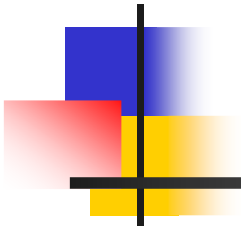


Protótipo de um Ambiente Ladder para os microcontroladores 8051 e PIC16F873



Aluno: Sidnei Alexandre de Almeida

Orientador: Antônio Carlos Tavares



Roteiro

- Introdução
- Motivação
- Diagramas Ladder
- Microcontroladores
- 8051
- PIC16F873
- Compiladores
- Tradução de expressões booleanas
- Especificação
- Estudo de caso
- Conclusão



Introdução

- Implementar um ambiente de programação Ladder para os microcontroladores 8051 e PIC16F873
- Este ambiente deve gerar o código assembly para o respectivo microcontrolador

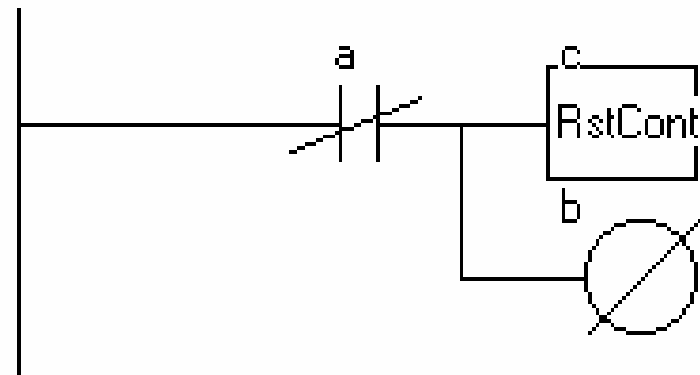


Motivação

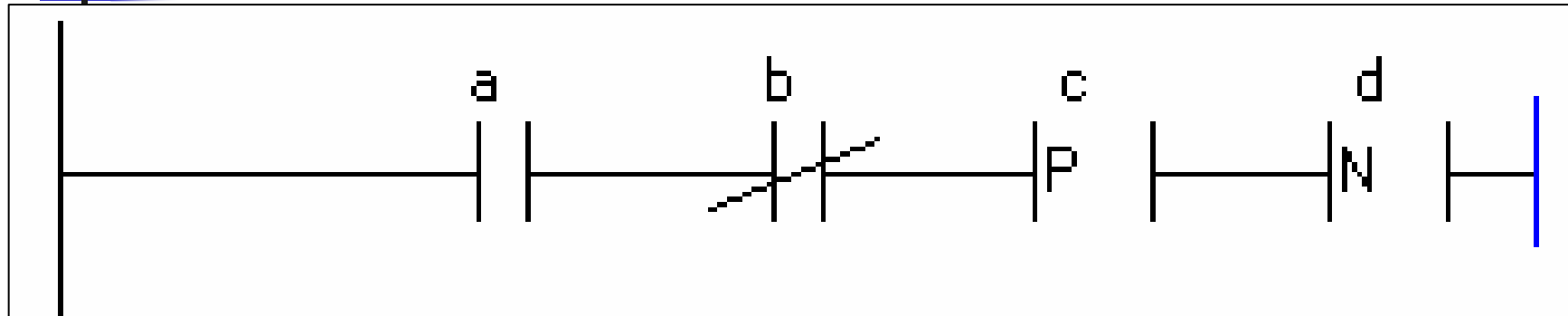
- Alta complexidade das linguagens assembly
- Ladder é uma linguagem simbólica simples orientada a programação de controladores lógicos programáveis
- Linguagens visuais são de melhor entendimento

Diagramas Ladder

- Linguagem orientada a programação de controladores lógicos programáveis
- O desenho assemelha-se a uma escada de relés
- Consiste de quatro símbolos principais: Contatos, bobinas e caixas de função e elementos de ligação



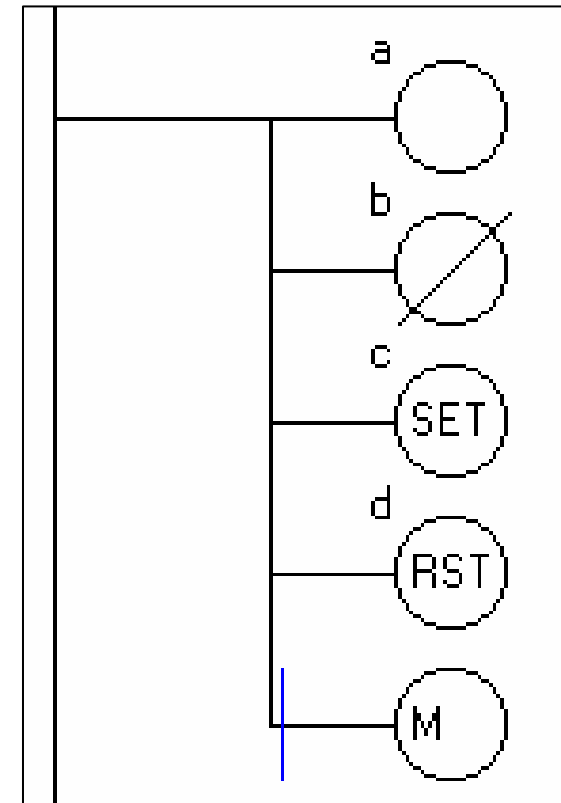
Diagramas Ladder (Contatos)



- Contato Normalmente aberto
- Contato Normalmente fechado
- Contato sensor de transição positiva
- Contato sensor de transição negativa

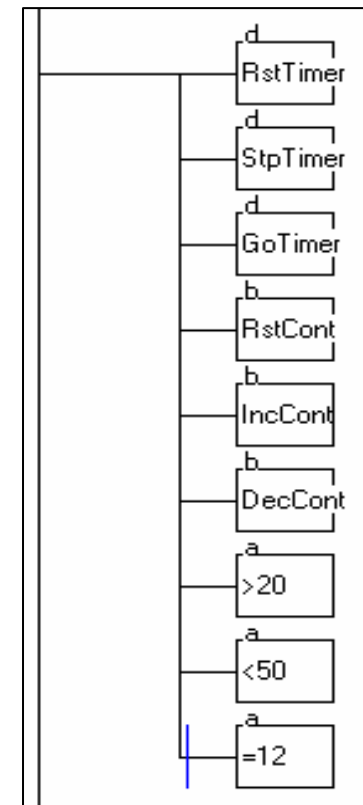
Diagramas Ladder (Bobinas)

- Bobinas Normal
- Bobinas Negada
- Bobinas Set
- Bobinas Reset
- Bobinas Retentivas (memórias auxiliares)

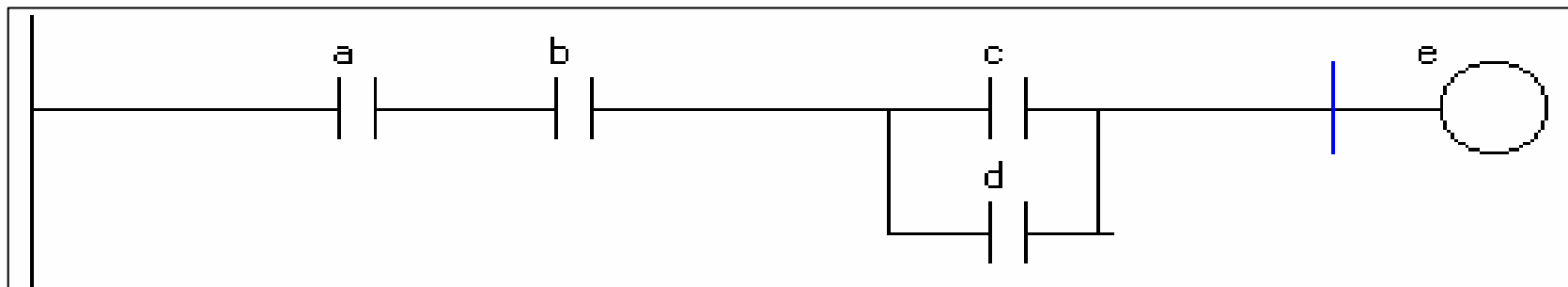


Diagramas Ladder (Funções)

- Caixas de função permitem extensibilidade ao diagrama
- Temporizadores
- A/D
- Contadores



Diagramas Ladder (avaliação)



- Contatos em seqüência representam operação lógica AND
- Contatos paralelos representam operação lógica OR
- Bobinas e funções não influenciam o comportamento do circuito



Microcontroladores

- Microcomputadores em um só chip
- São programáveis
- Possuem entradas, saídas, memória, temporizadores e demais funcionalidades
- São baratos e podem substituir o uso de um computador em pequenas aplicações



8051

- Arquitetura Von Neumann
- Cisc (Conjunto complexo de instruções)
- 32 portas de entrada e saída
- 2 temporizadores/contadores
- Freqüência de operação - 12 MHz
- Memória



PIC16F873

- Arquitetura Harvard
- Risc (35 instruções)
- 22 portas de entrada e saída
- Canal analógico/digital de 10 bits
- 3 temporizadores
- Frequência de operação 20 MHz
- Memória



Tradução de expressões booleanas

$E \rightarrow E_1 \text{ or } E_2$

$\{E.local := novotemporario;$

$Emitir (E.local := E_1.local \text{ or } E_2.local)\}$

$E \rightarrow E_1 \text{ and } E_2$

$\{E.local := novotemporario;$

$Emitir (E.local := E_1.local \text{ and } E_2.local)\}$

$E \rightarrow \text{not } E_1$

$\{E.local := novotemporario;$

$Emitir (E.local := \text{not } E_1.local)\}$

$E \rightarrow (E_1)$

$\{E.local := E_1.local\}$



Tradução de expressões booleanas (exemplo)

- $(X \text{ and } Y) \text{ or } (\text{not } X \text{ and } \text{not } Y)$
- $T0 := X \text{ and } Y$ *(T0) or (not(X) and not(Y))*
- $T1 := \text{not } X$ *(T0) or (T1 and not (Y))*
- $T2 := \text{not } Y$ *T0 or (T1 and T2)*
- $T3 := T1 \text{ and } T2$ *T0 or T3*
- $T4 := T0 \text{ or } T3$



Tradução de expressões booleanas (8051)

$E \rightarrow E_1 \text{ or } E_2$

{E.local:= novotemporario;

Emitir ('MOV C,'E1.Local

'ORL C,'E2.Local

'MOV 'E.local','C')}

$E \rightarrow E_1 \text{ and } E_2$

{E.local:= novotemporario;

Emitir (MOV C,'E1.Local

'ANL C,'E2.Local

'MOV 'E.local','C')}

$E \rightarrow \text{not } E_1$

{E.local:= novotemporario;

Emitir ('MOV C,' E1.local

'CPL C'

'MOV 'E.Local','C')}

$E \rightarrow (E_1)$

{E.local:= E₁.local}



Tradução de expressões booleanas (exemplo)

- (X and Y) or (not X and not Y)
- MOV C, X
- ANL C,Y
- MOV T0,C
 - *(T0) or (not(X) and not(Y))*
- MOV C, X
- CPL C
- MOV T1, X
 - *(T0) or (T1 and not (Y))*
- MOV C, Y
- CPL C
- MOV T2, X
 - *T0 or (T1 and T2)*
- MOV C, T1
- ANL C,T2
- MOV T3,C
 - *T0 or T3*
- MOV C,T0
- ORL C,T3
- MOV T4,C



Tradução de expressões booleanas (PIC16F873)

$E \rightarrow E_1 \text{ or } E_2$

{E.local:= novotemporario;

Emitir ('BCF E.Local

BTFS E1.Local

BSF E.Local

BTFS E2.Local

BSF E.Local}

$E \rightarrow E_1 \text{ and } E_2$

{E.local:= novotemporario;

Emitir (BSF E.local

BTFS E1.Local

BCF E.Local

BTFS E2.Local

BCF E.local)}

$E \rightarrow \text{not } E_1$

{E.local:= novotemporario;

Emitir ('BTFS E1.local

BCF E.Local

BTFS E1.Local

BSF E.Local)}

$E \rightarrow (E_1)$

{E.local:= E₁.local}



Tradução de expressões booleanas (exemplo)

- $(X \text{ and } Y) \text{ or } (\text{not } X \text{ and not } Y)$

BSF T0

BTFSS X

BCF T0

BTFSS Y

BCF T0

(T0) or (not(X) and not(Y))

BTFSC X

BCF T1

BTFSS X

BSF T1

(T0) or (T1 and not (Y))

BTFSC Y

BCF T2

BTFSS Y

BSF T2

- *T0 or (T1 and T2)*

BSF T3

BTFSS T1

BCF T3

BTFSS T2

BCF T3

- *T0 or T3*

BCF T4

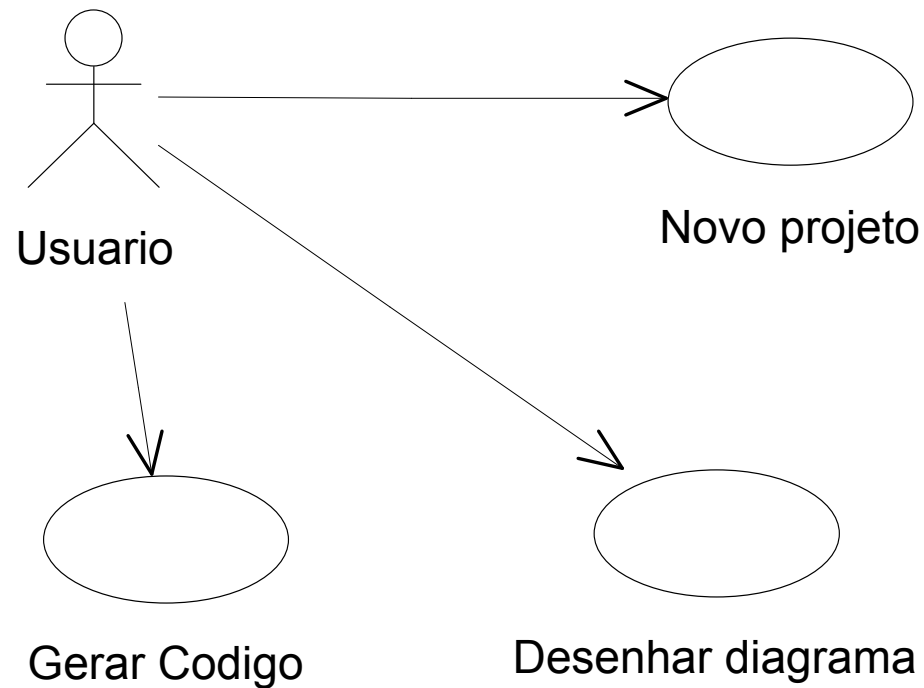
BTFSC T0

BSF T4

BTFSC T3

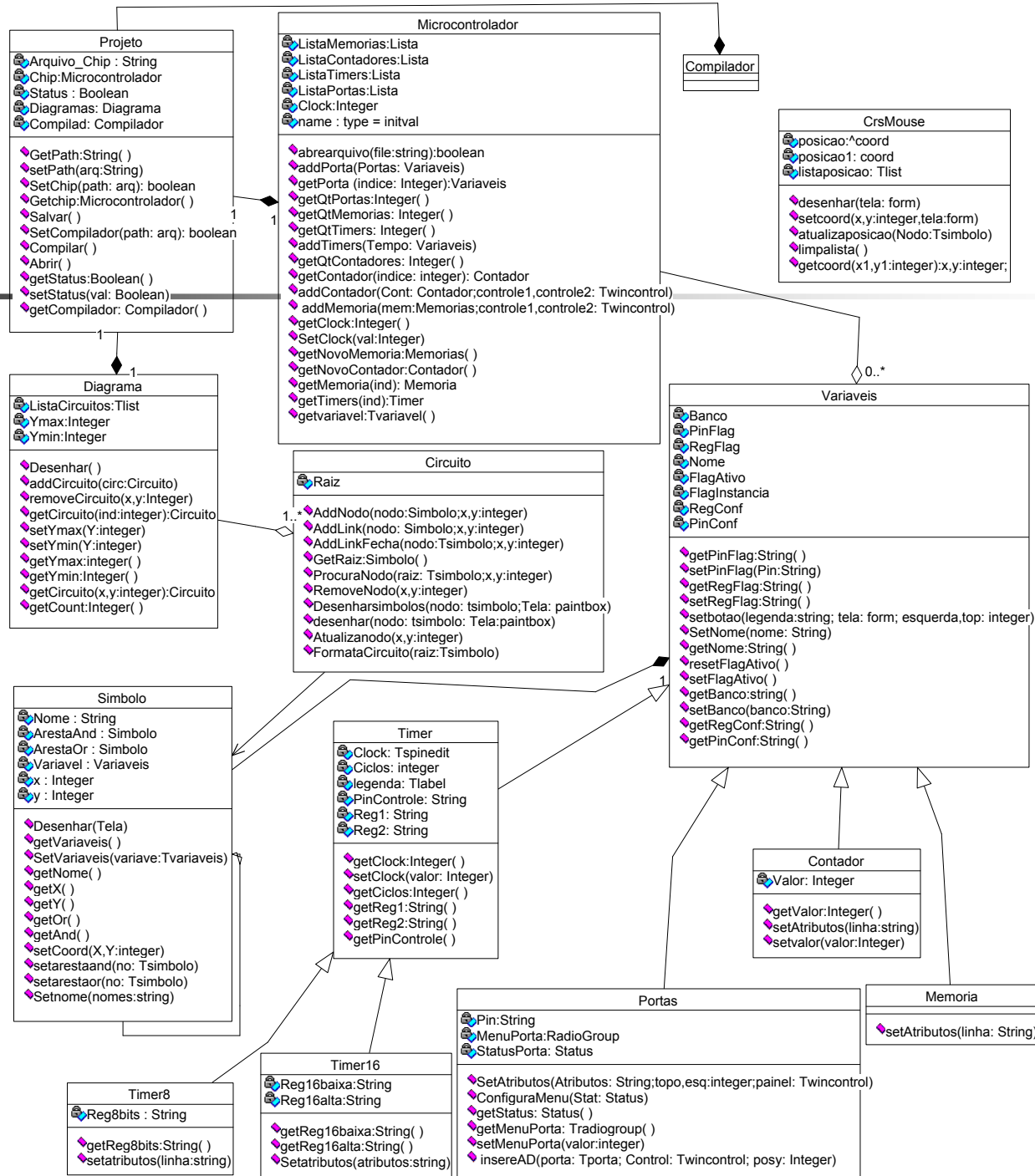
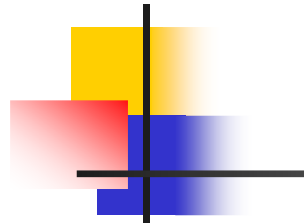
BSF T4

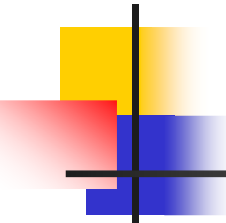
Especificação (Diagrama de casos de uso)



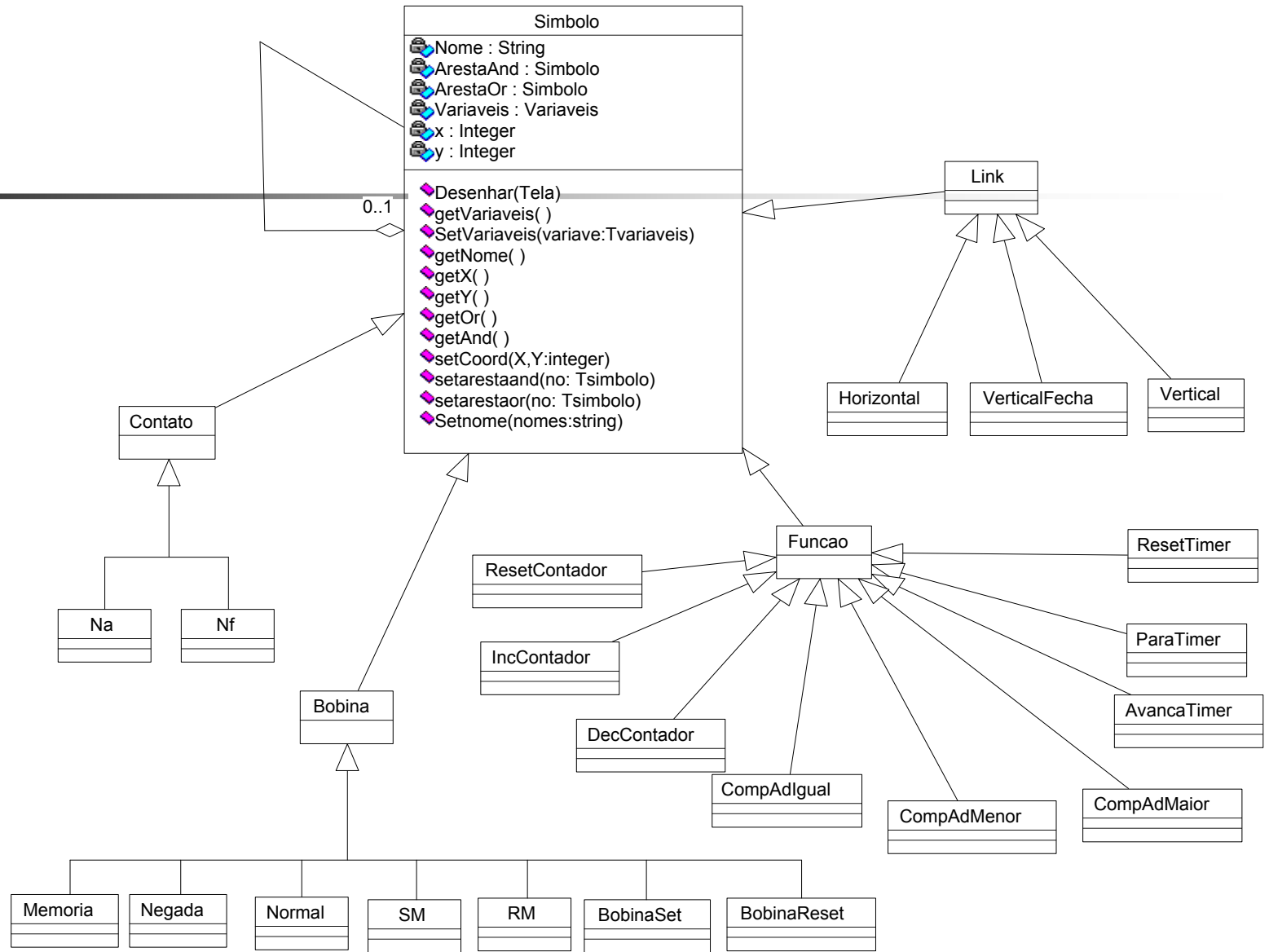
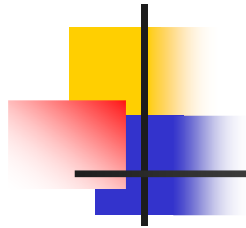


Especificação (Diagrama de Classes (Escopo Geral))



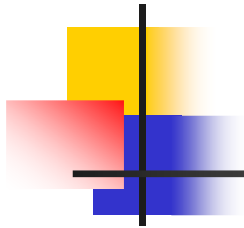


Especificação (Diagrama da Classe Símbolo)



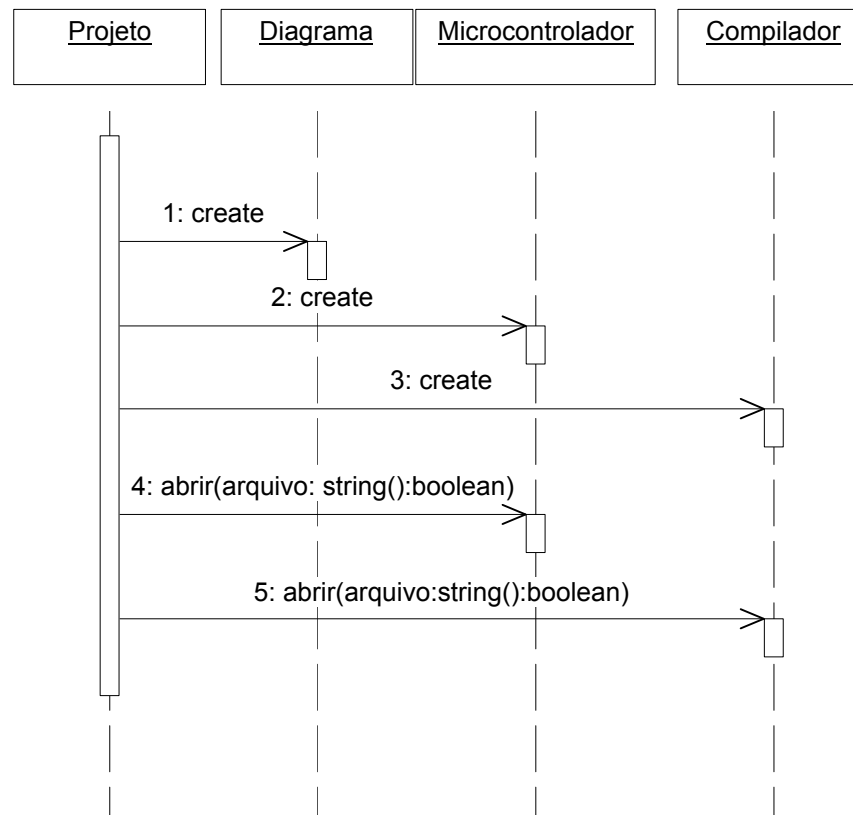


Especificação (Diagrama da Classe Compilador)

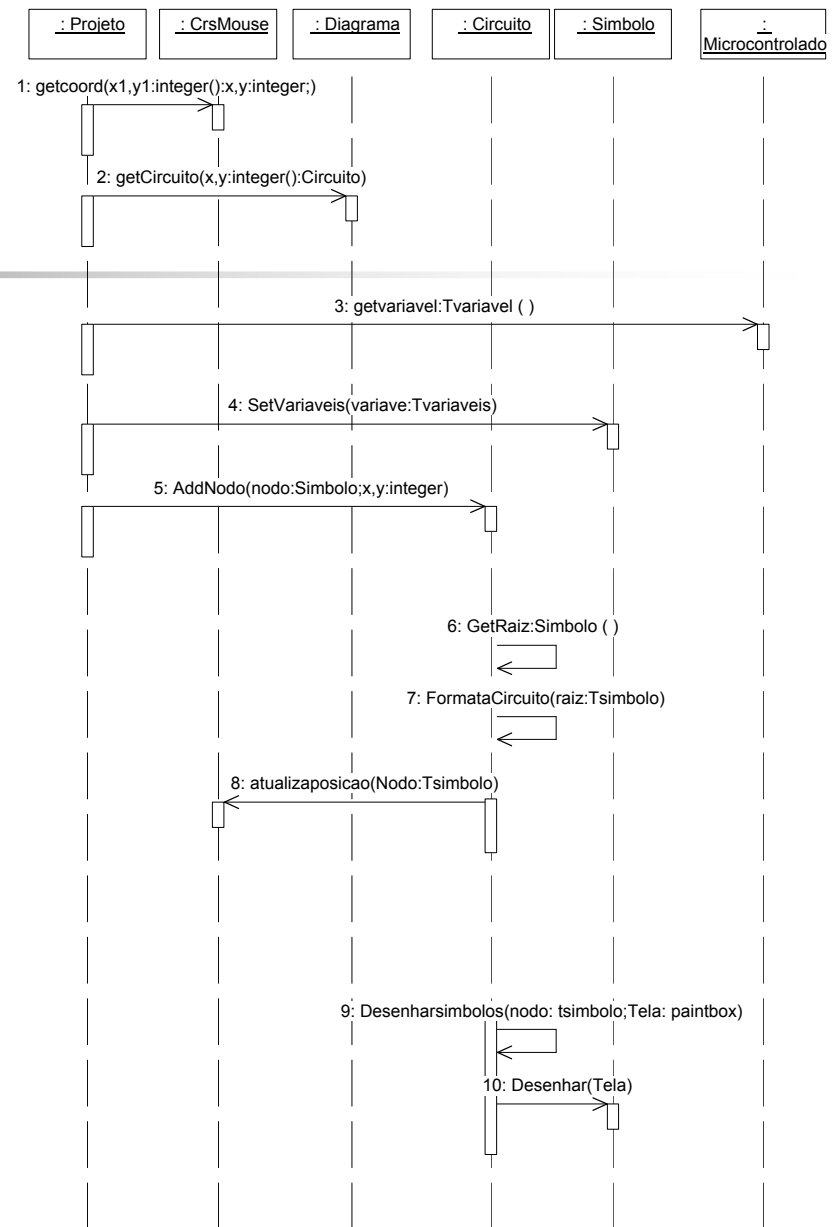


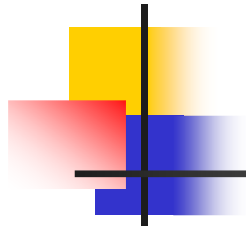
Compilador	
	AsemAnd:String
	AsemOr:String
	AsemNot:String
	AsemInsMem:String
	AsemInstCont:String
	AsemInstTimer8:String
	AsemInstTimer16:String
	AsemInstInput:String
	AsemInstOutput:String
	AsemInstAd:String
	FParaTimer:String
	FATimer:String
	RetomaTeste:String
	FRstTimer:String
	FRstCont:String
	FIncCont:String
	FDecCont:String
	FCmpAdMenor:String
	FCmpAdMaior:String
	FCmpAdIguar:String
	Saída:String
	SNegada:String
	SSet:String
	SReset:String
	TestaTimer8:String
	TestaTimer16:String
	StartContador:String
	StartTimer8:String
	StartTimer16:String
	StartInput:String
	StartOutput:String
	StartMemoria:String
	StartAd:String
	Cabecalho:String
	VaiPara:String
	Logica:Stringlist
	SeVaiPara:String
	Inicializacao:Stringlist
	Testes:StringList
	Instanciação:Stringlist
	Subrotina:String
	Expressao:String
	ListaAux:Stringlist
	Interpretador(linha:string):string
	avaliaexpressao(linha:string;referencia:integer):boolean
	convertecircuito(circ:Tcircuito):string
	expressaologica(expressao:string):string
	analizadorlexica(simb:Tsimbolo)
	novotemp:string()
	getexpressao:string()
	setexpressao(parte:string)
	obtervariavel(arg:string):Tvariaveis
	compilar(xip:Tmicrocontrolador;ladder:Tdiagrama):Tstringlist
	geracod(argumento:string;op1,op2,op3,op4:string):string
	acaosemantinot(op:string):string
	acaosemanticaand(op1,op2:string):string
	acaosemanticaor(op1,op2:string):string
	acaosemanticasaída(op1,op2,op3:string)
	decifra(simb:Tsimbolo)
	operacaologica(lista:Tstringlist):Tstringlist
	operacaoOr(lista:Tstringlist):Tstringlist
	operacaoAnd(lista:Tstringlist):Tstringlist
	transforma(simb:Tsimbolo):Tlist
	desmembra(lista:tstringlist;op:string)
	inicializa(xip:Tmicrocontrolador):Tstringlist
	abrir(arquivo:string)

Especificação (Diagrama de Seqüência Novo Projeto)

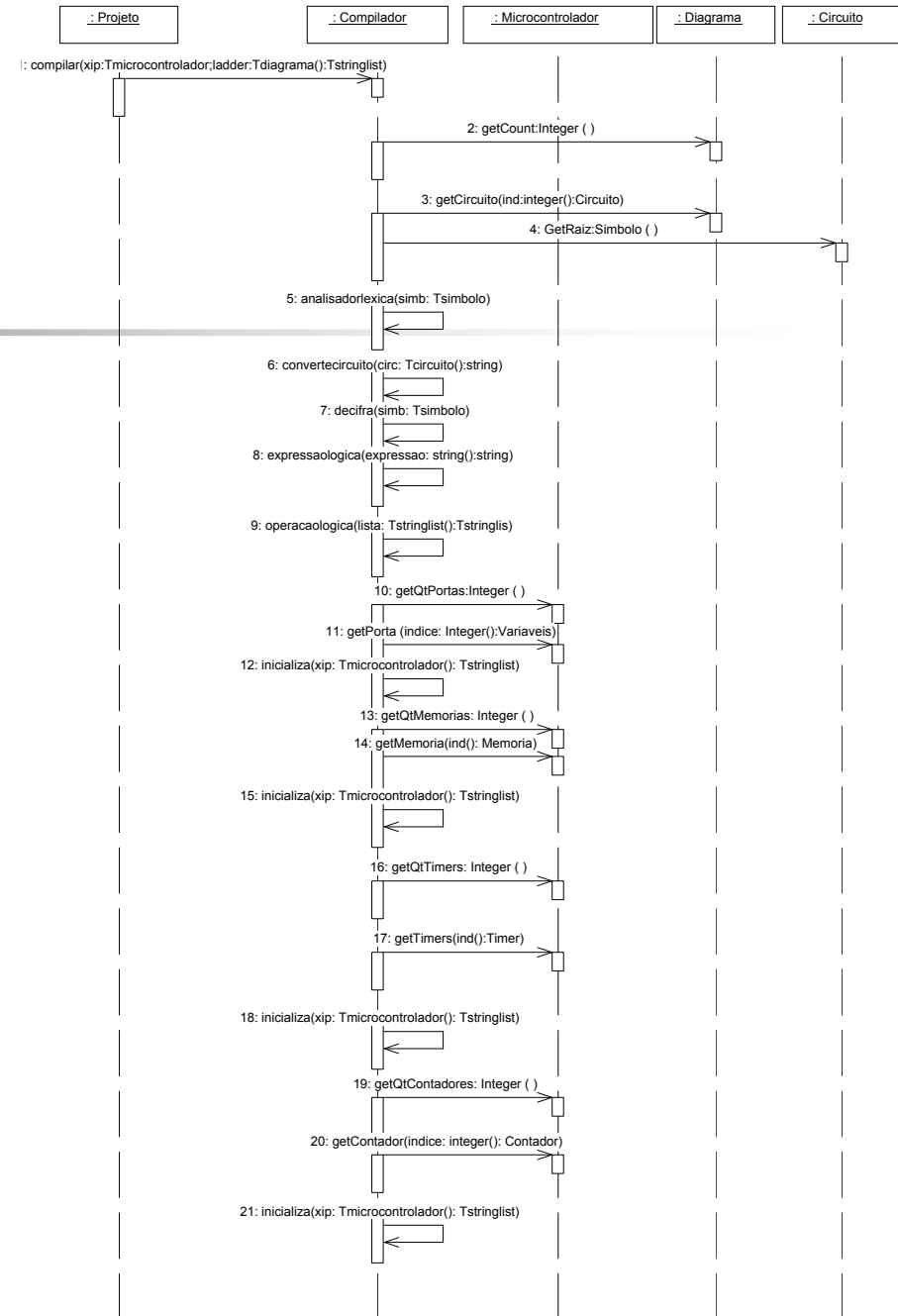


Especificação (Diagrama de Seqüência Desenhar Diagrama)





Especificação (Diagrama de Seqüência)

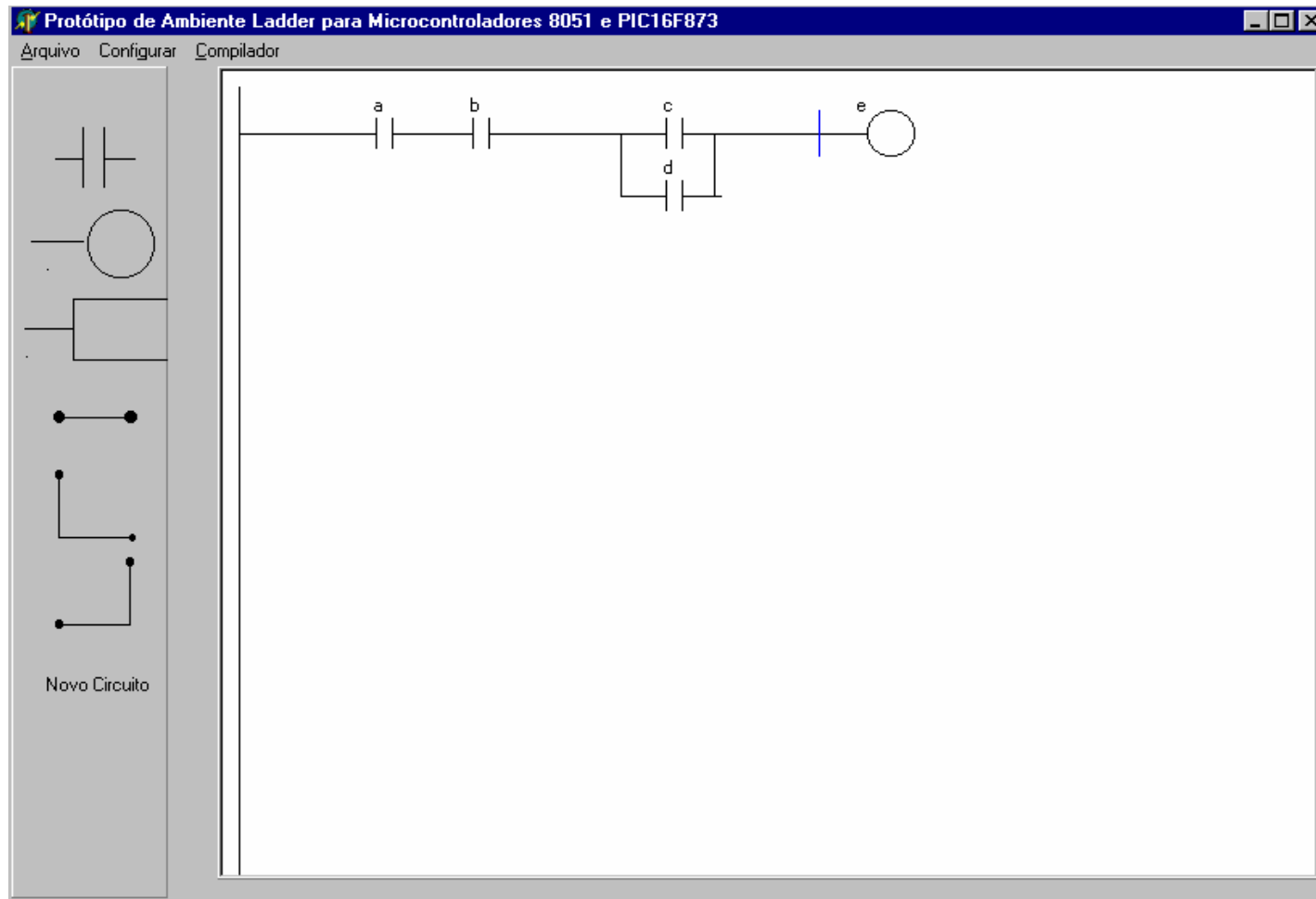




Estudo de caso


- Gerar código para o PIC16F873 que resolva a seguinte expressão booleana
- IF (A and B) and (C or D) then E

Estudo de caso (Tela de abertura)



Estudo de caso (Criando um novo projeto)



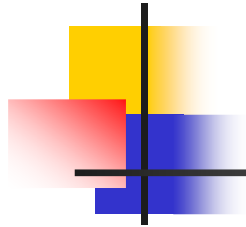


Estudo de caso (Declarando variáveis)

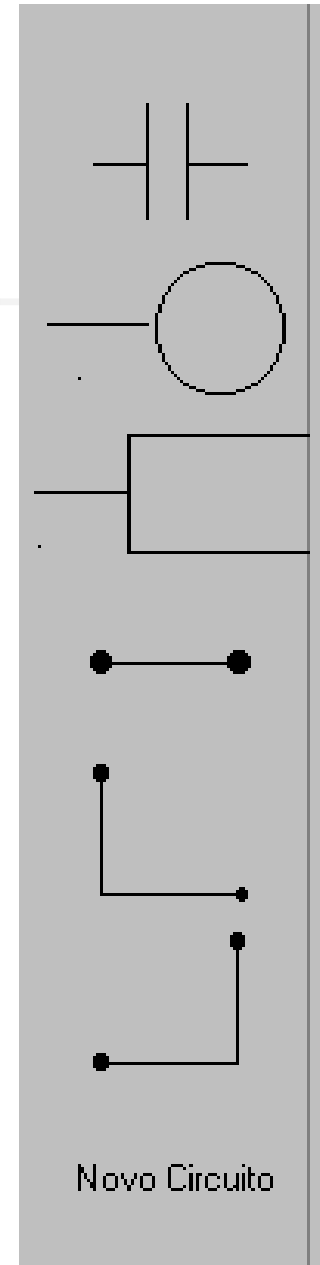
Configuração de Portas

Ra0	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída <input type="radio"/> A/D
Ra1	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída
Ra2	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída
Ra3	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída
Ra4	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída
Ra5	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída
Rb0	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída
Rb1	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída
Rb2	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída
Rb3	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída <input type="radio"/> ST
Rb4	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída <input type="radio"/> ST
Rb5	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída <input type="radio"/> ST
Rb6	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída <input type="radio"/> ST
Rb7	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída <input type="radio"/> ST
Pc0	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída
Pc1	<input type="text"/>	<input type="radio"/> Entrada <input type="radio"/> Saída

OK Cancelar




Estudo de caso (Desenhando o diagrama)

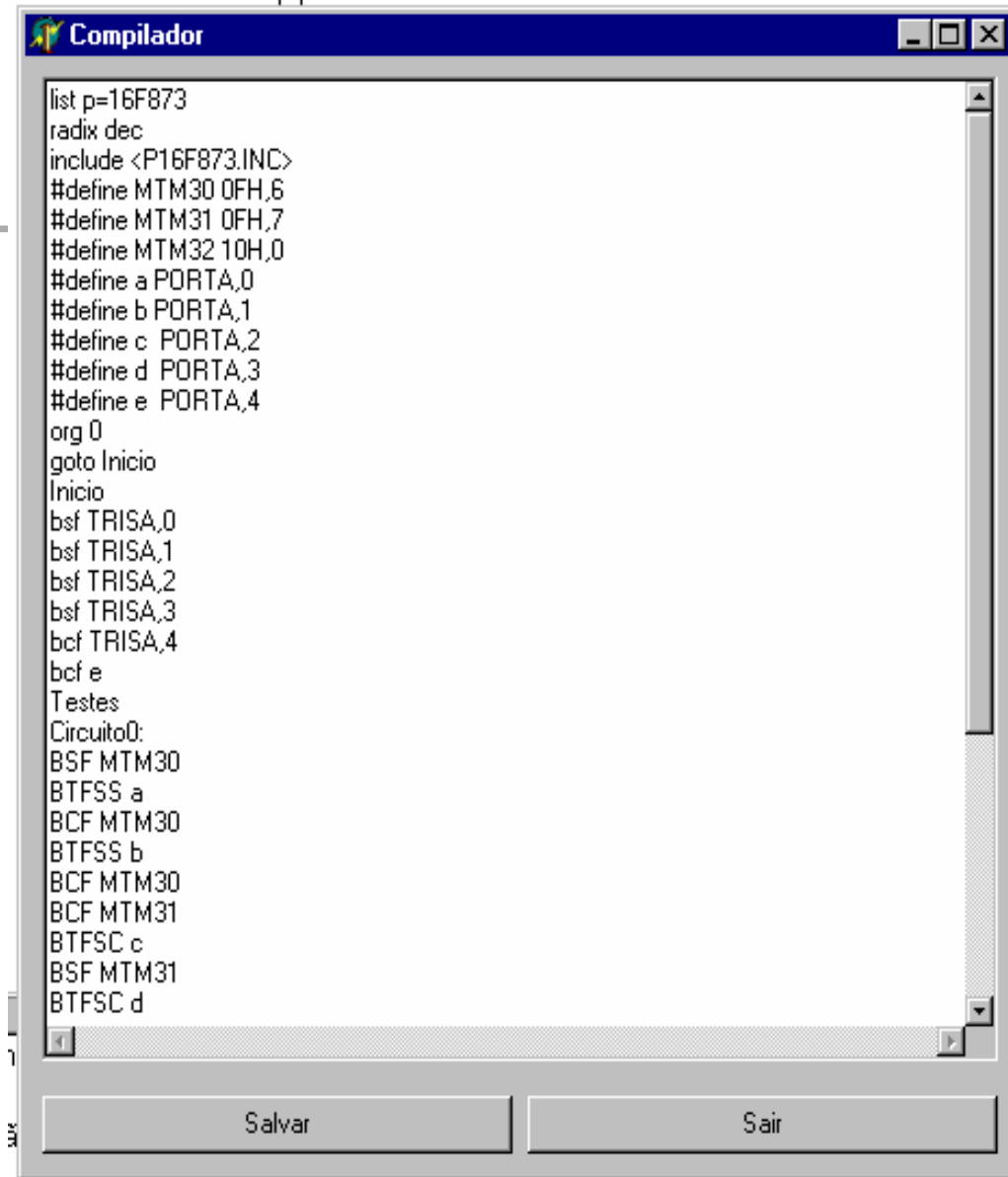


Estudo de caso (Escolhendo variáveis)

The image shows a software window titled "Escolha a porta" (Choose the port). The window has a dark blue title bar with a close button (X) on the right. Below the title bar is a tabbed interface with the following tabs: "Entradas" (selected), "Timers", "Saídas", "Contadores", "Memória", "Comparador AD", and "Sensor de Transição". The main area of the window is a large, empty gray rectangle. In the bottom-left corner, there is a checkbox labeled "Normalmente Fechada" (Normally Closed). To its right is a section titled "Tipo de Saída" (Output Type) with three radio button options: "Set", "Reset", and "Normal". The "Set" option is currently selected. In the bottom-right corner, there is a "Cancelar" (Cancel) button.



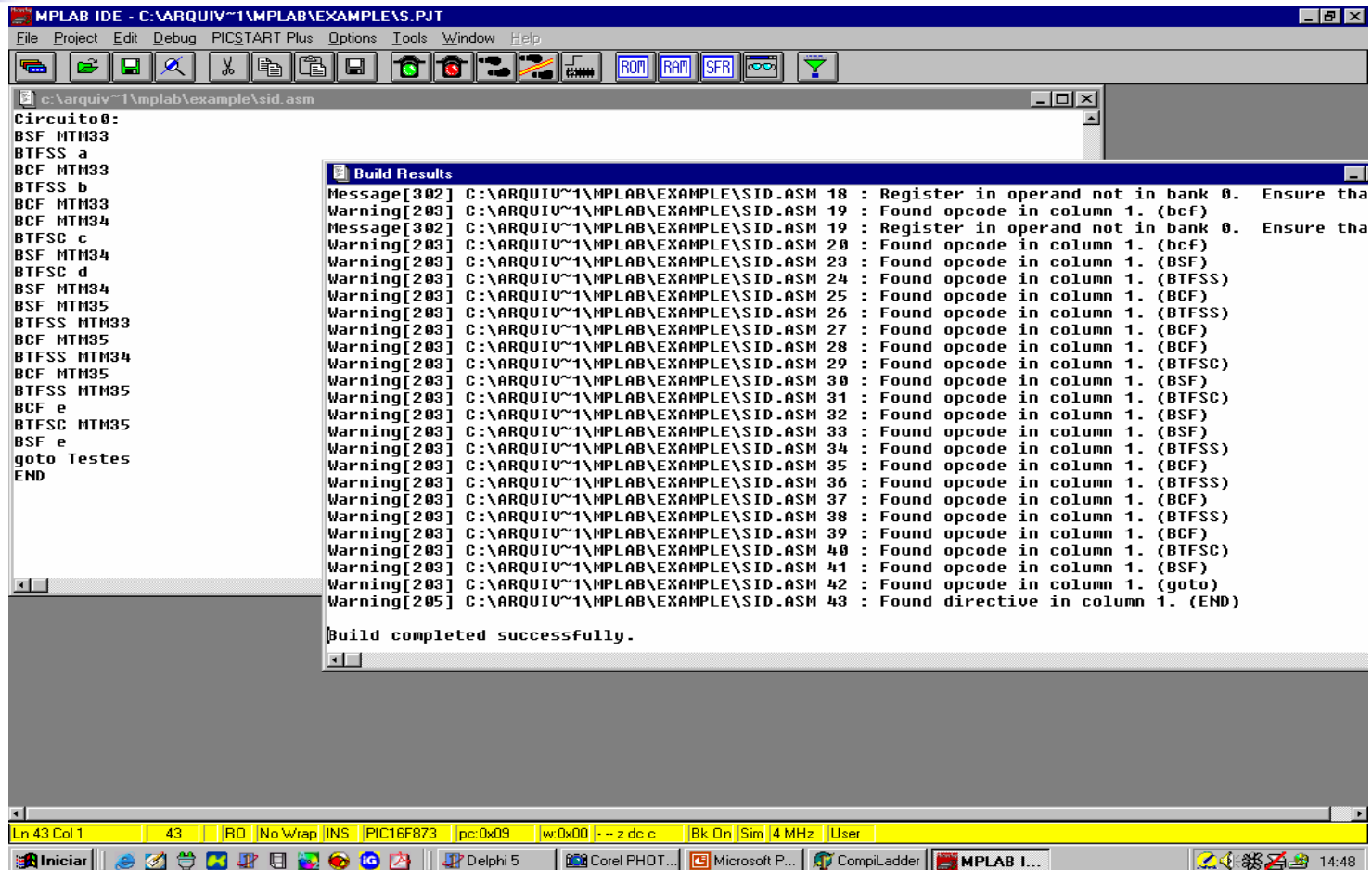
Estudo de caso (Compilando o diagrama)



```
list p=16F873
radix dec
include <P16F873.INC>
#define MTM30 0FH,6
#define MTM31 0FH,7
#define MTM32 10H,0
#define a PORTA,0
#define b PORTA,1
#define c PORTA,2
#define d PORTA,3
#define e PORTA,4
org 0
goto Inicio
Inicio
bsf TRISA,0
bsf TRISA,1
bsf TRISA,2
bsf TRISA,3
bcf TRISA,4
bcf e
Testes
Circuito0:
BSF MTM30
BTFSS a
BCF MTM30
BTFSS b
BCF MTM30
BCF MTM31
BTFSC c
BSF MTM31
BTFSC d
```

Salvar Sair

Estudo de caso (Montando o código)



The screenshot displays the MPLAB IDE interface. The main window shows assembly code for 'Circuito0' with instructions like BSF, BTFSS, BCF, and BTFSC. A 'Build Results' window is open, showing a list of warnings and messages. The warnings indicate that various opcodes (bcf, bsf, btfss, bcf, btfsc) were found in column 1, which is not a valid register operand. The build process completed successfully.

```
C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM
Circuito0:
BSF MTM33
BTFSS a
BCF MTM33
BTFSS b
BCF MTM33
BCF MTM34
BTFSC c
BSF MTM34
BTFSC d
BSF MTM34
BSF MTM35
BTFSS MTM33
BCF MTM35
BTFSS MTM34
BCF MTM35
BTFSS MTM35
BCF e
BTFSC MTM35
BSF e
goto Testes
END
```

```
Build Results
Message[302] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 18 : Register in operand not in bank 0. Ensure tha
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 19 : Found opcode in column 1. (bcf)
Message[302] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 19 : Register in operand not in bank 0. Ensure tha
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 20 : Found opcode in column 1. (bcf)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 23 : Found opcode in column 1. (BSF)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 24 : Found opcode in column 1. (BTFSS)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 25 : Found opcode in column 1. (BCF)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 26 : Found opcode in column 1. (BTFSS)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 27 : Found opcode in column 1. (BCF)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 28 : Found opcode in column 1. (BCF)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 29 : Found opcode in column 1. (BTFSC)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 30 : Found opcode in column 1. (BSF)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 31 : Found opcode in column 1. (BTFSC)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 32 : Found opcode in column 1. (BSF)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 33 : Found opcode in column 1. (BSF)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 34 : Found opcode in column 1. (BTFSS)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 35 : Found opcode in column 1. (BCF)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 36 : Found opcode in column 1. (BTFSS)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 37 : Found opcode in column 1. (BCF)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 38 : Found opcode in column 1. (BTFSS)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 39 : Found opcode in column 1. (BCF)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 40 : Found opcode in column 1. (BTFSC)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 41 : Found opcode in column 1. (BSF)
Warning[203] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 42 : Found opcode in column 1. (goto)
Warning[205] C:\ARQUIV~1\MPLAB\EXAMPLE\SID.ASM 43 : Found directive in column 1. (END)

Build completed successfully.
```



Resultados

- Na montagem o programa não apresentou erros de sintaxe
- Testando passo a passo o programa não apresentou erros de semântica



Conclusões

- A ferramenta auxilia na construção de aplicações embedded
- A linguagem Ladder abstrai a linguagem assembly dos microcontroladores
- O código precisa ser otimizado



Sugestões

- Adicionar um simulador ao projeto
- Incluir novos microcontroladores