



# MICROPROCESSADORES I

**KIT 8051 – Sistema de Desenvolvimento 8051**

**Keil  $\mu$ Vision 3: Depuração e Simulação II**

**Autores: Adriano Tavares, Jorge Cabral, José Mendes**

# 1 Objectivo

Apresentar um guia básico de como implementar em linguagem *assembly* os fluxogramas desenvolvidos nas aulas. Neste guia, o fluxograma que soluciona o exercício do slide A4-34 será codificado para *assembly* do MCS-51 e iremos usar o Keil  $\mu$ Vision3 para simular e depurar programas em *assembly* com mais que um ficheiro de código fonte.

## 1.1 Codificação do fluxograma

O fluxograma que soluciona o exercício proposto no slide A4-34 foi já discutido e apresentado nas aulas teóricas.

O fluxograma e a respectiva conversão para linguagem *assembly* são apresentados na Figura 1, que se segue:

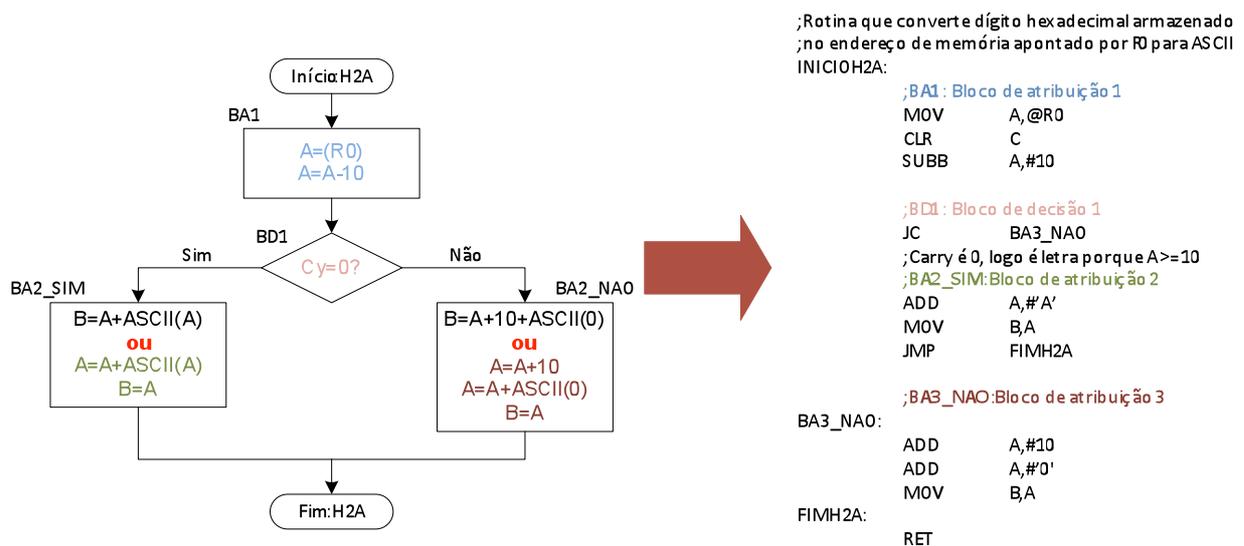


Figura 1 – Fluxograma e respectiva codificação

Salientar apenas que o fluxograma foi codificado como sendo uma subrotina, cujo parâmetro de entrada é o registo R0 que contém o endereço da memória de dados interna onde está armazenado o dígito hexadecimal a converter e o parâmetro de saída é o registo B. A subrotina utiliza ainda o registo Acumulador como registo auxiliar.

## 1.2 Descrição

O ambiente  $\mu$ Vision3 permite três formas diferentes de simulação:

- Simulação passo-a-passo,
- Simulação contínua com *breakpoints*,



### 1.3 Criar um projecto

Crie e “assemble” o projecto seguinte, certificando que a configuração do *debug* é idêntica à apresentada na Figura 2.

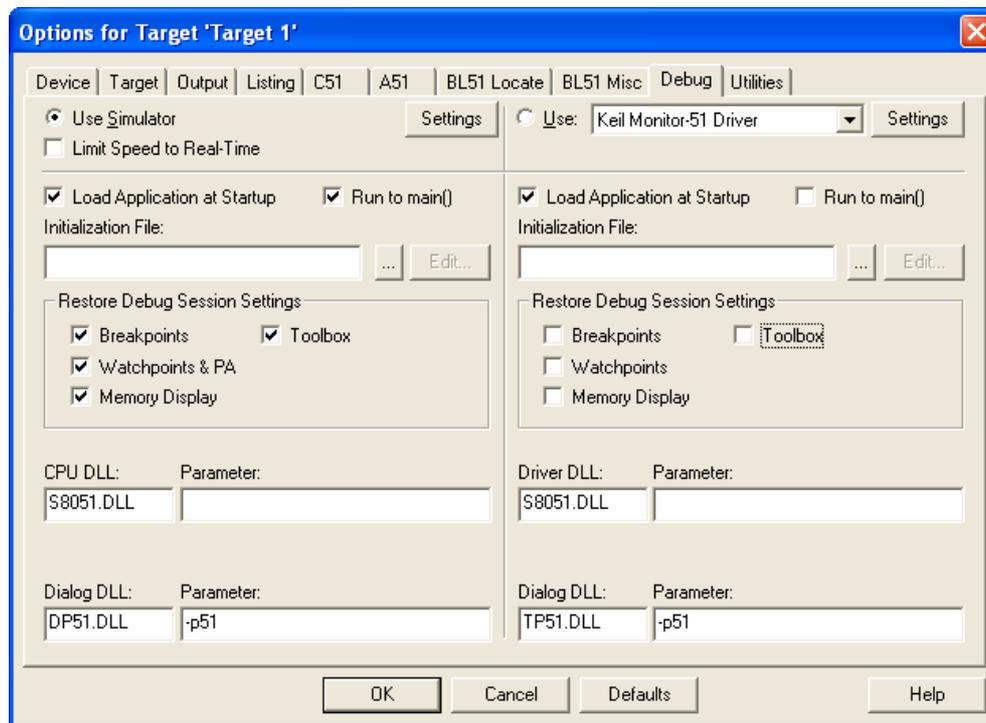


Figura 2 – Configuração de *debug*

Seguir os passos habituais para criação de um projecto mas agora com dois ficheiros de código: h2a.a51 e main.a51.

No ficheiro h2a.a51 está colocado o código *assembly* que implementa a subrotina que permite converter um dígito hexadecimal para ASCII. O código *assembly* presente neste ficheiro foi apresentado na Figura 1, com a excepção de algumas directivas para o *assembler* da família MCS-51. O ficheiro h2a.a51 é apresentado na Figura 3.

No ficheiro main.a51 está o código principal, onde se armazenam os valores dos dígitos hexadecimais na memória e se invoca, através de um *loop*, a subrotina implementada no ficheiro h2a.a51, guardando o resultado da conversão de dígito hexadecimal para o correspondente ASCII na memória de dados externa. O ficheiro main.a51 é apresentado na Figura 4.

```

01      NAME      H2A
02
03      PUBLIC   INICIOH2A
04
05      ;Rotina que converte digito hexadecimal armazenado
06      ;no endereço de memória apontado por R0 para ASCII
07      INICIOH2A:
08      ;BA1: Bloco de atribuição 1
09      MOV      A,@R0
10      CLR      C
11      SUBB    A,#10
12
13      ;ED1: Bloco de decisão 1
14      JC      BA3_NAO
15      ;Carry é 0, logo é letra porque A>=10
16      ;BA2_SIM: Bloco de atribuição 2
17      ADD     A,#'A'
18      MOV     B,A
19      JMP     FIMH2A
20
21      ;BA3_NAO: Bloco de atribuição 3
22      BA3_NAO:
23      ADD     A,#10
24      ADD     A,#'0'
25      MOV     B,A
26      FIMH2A:
27      RET
28
29      END

```

Figura 3 – Ficheiro h2a.a51

```

01      NAME      MAIN
02
03      EXTRN    CODE      (INICIOH2A)
04
05      NO_LOOPS EQU      5
06
07      CSEG    AT      0H
08      LJMP   START
09
10      CSEG    AT      30H
11      START:
12      MOV     20H,#05H
13      MOV     21H,#08H
14      MOV     22H,#0AH
15      MOV     23H,#0EH
16      MOV     24H,#07H
17
18      MOV     R2,#NO_LOOPS
19      MOV     R0,#20H
20      MOV     R1,#8H
21      LOOP:
22      MOV     A,@R0
23      PUSH  ACC
24      LCALL  INICIOH2A
25      MOV     A,B
26      MOVX   @R1,A
27      POP   ACC
28      INC   R0
29      INC   R1
30      DJNZ  R2,LOOP
31
32      SJMP  $
33      END

```

Figura 4 - Ficheiro main.a51

