

Engenharia Electrotécnica/ Engenharia Informática
Microprocessadores e Aplicações

Conversão de Binário para BCD

A conversão de um número binário para BCD depende do valor a ser convertido:

- caso a: 0 a 99 (2 algarismos BCD)
- caso b: 0 a 255 (1 byte)
- caso c: 0 a 999 (3 algarismos BCD)
- caso d: 0 a 9999 (4 algarismos BCD)

CASO a: Em R4 existe um número de 0 a 99. Converter o número para BCD e colocar o resultado em R6.

Ao dividir o número por 10, as unidades são colocadas em B e as dezenas em A.

```
BINBCD_99:MOV A,R4
          MOV B,#10
          DIV AB ; A=0X B=0Y
          SWAP A ; A=X0 B=0Y
          ADD A,B ; A=XY B=0Y
          MOV R6,A ;coloca em R6 o valor de A
          RET
```

CASO b: Em R4 existe um número de 0 a 255. Converter o número para BCD e colocar o resultado em R7 e R6. Divide-se o número por 100 para separar as centenas. Assim já é possível utilizar a solução do caso a.

```
BINBCD_255:MOV A,R4
           MOV B,#100
           DIV AB ;separar as centenas (dividir por 100)
           MOV R7,A ;R7 guarda o quociente (digito BCD mais
sigf.)
           MOV A,B ;igual a BINBCD_99
           MOV B,#10
           DIV AB
           SWAP A
           ADD A,B
           MOV R6,A
           RET
```



Engenharia Electrotécnica/ Engenharia Informática
Microprocessadores e Aplicações

CASO c: Em R5 e R4 existe um número de 0 a 999. Converter o número para BCD e colocar o resultado em R7 e R6. A solução utilizada consiste em somar 256 em R7 e R6 e decrementar R5 até que este seja 0; depois utilize-se uma rotina idêntica à BINBCD_255.

```
BINBCD_999:CLR A ;Coloca A a zero
            MOV R7,A ;Coloca R7 e R6 a zero para
            MOV R6,A ;receber os resultados
            MOV A,R5
            JZ LB1 ;se R5=0 (0<= num <=255), seguir adiante;
LB2: MOV A,R6
            ADD A,#56H ;somar 256 BCD e decrementar
            DA A ;R5 até que este chegue
            MOV R6,A ;a zero
            MOV A,R7
            ADDC A,#2
            DA A
            MOV R7,A
            DJNZ R5,LB2 ;decrementar R5;
LB1: MOV A,R4 ;parecido com BINBCD_255 (0<= num <=255)
            MOV B,#100
            DIV AB
            ADD A,R7
            DA A
            MOV R7,A
            MOV A,B
            MOV B,#10
            DIV AB
            SWAP A
            ADD A,B
            ADD A,R6
            DA A
            MOV R6,A
            MOV A,R7
            ADDC A,#0
            DA A
            MOV R7,A
            RET
```

CASO d: Em R5 e R4 existe um número do 0 a 9999. Converter o número para BCD e colocar o resultado em R7 e R6.

A solução adoptada no Caso c sugere um loop até que R5 seja zero. Mas para números grandes esse loop pode ser excessivo em termos de tempo. Uma solução a melhor consiste usar divisões por 1000, 100 e 10. O problema agora será como realizar divisões de 2 números de 16 bits utilizando a divisão de 8 bits disponível.



Engenharia Electrotécnica/ Engenharia Informática
Microprocessadores e Aplicações

Uma maneira de solucionar este problema é converter o byte menos significativo para BCD e de seguida converter também o byte mais significativo para BCD, multiplicá-lo por 256 e somar com o anterior. Uma multiplicação por 256 é fácil: multiplicar por 200 (multiplicar por 2 e deslocar 2 posições para a esquerda); multiplicar por 50 (multiplicar por 5 e deslocar 1); multiplicar por 6 e por último somar os três resultados.

Com esta solução pode-se pensar num caso e mais genérico:

CASO e: Em R4 e R3 existe um número binário de 16 bits (0 a 9999). Converter o número para BCD e colocar o seu resultado em R7, R6 e R5. Na solução serão utilizadas algumas subrotinas auxiliares, tais como:

Converter um byte do ACC para BCD e colocar o resultado em AUXH e AUXL

```
BCD8:MOV B,#100 ;parecido com BINBCD_255
      DIV AB
      MOV AUXH,A
      MOV A,B
      MOV B,#10
      DIV A,B
      SWAP A
      ADD A,B
      MOV AUXL,A
      RET
```

Receber um número BCD em R2 e R1 e o incrementá-lo o número de vezes especificado em R0 (R0 > 0).

```
ROT_INC: MOV R1AUX,R1
          MOV R2AUX,R2
          DEC R0
R_INC:  MOV A,R1AUX
          ADD A,R1
          DA A
          MOV R1,A
          MOV A,R2AUX
          ADDC A,R2
          DA A
          MOV R2,A
          DJNZ R0,R_INC
          RET
```

Somar dois números BCD, em que um número está em R2 e R1 e o outro está indicado pelo ponteiro R0. O resultado deve ser guardado no endereço apontado por R0.



Engenharia Electrotécnica/ Engenharia Informática

Microprocessadores e Aplicações

$[@(R0+1) \text{ e } @R0] \text{ } [@(R0+1) \text{ e } @R0] + [R2 \text{ e } R1]$

```
SOMA:MOV A,@R0
      ADD A,R1 ;soma os LSBs
      DA A ;ajuste decimal da soma
      MOV @R0,A ;guarda a soma em @R0
      INC R0 ;vai para proximo endereço
      MOV A,@R0
      ADDC A,R2 ;soma os MSBs com Carry
      DA A ;ajuste decimal da soma
      MOV @R0,A ;guarda o MSB da soma
      RET
```

Subrotina principal:

Os valores binários colocados em (R4, R3) serão colocados em BCD(R7, R6, R5)

```
BCD16:MOV A,R3 ;coloca em A o LSB
      ACALL BCD8 ;converte o LSB p/ BCD
              ;e coloca o resultado em AUXH e AUXL
      MOV R5,AUXL ;o LSB em BCD vai para R5 e R6
      MOV R6,AUXH
      MOV A,R4 ;coloca em A o MSB
      ACALL BCD8 ;converte o LSB p/ BCD
              ;e coloca o resultado em AUXH e AUXL
      MOV R2,AUXH ;o MSB em BCD vai para R1 e R2
      MOV R1,AUXL
      MOV R0,#6
      ACALL ROT_INC ;Calcula BCD(MSB)*6
      MOV R0,#5 ;Para que a soma seja colocada em R5 e R6
      ACALL SOMA ;(R6,R5) <- (R6,R5) + 6*BCD(MSB)
      CLR A ;Coloca A a zero
      RLC A ;houve carry? (por segurança)
      MOV R7,A ;coloca carry em R7;
      MOV R2,AUXH ;AUXH e AUXL ainda guardam o BCD(MSB)
      MOV R1,AUXL
      MOV R0,#5
      ACALL ROT_INC ;Calcula BCD(MSB)*5
              ;A R2 R1 deslocar
      MOV A,R2; 0X 0X YZ uma
      MOV R0,#1
      XCHD A,@R0 ; 0Z 0X YX posicao BCD
      SWAP A ; Z0 0X YX para a
      XCH A,R1 ; YX 0X Z0 esquerda
      SWAP A ; XY 0X Z0 (multipl. por 10)
      XCH A,R2 ; 0X XY Z0
      MOV R0,#5 ;Para que a soma seja colocada em R5 e R6
      ACALL SOMA ;(R6,R5) <- (R6,R5)+ 6*BCD(MSB)+50*BCD(MSB)
      MOV A,R7
```



Engenharia Electrotécnica/ Engenharia Informática
Microprocessadores e Aplicações

```
ADDC A,#0 ;acrescenta o carry a R7
DA A
MOV R7,A
MOV R2,AUXH ;AUXH e AUXL ainda guardam o BCD(MSB)
MOV R1,AUXL
MOV R0,#2
ACALL ROT_INC ;Calcula BCD(MSB)*2
MOV R0,#6 ;Para que a soma seja colocada em R6 e R7
ACALL SOMA
RET
```

Deve-se notar que existem muitas outras rotinas para converter BIN para BCD.