

UNIVERSIDADE REGIONAL DE BLUMENAU  
CENTRO DE CIÊNCIAS EXATAS E NATURAIS  
CURSO DE CIÊNCIAS DA COMPUTAÇÃO  
TRABALHO DE CONCLUSÃO DE CURSO

**PROTÓTIPO GERADOR DE  
CÓDIGO ASSEMBLY PARA O  
MICROCONTROLADOR PIC16F84A  
A PARTIR DE FLUXOGRAMAS**

EDUARDO SALES PINHEIRO  
ACADÊMICO  
MIGUEL ALEXANDRE WISINTAINER  
ORIENTADOR

BLUMENAU, JULHO/2004

## Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Trabalhos correlatos
- Desenvolvimento do trabalho
- Funcionalidade da ferramenta
- Resultados/Discussões
- Conclusões
- Limitações/Sugestões

## Introdução

- As pessoas encontram-se rodeadas de aparelhos eletrônicos que possuem dentro de si um microcontrolador, e nem mesmo têm-se consciência disto;
- Microcontroladores são componentes eletrônicos utilizados para controlar um ou mais processos;
- O trabalho consiste em implementar um software de suporte para simplificar a programação para microcontroladores devido ao alto grau de dificuldade encontrado durante o aprendizado da linguagem Assembly;

## Introdução

- Este trabalho é um complemento ao trabalho desenvolvido por Fontanive (1999), que consiste num editor fluxogramático que gera código Assembly a partir de um fluxograma;
- Aqui são implementados os registradores do PIC16F84A, o tratamento de interrupções, a chamada de sub-rotinas e a interface com outros periféricos através de uma saída para escrita em serial;

## Introdução

- O código fonte (Assembly) gerado pelo protótipo em um arquivo texto é transformado em código objeto para o microcontrolador através do montador *MPASM*, que está integrado ao aplicativo *MPLAB*;
- Para simular o programa gerado através da ferramenta é utilizado o aplicativo Proteus.

## Objetivos

- Desenvolver um protótipo que, a partir de um fluxograma, gere código Assembly.
- Gerar código para um conjunto de instruções em Assembly para o microcontrolador PIC16F84A;
- Permitir realizar modificações no código gerado, desvinculando este código do fluxograma.

# Fundamentação Teórica

## Microcontroladores

- São componentes eletrônicos utilizados para controlar um ou mais processos: botões, *display's* e *led's*;
- Arquitetura *Havard*: consiste em dois barramentos internos, sendo um de dados e outro de instruções;
- Tecnologia RISC que significa *Reduce Instruction Set Computer* (Computador com *Set* de instruções reduzido). No PIC16F84A são 35;
- Tratador de interrupção e rotinas de pausa.

# Fundamentação Teórica

## Fluxogramas

- São representações de um fluxo de dados através de uma linguagem simbólica de programação que o computador entende;
- Ferramenta gráfica, que permite representar qualquer tipo de solução de problema (simples ou complexos) e é composto por um conjunto de símbolos pré-definidos;
- Simplifica o entendimento do problema.

# Fundamentação Teórica

## ExpressFlowChart Suite

- É um componente que permite a criação de aplicações que contenham quadros, esquemas, hierarquias, gráficos, etc;
- As bibliotecas são utilizadas para desenhar os objetos do fluxograma;
- Possui diversas propriedades e métodos para manipulação de objetos gráficos;

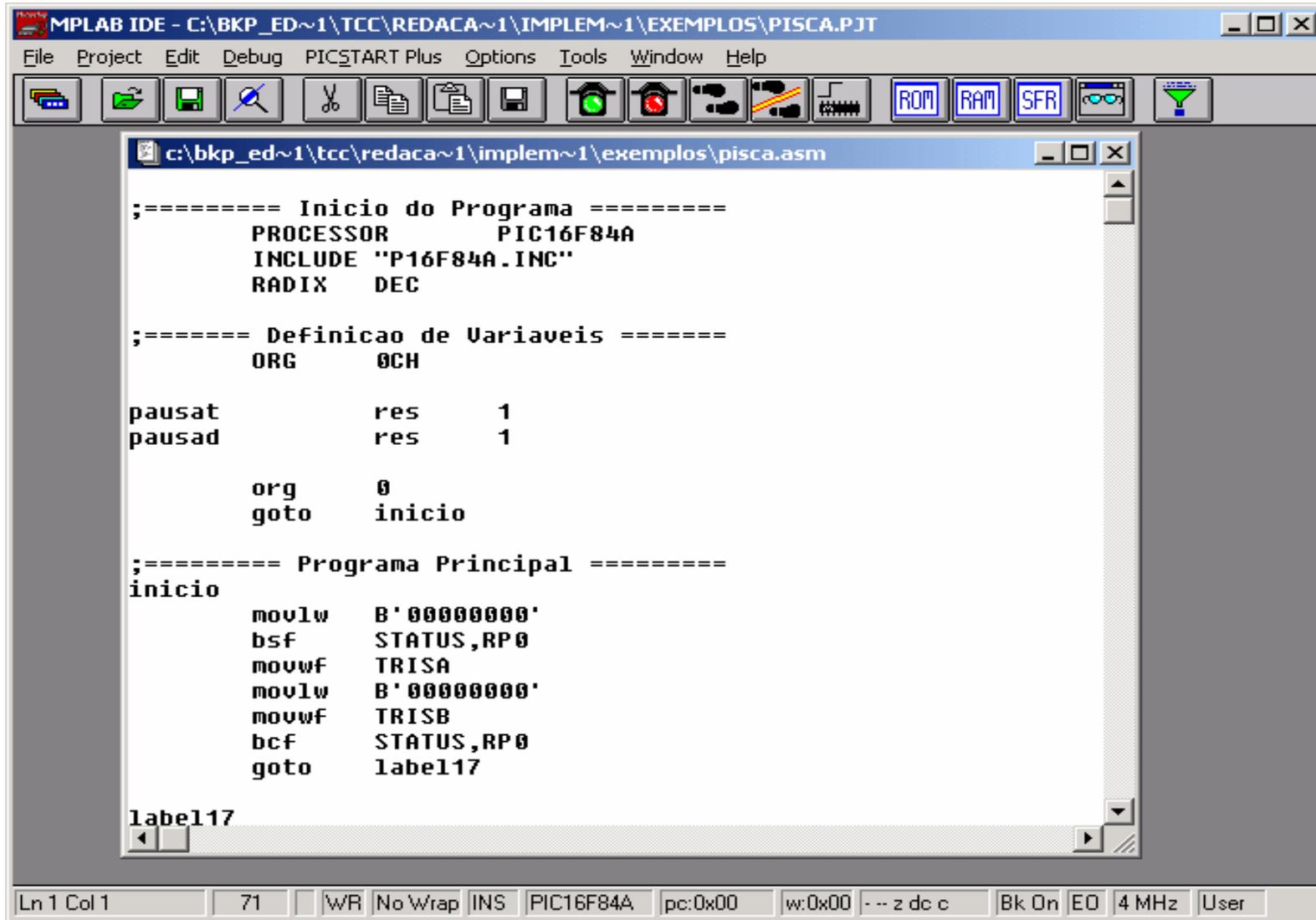
# Fundamentação Teórica

## MPLAB

- É um ambiente de desenvolvimento para programas PIC que roda na plataforma Windows e junta num mesmo ambiente, o gerenciamento de projetos, a compilação, a simulação, a emulação e a gravação” do PIC;
- Utilizado neste trabalho para gerar o código objeto para o microcontrolador a partir do código Assembly gerado pelo protótipo;

# PROTÓTIPO GERADOR DE CÓDIGO ASSEMBLY PARA O MICROCONTROLADOR PIC16F84A A PARTIR DE FLUXOGRAMAS

## MPLAB



The screenshot displays the MPLAB IDE interface. The title bar reads "MPLAB IDE - C:\BKP\_ED~1\TCC\REDACA~1\IMPLEM~1\EXEMPLOS\PISCA.PJT". The menu bar includes "File", "Project", "Edit", "Debug", "PICSTART Plus", "Options", "Tools", "Window", and "Help". The toolbar contains various icons for file operations, debugging, and hardware simulation. The main window shows the assembly code for "pisca.asm".

```
===== Inicio do Programa =====  
PROCESSOR      PIC16F84A  
INCLUDE "P16F84A.INC"  
RADIX      DEC  
  
===== Definicao de Variaveis =====  
ORG      0CH  
  
pausat      res      1  
pausad      res      1  
  
org      0  
goto      inicio  
  
===== Programa Principal =====  
inicio  
    movlw   B'00000000'  
    bsf     STATUS,RP0  
    movwf   TRISA  
    movlw   B'00000000'  
    movwf   TRISB  
    bcf     STATUS,RP0  
    goto    label17  
  
label17
```

The status bar at the bottom shows "Ln 1 Col 1", "71", "WR No Wrap INS", "PIC16F84A", "pc:0x00", "w:0x00", "-- z dc c", "Bk On EO", "4 MHz", and "User".

# Fundamentação Teórica

## Proteus

- É um aplicativo simulador interativo onde é possível criar e simular o funcionamento de circuitos eletrônicos sem a necessidade de se criar um circuito físico. Com isto, o tempo de construção e adaptação se torna muito menor;
- Entre as características do *Proteus* pode-se citar:
  - a) capacidade de importar figuras e outros componentes;
  - b) de simular várias famílias de microcontroladores;
  - c) de selecionar objetos e atribuir suas propriedades;
  - d) suporte a circuitos secundários, que podem ser ligados ao principal;
  - e) suporte total a barramentos incluindo pinos, terminais, portas e fios;
- Utilizado neste trabalho para validar o código gerado pelo protótipo.

# PROTÓTIPO GERADOR DE CÓDIGO ASSEMBLY PARA O MICROCONTROLADOR PIC16F84A A PARTIR DE FLUXOGRAMAS

## Proteus

The screenshot displays the Proteus ISIS Professional interface. The main workspace shows a circuit diagram for a PIC16F84A microcontroller (U1) connected to an LCD1 (HELPO-D-202-SF) and a terminal (VT1). The PIC16F84A is connected to an oscillator circuit consisting of a crystal (X1) and two capacitors (C1, C2). The LCD1 is connected to the PIC16F84A via a parallel interface. The terminal (VT1) is connected to the PIC16F84A via a serial interface.

**PIC and Serial LCD Displays**

This design demonstrates how to control a serial LCD display using a PIC16F84A. The simple assembly file for this display has a HEX file available from Labcenter Electronics ([lce@labcenter.co.uk](mailto:lce@labcenter.co.uk)).

The PIC code will use the USART to receive the commands and to send characters back to the terminal. The assembly file will be used to send the commands to the terminal, which are processed separately.

**Labcenter Electronics**

Labcenter Electronics, 53-55 Main Street, Grassington, North Yorkshire, BD23 5AA  
Fax: +44 (0)1756 752257 Tel: +44 (0)1756 753440  
Email: [lce@labcenter.co.uk](mailto:lce@labcenter.co.uk) WWW: <http://www.labcenter.co.uk/>

**ISIS**

ROOT SHEET 1

+4600 +1000 th

## Trabalhos Correlatos

### Fontanive

- Consiste em um editor fluxogramático para geração de código Assembly;
- Falhas/Deficiências detectadas neste software:
  - a) não possui a instrução de **interrupção** que é essencial para o desenvolvimento de uma boa aplicação;
  - b) ocorrem algumas violações de acesso que inviabilizam o uso contínuo do protótipo;
  - c) não possui a instrução de **pausa** que é essencial para o controle de algumas ações.

# PROTÓTIPO GERADOR DE CÓDIGO ASSEMBLY PARA O MICROCONTROLADOR PIC16F84A A PARTIR DE FLUXOGRAMAS

## Fontavive

Flowchart Editor - PIC16F84

New Open Save Close Exit Process Condition GoTo END Zoom 100 Delete Variables I/O Asm File

Flowchart Editor Assembly File

```
; Begin  
list p=16F84  
radix dec  
include <P16F84.INC>
```

```
; Variables
```

```
====  
org 0x000  
movlw 0xFF  
tris PORTB  
tris PORTA
```

# Trabalhos Correlatos

## CH Basic

- Simplifica o desenvolvimento e escrita de programas para PIC;
- Contém um conjunto básico de instruções (em Basic) e possui componentes visuais em forma de caixa de diálogo que auxiliam e contribuem para reduzir o tempo de desenvolvimento de um software para o PIC;
- Benefícios:
  - a) redução do tempo que leva para escrever programas para PIC;
  - b) não é necessário que o programador aprenda a linguagem Assembly para desenvolver seus programas;
  - c) possui comandos simples e intuitivos;
- Utilizado para verificar as estruturas dos comandos gerados na linguagem Assembly.

# PROTÓTIPO GERADOR DE CÓDIGO ASSEMBLY PARA O MICROCONTROLADOR PIC16F84A A PARTIR DE FLUXOGRAMAS

## CH Basic

The screenshot displays the CH Basic - Flash Edition (Evaluation Version) software interface. The main window shows a Basic program with several errors. A dialog box titled "Interrupt\_PortB" is open, allowing configuration of the Port B Change Interrupt. The "Error Report" window shows the error messages.

**CH Basic - Flash Edition (Evaluation Version) C:\Arquivos de programas\CH Basic - Flash Edition V3.7.0\TEMP\_B5C.bsc**

File Edit Commands Tools Options Help

'Flash Basic

```
IF a > 1 THEN
  ' < Error in Basic line.
  Interrupt_Ext 1, rising
  Interrupt_PortB 1, 0
  Float <<< Error in Basic line.
  RETURN
END PROC <<< Error in Basic line.
Interrupt_Enable
Interrupt_Ext 1, falling
NEXT <<< Error in Basic line.
Counter Read <<< Error in Basic line.
Delay , 0 <<<
```

**Interrupt\_PortB**

Port B Change Interrupt

Interrupt:  Enable  Disable

Port B Pull-up:  Enable

Cancel OK

**Error Report**

Errors: 1

```
> Error in Basic line
```

**Compiler Information:**

PIC 16F84A 20 MHz (Crystal oscillator)  
Max 1024 Lines of Program Memory  
Max 68 Bytes of Data Memory

Basic lines : 12  
Program Memory : 38 (96,3% free)  
Data Memory : 8 (88,2% free)

; 0x0 - 0xE = occupied by compiler

```
1 MOVF A, W
1 MOVWF S1
1 MOVLW 1
1 MOVWF AUX
1 MOVF S1, W
1 MOVWF AUX1_L
1 CALL CREUS
1 MOVF AUX, W
1 BTFSS AUX, 0
1 GOTO ENDIFLAB1
2 ; < ERROR IN BASIC LINE.
3 BSF OPTION_REG, INTEDG
3 BSF INTCON, INTE
4 BCF OPTION_REG, NOT_RBPU
4 BSF INTCON, RBIE
5 ; Error in Basic line : F1
6 RETURN
7 ; Error in Basic line : END P
8 BSF INTCON, GIE
9 BCF OPTION_REG, INTEDG
9 BSF INTCON, INTE
10 ; Error in Basic line : NEXT
11 ; Error in Basic line : Count
12 ; Error in Basic line : Delay
```

**Variables**

BYTE a

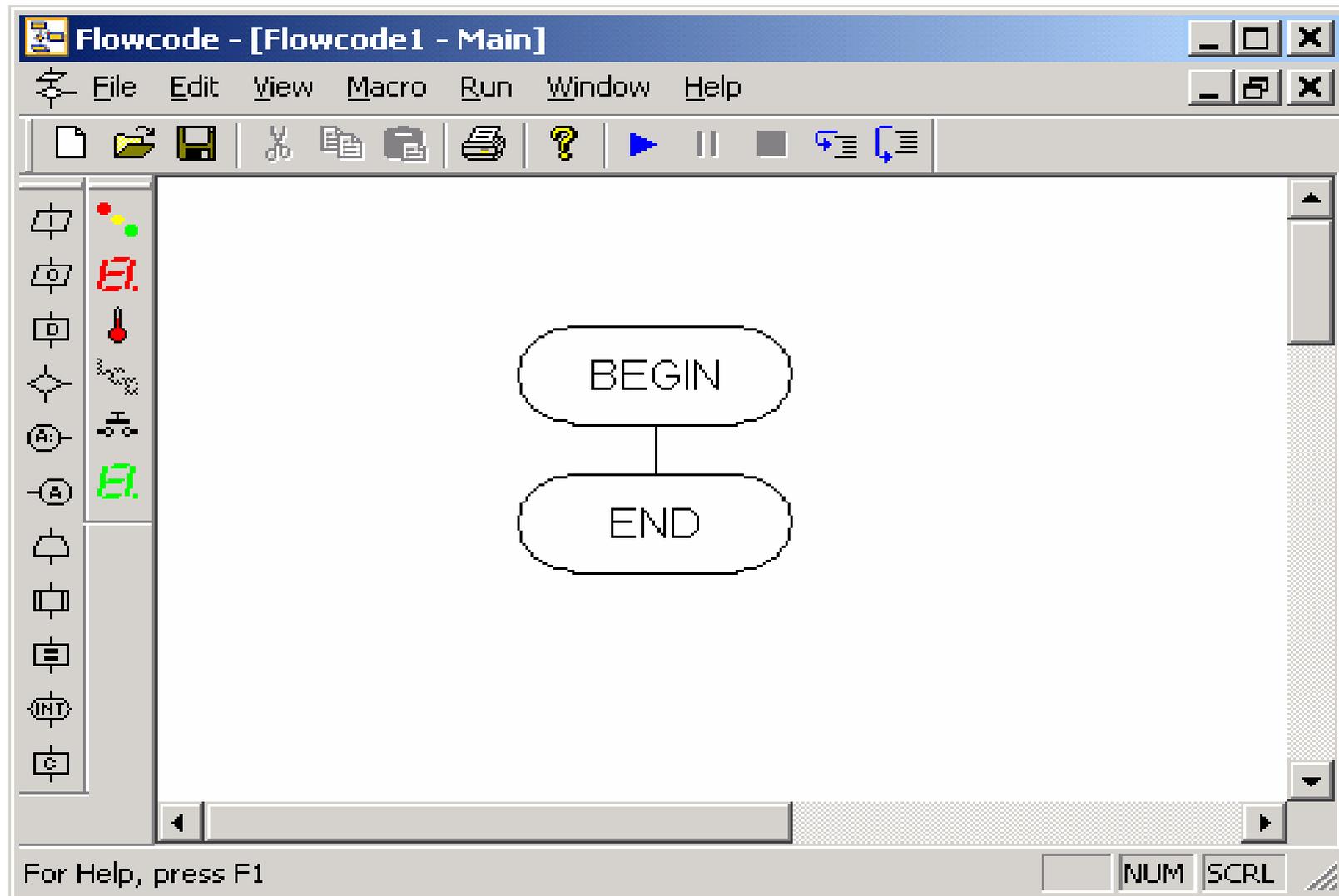
# Trabalhos Correlatos

## FLOWCODE

- Permite criar fluxogramas e simular a sua execução passo a passo;
- Gerar código objeto para PIC e/ou gerar código Assembly em arquivo texto;
- Necessidade de programar em Assembly;
- Deficiências.

PROTÓTIPO GERADOR DE CÓDIGO ASSEMBLY PARA O  
MICROCONTROLADOR PIC16F84A A PARTIR DE FLUXOGRAMAS

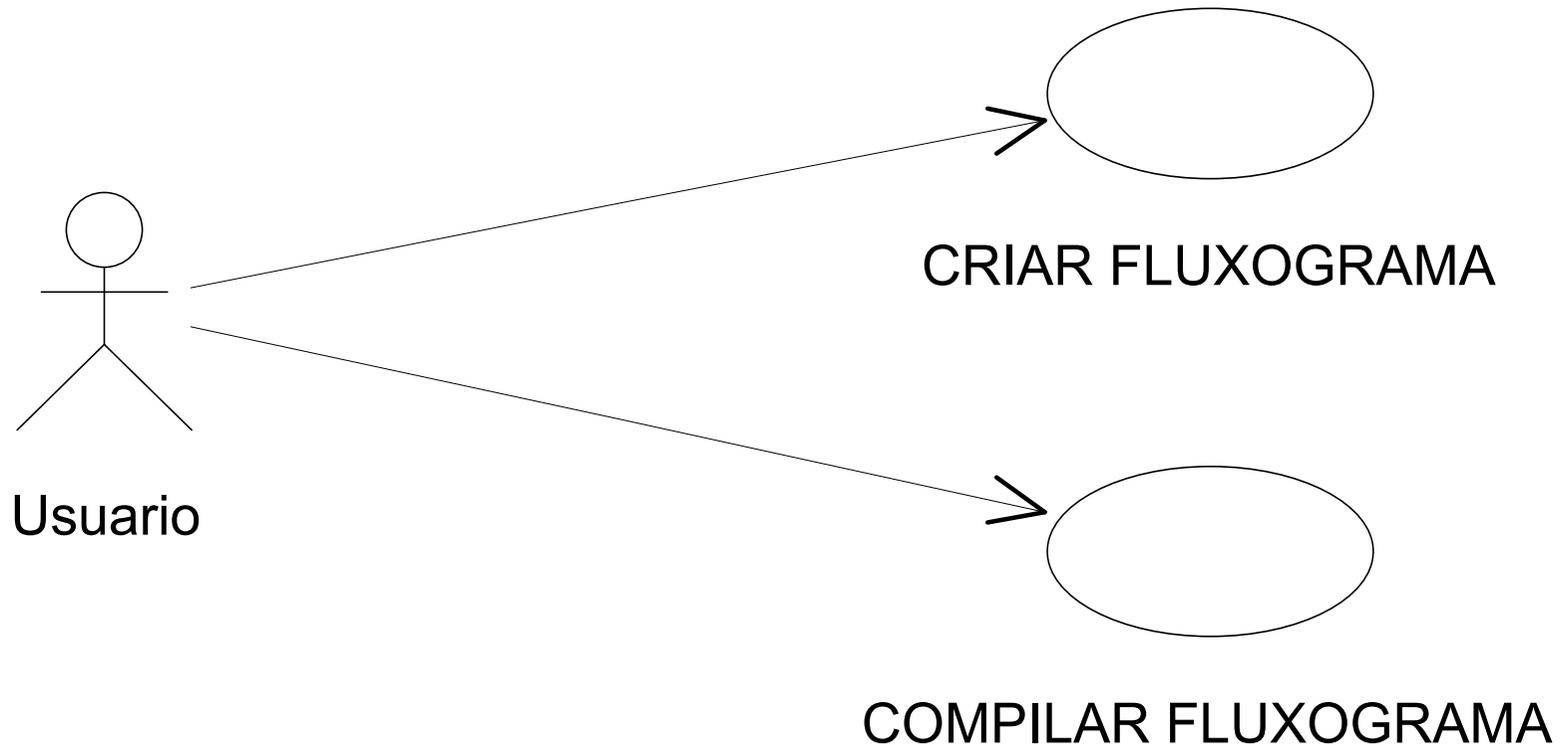
# FLOWCODE



## Desenvolvimento - Requisitos

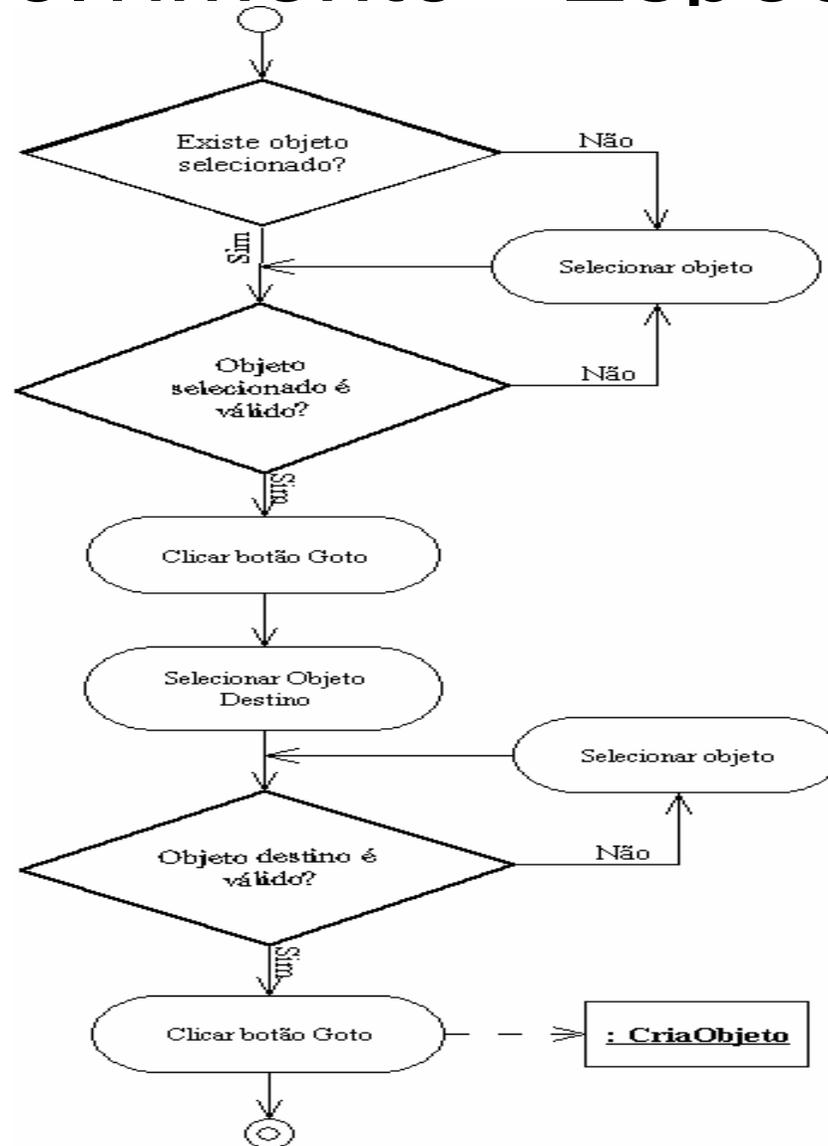
- criar dois arquivos de saída, um contendo o fluxograma e outro contendo o código gerado (Requisito Funcional - RF);
- disponibilizar recursos para trabalhar com registradores, variáveis e literais (RF);
- gerar código Assembly a partir do fluxograma, respeitando as ligações definidas pelo usuário (RF);
- permitir realizar modificações no código gerado, desvinculando este código do fluxograma (RF);
- verificar ligações entre os componentes gráficos do fluxograma durante sua construção (RF);
- verificar sintaxe através do emprego de técnicas de compiladores (RF).

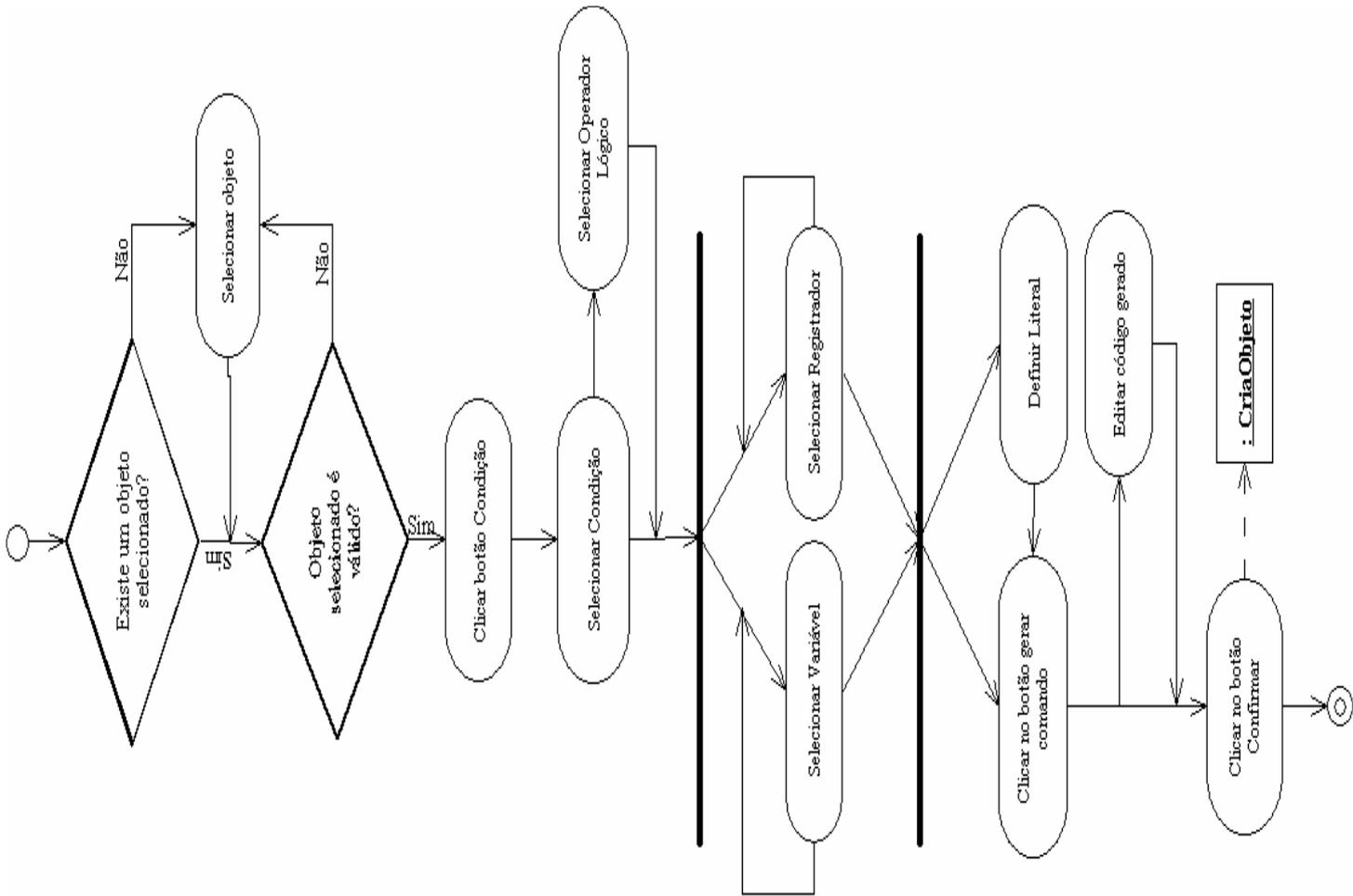
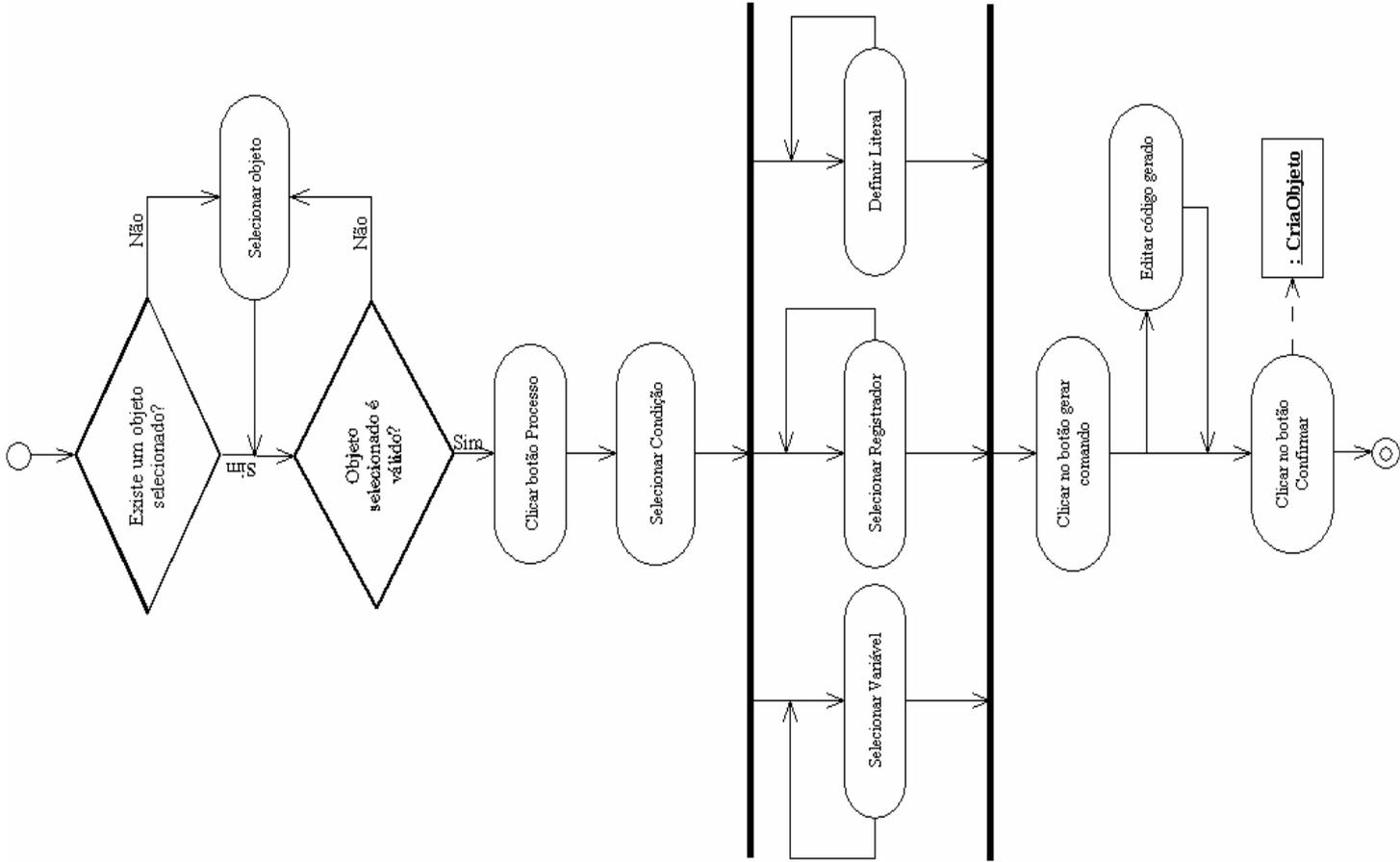
# Desenvolvimento - Especificação



PROTÓTIPO GERADOR DE CÓDIGO ASSEMBLY PARA O  
MICROCONTROLADOR PIC16F84A A PARTIR DE FLUXOGRAMAS

# Desenvolvimento - Especificação





# Desenvolvimento - Implementação

## Técnicas e Ferramentas

- ExpressflowChart Suite: objetos do fluxograma;
- Algoritmos Recursivos: compilar o fluxograma;

# Desenvolvimento - Implementação

```
Procedure TFrmPrincipal.Compilar1Click(Sender: TObject);  
var  
  Xfluxograma : TdxFlowChart;  
  i : integer;  
begin  
  { DIVERSAS CONSISTENCIAS .....}  
  VerificaNo(xfluxograma, xfluxograma.Objects[0].CustomData);  
  { DIVERSAS CONSISTENCIAS .....}  
end;
```

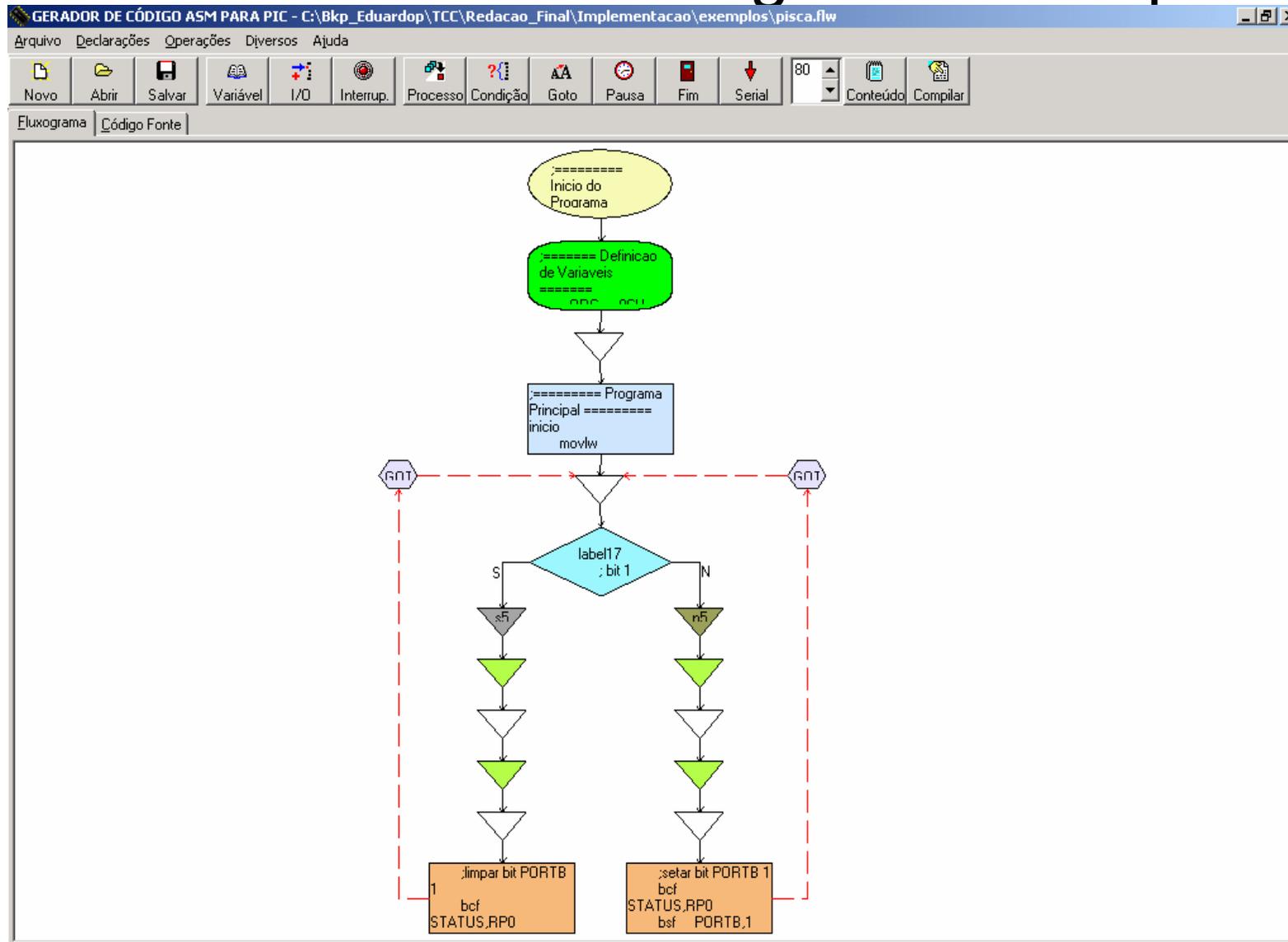
PROTÓTIPO GERADOR DE CÓDIGO ASSEMBLY PARA O  
MICROCONTROLADOR PIC16F84A A PARTIR DE FLUXOGRAMAS

# Desenvolvimento - Implementação

```
procedure TFrmPrincipal.VerificaNo(fluxograma: TdxFlowChart; xno: String);
var x: Integer;
begin
  { Varre todas as conexões verificando o conteúdo dos objetos no fluxograma. }
  for x := 0 to fluxograma.ConnectionCount - 1 do
  begin
    // Tratamento especial para GOTO.
    If (fluxograma.Connections[x].ObjectSource.CustomData = xno) and
      (fluxograma.Connections[x].PenStyle <> psDash) and (not gFalhou) then
    begin
      { Verifica se todos os blocos finais estão criados. }
      ....
      { Só adiciona o texto ao Memo se o objeto possuir algum texto. }
      if (fluxograma.Connections[x].ObjectDest.Text <> "") then
        MemoFonte.Lines.Text := MemoFonte.Lines.Text + "#13" +
          (fluxograma.Connections[x].ObjectDest.Text);
      VerificaNo(fluxograma, fluxograma.Connections[x].ObjectDest.CustomData);
    end;
  end;
end;
```

# PROTÓTIPO GERADOR DE CÓDIGO ASSEMBLY PARA O MICROCONTROLADOR PIC16F84A A PARTIR DE FLUXOGRAMAS

## Funcionalidade - Fluxograma Exemplo



PROTÓTIPO GERADOR DE CÓDIGO ASSEMBLY PARA O  
MICROCONTROLADOR PIC16F84A A PARTIR DE FLUXOGRAMAS

## Funcionalidade - Código Fonte Gerado

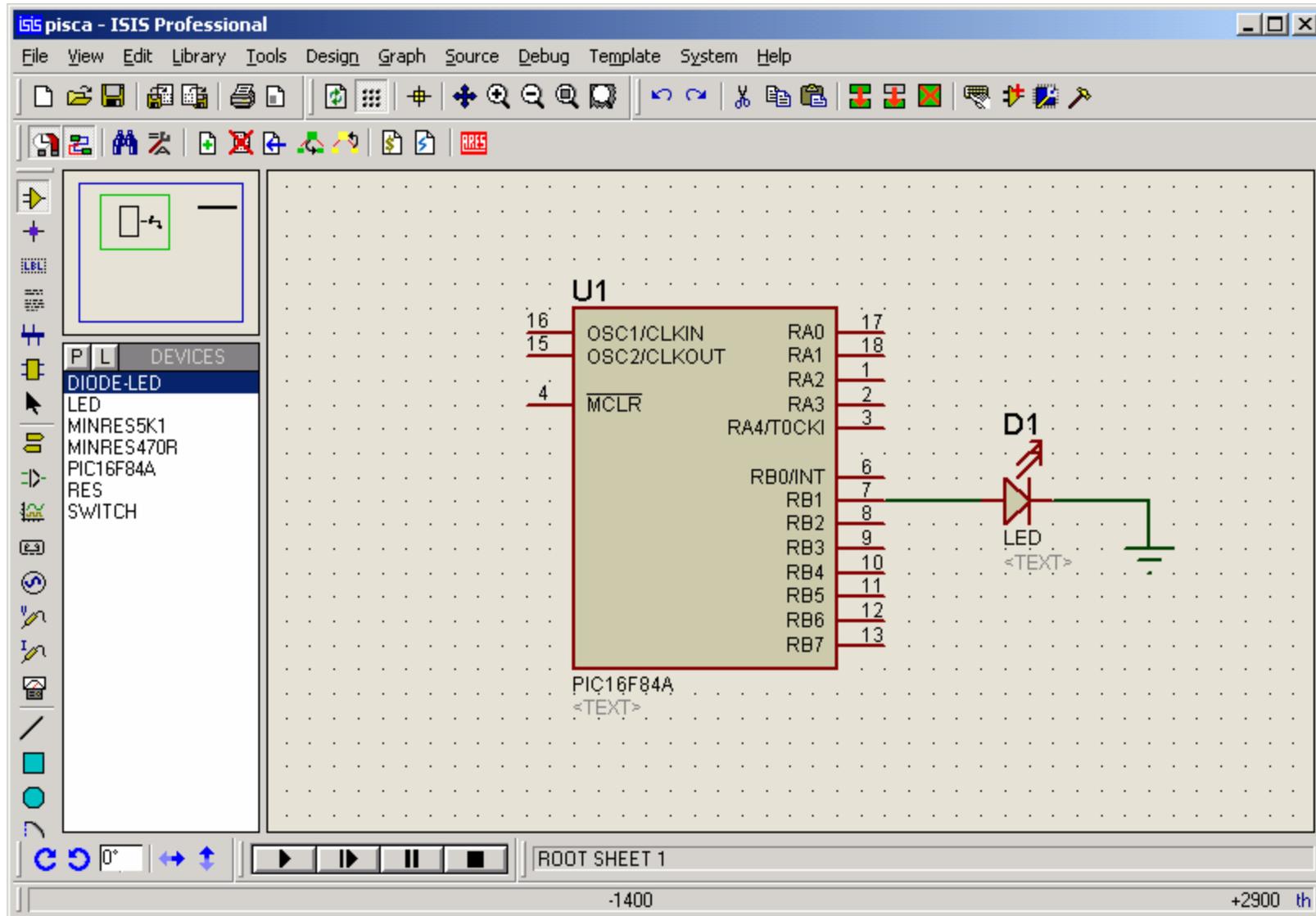
```
===== Inicio do Programa =====  
PROCESSOR  
PIC16F84A  
INCLUDE "P16F84A.INC"  
RADIX DEC  
;=== Definicao de Variaveis ===  
ORG 0CH  
pausat res 1  
pausad res 1  
org 0  
goto inicio  
  
===== Programa Principal =====  
inicio  
movlw B'00000000'  
bsf STATUS,RP0  
movwf TRISA  
movlw B'00000000'  
movwf TRISB  
bcf STATUS,RP0  
goto label17
```

```
label17  
; bit 1 do registradorl  
PORTB = set!  
bcf STATUS,RP0  
btfss PORTB,1  
goto n5  
goto s5  
  
s5  
call pausap  
call pausap  
;limpar bit PORTB 1  
bcf STATUS,RP0  
bcf PORTB,1  
goto label17  
  
n5  
call pausap  
call pausap  
  
;setar bit PORTB 1  
bcf STATUS,RP0  
bsf PORTB,1
```

```
goto label17  
  
;= Inicio da Rotina de Pausa =  
pausap  
movlw 255  
movwf pausat  
pausap1  
movlw 253  
movwf pausad  
pausap2  
nop  
decfsz pausad,f  
goto pausap2  
decfsz pausat,f  
goto pausap1  
return  
  
;== Fim da Rotina de Pausa ==  
  
fim goto fim  
END
```

# PROTÓTIPO GERADOR DE CÓDIGO ASSEMBLY PARA O MICROCONTROLADOR PIC16F84A A PARTIR DE FLUXOGRAMAS

## Funcionalidade - Circuito



PROTÓTIPO GERADOR DE CÓDIGO ASSEMBLY PARA O  
MICROCONTROLADOR PIC16F84A A PARTIR DE FLUXOGRAMAS

## Resultados/Discussões

Funções	Protótipo	Fontanive
Consistência de Variáveis	X	
Impressão em serial	X	
Integração com outros aplicativos	X	
Maior área de edição do fluxograma	X	
Opção de escolha do local para salvar o fluxograma e o código fonte	X	
Opção de imprimir código fonte	X	
Opção para desfazer a seleção de todos objetos selecionados	X	
Opção para excluir os objetos criados	X	X
Opção para selecionar todos os objetos	X	
Opção para trabalhar com variáveis e literais (números)	X	X
Opção para trabalhar com registradores	X	
Opção para trabalhar com variáveis	X	X
Opção para visualizar o conteúdo dos objetos sem redimensioná-los	X	
Possibilidade de alterar o código Assembly gerado	X	
Proporcionalidade entre os objetos	X	
Rotina de fim	X	X
Rotina de goto	X	X
Rotina de pausa	X	
Software em língua portuguesa	X	
Tratador de interrupção	X	
Zoom	X	X

## Conclusões

- Os objetivos e os requisitos previamente definidos foram todos atingidos;
- A ferramenta introduz uma nova alternativa para o aprendizado da linguagem Assembly;
- Pode ser empregado em disciplinas como Prática em Arquitetura de Computadores, Sensores e Atuadores ou Automação e Controle.

## Limitações/Sugestões

- LIMITAÇÕES:
  - a) ao excluir um bloco de goto, não exclui os *labels* criados nos objetos de origem e destino;
  - b) não carrega as variáveis cadastradas e as definições de entrada/saída quando abre um fluxograma existente;
  - c) não permite imprimir o fluxograma, devido a limitações do componente ExpressFlowChart;
- SUGESTÕES:
  - a) demonstrar a execução passo a passo do fluxograma através de animações;
  - b) disponibilizar a geração do código para outros modelos de microcontroladores;
  - c) implementar recursos de macros para dispositivos;

## Comentários de usuários

- Ferramenta muito interessante, auxilia no aprendizado dos comandos da linguagem Assembly...” (PIRES, 2004).
- “Deveria haver opções no botão direito do mouse, como excluir, mover, modificar...” (JACINTO, 2004).
- “Poderia ter opções de voltar (ctrl+z), copiar(ctrl+c) e colar(ctrl+v)” (JACINTO, 2004).
- “O protótipo está excelente, é uma ótima ferramenta e pode auxiliar na resolução dos exercícios da matéria. (SAVI, 2004).