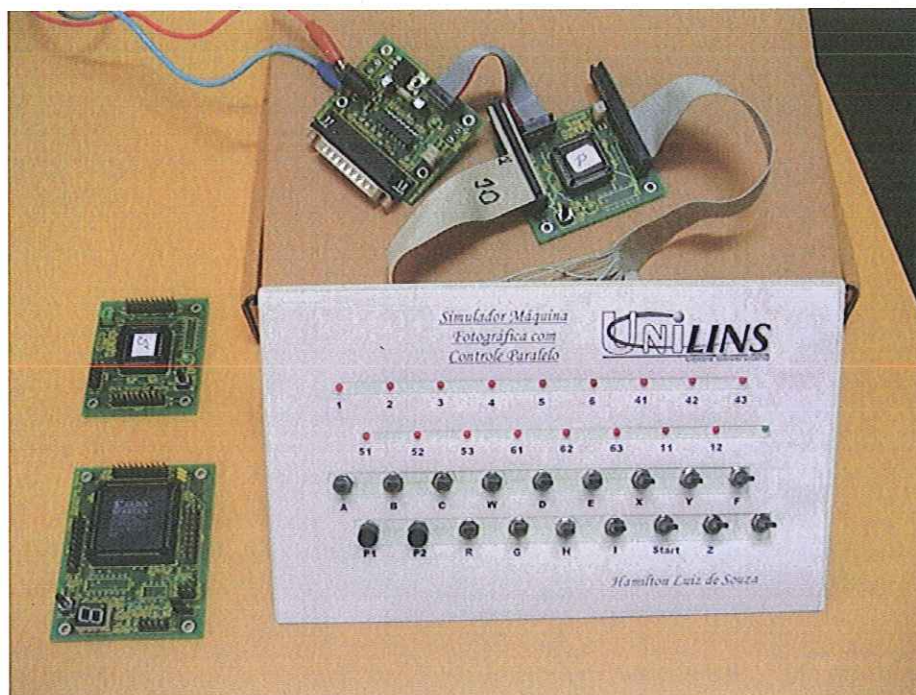


Figura 6.6- Dispositivo simulador

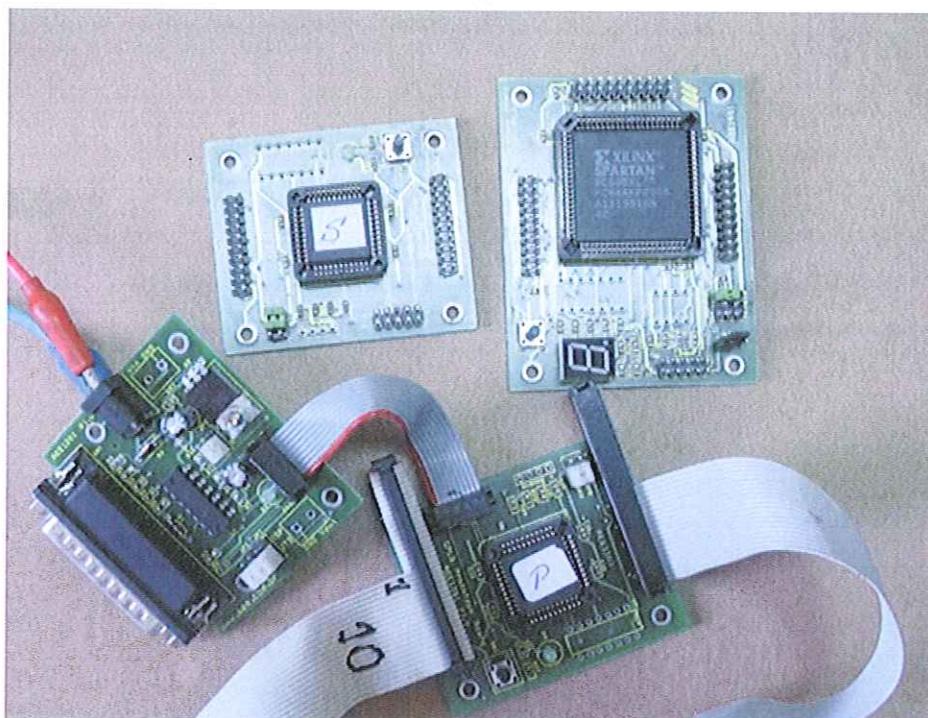


Figura 6.7- Placas AEE 1201 e AEE 1301, fonte controladora e placa de prototipação, respectivamente

O processo tem início com o acionamento da chave “START” que é indicado pelo “LED 1” aceso (Fig. 6.8).

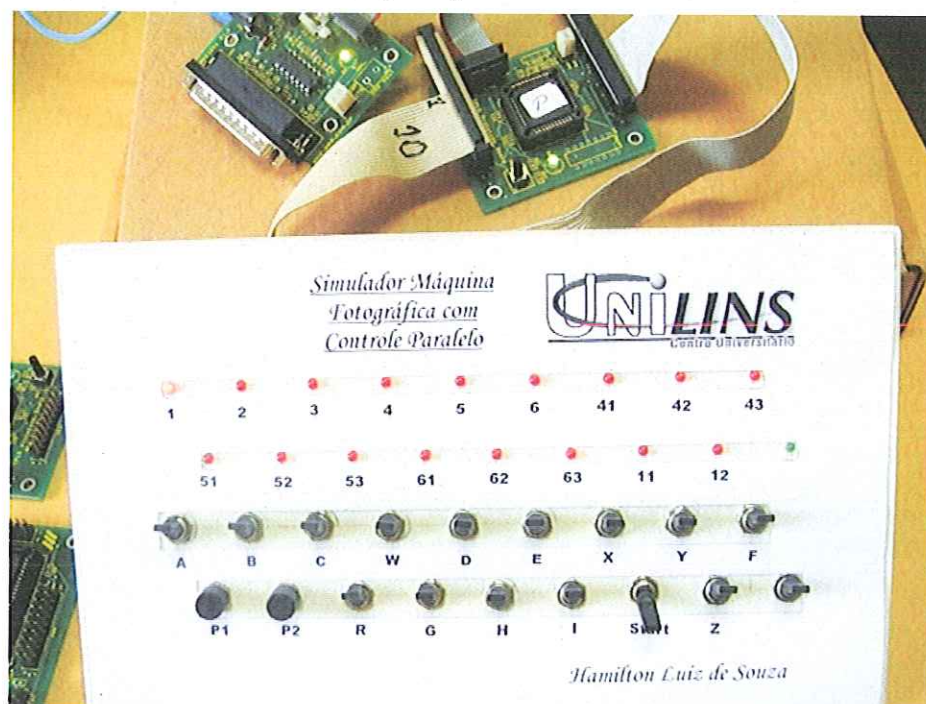


Figura 6.8- Chave “START” e LED “1” ativos indicando início do processo

Com a máquina acoplada à cadeira, chave “A” ativa, e o sensor indicado pela chave “B” tendo indicado que o processo foi iniciado, temos a abertura da lente que é representado pelo LED “2” (Fig. 6.9).

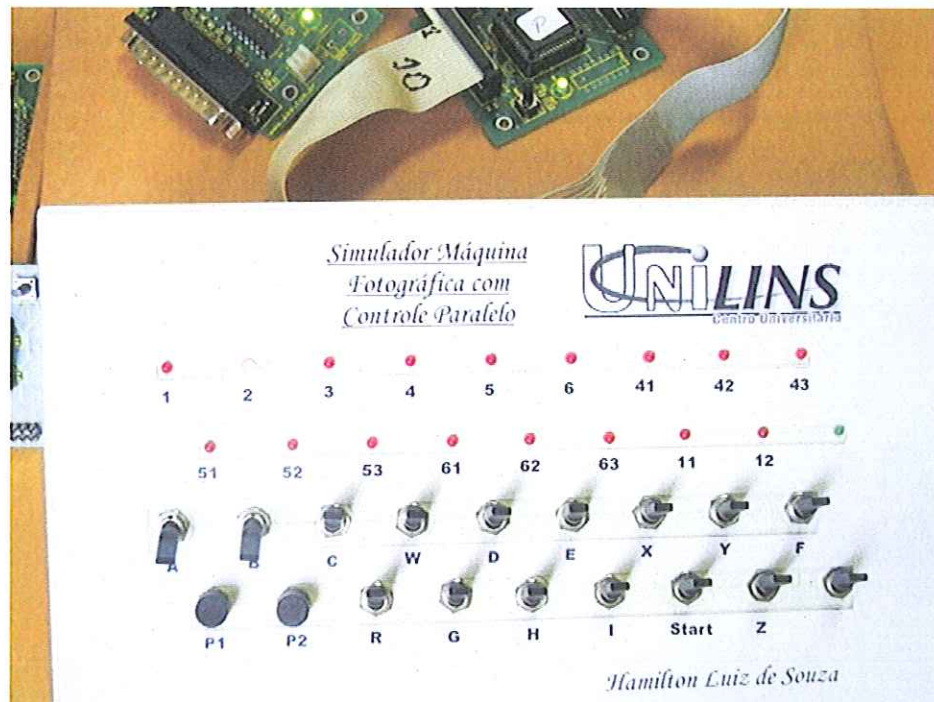


Figura 6.9- Chaves “A” e “B” ativas e LED “2” aceso, indicando que o processo “abrir lente” foi ativado

O sensor representado pela chave “C” indica que a lente foi aberta e passamos a visualizar a imagem, LED “3” (Fig. 6.10).

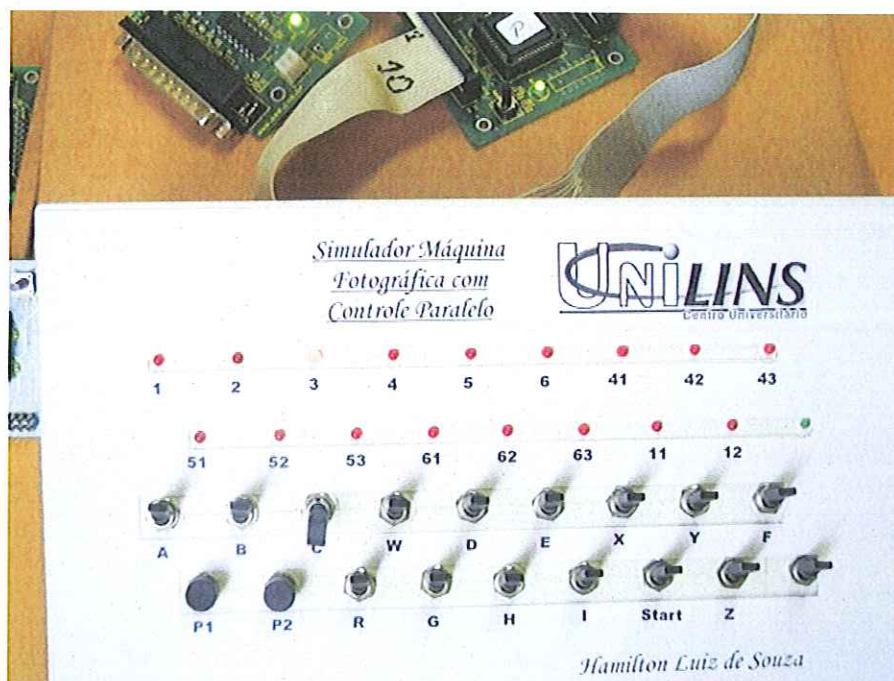


Fig. 6.10- Seqüência indicando que a lente está aberta

A chave "W" dispara os processos:

- posicionamento da câmara com movimentos para a direita e esquerda, LED "4";
- ajuste de foco com movimentos para frente e para traz, LED "6"; e
- posicionamento da câmara com movimentos para cima ou para baixo, LED "5".

Este disparo é apresentado na figura 6.11.

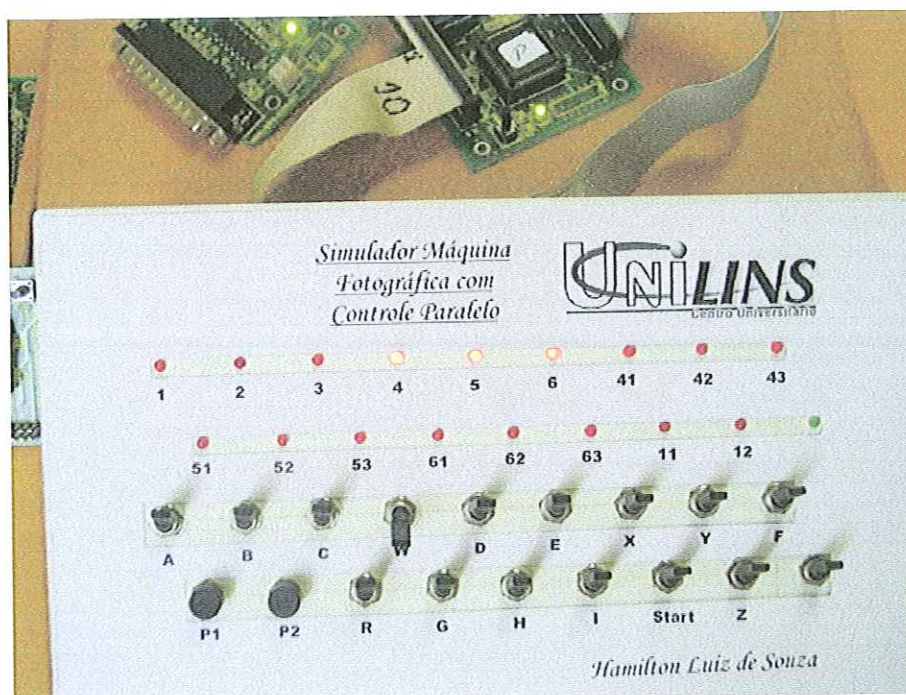


Figura 6.11- Estado indicando estado de espera do dispositivo

O acionamento das chaves "D" e "X" (fig. 6.12), provocam um movimento simultâneo para a direita e para cima da câmera, indicados pelos LEDs "42" e "52", sem o movimento de foco, como indicado pelo LED "61" (Fig. 6.13).

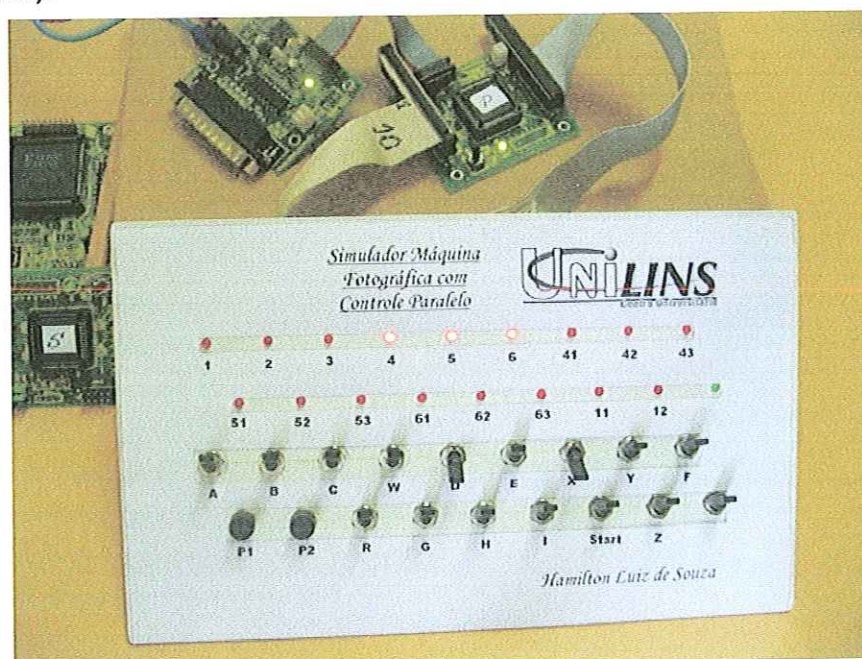


Figura 6.12- Chaves "D" e "X" ativadas



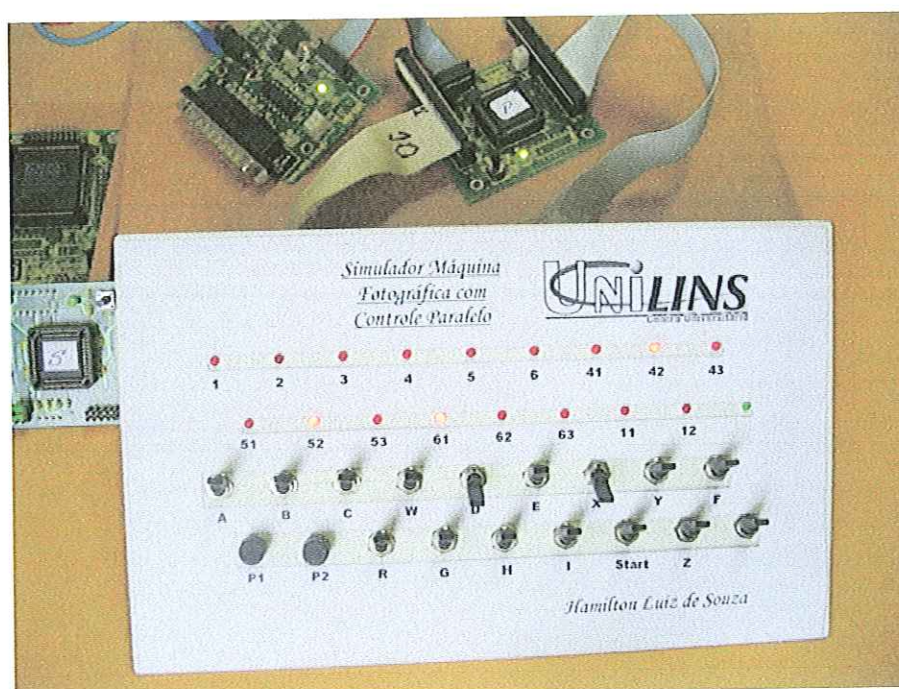


Figura 6.13- Estado indicando movimentos simultâneos

Se a chave “G” não for acionada o processo retorna ao passo de espera dos movimentos (Fig. 6.14), caso contrário temos o acionamento do obturador e o armazenamento da imagem, indicados pelo LED “11” ativo (Fig. 6.15).

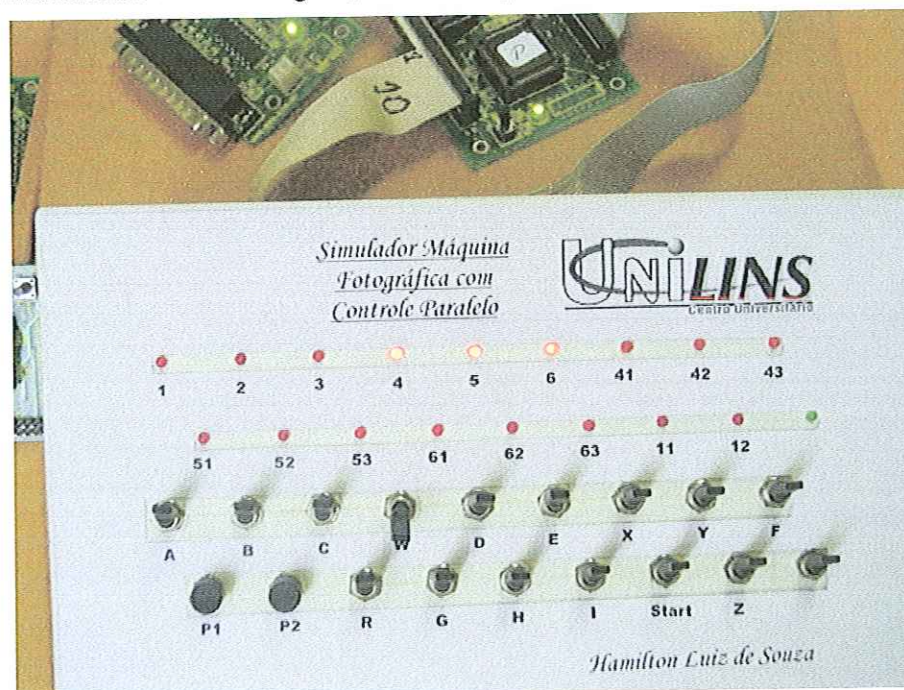


Figura 6.14- Chave “G” não acionada retorna ao estado de espera de movimento

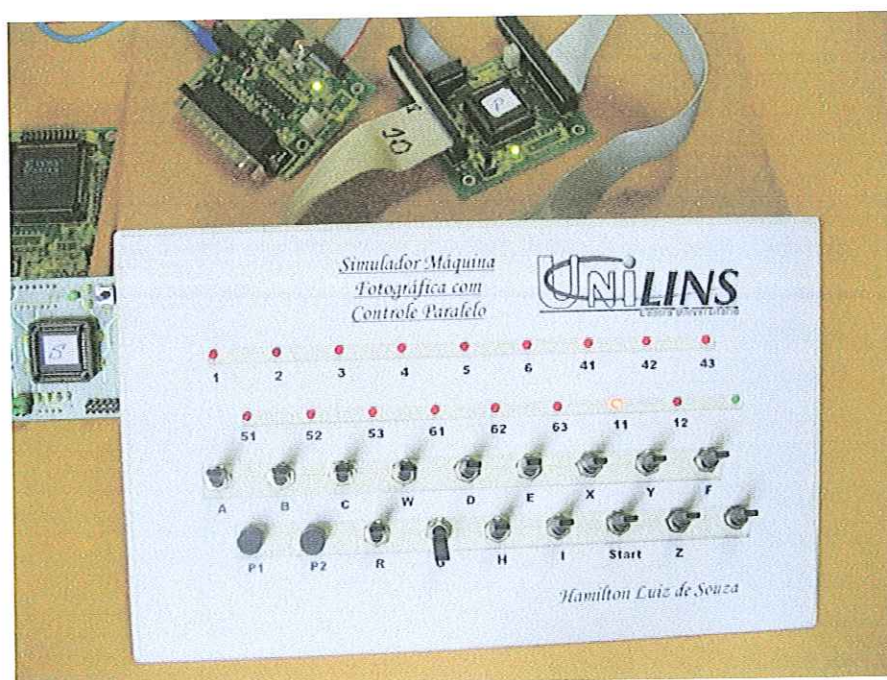


Fig. 6.15- Chave "G" acionada, disparo do obturador e armazenagem da imagem

O sensor representado pela chave "h" indica que a imagem foi impressa (Fig. 6.16) e que o sistema está pronto para retornar ao início do processo, através do acionamento da chave "I" (Fig. 6.17).

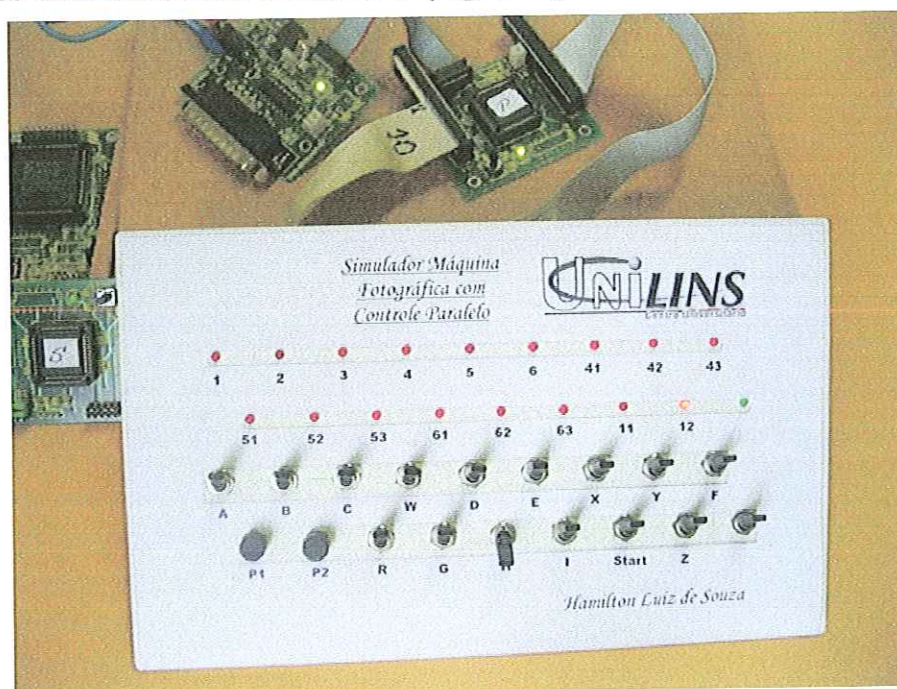


Figura. 6.16- Impressão da foto

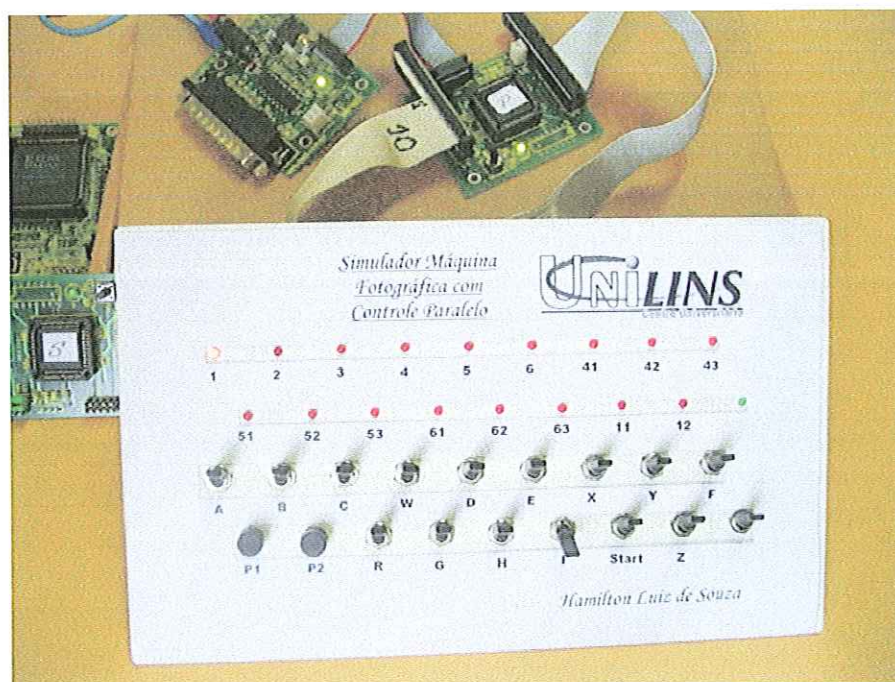


Figura 6.17 – Estado inicial do processo

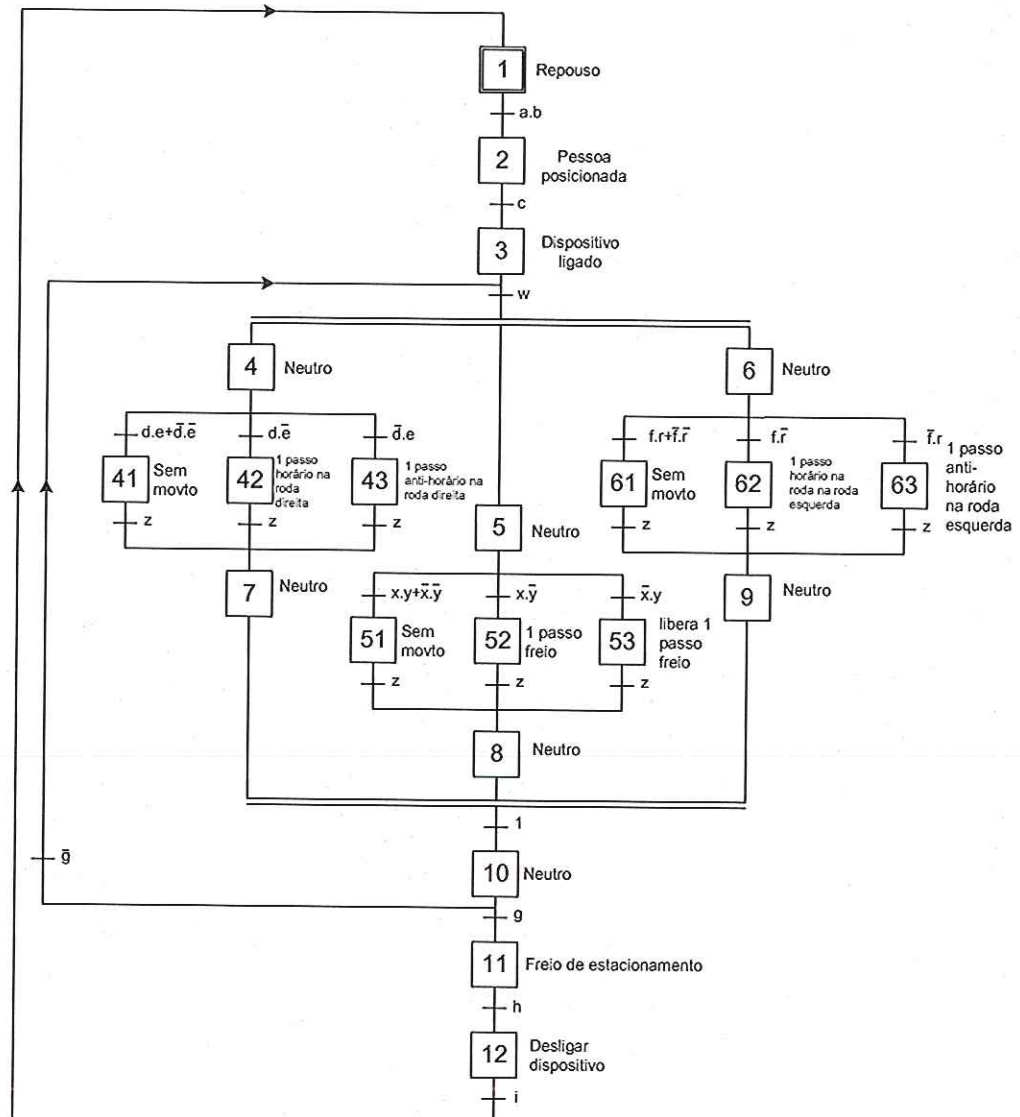
O mesmo GRAFCET, sem alteração do dispositivo assistivo, adapta-se perfeitamente, por exemplo, para controlar uma cadeira de rodas motorizada. A cadeira motorizada terá, de forma simplista, as seguintes funcionalidades:

- curva para a direita – A roda esquerda sofre um aumento de velocidade no sentido horário, enquanto a direita uma diminuição;
- curva para a esquerda – A roda direita sofre um aumento de velocidade no sentido horário, enquanto a roda esquerda uma diminuição;
- movimento retilíneo – As rodas mantêm a mesma velocidade de rotação, mantendo a cadeira em movimento retilíneo constante;
- freio – As rodas sofrem uma diminuição da rotação, de tal ordem que promova a frenagem da cadeira; e
- freio de estacionamento – O freio de estacionamento impossibilita a movimentação da cadeira, literalmente, travando as rodas.

Os controles de tal ordem que garanta a integridade do condutor deverá ser implementado em circuitos controladores próprios. As similaridades dos processos podem ser observados na figura 6.18.

O GRAFCET proposto e implementado foi testado nas várias possibilidades de acionamentos dos comandos do autômato, enfatizando a repetitividade, onde a fizemos, para cada acionamento uma repetição de dez vezes sendo que, em instante algum o dispositivo perdeu o controle dos comandos.

## GRAFSET Cadeira de Rodas Controle Paralelo



### Sensores posicionados nos atuadores

a- Bateria com carga  
 c- Aciona dispositivo  
 z- Movimento realizado  
 h- Cadeira parada  
 i- Sem sinal

### Sensores acionados pelo usuário

b- Inicia processo  
 d- } Controle de movimento ds roda direita  
 e- }  
 x- } Controle de movimento ds roda esquerda  
 y- }  
 f- } Controle de freio  
 r- }  
 g- posição central do controle

Figura 6.18- GRAFCET par controle de Cadeira de Rodas Motorizada

## 7. CONCLUSÕES

Este trabalho mostra-nos a escassez de equipamentos e dispositivos assistenciais, necessários para a vida diária, profissionalização e lazer do portador de deficiência. Poucos dispositivos despertaram interesse, principalmente porque atendem, em sua grande maioria, a interesses comerciais, ficando em segundo plano as reais necessidades do deficiente.

O número crescente de interfaces de comando que estão sendo disponibilizadas no mercado traz a necessidade dos sistemas robotizados integrarem-se ao contexto e ambiente do portador de restrição. Por exemplo, a unidade de reconhecimento de voz não deve e não pode continuar sendo limitada ao domínio do computador que está executando o software, mas sim incorporado e compatível aos equipamentos que se deseja acionar. Por outro lado, os avanços das técnicas médicas e da tecnologia envolvida na recuperação da sanidade física das pessoas, envolvidas em acidentes, têm sido grande, o que leva a um crescente número de pessoas portadoras de restrições físicas, em consequência destes acidentes, necessitarem de reabilitação para efetiva independência e reintegração de suas atividades sociais e profissionais, limitadas à suas capacidades.

O desenvolvimento de dispositivos a preços acessíveis é fundamental para a recuperação da independência e aplicação plena da cidadania do indivíduo junto à sociedade.

A utilização do GRAFCET na fase de desenvolvimento e prototipação mostra-se eficaz como ferramenta, visto que proporciona segurança, facilidade de uso e ampla visão do conjunto como um todo, possibilitando visualizar o que e como será possível se colocar em prática as expectativas do usuário final.

O desenvolvimento do autômato que propomos vem ao encontro das principais características citadas acima, pois possibilitará redução de custos, maior aplicabilidade do dispositivo, além de não ser dependente do uso de computador para controle.

Aliado a todos estes fatores há a necessidade do domínio das técnicas envolvidas no desenvolvimento e concepção de dispositivos assistenciais pelos técnicos nacionais, não podemos ficar a mercê de políticas de importação o que, via de regra, torna os equipamentos mais caros além de impossibilitar o treinamento ou pelo menos o conhecimento do usuário final antes da aquisição.

O modelo concebido serviu para validar a ferramenta desenvolvida, unicamente para este trabalho, e o dispositivo FPGA mostrou ser robusto, de fácil manipulação, que, juntamente com as ferramentas disponíveis no mercado tornam-no o principal componente no conjunto.

Para trabalhos futuros sugerimos o aperfeiçoamento da ferramenta desenvolvida, onde além da possibilidade de transformação do esquema GRAFCET em circuito digital, permita também que seu equivalente seja escrito em linguagem VHDL, de tal forma que no produto final tenhamos um programa otimizado, objetivo que não é atingido com as ferramentas hoje disponíveis.

## Referências

- \_\_\_\_\_. **A Layman's View of Reconfigurable Computing**. Disponível em: <<http://www.vcc.com/simple.html>>. Acesso em: 10 ago. 2000.
- \_\_\_\_\_. Click Website - Tecnologia Assistiva. Disponível em: <<http://www.click.com.br>>. Acesso em: 10 ago. 2000.
- \_\_\_\_\_. Explore the World of Assistive Technology. Disponível em: <[www.abledata.com](http://www.abledata.com)>. Acesso em: 10 ago. 2000.
- \_\_\_\_\_. **Reconfigurable Computing Overview**. Giga Operations. Disponível em: <<http://www.reconfig.com/giga/rintro.html>>. Acesso em: 21 ago. 2000.
- \_\_\_\_\_. **Rede Entreamigos. Informações Básicas sobre Tecnologia Assistiva**. Disponível em: <<http://www.entreamigos.com.br>>. Acesso em: 21 ago. 2000.
- \_\_\_\_\_. **RESNA, Results of 1996 Consumer Survey, 1996**. Disponível em: <<http://www.resna.org>>. Acesso em: 10 ago. 2000.
- \_\_\_\_\_. **Xilinx XC4000 FPGA DataBook**. 1994. cap.8, p. 21-35.
- Anonymous. **Cochlear Implants**: NIH Consensus Statement May 2-4, 7 (2) :1-25, 1988. Disponível em: <<http://www.bae.ncsu.edu/research/blanchard/www/465/textbook/nervours/projects/cocheat>>. Acesso em: 10 ago. 2000.
- Bossy, J. C.; Brard, P.; Faugère, P.; Merlaud, C. **LE GRAFCET : Sa Pratique et ses Applications**. Paris: Educavivre, 1979.
- Brown, S.; Rose, J. **FPGA and CPLD Architectures**: A tutorial. IEEE Design & Test Computers. Summer 1996.
- Buell D.A., Arnold, J.M., Kleinfelder W.J. **Splash2**: FPGAs in a Custom Computing Machine. IEEE Computer Society Press. 1996
- Busnel, M. et al. **The Robotized Workstation "MASTER" for Users with Tetraplegia**: Description and Evaluation. Journal of Rehabilitation Research and Development, v. 36, n° 3, 1999.



- Busnel, M.; Morvan, J. B. **Evaluation d'un système robotisé: le Robot MASTER 1**, Paris: CTNERHI, 1996.
- Chan, P.K. et al **Tutorial 2: Rapid System Prototyping using Field-Programmable Devices**. 31st Design Automation Conference. June 6-10, 1994. San Diego, California.
- David, R. **GRAF CET: A Powerful Tool for Specification of Logic Controllers**, IEEE Transaction on Control, Systems Tecnology, v.3 n.3, 1995.
- Détriché, J. M.; Lesigne, B. **The robotized system MASTER**. Proceeding of the Third Japanese-Frence Biomedical Technology Symposium, Himeji Dokkyo University, Japan. 1989.
- Devars, J. G.B.; Albert, F. **A Fast Maximum Homogeneity Smoothing**. Group ESIEE Paris
- Georgini, M. **Automação Aplicada: Descrição e Implementação de Sistemas Seqüenciais com PLCs**. São Paulo: Érica, 2000.
- Gonzalez, R.C. **Digital Image Processing**. Addison Wesley Publishing Company, 1977.
- Guittet, J. et al **The SPARTACUS telethesis: manipulator control studies**. Bull Prosthet Res, 1979.
- Hagbebell, C. H. **SYNTHESE DE LA COMMANDE DES SYSTEMES A EVENEMENTS DISCRETS PAR GRAFCET**, Université de Reims Champagne Ardenne, 1999.
- Ide, T. **Medical Robotics**. Nov 11, 1996. Disponível em: <<http://www.yamanashi-med.ac.jp/~orthop/res/robot/robote.htm>>. Acesso em: 10 ago. 2000.
- Kwee, H. H., et al **First experimentation of the SPARTACUS telethesis in a clinical environment**. Paraplegia, 1983.
- Lupton, D.; Seymor, W. **Technology, selfhood and physucal disabiity**. Soc Sci Med, 2000.
- McLaurin, C.A.; Brubaker, C.E. **Biomechanics and the Wheelchair: " Prosthetics and Orthotics International**, v.15, n.1, p. 24-37, abr 1991.
- Mello, M.A.F. **Tecnologia Assistiva**. In: Greve J.M.A., Amatuzzi M.M.- Medicina de Reabilitação aplicada à Ortopedia e Traumatologia. São Paulo: Roca, 1999.
- Milner, M. **"Rehabilitation Technology: Exploitation of R & D and Current Technologies," International Journal of Rehabilitation Research**, v. 16, n. 4, p. 253-263, Dec 1993.
- Moraes, J.C.T.B. **Engenharia de Reabilitação e Dispositivos Assistenciais: Estado da Arte e Perspectivas**. In: Anais do XVII Congresso Brasileiro de Engenharia Biomédica. Florianópolis, 2000.
- Morin, G.; Vincent, C.; **L'utilisation ou non des aides techniques: comparacion d'un modele americain besoins de la realite quebecoise**. Can J Occup Ther, 1999.
- Niblack, W. **An Introduction to Digital Image Processing**.

- Oshio, M.S.; Mourão, M.; Masiero, D. **Adaptações Alternativas de Vida Diária em Terapia Ocupacional para Portadores de Lesão Medular de Baixa Renda**. In: Programas e Resumos do VI Congresso Brasileiro de Terapia Ocupacional. Águas de Lindóia, 1999.
- Philot, G.M. **Recursos Tecnológicos e Equipamentos**. In: Anais do V Congresso Brasileiro de Terapia Ocupacional. Belo Horizonte, 1997.
- Rocha, E.F. **Recursos Tecnológicos e Equipamentos**. In: Anais do V Congresso Brasileiro de Terapia Ocupacional. Belo Horizonte, 1997.
- Scherer, M.J. **Living in the State of Stuck**. Cambridge: Brookline Books, 1993.
- Siddiqi, N.A. **"A Computer-Aided Walking Rehabilitation Robot"**, *American Journal of Physical Medicine & Rehabilitation*, v. 73, n. 3, p. 212-216, Jun 1994.
- Simmons, F. B. **"History of Cochlear Implants in the United States: A Personal Perspective"**, Cochlear Implants, Ed. R. Schindler, M. Merzenich, Raven Press, 1985.
- Taylor, B. **Math Operators Implemented in Xilinx 4K FPGA's**: Video Imaging Workshop – DSPX 1996. Giga Operations Corporation
- Taylor, B.; Varga, B.O. **Low Cost, Real Time Video Imaging is enable in SRAM based FPGA's**. Giga Operations Corporation
- Terroso, A. R. **Dispositivos Lógicos Programáveis e Linguagem de Descrição de Hardware**. Porto Alegre, 1998.
- Topping, M.; Smith, J. **The development of HANDY 1, a rehabilitation robotic system to assist the severely disabled**. Industr. Robot. 1998.
- Trombly, C.A.; Scott, A.D. **Estrutura da Terapia**. In: Trombly C.A. Terapia Ocupacional para Disfunções Físicas.
- Vieira; E.B. **Atividades da Vida Diária**. In: Manual de Gerontologia: Um Guia Teórico-Prático para Profissionais, Cuidadores e Familiares. Rio de Janeiro: Revinter, 1996.
- Villasenor, J.; Willian, H.J.; Smith, M.; **Configurable Computing**. Scientific American. Jun. 1997.
- Vincent, C. **Practices of recycling assistive technology in Quebec**. Can J Occup Ther, 1999.
- Webster, J.G. et al **Electronic Devices for Rehabilitation**. New York: John Wiley and Sons, 1985.
- Yamamoto, H. **Unidade de Referência a Recursos para a Educação Especial: Da Elaboração e do Acesso dos Materiais Didáticos ao Aluno com Deficiência Visual**. Dissertação de mestrado apresentada na Universidade Estadual do Paraná, Brasil, 1995.

## Apêndice 1 – Listagem dos programas do Sistema GRAFCET → Circuito Digital

```
program Grafcet;

uses
  Forms,
  Inicio in 'Inicio.pas' {Abertura},
  About in 'About.pas' {Sobre},
  Descricao in 'Descricao.pas' {Descr_Obj},
  Digital in 'Digital.pas' {Sist_Digital},
  Titul in 'Titul.pas' {Titulo};

{$R *.RES}

begin
  Application.Initialize;
  Application.CreateForm(TAbertura, Abertura);
  Application.CreateForm(TSobre, Sobre);
  Application.CreateForm(TDescr_Obj, Descr_Obj);
  Application.CreateForm(TSist_Digital, Sist_Digital);
  Application.CreateForm(TTitulo, Titulo);
  Application.Run;
end.
```

```
unit Inicio;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Menus, ExtCtrls, ComCtrls, Buttons, Db, DBTables, Printers, ToolWin,
  ImgList;
type
  TAbertura = class(TForm)
    MainMenu1: TMainMenu;
    Arquivo1: TMenuItem;
    Abrir1: TMenuItem;
    Abrir2: TMenuItem;
    Salvar1: TMenuItem;
    SalvarComo1: TMenuItem;
    N1: TMenuItem;
    Sair1: TMenuItem;
    Editar1: TMenuItem;
    Cortar1: TMenuItem;
    Copiar1: TMenuItem;
    Colar1: TMenuItem;
    Sobre1: TMenuItem;
    ScrollBox1: TScrollBox;
    StatusBar1: TStatusBar;
    Image: TImage;
    OpenFileDialog1: TOpenDialog;
    SaveDialog1: TSaveDialog;
    Tab_Graficet: TTable;
    Visualizar1: TMenuItem;
    Imprimir1: TMenuItem;
    N2: TMenuItem;
    PrintDialog1: TPrintDialog;
    ToolbarImages: TImageList;
    Ferramentas: TToolBar;
    ToolButton1: TToolButton;
    ToolButton2: TToolButton;
    ToolButton3: TToolButton;
    ToolButton4: TToolButton;
    ToolButton5: TToolButton;
    ToolButton6: TToolButton;
    ToolButton7: TToolButton;
    EsquemaDigital1: TMenuItem;
    Ferramentas1: TMenuItem;
    Visvel1: TMenuItem;
    ScrollBox2: TScrollBox;
    Panel1: TPanel;
    AcaoButton: TSpeedButton;
    TransicaoButton: TSpeedButton;
    XorOpenButton: TSpeedButton;
    XorCloseButton: TSpeedButton;
    AndBiOpenButton: TSpeedButton;
```

```
AndTriOpenButton: TSpeedButton;
AndBiCloseButton: TSpeedButton;
AndTriCloseButton: TSpeedButton;
LinhaVertButton: TSpeedButton;
LinhaHorDirButton: TSpeedButton;
DirAbaButton: TSpeedButton;
EsqAbaButton: TSpeedButton;
DirAciButton: TSpeedButton;
EsqAciButton: TSpeedButton;
SpeedButton1: TSpeedButton;
linhaHorEsqButton: TSpeedButton;
BarradeObjetos: TImageList;
Tab_Digital: TTable;
XorTriOpen: TSpeedButton;
XorTriClose: TSpeedButton;
procedure Sair2Click(Sender: TObject);
procedure Sobre1Click(Sender: TObject);
procedure FormClick(Sender: TObject);
procedure FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure FormCreate(Sender: TObject);
procedure AcaoButtonClick(Sender: TObject);
procedure TransicaoButtonClick(Sender: TObject);
procedure XorOpenButtonClick(Sender: TObject);
procedure FormMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure SalvarComo1Click(Sender: TObject);
procedure Salvar1Click(Sender: TObject);
procedure Sair1Click(Sender: TObject);
procedure Abrir2Click(Sender: TObject);
procedure XorCloseButtonClick(Sender: TObject);
procedure AndBiOpenButtonClick(Sender: TObject);
procedure AndTriOpenButtonClick(Sender: TObject);
procedure AndBiCloseButtonClick(Sender: TObject);
procedure AndTriCloseButtonClick(Sender: TObject);
procedure LinhaVertButtonClick(Sender: TObject);
procedure LinhaHorDirButtonClick(Sender: TObject);
procedure DirAbaButtonClick(Sender: TObject);
procedure EsqAbaButtonClick(Sender: TObject);
procedure DirAciButtonClick(Sender: TObject);
procedure EsqAciButtonClick(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure linhaHorEsqButtonClick(Sender: TObject);
procedure EsquemaDigital1Click(Sender: TObject);
procedure Imprimir1Click(Sender: TObject);
procedure Visvel1Click(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure XorTriOpenClick(Sender: TObject);
procedure XorTriCloseClick(Sender: TObject);
private
```

```

    { Private declarations }
public
    { Public declarations }
    X1,Y1,X1_Ant,Y1_Ant,Fig,Num,Obj: integer;
    X1_Mensg,Y1_Mensg,X2_Mensg,Y2_Mensg: integer;
    Seq: TRect;
    XH, YH: integer;
    Trans: string;
    Acao_OK, Trans_OK, Ferram, Comp_OK: Boolean;
    CX_I,CY_I,CX_F,CY_F,Sequencia: integer;
    Num_Acao, Guarda_Mem: Integer;
    G1, G2, G3, G4, G5, G6: string;
    BrushStyle: TBrushStyle;
    PenStyle: TPenStyle;
    PenWide: Integer;
    Drawing: Boolean;
    Origin, MovePt: TPoint;
    CurrentFile: string;
    procedure SaveStyles;
    procedure RestoreStyles;
    procedure Fig_Acao(X1,Y1: integer);
    procedure Fig_Transicao(X1,Y1: integer);
    procedure Fig_XOR_Open(X1,Y1: integer);
    procedure Fig_XOR_Close(X1,Y1: integer);
    procedure Fig_XorTri_Open(X1,Y1: integer);
    procedure Fig_XorTri_Close(X1,Y1: integer);
    procedure Fig_AndBi_Open(X1,Y1: integer);
    procedure Fig_AndTri_Open(X1,Y1: integer);
    procedure Fig_AndBi_Close(X1,Y1: integer);
    procedure Fig_AndTri_Close(X1,Y1: integer);
    procedure Fig_LinhaVertAba(X1,Y1: integer);
    procedure Fig_LinhaHorDir(X1,Y1: integer);
    procedure Fig_LinhaHorEsq(X1,Y1: integer);
    procedure Fig_LinhaVertAci(X1,Y1: integer);
    procedure Fig_LinhaDirAba(X1,Y1: integer);
    procedure Fig_LinhaEsqAba(X1,Y1: integer);
    procedure Fig_LinhaDirAci(X1,Y1: integer);
    procedure Fig_LinhaEsqAci(X1,Y1: integer);
    procedure Gravacao_Dados(XIG, YIG, Sequenc, XFG, YFG, Tip: Integer;
        Descr1, Descr2, Descr3, Stat1, Stat2,
        Stat3: string; Numero: Integer; Apelido:String);
    procedure Gravacao_Digit(Sequenc, Tipo,X_MEU,Y_MEU:integer);
    procedure Regrava_Digit(Sequenc:integer; Ent1, Ent2, Ent3,
        St1, St2, St3:string);
    procedure Fechamento(XFF,YFF: integer);
    function Verifica_Componente(XVC,YVC:Integer):Integer;
end;
var
    Abertura: TAbertura;
implementation

```

```

uses About, Descricao, Digital, Titul;
{$R *.DFM}
// Procedimentos e Funções Particulares
function TAbertura.Verifica_Componente(XVC,YVC: Integer):Integer;
begin
  with Tab_Grafcet do
  begin
    open;
    first;
    Verifica_Componente:= 0;
    while (not EOF) do
    begin
      if ((XVC = fieldbyname('Coord_X_Fim').asInteger) and
        (YVC = fieldbyname('Coord_Y_Fim').asInteger)) then
      begin
        Verifica_Componente:= fieldbyname('Tipo').asInteger;
        break;
      end;
    next;
    end;
    close;
  end;
end;
procedure TAbertura.Fechamento(XFF: integer;
                                YFF: integer);
var
  Memor: integer;
begin
  with Tab_Grafcet do
  begin
    open;
    first;
    while not EOF do
    begin
      if (XFF = fieldbyname('Coord_X_Inic').asinteger) and
        (YFF = fieldbyname('Coord_Y_Inic').asinteger) then
      begin
        Memor:= fieldbyname('Num_Memoria').asinteger;
        Tab_Digital.open;
        Tab_Digital.First;
        while not Tab_Digital.eof do
        begin
          if Memor = Tab_Digital.fieldbyname('Cod_Obj').asinteger then
          begin
            Tab_Digital.edit;
            Tab_Digital.fieldbyname('Entrada1').asstring:= G1;
            Tab_Digital.fieldbyname('Entrada2').asstring:= G2;
            Tab_Digital.fieldbyname('Entrada3').asstring:= G3;
            Tab_Digital.fieldbyname('Nega_Ent1').asstring:= G4;
            Tab_Digital.fieldbyname('Nega_Ent2').asstring:= G5;
          end;
        end;
      end;
    end;
  end;
end;

```

```

    Tab_Digital.fieldbyname('Nega_Ent3').asString:= G6;
    Tab_Digital.fieldbyname('Reset1').asinteger:= Guarda_Mem;
    Tab_Digital.fieldbyname('Anterior1').asinteger:= Guarda_Mem;
    Tab_Digital.refresh;
    Tab_Digital.findkey([Guarda_Mem]);
    Tab_Digital.edit;
    Tab_Digital.fieldbyname('Proximo1').asinteger:= Memor;
    Tab_Digital.fieldbyname('Reset1').asinteger:= Memor;
    end;
    Tab_Digital.next;
end;
Tab_Digital.close;
break;
end;
next;
end;
close;
end;
end;
procedure TAbertura.Gravacao_Dados(XIG:Integer; YIG:Integer;
Sequenc:Integer;
                                XFG:Integer; YFG:Integer; Tip: Integer;
                                Descr1:string; Descr2:string;
                                Descr3: string;
                                Stat1: string;
                                stat2: string;
                                stat3: string;
                                Numero: integer;
                                Apelido:string);
begin
    with Tab_Grafcet do
        begin
            open;
            if findkey([XIG,YIG,Sequenc]) then
                begin
                    MessageDlg('Componente já cadastrado', mtInformation, [mbOK], 0);
                end
            else
                begin
                    // Sequencia:= Sequencia + 1;
                    insert;
                    FieldByName('Tipo').asInteger:= Tip;
                    FieldByName('Coord_X_Inic').asInteger:= XIG;
                    FieldByName('Coord_Y_Inic').asInteger:= YIG;
                    FieldByName('Coord_X_Fim').asInteger:= XFG;
                    FieldByName('Coord_Y_Fim').asInteger:= YFG;
                    FieldByName('Sequencia').asInteger:= Sequenc;
                    FieldByName('Descricao1').asString:= Descr1;
                    FieldByName('Descricao2').asString:= Descr2;
                    FieldByName('Descricao3').asString:= Descr3;
                end
            end;
        end;
    end;
end;

```



```

        FieldByName('Num_Memoria').asInteger:= Numero;
        FieldByName('Nega_Descr1').asString:= Stat1;
        FieldByName('Nega_Descr2').asString:= Stat2;
        FieldByName('Nega_Descr3').asString:= Stat3;
        fieldByName('Apelido').asString:= Apelido;
    end;
    post;
    close;
end;
end;
procedure TAbertura.Gravacao_Digit(Sequenc:integer;
        Tipo:integer;
        X_MEU:integer;
        Y_MEU:integer);
//        N_Prox:integer);
var
    E1,E2,E3,S1,S2,S3:string;
    XM,YM,P: integer;
begin
    E1:="";
    E2:="";
    E3:="";
    S1:="";
    S2:="";
    S3:="";
    XM:=0;
    YM:=0;
    P:= 0;
    with Tab_Grafcet do
    begin
        open;
        while not eof do
        begin
            if (fieldbyname('Coord_X_Fim').asInteger = X_MEU) and
                (fieldbyname('Coord_Y_Fim').asInteger = Y_MEU) then
            begin
                E1:= fieldbyname('Descricao1').asString;
                E2:= fieldbyname('Descricao2').asString;
                E3:= fieldbyname('Descricao3').asString;
                S1:= fieldbyname('Nega_Descr1').asString;
                S2:= fieldbyname('Nega_Descr2').asString;
                S3:= fieldbyname('Nega_Descr3').asString;
                XM:= fieldbyname('Coord_X_Inic').asinteger;
                YM:= fieldbyname('Coord_Y_Inic').asinteger;
                break;
            end
            else
            begin
                next;
            end;
        end;
    end;
end;

```

```

end;
first;
while not eof do
begin
  if (fieldbyname('Coord_X_Fim').asInteger = XM) and
    (fieldbyname('Coord_Y_Fim').asInteger = YM) then
  begin
    P:= fieldbyname('Num_Memoria').asinteger;
    break;
  end
  else
  begin
    next;
  end;
end;
close;
end;
with Tab_Digital do
begin
  open;
  if findkey([Sequenc]) then
  begin
    MessageDlg('Componente já cadastrado', mtInformation, [mbOK], 0);
  end
  else
  begin
    insert;
    FieldByName('Tipo').asInteger:= Tipo;
    FieldByName('Cod_Obj').asInteger:= Sequenc;
    fieldbyname('Entrada1').asstring:= E1;
    fieldbyname('Entrada2').asstring:= E2;
    fieldbyname('Entrada3').asstring:= E3;
    fieldbyname('Nega_Ent1').asstring:= S1;
    fieldbyname('Nega_Ent2').asstring:= S2;
    fieldbyname('Nega_Ent3').asstring:= S3;
    fieldbyname('Anterior1').asinteger:= P;
    fieldbyname('Reset1').asinteger:= P;
    post;
    first;
    while not eof do
    begin
      if fieldbyname('Cod_Obj').asinteger = P then
      begin
        edit;
        fieldbyname('Proximo1').asinteger:= Sequenc;
        refresh;
        break;
      end
      else
      begin

```

```

        next
    end;
end;
end;
close;
end;
end;
procedure TAbertura.Regrava_Digit(Sequenc:integer;
    Ent1: string;
    Ent2: string;
    Ent3: string;
    St1: string;
    St2: string;
    St3: string);
begin
    with Tab_Digital do
    begin
        open;
        if findkey([Sequenc]) then
        begin
            edit;
            fieldbyname('Entrada1').asstring:= Ent1;
            fieldbyname('Entrada2').asstring:= Ent2;
            fieldbyname('Entrada3').asstring:= Ent3;
            fieldbyname('Nega_Ent1').asstring:= St1;
            fieldbyname('Nega_Ent2').asstring:= St2;
            fieldbyname('Nega_Ent3').asstring:= St3;
            refresh;
            close;
        end
        else
        begin
            MessageDlg('Componente não cadastrado', mtInformation, [mbOK], 0);
        end;
    end;
end;
procedure TAbertura.SaveStyles;
begin
    with Image.Canvas do
    begin
        BrushStyle := Brush.Style;
        PenStyle := Pen.Style;
        PenWide := Pen.Width;
    end;
end;
procedure TAbertura.RestoreStyles;
begin
    with Image.Canvas do
    begin
        Brush.Style := BrushStyle;

```

```

    Pen.Style := PenStyle;
    Pen.Width := PenWide;
end;
end;
procedure TAbertura.Fig_Acao(X1: integer; Y1: integer);
begin
    Comp_OK:= false;
    with Abertura do
    begin
        if Sequencia = 0 then
        begin
            // X1:= round(X1/10)*10;
            // Y1:= round(Y1/10)*10;
            CX_I:= X1;
            CY_I:= Y1;
            Comp_OK:= true;
        end
        else
        begin
            CX_I:= X1;
            CY_I:= Y1;
            if Verifica_Componente(CX_I,CY_I) <> 1 then
                Comp_OK:= true;
            end;
            if Comp_OK then
            begin
                if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
                begin
                    X1:= round(X1_Ant/10)*10;
                    Y1:= round(Y1_Ant/10)*10;
                end;
                Image.Canvas.MoveTo(X1,Y1);
                Image.Canvas.Brush.Color := clWhite;
                Image.Canvas.Pen.Color:= clBlack;
                Image.Canvas.Lineto(X1,Y1+10);
                if Num = 0 then
                    Image.Canvas.Rectangle(X1-11,Y1+9,X1+11,Y1+31);
                Image.Canvas.Rectangle(X1-10,Y1+10,X1+10,Y1+30);
                X1_Mensg:= X1-9;
                Y1_Mensg:= Y1+11;
                X2_Mensg:= X1+9;
                Y2_Mensg:= Y1+29;
                Image.Canvas.MoveTo(X1,Y1+30);
                Image.Canvas.Lineto(X1,Y1+40);
                Y1:=Y1+40;
                Image.Canvas.Brush.Color := clGray;
                Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
                X1_Ant:=X1;
                Y1_Ant:=Y1;
                CX_F:= X1_Ant;
            end;
        end;
    end;
end;

```

```

        CY_F:= Y1_Ant;
    // Escreve Número da Ação
        Seq := Rect(X1_Mensg,Y1_Mensg,X2_Mensg,Y2_Mensg);
        Image.Canvas.Brush.Color:= clWhite;
        Num:= Num+1;
    {
        if Num < 10 then
            Image.Canvas.TextRect(Seq,X1_Mensg+6,Y1_Mensg+3,IntToStr(Num))
        else

Image.Canvas.TextRect(Seq,X1_Mensg+3,Y1_Mensg+3,IntToStr(Num));}
    // Fim Numeração
    // Recebe Descrição
        Descr_Obj.Edit_Apelido.Visible := true;
        Descr_Obj.showmodal;

Image.Canvas.TextRect(Seq,X1_Mensg,Y1_Mensg+2,Descr_Obj.Edit_Apelido.
Text);
        Seq:= Rect(X1_Mensg+22,Y1_Mensg,X2_Mensg+200,Y2_Mensg);

Image.Canvas.TextRect(Seq,X1_Mensg+23,Y1_Mensg+2,Descr_Obj.DescrEdit
1.text);
    // Fim Descrição
    // Grava na tabela Dados
        Sequencia:= Sequencia + 1;
        Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig,
            Descr_Obj.DescrEdit1.Text , " ", " ", " ", " ",
Num,Descr_Obj.Edit_Apelido.text);

    // Grava na tabela Digital
        Gravacao_Digit(Num_Acao, 1,CX_I, CY_I);
        Num_Acao:= Num_Acao + 1;
        Guarda_Mem:= Num_Acao - 1;
    // Fim da gravação
        Obj:= Obj+1;
        Descr_Obj.Edit_Apelido.Visible := false;
    end
    else
    begin
        MessageDlg('Não é permitida a colocação de Ação neste caso',
mtlInformation,
            [mbOk], 0);
    end;
    end;
// end;
end;
procedure TAbertura.Fig_Transicao(X1,Y1: integer);
var
    S1, S2, S3: string;
begin
    Comp_OK:= false;

```

```

with Abertura do
begin
  if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
  begin
    X1:=X1_Ant;
    Y1:= Y1_Ant;
    CX_I:= X1;
    CY_I:= Y1;
  end;
  if Verifica_Componente(CX_I,CY_I) <> 2 then
    Comp_OK:= true;
  if Comp_OK then
  begin
    Image.Canvas.Moveto(X1,Y1);
    Image.Canvas.Brush.Color := clWhite;
    Image.Canvas.Lineto(X1,Y1+20);
    Image.Canvas.Lineto(X1-5,Y1+20);
    Image.Canvas.Lineto(X1+5,Y1+20);
    Image.Canvas.Moveto(X1,Y1+20);
    Image.Canvas.Lineto(X1,Y1+40);
    Y1:=Y1+40;
    Image.Canvas.Brush.Color := clGray;
    Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
    X1_Ant:=X1;
    Y1_Ant:=Y1;
    CX_F:=X1_Ant;
    CY_F:=Y1_Ant;
  // Recebe Descrição
    Descr_Obj.Edit_Apelido.Visible := true;
    Descr_Obj.showmodal;
    X1_Mensg:= X1-18;
    Y1_Mensg:= Y1-30;
    X2_Mensg:= X1+11;
    Y2_Mensg:= Y1+5;
    Image.Canvas.Brush.Color:= clWhite;
    Trans:= Descr_Obj.DescrEdit1.text;
    if Descr_Obj.DescrEdit2.text <> " then
      Trans:= Trans + ';' + Descr_Obj.DescrEdit2.text;
    if Descr_Obj.DescrEdit3.text <> " then
      Trans:= Trans + ';' + Descr_Obj.DescrEdit3.text;
    Image.Canvas.TextOut(X1_Mensg+23,Y1_Mensg+4,Trans);
  // Recebe Negação da ação
    if Descr_Obj.Descr1CheckBox1.Checked then
      Trans:= StringOfChar('_',length(descr_Obj.DescrEdit1.Text))
    else
      Trans:= '';
    if Descr_Obj.Descr2CheckBox2.Checked then
      Trans:= Trans + '+' +StringOfChar('_',length(descr_Obj.DescrEdit2.Text))
    else
      Trans:= Trans + '+' + '';
  
```

```

if Descr_Obj.Descr3CheckBox3.Checked then
  Trans:= Trans + ''+StringOfChar('_',length(descr_Obj.DescrEdit3.Text))
else
  Trans:= Trans + '+' ;
Image.Canvas.TextOut(X1_Mensg+23,Y1_Mensg-10,Trans);
// Fim Descrição
// Grava na tabela Dados
Sequencia:= Sequencia + 1;
if Descr_Obj.Descr1CheckBox1.Checked then
  S1:= 'N'
else
  S1:= '';
if Descr_Obj.Descr2CheckBox2.Checked then
  S2:= 'N'
else
  S2:= '';
if Descr_Obj.Descr3CheckBox3.Checked then
  S3:= 'N'
else
  S3:= '';
Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig,
  Descr_Obj.DescrEdit1.Text , Descr_Obj.DescrEdit2.Text,
  Descr_Obj.DescrEdit3.Text,S1,S2,S3,Num,"");
{ Guarda informações para FECHAMENTO }
G1:= Descr_Obj.DescrEdit1.Text;
G2:= Descr_Obj.DescrEdit2.Text;
G3:= Descr_Obj.DescrEdit3.Text;
G4:= S1;
G5:= S2;
G6:= S3;
// Fim da gravação
end
else
begin
  MessageDlg('Não é permitido DUAS TRANSIÇÕES subseqüentes.',
mtInformation,
  [mbOk], 0);
  end;
end;
end;
procedure TAbertura.Fig_XOR_Open(X1,Y1: integer);
begin
  with Abertura do
  begin
    if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
    begin
      X1:=X1_Ant;
      Y1:= Y1_Ant;
      CX_I:= X1;
      CY_I:= Y1;
    end;
  end;
end;

```

```

end;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Brush.Color := clWhite;
Image.Canvas.Lineto(X1,Y1+20);
Image.Canvas.Lineto(X1-80,Y1+20);
Image.Canvas.Lineto(X1-80,Y1+40);
Y1:= Y1+40;
X1:= X1-80;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
X1_Ant:= X1;
Y1_Ant:= Y1;
CX_F:=X1_Ant;
CY_F:=Y1_Ant;
Image.Canvas.Moveto(X1,Y1-20);
Image.Canvas.Lineto(X1+160,Y1-20);
Image.Canvas.Lineto(X1+160,Y1);
X1:= X1+160;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
end;
// Grava na tabela
// Gravação da perna esquerda
Sequencia:= Sequencia + 1;
Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig," ", " ",
               " ", " ", Num, "");
// Gravação da perna direita
Sequencia:= Sequencia + 1;
Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F + 160, CY_F, Fig," ", " ",
               " ", " ", Num, "");
// Fim da gravação
end;
procedure TAbertura.Fig_XOR_Close(X1,Y1: integer);
begin
with Abertura do
begin
if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
begin
X1:=X1_Ant;
Y1:= Y1_Ant;
CX_I:= X1;
CY_I:= Y1;
end;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Brush.Color := clWhite;
Image.Canvas.Lineto(X1,Y1+20);
Image.Canvas.Lineto(X1+160,Y1+20);
Image.Canvas.Lineto(X1+160,Y1);
X1:= X1+160;
Image.Canvas.Brush.Color := clGray;

```



```

Image.Canvas.Ellipse(X1-2,Y1+1,X1+2,Y1-3);
Image.Canvas.Brush.Color := clWhite;
Image.Canvas.Moveto(X1-80,Y1+20);
Image.Canvas.Lineto(X1-80,Y1+40);
Y1:= Y1+40;
X1:= X1-80;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
X1_Ant:=X1;
Y1_Ant:=Y1;
CX_F:=X1_Ant;
CY_F:=Y1_Ant;
end;
// Grava na tabela
Sequencia:= Sequencia + 1;
Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig," ", " ",
               ", ", " ", Num, "");

// Fim da gravação
end;
procedure TAbertura.Fig_XorTri_Open(X1,Y1: integer);
begin
with Abertura do
begin
if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
begin
X1:=X1_Ant;
Y1:= Y1_Ant;
CX_I:= X1;
CY_I:= Y1;
end;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Brush.Color := clWhite;
Image.Canvas.Lineto(X1,Y1+20);
X1:= X1-160;
Y1:= Y1+20;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Lineto(X1+320,Y1);
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Lineto(X1,Y1+20);
Image.Canvas.Moveto(X1+160,Y1);
Image.Canvas.Lineto(X1+160,Y1+20);
Image.Canvas.Moveto(X1+320,Y1);
Image.Canvas.Lineto(X1+320,Y1+20);
Y1:= Y1+20;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
Image.Canvas.Brush.Color := clWhite;
X1_Ant:= X1;
Y1_Ant:= Y1;

```

```

CX_F:=X1_Ant;
CY_F:=Y1_Ant;
X1:= X1+160;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
Image.Canvas.Brush.Color := clWhite;
X1:= X1+160;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
Image.Canvas.Brush.Color := clWhite;
end;
// Grava na tabela perna direita

    Sequencia:= Sequencia + 1;
    Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig," ", " ",
        " ", " ", "Num, ");
// Grava na tabela perna central
    Sequencia:= Sequencia + 1;
    Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F + 160, CY_F, Fig," ", " ",
        " ", " ", "Num, ");
// Grava na tabela perna esquerda
    Sequencia:= Sequencia + 1;
    Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F + 320, CY_F, Fig," ", " ",
        " ", " ", "Num, ");
// Fim da gravação
end;
procedure TAbertura.Fig_XorTri_Close(X1,Y1: integer);
begin
with Abertura do
begin
if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
begin
X1:=X1_Ant;
Y1:= Y1_Ant;
CX_I:= X1;
CY_I:= Y1;
end;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Lineto(X1,Y1+20);
Image.Canvas.Lineto(X1+320,Y1+20);
Image.Canvas.Lineto(X1+320,Y1);
Image.Canvas.Moveto(X1+160,Y1+20);
Image.Canvas.Lineto(X1+160,Y1);
{ Y1:= Y1+12;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Lineto(X1+320,Y1);}
Image.Canvas.Moveto(X1+160,Y1+20);
Image.Canvas.Lineto(X1+160,Y1+40);
Y1:=Y1+40;
X1:= X1+160;

```

```

X1_Ant:= X1;
Y1_Ant:= Y1;
CX_F:=X1_Ant;
CY_F:=Y1_Ant;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
Y1:= Y1-20;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1-2);
X1:= X1+160;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1-2);
X1:= X1-320;
end;
// Grava na tabela
Sequencia:= Sequencia + 1;
Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig," ", " ",
               " ", " ", Num, "");
// Fim da gravação
end;
procedure TAbertura.Fig_AndBi_Open(X1,Y1: integer);
begin
with Abertura do
begin
if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
begin
X1:=X1_Ant;
Y1:= Y1_Ant;
CX_I:= X1;
CY_I:= Y1;
end;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Brush.Color := clWhite;
Image.Canvas.Lineto(X1,Y1+18);
Image.Canvas.Moveto(X1-80,Y1+18);
Image.Canvas.LineTo(X1+80,Y1+18);
Image.Canvas.Moveto(X1,Y1+20);
Image.Canvas.Lineto(X1-80,Y1+20);
Image.Canvas.Lineto(X1-80,Y1+40);
Y1:= Y1+40;
X1:= X1-80;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
X1_Ant:= X1;
Y1_Ant:= Y1;
CX_F:=X1_Ant;
CY_F:=Y1_Ant;
Image.Canvas.Moveto(X1,Y1-20);
Image.Canvas.Lineto(X1+160,Y1-20);
Image.Canvas.Lineto(X1+160,Y1);

```

```

X1:= X1+160;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
end;
// Grava na tabela
// Gravação da perna esquerda
Sequencia:= Sequencia + 1;
Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig,"", "", "",
               "", "", "", Num, "");
// Gravação da perna direita
Sequencia:= Sequencia + 1;
Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F + 160, CY_F, Fig,"", "", "",
               "", "", "", Num, "");
// Fim da gravação
end;
procedure TAbertura.Fig_AndTri_Open(X1,Y1: integer);
begin
with Abertura do
begin
if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
begin
X1:=X1_Ant;
Y1:= Y1_Ant;
CX_I:= X1;
CY_I:= Y1;
end;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Brush.Color := clWhite;
Image.Canvas.Lineto(X1,Y1+20);
X1:= X1-160;
Y1:= Y1+20;
Image.Canvas.Moveto(X1,Y1-3);
Image.Canvas.Lineto(X1+320,Y1-3);
Y1:= Y1;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Lineto(X1+320,Y1);
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Lineto(X1,Y1+20);
Image.Canvas.Moveto(X1+160,Y1);
Image.Canvas.Lineto(X1+160,Y1+20);
Image.Canvas.Moveto(X1+320,Y1);
Image.Canvas.Lineto(X1+320,Y1+20);
Y1:= Y1+20;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
Image.Canvas.Brush.Color := clWhite;
X1_Ant:= X1;
Y1_Ant:= Y1;
CX_F:=X1_Ant;
CY_F:=Y1_Ant;

```

```

X1:= X1+160;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
Image.Canvas.Brush.Color := clWhite;
X1:= X1+160;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
Image.Canvas.Brush.Color := clWhite;
end;
// Grava na tabela perna direita
Sequencia:= Sequencia + 1;
Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig," ", " ",
               ", ", " ", Num, "");
// Grava na tabela perna central
Sequencia:= Sequencia + 1;
Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F + 160, CY_F, Fig," ", " ",
               ", ", " ", Num, "");
// Grava na tabela perna esquerda
Sequencia:= Sequencia + 1;
Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F + 320, CY_F, Fig," ", " ",
               ", ", " ", Num, "");
// Fim da gravação
end;
procedure TAbertura.Fig_AndBi_Close(X1,Y1: integer);
begin
with Abertura do
begin
if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
begin
X1:=X1_Ant;
Y1:= Y1_Ant;
CX_I:= X1;
CY_I:= Y1;
end;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Lineto(X1,Y1+20);
Image.Canvas.Lineto(X1+160,Y1+20);
Image.Canvas.Lineto(X1+160,Y1);
Y1:= Y1+20;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Lineto(X1+160,Y1);
Image.Canvas.Moveto(X1+80,Y1);
Image.Canvas.Lineto(X1+80,Y1+20);
Y1:=Y1+20;
X1:= X1+80;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
X1_Ant:=X1;
Y1_Ant:=Y1;
CX_F:=X1_Ant;

```

```

CY_F:=Y1_Ant;
X1:= X1-80;
Y1:= Y1-20;
X1:= X1+160;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1+2);
end;
// Grava na tabela
Sequencia:= Sequencia + 1;
Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig,"", "",
               "", "", "Num, ");
// Fim da gravação
end;
procedure TAbertura.Fig_AndTri_Close(X1,Y1: integer);
begin
with Abertura do
begin
if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
begin
X1:=X1_Ant;
Y1:= Y1_Ant;
CX_I:= X1;
CY_I:= Y1;
end;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Lineto(X1,Y1+20);
Image.Canvas.Lineto(X1+320,Y1+20);
Image.Canvas.Lineto(X1+320,Y1);
Image.Canvas.Moveto(X1+160,Y1+20);
Image.Canvas.Lineto(X1+160,Y1);
Y1:= Y1+23;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Lineto(X1+320,Y1);
Image.Canvas.Moveto(X1+160,Y1);
Image.Canvas.Lineto(X1+160,Y1+17);
Y1:=Y1+20;
X1:= X1+160;
X1_Ant:= X1;
Y1_Ant:= Y1;
CX_F:=X1_Ant;
CY_F:=Y1_Ant;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-5,X1+2,Y1-1);
Y1:= Y1-20;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1-2);
X1:= X1+160;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-2,X1+2,Y1-2);
X1:= X1-320;

```

```

end;
  // Grava na tabela
  Sequencia:= Sequencia + 1;
  Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig,"", "", "",
    "", "", "Num, ");
  // Fim da gravação
end;
procedure TAbertura.Fig_LinhaVertAba(X1,Y1: integer);
begin
  with Abertura do
  begin
    if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
    begin
      X1:=X1_Ant;
      Y1:= Y1_Ant;
      CX_I:= X1;
      CY_I:= Y1;
    end;
    Image.Canvas.Moveto(X1,Y1);
    Image.Canvas.Lineto(X1,Y1+40);
    Image.Canvas.Moveto(X1,Y1+20);
    Image.Canvas.Brush.Color:= clYellow;
    Image.Canvas.Polygon([Point(X1+3,Y1+10),Point(X1-
3,Y1+10),Point(X1,Y1+20)]);
    Image.Canvas.Brush.Color:= clWhite;
    X1_Ant:= X1;
    Y1_Ant:= Y1+40;
    CX_F:=X1_Ant;
    CY_F:=Y1_Ant;
    Image.Canvas.Brush.Color := clGray;
    Image.Canvas.Ellipse(X1-2,Y1+38,X1+2,Y1+42);
  end;
  // Grava na tabela
  Sequencia:= Sequencia + 1;
  Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig,"", "", "",
    "", "", "Num, ");
  Fechamento(CX_F, CY_F);
  // Fim da gravação
end;
procedure TAbertura.Fig_LinhaHorDir(X1,Y1: integer);
begin
  with Abertura do
  begin
    if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
    begin
      X1:=X1_Ant;
      Y1:= Y1_Ant;

      CX_I:= X1;
      CY_I:= Y1;

```



```

end;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Lineto(X1+40,Y1);
Image.Canvas.Moveto(X1+10,Y1);
Image.Canvas.Brush.Color:= clYellow;
Image.Canvas.Polygon([Point(X1+10,Y1+3),Point(X1+10,Y1-
3),Point(X1+20,Y1)]);
Image.Canvas.Brush.Color:= clWhite;
X1_Ant:= X1+40;
Y1_Ant:= Y1;
CX_F:=X1_Ant;
CY_F:=Y1_Ant;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1+38,Y1-2,X1+42,Y1+2);
end;
// Grava na tabela
  Sequencia:= Sequencia + 1;
  Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig,"", "", "",
    "", "", "", Num, "");
  Fechamento(CX_F, CY_F);
// Fim da gravação
// ---> Começa aqui
Tab_Grafcet.open;
Tab_Grafcet.first;
while not Tab_Grafcet.Eof do
begin
  if (Tab_Grafcet.fieldbyname('Coord_X_Inic').asinteger = X1_Ant - 30) and
    (Tab_Grafcet.fieldbyname('Coord_Y_Inic').asinteger = Y1_Ant - 45) then
  begin
    messagedlg('Achei', mtInformation,
      [mbOk], 0);
    break;
  end;
  Tab_Grafcet.next;
end;
Tab_Grafcet.close;
// -----> Termina aqui
end;
procedure TAbertura.Fig_LinhaVertAci(X1,Y1: integer);
begin
  with Abertura do
  begin
    if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
    begin
      X1:=X1_Ant;
      Y1:= Y1_Ant;
      CX_I:= X1;
      CY_I:= Y1;
    end;
    Image.Canvas.Moveto(X1,Y1);

```



```

Image.Canvas.Lineto(X1,Y1-40);
Image.Canvas.Moveto(X1,Y1-20);
Image.Canvas.Brush.Color:= clYellow;
Image.Canvas.Polygon([Point(X1+3,Y1-10),Point(X1-3,Y1-10),Point(X1,Y1-
20)]);
Image.Canvas.Brush.Color:= clWhite;
X1_Ant:= X1;
Y1_Ant:= Y1-40;
CX_F:=X1_Ant;
CY_F:=Y1_Ant;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-38,X1+2,Y1-42);
end;
// Grava na tabela
Sequencia:= Sequencia + 1;
Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig,"", "", "",
",", ", ", "Num, ");
Fechamento(CX_F, CY_F);
// Fim da gravação
end;
procedure TAbertura.Fig_LinhaHorEsq(X1,Y1: integer);
begin
with Abertura do
begin
if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
begin
X1:=X1_Ant;
Y1:= Y1_Ant;
CX_I:= X1;
CY_I:= Y1;
end;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Lineto(X1-40,Y1);
Image.Canvas.Moveto(X1-20,Y1);
Image.Canvas.Brush.Color:= clYellow;
Image.Canvas.Polygon([Point(X1-10,Y1+3),Point(X1-10,Y1-3),Point(X1-
20,Y1)]);
Image.Canvas.Brush.Color:= clWhite;
X1_Ant:= X1-40;
Y1_Ant:= Y1;
CX_F:=X1_Ant;
CY_F:=Y1_Ant;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-38,Y1-2,X1-42,Y1+2);
end;
// Grava na tabela
Sequencia:= Sequencia + 1;
Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig,"", "", "",
",", ", ", "Num, ");
Fechamento(CX_F, CY_F);

```

```

// Fim da gravação
end;
procedure TAbertura.Fig_LinhaDirAba(X1,Y1: integer);
begin
  with Abertura do
  begin
    if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
    begin
      X1:=X1_Ant;
      Y1:= Y1_Ant;
      CX_I:= X1;
      CY_I:= Y1;
    end;
    Image.Canvas.Moveto(X1,Y1);
    Image.Canvas.Lineto(X1+40,Y1);
    X1:= X1+40;
    Image.Canvas.Lineto(X1,Y1+40);
    Image.Canvas.Moveto(X1,Y1+20);
    Image.Canvas.Brush.Color:= clYellow;
    Image.Canvas.Polygon([Point(X1+3,Y1+10),Point(X1-
3,Y1+10),Point(X1,Y1+20)]);
    Image.Canvas.Brush.Color:= clWhite;
    X1_Ant:= X1;
    Y1_Ant:= Y1+40;
    CX_F:=X1_Ant;
    CY_F:=Y1_Ant;
    Image.Canvas.Brush.Color := clGray;
    Image.Canvas.Ellipse(X1-2,Y1+38,X1+2,Y1+42);
  end;
  // Grava na tabela
  Sequencia:= Sequencia + 1;
  Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig," ", " ",
  " ", " ", " ",Num, "");
  Fechamento(CX_F, CY_F);
  // Fim da gravação
end;
procedure TAbertura.Fig_LinhaEsqAba(X1,Y1: integer);
begin
  with Abertura do
  begin
    if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
    begin
      X1:=X1_Ant;
      Y1:= Y1_Ant;
      CX_I:= X1;
      CY_I:= Y1;
    end;
    Image.Canvas.Moveto(X1,Y1);
    Image.Canvas.Lineto(X1-40,Y1);
    X1:= X1-30;

```

```

Image.Canvas.Lineto(X1,Y1+40);
Image.Canvas.Moveto(X1,Y1+20);
Image.Canvas.Brush.Color:= clYellow;
Image.Canvas.Polygon([Point(X1+3,Y1+10),Point(X1-
3,Y1+10),Point(X1,Y1+20)]);
Image.Canvas.Brush.Color:= clWhite;
X1_Ant:= X1;
Y1_Ant:= Y1+40;
CX_F:=X1_Ant;
CY_F:=Y1_Ant;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1+38,X1+2,Y1+42);
end;
// Grava na tabela
Sequencia:= Sequencia + 1;
Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig,"", "",
",", "", "Num,");
Fechamento(CX_F, CY_F);
// Fim da gravação
end;
procedure TAbertura.Fig_LinhaDirAci(X1,Y1: integer);
begin
with Abertura do
begin
if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
begin
X1:=X1_Ant;
Y1:= Y1_Ant;
CX_I:= X1;
CY_I:= Y1;
end;
Image.Canvas.Moveto(X1,Y1);
Image.Canvas.Lineto(X1+40,Y1);
X1:= X1+30;
Image.Canvas.Lineto(X1,Y1-40);
Image.Canvas.Moveto(X1,Y1-20);
Image.Canvas.Brush.Color:= clYellow;
Image.Canvas.Polygon([Point(X1+3,Y1-10),Point(X1-3,Y1-10),Point(X1,Y1-
20)]);
Image.Canvas.Brush.Color:= clWhite;
X1_Ant:= X1;
Y1_Ant:= Y1-40;
CX_F:=X1_Ant;
CY_F:=Y1_Ant;
Image.Canvas.Brush.Color := clGray;
Image.Canvas.Ellipse(X1-2,Y1-38,X1+2,Y1-42);
end;
// Grava na tabela
Sequencia:= Sequencia + 1;
Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig,"", "",

```

```

        ", ", ", Num, ");
        Fechamento(CX_F, CY_F);
    // Fim da gravação
end;
procedure TAbertura.Fig_LinhaEsqAci(X1,Y1: integer);
begin
    with Abertura do
    begin
        if (Abs(X1_Ant - X1) < 10) and (Abs(Y1_Ant - Y1) < 10) then
        begin
            X1:=X1_Ant;
            Y1:= Y1_Ant;
            CX_I:= X1;
            CY_I:= Y1;
        end;
        Image.Canvas.Moveto(X1,Y1);
        Image.Canvas.Lineto(X1-40,Y1);
        X1:= X1-40;
        Image.Canvas.Lineto(X1,Y1-40);
        Image.Canvas.Moveto(X1,Y1-20);
        Image.Canvas.Brush.Color:= clYellow;
        Image.Canvas.Polygon([Point(X1+3,Y1-10),Point(X1-3,Y1-10),Point(X1,Y1-
20)]);
        Image.Canvas.Brush.Color:= clWhite;
        X1_Ant:= X1;
        Y1_Ant:= Y1-42;
        CX_F:=X1_Ant;
        CY_F:=Y1_Ant;
        Image.Canvas.Brush.Color := clGray;
        Image.Canvas.Ellipse(X1-2,Y1-38,X1+2,Y1-42);
    end;
    // Grava na tabela
        Sequencia:= Sequencia + 1;
        Gravacao_Dados(CX_I, CY_I, Sequencia, CX_F, CY_F, Fig,"", "",
        ", ", ", Num, ");
        Fechamento(CX_F, CY_F);
    // Fim da gravação
end;

// Final dos Procedimentos particulares
procedure TAbertura.Sair2Click(Sender: TObject);
begin
    close;
end;
procedure TAbertura.Sobre1Click(Sender: TObject);
begin
    Sobre.showmodal;
end;
procedure TAbertura.FormClick(Sender: TObject);
var

```

```

X11, Y11: integer;
begin
  Case Fig of
    0:
      begin
        with Tab_Grafcet do
          open;
          begin
            for X11:= X1 - 10 to X1 + 10 do
              Begin
                for Y11:= Y1 - 10 to Y1 + 10 do
                  begin
                    Tab_Grafcet.first;
                    while not Tab_Grafcet.EOF do
                      begin
                        if (Tab_Grafcet.fieldbyname('Coord_X_Fim').asinteger = X11) and
                           (Tab_Grafcet.fieldbyname('Coord_Y_Fim').asinteger = Y11) then
                          begin
                            X1_Ant:= Tab_Grafcet.FieldByName('Coord_X_Fim').asinteger;
                            Y1_Ant:= Tab_Grafcet.FieldByName('Coord_Y_Fim').asinteger;
                            break;
                          end
                        else
                          begin
                            Tab_Grafcet.next
                          end;
                        end;
                      end;
                    end;
                  Tab_Grafcet.close;
                end;
              end;
            end;
          end;
          Tab_Grafcet.close;
        end;
      end;
      1: Fig_Acao(X1,Y1);
      2: Fig_Transicao(X1,Y1);
      3: Fig_XOR_Open(X1,Y1);
      4: Fig_XOR_Close(X1,Y1);
      5: Fig_XorTri_Open(X1,Y1);
      6: Fig_XorTri_Close(X1,Y1);
      7: Fig_AndBi_Open(X1,Y1);
      8: Fig_AndTri_Open(X1,Y1);
      9: Fig_AndBi_Close(X1,Y1);
      10: Fig_AndTri_Close(X1,Y1);
      90: Fig_LinhaVertAba(X1,Y1);
      91: Fig_LinhaHorDir(X1,Y1);
      92: Fig_LinhaDirAba(X1,Y1);
      93: Fig_LinhaEsqAba(X1,Y1);
      94: Fig_LinhaDirAci(X1,Y1);
      95: Fig_LinhaEsqAci(X1,Y1);
      96: Fig_LinhaVertAci(X1,Y1);
      97: Fig_LinhaHorEsq(X1,Y1);

```

```
end;
if Fig < 89 then
  Fig:=0;
end;
procedure TAbertura.FormMouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
// StatusBar1.Panels[1].Text := Format('Corrente: (%d, %d)', [X, Y]);
end;
procedure TAbertura.FormCreate(Sender: TObject);
var
  Bitmap: TBitmap;
begin
  Num:= 0;
  Num_Acao:= 1;
  Sequencia:= 0;
  Ferram:= false;
  Acao_OK:= true;
  Trans_OK:= false;
  XH:= 30;
  YH:= 60;
  Obj:= 0;
  Bitmap := TBitmap.Create;
  Bitmap.Width := 1230;
  Bitmap.Height := 1645;
  Image.Picture.Graphic := Bitmap;
  with Tab_Grafcet do
  begin
    open;
    first;
    while not eof do
    begin
      delete;
    end;
    refresh;
    close;
  end;
  with Tab_Digital do
  begin
    open;
    first;
    while not eof do
    begin
      delete;
    end;
    refresh;
    close;
  end;
end;
end;
procedure TAbertura.AcaoButtonClick(Sender: TObject);
```



```
    RestoreStyles;
  end;
end;
procedure TAbertura.XorCloseButtonClick(Sender: TObject);
begin
  Fig:= 4;
end;
procedure TAbertura.AndBiOpenButtonClick(Sender: TObject);
begin
  Fig:= 7;
end;
procedure TAbertura.AndTriOpenButtonClick(Sender: TObject);
begin
  Fig:= 8;
end;
procedure TAbertura.AndBiCloseButtonClick(Sender: TObject);
begin
  Fig:= 9;
end;
procedure TAbertura.AndTriCloseButtonClick(Sender: TObject);
begin
  Fig:= 10;
end;
procedure TAbertura.LinhaVertButtonClick(Sender: TObject);
begin
  Fig:= 90;
end;
procedure TAbertura.LinhaHorDirButtonClick(Sender: TObject);
begin
  Fig:= 91;
end;
procedure TAbertura.DirAbaButtonClick(Sender: TObject);
begin
  Fig:= 92;
end;
procedure TAbertura.EsqAbaButtonClick(Sender: TObject);
begin
  Fig:= 93;
end;
procedure TAbertura.DirAciButtonClick(Sender: TObject);
begin
  Fig:= 94;
end;
procedure TAbertura.EsqAciButtonClick(Sender: TObject);
begin
  Fig:= 95;
end;
procedure TAbertura.SpeedButton1Click(Sender: TObject);
begin
  Fig:= 96;
```



```
end;
procedure TAbertura.linhaHorEsqButtonClick(Sender: TObject);
begin
  Fig:= 97;
end;
procedure TAbertura.EsquemaDigital1Click(Sender: TObject);
begin
  Sist_Digital.showmodal;
end;
procedure TAbertura.Imprimir1Click(Sender: TObject);
var
  Bmp: TBitmap;
begin
  if printdialog1.execute then
  begin
    Bmp := TBitmap.Create;
    Image.Picture.SaveToFile('C:\HlsGrafcet.tmp');
    try
      Bmp.LoadFromFile('C:\HlsGrafcet.tmp');
      with Printer do
      begin
        BeginDoc;
        Canvas.Draw((PageWidth - Bmp.Width) div 2,
                    (PageHeight - Bmp.Height) div 2,
                    Bmp);
        EndDoc;
      end;
    finally
      Bmp.Free;
    end;
  end;
end;
procedure TAbertura.Visvel1Click(Sender: TObject);
begin
  if not Ferram then
  begin
    Visvel1.ImageIndex:= 7;
    Ferramentas.Visible:= true;
    Ferram:= true;
  end
  else
  begin
    Visvel1.ImageIndex:= 8;
    Ferramentas.Visible:= false;
    Ferram:= false;
  end;
end;
procedure TAbertura.FormActivate(Sender: TObject);
begin
  if Titulo.Edit1.text = 'TÍTULO' then
```

```
begin
  Titulo.showmodal;
end;
Image.Canvas.Font.Height:= 30;
Image.Canvas.Font.Color:= clBlue;
Image.Canvas.TextOut(500,30,Titulo.Edit1.Text);
Image.Canvas.Font.Height:= 11;
Image.Canvas.Font.Color:= clBlack;
end;
procedure TAbertura.XorTriOpenClick(Sender: TObject);
begin
  Fig:= 5;
end;
procedure TAbertura.XorTriCloseClick(Sender: TObject);
begin
  Fig:= 6;
end;
end.
```

```
unit About;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, jpeg, ExtCtrls, Buttons;
type
  TSobre = class(TForm)
    Image1: TImage;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Sobre: TSobre;
implementation
{$R *.DFM}
procedure TSobre.BitBtn1Click(Sender: TObject);
begin
  close;
end;
end.
```

```

unit Descricao;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, Mask, ExtCtrls;
type
  TDescr_Obj = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    DescrEdit1: TEdit;
    Label7: TLabel;
    Label8: TLabel;
    BitBtn1: TBitBtn;
    DescrEdit2: TEdit;
    DescrEdit3: TEdit;
    Descr1CheckBox1: TCheckBox;
    Descr2CheckBox2: TCheckBox;
    Descr3CheckBox3: TCheckBox;
    Label9: TLabel;
    Shape1: TShape;
    Shape2: TShape;
    Edit_Apelido: TEdit;
    procedure FormActivate(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure DescrEdit1Exit(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Descr_Obj: TDescr_Obj;
implementation
uses Inicio;
{$R *.DFM}
procedure TDescr_Obj.FormActivate(Sender: TObject);
begin
  Edit_Apelido.setfocus;
  Edit_Apelido.text:="";
  DescrEdit1.text:="";
  Descr1CheckBox1.Checked:= false;
  Descr2CheckBox2.Checked:= false;
  Descr3CheckBox3.Checked:= false;
  if Abertura.Fig = 1 then
  begin
    DescrEdit2.enabled:= false;

```

```
DescrEdit2.visible:= false;
DescrEdit3.enabled:= false;
DescrEdit3.visible:= false;
label9.visible := false;
shape1.visible := false;
shape2.Visible := false;
label6.top:= 78;
label7.caption:= 'Ação';
label8.caption:= IntToStr(Abertura.Num);
Descr1CheckBox1.Visible:= false;
Descr2CheckBox2.Visible:= false;
Descr3CheckBox3.Visible:= false;
end
else
begin
  label6.top:= 100;
  DescrEdit2.enabled:= true;
  DescrEdit2.visible:= true;
  DescrEdit3.enabled:= true;
  DescrEdit3.visible:= true;
  DescrEdit2.text:= '1';
  DescrEdit3.text:= '1';
  label9.visible := true;
  shape1.visible := true;
  shape2.Visible := true;
  label7.caption:= 'Transição';
  label8.caption:= '.';
  Descr1CheckBox1.Visible:= true;
  Descr2CheckBox2.Visible:= true;
  Descr3CheckBox3.Visible:= true;
end;
end;
procedure TDescr_Obj.BitBtn1Click(Sender: TObject);
begin
  close;
end;
procedure TDescr_Obj.DescrEdit1Exit(Sender: TObject);
begin
  if (DescrEdit1.text = "") and (Abertura.Fig = 2) then
  begin
    MessageDlg('A TRANSIÇÃO deve ter pelo menos a primeira mudança de
estado', mtInformation,
      [mbOk], 0);
    DescrEdit1.setfocus;
  end;
end;
end.
```

```

unit Digital;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, Menus, StdCtrls, Printers, ComCtrls, ImgList, ToolWin, Db,
  DBTables;
type
  TSist_Digital = class(TForm)
    MainMenu1: TMainMenu;
    Arquivo1: TMenuItem;
    Imprimir1: TMenuItem;
    N1: TMenuItem;
    Fechar1: TMenuItem;
    PrintDialog1: TPrintDialog;
    ToolBar1: TToolBar;
    ToolbarImages: TImageList;
    ToolButton1: TToolButton;
    ToolButton2: TToolButton;
    Table1: TTable;
    ScrollBox1: TScrollBox;
    Image: TImage;
    procedure FormCreate(Sender: TObject);
    procedure Fechar1Click(Sender: TObject);
    procedure Imprimir1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }

    XH,YH,Obj: Integer;
  end;
var
  Sist_Digital: TSist_Digital;
implementation
{$R *.DFM}
procedure TSist_Digital.FormCreate(Sender: TObject);
var
  Bitmap: TBitmap;
begin
  XH:= 30;
  YH:= 30;
  Obj:= 0;
  Bitmap := TBitmap.Create;
  Bitmap.Width := 1230;
  Bitmap.Height := 1645;
  Image.Picture.Graphic := Bitmap;
end;
procedure TSist_Digital.Fechar1Click(Sender: TObject);

```

```

begin
  Image.Picture.Free;
  close;
end;
procedure TSist_Digital.Imprimir1Click(Sender: TObject);
var
  Bmp: TBitmap;
begin
  if printdialog1.execute then
  begin
    Bmp := TBitmap.Create;
    Image.Picture.SaveToFile('C:\HlsGrafcet.tmp');
    try
      Bmp.LoadFromFile('C:\HlsGrafcet.tmp');
      with Printer do
      begin
        BeginDoc;
        Canvas.Draw((PageWidth - Bmp.Width) div 2,
                    (PageHeight - Bmp.Height) div 2,
                    Bmp);
        EndDoc;
      end;
    finally
      Bmp.Free;
    end;
  end;
end;
procedure TSist_Digital.FormActivate(Sender: TObject);
var
  Num: Integer;
begin
  with table1 do
  begin
    open;
    first;
    begin
      while not Eof do
      begin
        Num:= fieldbyname('Cod_Obj').asInteger;
        // Montagem do equivalente digital
        case fieldbyname('Tipo').asInteger of
          1:
            begin
              Image.Canvas.Pen.Color:= clMaroon;
              Image.Canvas.Pen.Style:= psDot;
              Image.Canvas.Rectangle(XH,YH,XH+115,YH+70);
              Image.Canvas.Pen.Style:= psSolid;
              Image.Canvas.Pen.Color:= clBlack;
              Image.Canvas.Rectangle(XH+70,YH+5,XH+105,YH+35);
            end;
        end;
      end;
    end;
  end;
end;

```

```
Image.Canvas.Moveto(XH+70,YH+20);
Image.Canvas.LineTo(XH+105,YH+20);
Image.Canvas.Moveto(XH+105,YH+12);
Image.Canvas.LineTo(XH+125,YH+12);
Image.Canvas.Pen.Style:= psSolid;
Image.Canvas.Pen.Color:= clRed;
Image.Canvas.Moveto(XH+60,YH+27);
Image.Canvas.LineTo(XH+70,YH+27);
Image.Canvas.MoveTo(XH+60, YH+27);
Image.Canvas.LineTo(XH+60,YH+45);
Image.Canvas.LineTo(XH+125,YH+45);
Image.Canvas.LineTo(XH+125,YH+40);
Image.Canvas.LineTo(XH+120,YH+45);
Image.Canvas.LineTo(XH+125,YH+50);
Image.Canvas.LineTo(XH+125,YH+45);
Image.Canvas.MoveTo(XH+110, YH+12);
Image.Canvas.LineTo(XH+110,YH+30);
Image.Canvas.LineTo(XH+110,YH+30);
Image.Canvas.LineTo(XH+125,YH+30);
Image.Canvas.LineTo(XH+120,YH+25);
Image.Canvas.LineTo(XH+120,YH+35);
Image.Canvas.LineTo(XH+125,YH+30);
Image.Canvas.Pen.Color:= clBlack;
Image.Canvas.Arc(XH+5,YH+10,XH+45,YH+60,0,0,0,0);
Image.Canvas.Pen.Color:= clWhite;
Image.Canvas.Rectangle(XH+5, YH+10,XH+25,YH+60);
Image.Canvas.Pen.Color:= clBlack;
Image.Canvas.Moveto(XH+25,YH+10);
Image.Canvas.LineTo(XH+25,YH+60);
Image.Canvas.Moveto(XH+45,YH+35);
Image.Canvas.LineTo(XH+50,YH+35);
Image.Canvas.LineTo(XH+50,YH+12);
Image.Canvas.LineTo(XH+70,YH+12);
Image.Canvas.Moveto(XH+25,YH+15);
Image.Canvas.LineTo(XH+5,YH+15);
Image.Canvas.LineTo(XH+5,YH+115);
Image.Canvas.LineTo(XH+25,YH+115);
Image.Canvas.Moveto(XH+25,YH+27);
Image.Canvas.LineTo(XH+10,YH+27);
Image.Canvas.LineTo(XH+10,YH+105);
Image.Canvas.LineTo(XH+25,YH+105);
Image.Canvas.Moveto(XH+25,YH+39);
Image.Canvas.LineTo(XH+15,YH+39);
Image.Canvas.LineTo(XH+15,YH+95);
Image.Canvas.LineTo(XH+25,YH+95);
Image.Canvas.Moveto(XH+25,YH+51);
Image.Canvas.LineTo(XH+20,YH+51);
Image.Canvas.LineTo(XH+20,YH+85);
Image.Canvas.LineTo(XH+25,YH+85);
Image.Canvas.TextOut(XH+80,YH+7,'M'+IntToStr(Num));
```



```

Image.Canvas.TextOut(XH+30,YH+75,Fieldbyname('Entrada1').asstring);
Image.Canvas.TextOut(XH+30,YH+87,Fieldbyname('Entrada2').asstring);
Image.Canvas.TextOut(XH+30,YH+99,Fieldbyname('Entrada3').asstring);
Image.Canvas.TextOut(XH+30,YH+110,'M'+
  IntToStr(Fieldbyname('Anterior1').asinteger));
Image.Canvas.TextOut(XH+130,YH+25,'M'+
  IntToStr(Fieldbyname('Anterior1').asinteger));
Image.Canvas.TextOut(XH+130,YH+5,'M'+
  IntToStr(FieldByName('Proximo1').asinteger));
Image.Canvas.TextOut(XH+130,YH+40,'M'+
  IntToStr(FieldByName('Proximo1').asinteger));
if FieldByName('Nega_Ent1').asstring = 'N' then
begin
Image.Canvas.Brush.Color := clBlack;
Image.Canvas.Ellipse(XH+5-2+19,YH+47+2,XH+5+2+19,YH+47+4+3);
Image.Canvas.Brush.Color := clWhite;
end;
if FieldByName('Nega_Ent2').asstring = 'N' then
begin
Image.Canvas.Brush.Color := clBlack;
Image.Canvas.Ellipse(XH+5-2+19,YH+35+2,XH+5+2+19,YH+35+4+3);
Image.Canvas.Brush.Color := clWhite;
end;
if FieldByName('Nega_Ent3').asstring = 'N' then
begin
Image.Canvas.Brush.Color := clBlack;
Image.Canvas.Ellipse(XH+5-2+19,YH+24+2,XH+5+2+19,YH+24+4+3);
Image.Canvas.Brush.Color := clWhite;
end;
end;
2:
begin
Image.Canvas.Pen.Color:= clMaroon;
Image.Canvas.Pen.Style:= psDot;
Image.Canvas.Rectangle(XH,YH,XH+115,YH+70);
Image.Canvas.Pen.Style:= psSolid;
Image.Canvas.Pen.Color:= clBlack;
Image.Canvas.Arc(XH+35,YH+10,XH+75,YH+60,0,0,0,0);
Image.Canvas.Pen.Color:= clWhite;
Image.Canvas.Rectangle(XH+25,YH+10,XH+55,YH+60);
Image.Canvas.Pen.Color:= clBlack;
Image.Canvas.Moveto(XH+55,YH+10);
Image.Canvas.LineTo(XH+55,YH+60);
Image.Canvas.Moveto(XH+75,YH+35);
Image.Canvas.LineTo(XH+125,YH+35);
Image.Canvas.Moveto(XH+55,YH+15);
Image.Canvas.LineTo(XH+5,YH+15);

```

```

Image.Canvas.LineTo(XH+5,YH+115);
Image.Canvas.LineTo(XH+25,YH+115);
Image.Canvas.Moveto(XH+55,YH+27);
Image.Canvas.LineTo(XH+10,YH+27);
Image.Canvas.LineTo(XH+10,YH+105);
Image.Canvas.LineTo(XH+25,YH+105);
Image.Canvas.Moveto(XH+55,YH+39);
Image.Canvas.LineTo(XH+15,YH+39);
Image.Canvas.LineTo(XH+15,YH+95);
Image.Canvas.LineTo(XH+25,YH+95);
Image.Canvas.Moveto(XH+55,YH+51);
Image.Canvas.LineTo(XH+20,YH+51);
Image.Canvas.LineTo(XH+20,YH+85);
Image.Canvas.LineTo(XH+25,YH+85);
Image.Canvas.TextOut(XH+30,YH+75,'M'+IntToStr(1));
Image.Canvas.TextOut(XH+30,YH+87,'M'+IntToStr(2));
Image.Canvas.TextOut(XH+30,YH+99,'M'+IntToStr(3));
Image.Canvas.TextOut(XH+30,YH+110,'M'+IntToStr(4));
Image.Canvas.TextOut(XH+130,YH+30,'M'+IntToStr(5));
end;
3:
begin
Image.Canvas.Brush.Color := clRed;
Image.Canvas.Pen.Color:= clMaroon;
Image.Canvas.Pen.Style:= psDot;
Image.Canvas.Rectangle(XH,YH,XH+115,YH+70);
Image.Canvas.Pen.Style:= psSolid;
Image.Canvas.Pen.Color:= clBlack;
Image.Canvas.Brush.Color := clYellow;
Image.Canvas.Rectangle(XH+45,YH+10,XH+75,YH+60);
Image.Canvas.TextOut(XH+50,YH+30,'OU');
Image.Canvas.Brush.Color := clWhite;
Image.Canvas.Moveto(XH+45,YH+15);
Image.Canvas.LineTo(XH+5,YH+15);
Image.Canvas.LineTo(XH+5,YH+115);
Image.Canvas.LineTo(XH+25,YH+115);
Image.Canvas.Moveto(XH+45,YH+27);
Image.Canvas.LineTo(XH+10,YH+27);
Image.Canvas.LineTo(XH+10,YH+105);
Image.Canvas.LineTo(XH+25,YH+105);
Image.Canvas.Moveto(XH+45,YH+39);
Image.Canvas.LineTo(XH+15,YH+39);
Image.Canvas.LineTo(XH+15,YH+95);
Image.Canvas.LineTo(XH+25,YH+95);
Image.Canvas.Moveto(XH+45,YH+51);
Image.Canvas.LineTo(XH+20,YH+51);
Image.Canvas.LineTo(XH+20,YH+85);
Image.Canvas.LineTo(XH+25,YH+85);
Image.Canvas.Moveto(XH+75,YH+35);
Image.Canvas.LineTo(XH+125,YH+35);

```

```
Image.Canvas.TextOut(XH+30,YH+75,'M'+IntToStr(1));
Image.Canvas.TextOut(XH+30,YH+87,'M'+IntToStr(2));
Image.Canvas.TextOut(XH+30,YH+99,'M'+IntToStr(3));
Image.Canvas.TextOut(XH+30,YH+110,'M'+IntToStr(4));
Image.Canvas.TextOut(XH+130,YH+30,'M'+IntToStr(5));
end;
end;
XH:= XH + 170;
if XH > 1200 then
begin
  YH:= YH+200;
  XH:= 30;
end;
next;
end;
end;
close;
end;
end;
procedure TSist_Digital.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
// Image.Picture.Free;
// Image.Picture.Destroy;
end;
end.
```

```
unit Titul;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons;
type
  TTitulo = class(TForm)
    Edit1: TEdit;
    Label1: TLabel;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Titulo: TTitulo;
implementation
{$R *.DFM}
procedure TTitulo.BitBtn1Click(Sender: TObject);
begin
  close;
end;
end.
```

## Apêndice 2 – Listagem do programa escrito em VHDL para gravação do FPGA

-- Vhdl model created from schematic esquema.sch - Tue Jun 29 21:03:09 2004

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
-- synopsys translate_off
LIBRARY UNISIM;
USE UNISIM.Vcomponents.ALL;
-- synopsys translate_on
```

```
ENTITY esquema IS
  PORT ( A : IN STD_LOGIC;
        B : IN STD_LOGIC;
        C : IN STD_LOGIC;
        D : IN STD_LOGIC;
        E : IN STD_LOGIC;
        F : IN STD_LOGIC;
        G : IN STD_LOGIC;
        H : IN STD_LOGIC;
        I : IN STD_LOGIC;
        P : IN STD_LOGIC;
        P2 : IN STD_LOGIC;
        R : IN STD_LOGIC;
        Start : IN STD_LOGIC;
        W : IN STD_LOGIC;
        X : IN STD_LOGIC;
        Y : IN STD_LOGIC;
        Z : IN STD_LOGIC;
        M1 : OUT STD_LOGIC;
        M10 : OUT STD_LOGIC;
        M11 : OUT STD_LOGIC;
        M12 : OUT STD_LOGIC;
        M2 : OUT STD_LOGIC;
```

```
M3 : OUT STD_LOGIC;  
M4 : OUT STD_LOGIC;  
M5 : OUT STD_LOGIC;  
M6 : OUT STD_LOGIC;  
M7 : OUT STD_LOGIC;  
M8 : OUT STD_LOGIC;  
M9 : OUT STD_LOGIC);
```

end esquema;

ARCHITECTURE SCHEMATIC OF esquema IS

```
SIGNAL XLXN_100 : STD_LOGIC;  
SIGNAL XLXN_102 : STD_LOGIC;  
SIGNAL XLXN_103 : STD_LOGIC;  
SIGNAL XLXN_104 : STD_LOGIC;  
SIGNAL XLXN_108 : STD_LOGIC;  
SIGNAL XLXN_109 : STD_LOGIC;  
SIGNAL XLXN_110 : STD_LOGIC;  
SIGNAL XLXN_111 : STD_LOGIC;  
SIGNAL XLXN_112 : STD_LOGIC;  
SIGNAL XLXN_113 : STD_LOGIC;  
SIGNAL XLXN_115 : STD_LOGIC;  
SIGNAL XLXN_117 : STD_LOGIC;  
SIGNAL XLXN_119 : STD_LOGIC;  
SIGNAL XLXN_126 : STD_LOGIC;  
SIGNAL XLXN_127 : STD_LOGIC;  
SIGNAL XLXN_128 : STD_LOGIC;  
SIGNAL XLXN_132 : STD_LOGIC;  
SIGNAL XLXN_134 : STD_LOGIC;  
SIGNAL XLXN_135 : STD_LOGIC;  
SIGNAL XLXN_19 : STD_LOGIC;  
SIGNAL XLXN_20 : STD_LOGIC;  
SIGNAL XLXN_27 : STD_LOGIC;  
SIGNAL XLXN_28 : STD_LOGIC;  
SIGNAL XLXN_30 : STD_LOGIC;  
SIGNAL XLXN_32 : STD_LOGIC;  
SIGNAL XLXN_33 : STD_LOGIC;  
SIGNAL XLXN_37 : STD_LOGIC;  
SIGNAL XLXN_38 : STD_LOGIC;  
SIGNAL XLXN_39 : STD_LOGIC;  
SIGNAL XLXN_40 : STD_LOGIC;  
SIGNAL XLXN_41 : STD_LOGIC;  
SIGNAL XLXN_42 : STD_LOGIC;  
SIGNAL XLXN_43 : STD_LOGIC;  
SIGNAL XLXN_44 : STD_LOGIC;  
SIGNAL XLXN_45 : STD_LOGIC;  
SIGNAL XLXN_46 : STD_LOGIC;  
SIGNAL XLXN_47 : STD_LOGIC;  
SIGNAL XLXN_48 : STD_LOGIC;  
SIGNAL XLXN_49 : STD_LOGIC;
```

```
SIGNAL XLXN_50      :      STD_LOGIC;
SIGNAL XLXN_51      :      STD_LOGIC;
SIGNAL XLXN_52      :      STD_LOGIC;
SIGNAL XLXN_53      :      STD_LOGIC;
SIGNAL XLXN_54      :      STD_LOGIC;
SIGNAL XLXN_57      :      STD_LOGIC;
SIGNAL XLXN_58      :      STD_LOGIC;
SIGNAL XLXN_59      :      STD_LOGIC;
SIGNAL XLXN_60      :      STD_LOGIC;
SIGNAL XLXN_61      :      STD_LOGIC;
SIGNAL XLXN_62      :      STD_LOGIC;
SIGNAL XLXN_63      :      STD_LOGIC;
SIGNAL XLXN_64      :      STD_LOGIC;
SIGNAL XLXN_65      :      STD_LOGIC;
SIGNAL XLXN_66      :      STD_LOGIC;
SIGNAL XLXN_67      :      STD_LOGIC;
SIGNAL XLXN_68      :      STD_LOGIC;
SIGNAL XLXN_69      :      STD_LOGIC;
SIGNAL XLXN_72      :      STD_LOGIC;
SIGNAL XLXN_73      :      STD_LOGIC;
SIGNAL XLXN_74      :      STD_LOGIC;
SIGNAL XLXN_75      :      STD_LOGIC;
SIGNAL XLXN_76      :      STD_LOGIC;
SIGNAL XLXN_77      :      STD_LOGIC;
SIGNAL XLXN_78      :      STD_LOGIC;
SIGNAL XLXN_79      :      STD_LOGIC;
SIGNAL XLXN_80      :      STD_LOGIC;
SIGNAL XLXN_81      :      STD_LOGIC;
SIGNAL XLXN_82      :      STD_LOGIC;
SIGNAL XLXN_84      :      STD_LOGIC;
SIGNAL XLXN_85      :      STD_LOGIC;
SIGNAL XLXN_87      :      STD_LOGIC;
SIGNAL XLXN_88      :      STD_LOGIC;
SIGNAL XLXN_89      :      STD_LOGIC;
SIGNAL XLXN_91      :      STD_LOGIC;
SIGNAL XLXN_93      :      STD_LOGIC;
SIGNAL XLXN_96      :      STD_LOGIC;
SIGNAL XLXN_97      :      STD_LOGIC;
SIGNAL XLXN_98      :      STD_LOGIC;
```

```
ATTRIBUTE BOX_TYPE : STRING;
```

```
COMPONENT AND2
```

```
  PORT ( I0      :      IN   STD_LOGIC;
         I1      :      IN   STD_LOGIC;
         O       :      OUT  STD_LOGIC);
END COMPONENT;
```

```
ATTRIBUTE BOX_TYPE OF AND2 : COMPONENT IS "BLACK_BOX";
COMPONENT AND3
```

```

PORT ( I0      :      IN   STD_LOGIC;
       I1      :      IN   STD_LOGIC;
       I2      :      IN   STD_LOGIC;
       O       :      OUT  STD_LOGIC);
END COMPONENT;

```

```

ATTRIBUTE BOX_TYPE OF AND3 : COMPONENT IS "BLACK_BOX";
COMPONENT AND4

```

```

PORT ( I0      :      IN   STD_LOGIC;
       I1      :      IN   STD_LOGIC;
       I2      :      IN   STD_LOGIC;
       I3      :      IN   STD_LOGIC;
       O       :      OUT  STD_LOGIC);
END COMPONENT;

```

```

ATTRIBUTE BOX_TYPE OF AND4 : COMPONENT IS "BLACK_BOX";
COMPONENT FDCP

```

```

PORT ( C       :      IN   STD_LOGIC;
       CLR     :      IN   STD_LOGIC;
       D       :      IN   STD_LOGIC;
       PRE     :      IN   STD_LOGIC;
       Q       :      OUT  STD_LOGIC);
END COMPONENT;

```

```

ATTRIBUTE BOX_TYPE OF FDCP : COMPONENT IS "BLACK_BOX";
COMPONENT GND

```

```

PORT ( G       :      OUT  STD_LOGIC);
END COMPONENT;

```

```

ATTRIBUTE BOX_TYPE OF GND : COMPONENT IS "BLACK_BOX";
COMPONENT INV

```

```

PORT ( I       :      IN   STD_LOGIC;
       O       :      OUT  STD_LOGIC);
END COMPONENT;

```

```

ATTRIBUTE BOX_TYPE OF INV : COMPONENT IS "BLACK_BOX";
COMPONENT OR2

```

```

PORT ( I0      :      IN   STD_LOGIC;
       I1      :      IN   STD_LOGIC;
       O       :      OUT  STD_LOGIC);
END COMPONENT;

```

```

ATTRIBUTE BOX_TYPE OF OR2 : COMPONENT IS "BLACK_BOX";
COMPONENT OR3

```

```

PORT ( I0      :      IN   STD_LOGIC;
       I1      :      IN   STD_LOGIC;
       I2      :      IN   STD_LOGIC;
       O       :      OUT  STD_LOGIC);
END COMPONENT;

```



```
ATTRIBUTE BOX_TYPE OF OR3 : COMPONENT IS "BLACK_BOX";  
COMPONENT OR4
```

```
    PORT ( I0      :      IN      STD_LOGIC;  
           I1      :      IN      STD_LOGIC;  
           I2      :      IN      STD_LOGIC;  
           I3      :      IN      STD_LOGIC;  
           O       :      OUT     STD_LOGIC);  
END COMPONENT;
```

```
ATTRIBUTE BOX_TYPE OF OR4 : COMPONENT IS "BLACK_BOX";  
BEGIN
```

```
XLXI_60 : AND2  
    PORT MAP (I0=>Z, I1=>XLXN_73, O=>XLXN_61);  
  
XLXI_59 : AND2  
    PORT MAP (I0=>Z, I1=>XLXN_74, O=>XLXN_60);  
  
XLXI_58 : AND2  
    PORT MAP (I0=>Z, I1=>XLXN_75, O=>XLXN_62);  
  
XLXI_57 : AND2  
    PORT MAP (I0=>Z, I1=>XLXN_78, O=>XLXN_64);  
  
XLXI_56 : AND2  
    PORT MAP (I0=>Z, I1=>XLXN_77, O=>XLXN_63);  
  
XLXI_96 : AND2  
    PORT MAP (I0=>W, I1=>XLXN_119, O=>XLXN_115);  
  
XLXI_55 : AND2  
    PORT MAP (I0=>Z, I1=>XLXN_76, O=>XLXN_65);  
  
XLXI_61 : AND2  
    PORT MAP (I0=>Z, I1=>XLXN_79, O=>XLXN_59);  
  
XLXI_62 : AND2  
    PORT MAP (I0=>Z, I1=>XLXN_80, O=>XLXN_57);  
  
XLXI_63 : AND2  
    PORT MAP (I0=>Z, I1=>XLXN_81, O=>XLXN_58);  
  
XLXI_82 : AND2  
    PORT MAP (I0=>XLXN_100, I1=>H, O=>XLXN_98);  
  
XLXI_90 : AND2  
    PORT MAP (I0=>C, I1=>XLXN_117, O=>XLXN_113);  
  
XLXI_92 : AND2  
    PORT MAP (I0=>XLXN_108, I1=>I, O=>XLXN_109);
```

XLXI\_79 : AND3

PORT MAP (I0=>P2, I1=>G, I2=>XLXN\_91, O=>XLXN\_97);

XLXI\_83 : AND3

PORT MAP (I0=>XLXN\_89, I1=>XLXN\_50, I2=>XLXN\_54, O=>XLXN\_69);

XLXI\_94 : AND3

PORT MAP (I0=>B, I1=>A, I2=>XLXN\_111, O=>XLXN\_112);

XLXI\_23 : AND4

PORT MAP (I0=>P, I1=>E, I2=>D, I3=>XLXN\_135, O=>XLXN\_39);

XLXI\_36 : AND4

PORT MAP (I0=>P, I1=>R, I2=>XLXN\_37, I3=>XLXN\_20, O=>XLXN\_45);

XLXI\_24 : AND4

PORT MAP (I0=>P, I1=>XLXN\_30, I2=>XLXN\_28, I3=>XLXN\_135,  
O=>XLXN\_40);

XLXI\_35 : AND4

PORT MAP (I0=>P, I1=>XLXN\_38, I2=>F, I3=>XLXN\_20, O=>XLXN\_46);

XLXI\_34 : AND4

PORT MAP (I0=>P, I1=>XLXN\_38, I2=>XLXN\_37, I3=>XLXN\_20,  
O=>XLXN\_44);

XLXI\_25 : AND4

PORT MAP (I0=>P, I1=>XLXN\_30, I2=>D, I3=>XLXN\_135, O=>XLXN\_51);

XLXI\_26 : AND4

PORT MAP (I0=>P, I1=>XLXN\_28, I2=>E, I3=>XLXN\_135, O=>XLXN\_52);

XLXI\_33 : AND4

PORT MAP (I0=>P, I1=>R, I2=>F, I3=>XLXN\_20, O=>XLXN\_43);

XLXI\_78 : AND4

PORT MAP (I0=>XLXN\_91, I1=>W, I2=>P2, I3=>XLXN\_93, O=>XLXN\_88);

XLXI\_103 : AND4

PORT MAP (I0=>P, I1=>XLXN\_32, I2=>Y, I3=>XLXN\_19, O=>XLXN\_132);

XLXI\_29 : AND4

PORT MAP (I0=>P, I1=>Y, I2=>X, I3=>XLXN\_19, O=>XLXN\_41);

XLXI\_30 : AND4

PORT MAP (I0=>P, I1=>XLXN\_33, I2=>XLXN\_32, I3=>XLXN\_19,  
O=>XLXN\_42);

XLXI\_31 : AND4  
PORT MAP (I0=>P, I1=>XLXN\_33, I2=>X, I3=>XLXN\_19, O=>XLXN\_49);

XLXI\_95 : FDCP  
PORT MAP (C=>XLXN\_127, CLR=>XLXN\_119, D=>XLXN\_127,  
PRE=>XLXN\_112,  
Q=>XLXN\_117);

XLXI\_67 : FDCP  
PORT MAP (C=>XLXN\_72, CLR=>XLXN\_91, D=>XLXN\_72,  
PRE=>XLXN\_66, Q=>XLXN\_54);

XLXI\_68 : FDCP  
PORT MAP (C=>XLXN\_72, CLR=>XLXN\_91, D=>XLXN\_72,  
PRE=>XLXN\_67, Q=>XLXN\_50);

XLXI\_69 : FDCP  
PORT MAP (C=>XLXN\_72, CLR=>XLXN\_91, D=>XLXN\_72,  
PRE=>XLXN\_68, Q=>XLXN\_89);

XLXI\_71 : FDCP  
PORT MAP (C=>XLXN\_72, CLR=>XLXN\_96, D=>XLXN\_72,  
PRE=>XLXN\_69, Q=>XLXN\_91);

XLXI\_80 : FDCP  
PORT MAP (C=>XLXN\_102, CLR=>XLXN\_108, D=>XLXN\_102,  
PRE=>XLXN\_97,  
Q=>XLXN\_100);

XLXI\_81 : FDCP  
PORT MAP (C=>XLXN\_103, CLR=>XLXN\_111, D=>XLXN\_103,  
PRE=>XLXN\_98,  
Q=>XLXN\_108);

XLXI\_89 : FDCP  
PORT MAP (C=>XLXN\_128, CLR=>XLXN\_104, D=>XLXN\_128,  
PRE=>XLXN\_113,  
Q=>XLXN\_119);

XLXI\_93 : FDCP  
PORT MAP (C=>XLXN\_126, CLR=>XLXN\_117, D=>XLXN\_126,  
PRE=>XLXN\_110,  
Q=>XLXN\_111);

XLXI\_47 : FDCP  
PORT MAP (C=>XLXN\_134, CLR=>XLXN\_50, D=>XLXN\_134,  
PRE=>XLXN\_49,  
Q=>XLXN\_74);

XLXI\_46 : FDCP

```
PORT MAP (C=>XLXN_134, CLR=>XLXN_50, D=>XLXN_134,  
PRE=>XLXN_132,  
Q=>XLXN_75);
```

```
XLXI_45 : FDCP  
PORT MAP (C=>XLXN_134, CLR=>XLXN_89, D=>XLXN_134,  
PRE=>XLXN_47,  
Q=>XLXN_78);
```

```
XLXI_44 : FDCP  
PORT MAP (C=>XLXN_134, CLR=>XLXN_89, D=>XLXN_134,  
PRE=>XLXN_46,  
Q=>XLXN_77);
```

```
XLXI_43 : FDCP  
PORT MAP (C=>XLXN_134, CLR=>XLXN_89, D=>XLXN_134,  
PRE=>XLXN_45,  
Q=>XLXN_76);
```

```
XLXI_48 : FDCP  
PORT MAP (C=>XLXN_134, CLR=>XLXN_50, D=>XLXN_134,  
PRE=>XLXN_48,  
Q=>XLXN_73);
```

```
XLXI_32 : FDCP  
PORT MAP (C=>XLXN_27, CLR=>XLXN_85, D=>XLXN_27,  
PRE=>XLXN_87, Q=>XLXN_20);
```

```
XLXI_28 : FDCP  
PORT MAP (C=>XLXN_27, CLR=>XLXN_84, D=>XLXN_27,  
PRE=>XLXN_87, Q=>XLXN_19);
```

```
XLXI_27 : FDCP  
PORT MAP (C=>XLXN_27, CLR=>XLXN_82, D=>XLXN_27,  
PRE=>XLXN_87,  
Q=>XLXN_135);
```

```
XLXI_52 : FDCP  
PORT MAP (C=>XLXN_134, CLR=>XLXN_54, D=>XLXN_134,  
PRE=>XLXN_52,  
Q=>XLXN_79);
```

```
XLXI_53 : FDCP  
PORT MAP (C=>XLXN_134, CLR=>XLXN_54, D=>XLXN_134,  
PRE=>XLXN_51,  
Q=>XLXN_80);
```

```
XLXI_54 : FDCP  
PORT MAP (C=>XLXN_134, CLR=>XLXN_54, D=>XLXN_134,  
PRE=>XLXN_53,
```

```
Q=>XLXN_81);

XLXI_99 : GND
  PORT MAP (G=>XLXN_128);

XLXI_87 : GND
  PORT MAP (G=>XLXN_102);

XLXI_100 : GND
  PORT MAP (G=>XLXN_127);

XLXI_86 : GND
  PORT MAP (G=>XLXN_103);

XLXI_101 : GND
  PORT MAP (G=>XLXN_126);

XLXI_76 : GND
  PORT MAP (G=>XLXN_27);

XLXI_73 : GND
  PORT MAP (G=>XLXN_134);

XLXI_70 : GND
  PORT MAP (G=>XLXN_72);

XLXI_102 : INV
  PORT MAP (I=>XLXN_117, O=>M2);

XLXI_98 : INV
  PORT MAP (I=>XLXN_119, O=>M3);

XLXI_97 : INV
  PORT MAP (I=>XLXN_111, O=>M1);

XLXI_104 : INV
  PORT MAP (I=>XLXN_135, O=>M4);

XLXI_105 : INV
  PORT MAP (I=>XLXN_19, O=>M5);

XLXI_37 : INV
  PORT MAP (I=>D, O=>XLXN_28);

XLXI_38 : INV
  PORT MAP (I=>E, O=>XLXN_30);

XLXI_39 : INV
  PORT MAP (I=>X, O=>XLXN_32);
```

XLXI\_106 : INV  
PORT MAP (I=>XLXN\_20, O=>M6);

XLXI\_40 : INV  
PORT MAP (I=>Y, O=>XLXN\_33);

XLXI\_41 : INV  
PORT MAP (I=>F, O=>XLXN\_37);

XLXI\_107 : INV  
PORT MAP (I=>XLXN\_54, O=>M7);

XLXI\_108 : INV  
PORT MAP (I=>XLXN\_50, O=>M8);

XLXI\_109 : INV  
PORT MAP (I=>XLXN\_68, O=>M9);

XLXI\_84 : INV  
PORT MAP (I=>G, O=>XLXN\_93);

XLXI\_42 : INV  
PORT MAP (I=>R, O=>XLXN\_38);

XLXI\_110 : INV  
PORT MAP (I=>XLXN\_91, O=>M10);

XLXI\_111 : INV  
PORT MAP (I=>XLXN\_100, O=>M11);

XLXI\_112 : INV  
PORT MAP (I=>XLXN\_108, O=>M12);

XLXI\_51 : OR2  
PORT MAP (I0=>XLXN\_40, I1=>XLXN\_39, O=>XLXN\_53);

XLXI\_50 : OR2  
PORT MAP (I0=>XLXN\_44, I1=>XLXN\_43, O=>XLXN\_47);

XLXI\_49 : OR2  
PORT MAP (I0=>XLXN\_42, I1=>XLXN\_41, O=>XLXN\_48);

XLXI\_91 : OR2  
PORT MAP (I0=>XLXN\_109, I1=>Start, O=>XLXN\_110);

XLXI\_77 : OR2  
PORT MAP (I0=>XLXN\_88, I1=>XLXN\_115, O=>XLXN\_87);

XLXI\_66 : OR3  
PORT MAP (I0=>XLXN\_65, I1=>XLXN\_63, I2=>XLXN\_64, O=>XLXN\_68);

```
XLXI_65 : OR3
  PORT MAP (I0=>XLXN_62, I1=>XLXN_60, I2=>XLXN_61, O=>XLXN_67);

XLXI_64 : OR3
  PORT MAP (I0=>XLXN_59, I1=>XLXN_57, I2=>XLXN_58, O=>XLXN_66);

XLXI_88 : OR3
  PORT MAP (I0=>XLXN_20, I1=>XLXN_19, I2=>XLXN_135,
O=>XLXN_104);

XLXI_72 : OR3
  PORT MAP (I0=>XLXN_75, I1=>XLXN_74, I2=>XLXN_73, O=>XLXN_84);

XLXI_74 : OR3
  PORT MAP (I0=>XLXN_76, I1=>XLXN_77, I2=>XLXN_78, O=>XLXN_85);

XLXI_75 : OR3
  PORT MAP (I0=>XLXN_79, I1=>XLXN_80, I2=>XLXN_81, O=>XLXN_82);

XLXI_85 : OR4
  PORT MAP (I0=>XLXN_20, I1=>XLXN_19, I2=>XLXN_135,
I3=>XLXN_100,
  O=>XLXN_96);

END SCHEMATIC;
```