НИУ ИТМО

Факультет ПИиКТ

Лабораторная работа №2

Реализация программного перевода чисел между системами счисления

Дисциплина: Архитектура компьютера

Студент: Ноговицын Михаил Павлович

Вариант: 9

Преподаватель: Тропченко Андрей Александрович

Санкт-Петербург

2020г.

**Цель работы:**

Изучение программно реализуемых алгоритмов перевода чисел между двоично-десятичной и двоичной системами счисления, написание соответствующих программ, изучение и реализация ввода-вывода на языках C и Ассемблер.

**Исходный текст программы целочисленного перевода (P2 хранит двоичное значение, P3 – обратно переведённое двоично-десятичное значение):**

#include <reg51.h>

int sum = 0;

int i, t, mult;

int pow (a, b) {

if (b == 0) {

return 1;

}

mult = 1;

for (t = 0; t < b; t++) {

mult = mult \* a;

}

return mult;

}

main() {

P2 = (P1 >> 4) \* 10 + (P1 & 0x0f);

for (i = 7; i >= 0; i--) {

if ((P2 >> i) & 0x01 == 1){

sum += pow(2, i);

P3 = sum;

}

}

P3 = sum;

return 0;

}

**Листинг:**

C51 COMPILER V9.60.0.0 CONVERTER 10/08/2020 05:29:49 PAGE 1

C51 COMPILER V9.60.0.0, COMPILATION OF MODULE CONVERTER

OBJECT MODULE PLACED IN .\Objects\Converter.obj

COMPILER INVOKED BY: C:\Keil\_v5\C51\BIN\C51.EXE Converter.c COMPACT ROM(COMPACT) OPTIMIZE(1,SPEED) BROWSE DEBUG OBJECTEX

-TEND CODE PRINT(.\Listings\Converter.lst) TABS(2) OBJECT(.\Objects\Converter.obj)

line level source

1 #include <reg51.h>

2

3 int sum = 0;

4 int i, t, mult;

5

6 int pow (a, b) {

7 1 if (b == 0) {

8 2 return 1;

9 2 }

10 1 mult = 1;

11 1 for (t = 0; t < b; t++) {

12 2 mult = mult \* a;

13 2 }

14 1 return mult;

15 1

16 1 }

17

18 main() {

19 1

20 1 P2 = (P1 >> 4) \* 10 + (P1 & 0x0f);

21 1

22 1 for (i = 7; i >= 0; i--) {

23 2

24 2 if ((P2 >> i) & 0x01 == 1){

25 3 sum += pow(2, i);

26 3 P3 = sum;

27 3 }

28 2

29 2 }

30 1

31 1 P3 = sum;

32 1

33 1

34 1

35 1

36 1 return 0;

37 1

38 1 }

39

40

41

42

43 //P3 = ((P2 << 4) + (P2 & 0x0f) \* 10) / 10;

C51 COMPILER V9.60.0.0 CONVERTER 10/08/2020 05:29:49 PAGE 2

ASSEMBLY LISTING OF GENERATED OBJECT CODE

; FUNCTION \_pow (BEGIN)

; SOURCE LINE # 6

0000 7800 R MOV R0,#LOW a

0002 EE MOV A,R6

0003 F2 MOVX @R0,A

0004 08 INC R0

0005 EF MOV A,R7

0006 F2 MOVX @R0,A

0007 7800 R MOV R0,#LOW b

0009 EC MOV A,R4

000A F2 MOVX @R0,A

000B 08 INC R0

000C ED MOV A,R5

000D F2 MOVX @R0,A

; SOURCE LINE # 7

000E 7800 R MOV R0,#LOW b

0010 E2 MOVX A,@R0

0011 FE MOV R6,A

0012 08 INC R0

0013 E2 MOVX A,@R0

0014 FF MOV R7,A

0015 EF MOV A,R7

0016 4E ORL A,R6

0017 7005 JNZ ?C0001

; SOURCE LINE # 8

0019 7E00 MOV R6,#00H

001B 7F01 MOV R7,#01H

001D 22 RET

; SOURCE LINE # 9

001E ?C0001:

; SOURCE LINE # 10

001E 7800 R MOV R0,#LOW mult

0020 E4 CLR A

0021 F2 MOVX @R0,A

0022 08 INC R0

0023 7401 MOV A,#01H

0025 F2 MOVX @R0,A

; SOURCE LINE # 11

0026 7800 R MOV R0,#LOW t

0028 E4 CLR A

0029 F2 MOVX @R0,A

002A 08 INC R0

002B E4 CLR A

002C F2 MOVX @R0,A

002D ?C0003:

002D 7800 R MOV R0,#LOW b

002F E2 MOVX A,@R0

0030 FE MOV R6,A

0031 08 INC R0

0032 E2 MOVX A,@R0

0033 FF MOV R7,A

0034 7800 R MOV R0,#LOW t

0036 E2 MOVX A,@R0

0037 FC MOV R4,A

0038 08 INC R0

0039 E2 MOVX A,@R0

003A FD MOV R5,A

003B C3 CLR C

003C ED MOV A,R5

C51 COMPILER V9.60.0.0 CONVERTER 10/08/2020 05:29:49 PAGE 3

003D 9F SUBB A,R7

003E EE MOV A,R6

003F 6480 XRL A,#080H

0041 F8 MOV R0,A

0042 EC MOV A,R4

0043 6480 XRL A,#080H

0045 98 SUBB A,R0

0046 5025 JNC ?C0004

; SOURCE LINE # 12

0048 7800 R MOV R0,#LOW mult

004A E2 MOVX A,@R0

004B FE MOV R6,A

004C 08 INC R0

004D E2 MOVX A,@R0

004E FF MOV R7,A

004F 7800 R MOV R0,#LOW a

0051 E2 MOVX A,@R0

0052 FC MOV R4,A

0053 08 INC R0

0054 E2 MOVX A,@R0

0055 FD MOV R5,A

0056 120000 E LCALL ?C?IMUL

0059 7800 R MOV R0,#LOW mult

005B EE MOV A,R6

005C F2 MOVX @R0,A

005D 08 INC R0

005E EF MOV A,R7

005F F2 MOVX @R0,A

; SOURCE LINE # 13

0060 7800 R MOV R0,#LOW t+01H

0062 E2 MOVX A,@R0

0063 2401 ADD A,#01H

0065 F2 MOVX @R0,A

0066 18 DEC R0

0067 E2 MOVX A,@R0

0068 3400 ADDC A,#00H

006A F2 MOVX @R0,A

006B 80C0 SJMP ?C0003

006D ?C0004:

; SOURCE LINE # 14

006D 7800 R MOV R0,#LOW mult

006F E2 MOVX A,@R0

0070 FE MOV R6,A

0071 08 INC R0

0072 E2 MOVX A,@R0

0073 FF MOV R7,A

; SOURCE LINE # 16

0074 ?C0002:

0074 22 RET

; FUNCTION \_pow (END)

; FUNCTION main (BEGIN)

; SOURCE LINE # 18

; SOURCE LINE # 20

0000 E590 MOV A,P1

0002 C4 SWAP A

0003 540F ANL A,#0FH

0005 FF MOV R7,A

0006 EF MOV A,R7

0007 75F00A MOV B,#0AH

000A A4 MUL AB

000B FF MOV R7,A

C51 COMPILER V9.60.0.0 CONVERTER 10/08/2020 05:29:49 PAGE 4

000C E590 MOV A,P1

000E 540F ANL A,#0FH

0010 FE MOV R6,A

0011 EF MOV A,R7

0012 2E ADD A,R6

0013 F5A0 MOV P2,A

; SOURCE LINE # 22

0015 7800 R MOV R0,#LOW i

0017 E4 CLR A

0018 F2 MOVX @R0,A

0019 08 INC R0

001A 7407 MOV A,#07H

001C F2 MOVX @R0,A

001D ?C0006:

001D 7800 R MOV R0,#LOW i

001F E2 MOVX A,@R0

0020 FE MOV R6,A

0021 08 INC R0

0022 E2 MOVX A,@R0

0023 FF MOV R7,A

0024 C3 CLR C

0025 EE MOV A,R6

0026 6480 XRL A,#080H

0028 9480 SUBB A,#080H

002A 4044 JC ?C0007

; SOURCE LINE # 24

002C 7800 R MOV R0,#LOW i

002E E2 MOVX A,@R0

002F FE MOV R6,A

0030 08 INC R0

0031 E2 MOVX A,@R0

0032 FF MOV R7,A

0033 A807 MOV R0,AR7

0035 E5A0 MOV A,P2

0037 08 INC R0

0038 8002 SJMP ?C0012

003A ?C0011:

003A C3 CLR C

003B 13 RRC A

003C ?C0012:

003C D8FC DJNZ R0,?C0011

003E FF MOV R7,A

003F EF MOV A,R7

0040 30E020 JNB ACC.0,?C0008

; SOURCE LINE # 25

0043 7800 R MOV R0,#LOW i

0045 E2 MOVX A,@R0

0046 FC MOV R4,A

0047 08 INC R0

0048 E2 MOVX A,@R0

0049 FD MOV R5,A

004A 7F02 MOV R7,#02H

004C 7E00 MOV R6,#00H

004E 120000 R LCALL \_pow

0051 7800 R MOV R0,#LOW sum+01H

0053 E2 MOVX A,@R0

0054 2F ADD A,R7

0055 F2 MOVX @R0,A

0056 18 DEC R0

0057 E2 MOVX A,@R0

0058 3E ADDC A,R6

0059 F2 MOVX @R0,A

C51 COMPILER V9.60.0.0 CONVERTER 10/08/2020 05:29:49 PAGE 5

; SOURCE LINE # 26

005A 7800 R MOV R0,#LOW sum

005C E2 MOVX A,@R0

005D FE MOV R6,A

005E 08 INC R0

005F E2 MOVX A,@R0

0060 FF MOV R7,A

0061 8FB0 MOV P3,R7

; SOURCE LINE # 27

; SOURCE LINE # 29

0063 ?C0008:

0063 7800 R MOV R0,#LOW i+01H

0065 E2 MOVX A,@R0

0066 24FF ADD A,#0FFH

0068 F2 MOVX @R0,A

0069 18 DEC R0

006A E2 MOVX A,@R0

006B 34FF ADDC A,#0FFH

006D F2 MOVX @R0,A

006E 80AD SJMP ?C0006

0070 ?C0007:

; SOURCE LINE # 31

0070 7800 R MOV R0,#LOW sum

0072 E2 MOVX A,@R0

0073 FE MOV R6,A

0074 08 INC R0

0075 E2 MOVX A,@R0

0076 FF MOV R7,A

0077 8FB0 MOV P3,R7

; SOURCE LINE # 36

0079 E4 CLR A

007A 7E00 MOV R6,#00H

007C 7F00 MOV R7,#00H

; SOURCE LINE # 38

007E 22 RET

; FUNCTION main (END)

MODULE INFORMATION: STATIC OVERLAYABLE

CODE SIZE = 244 ----

CONSTANT SIZE = ---- ----

XDATA SIZE = ---- ----

PDATA SIZE = 12 ----

DATA SIZE = ---- ----

IDATA SIZE = ---- ----

BIT SIZE = ---- ----

END OF MODULE INFORMATION.

C51 COMPILATION COMPLETE. 0 WARNING(S), 0 ERROR(S)

**Исходный текст программы дробного перевода (аргумент N < 1, P2 хранит двоичное значение, P3 – обратно переведённое двоично-десятичное значение):**

#include <reg51.h>

int y;

main() {

while(1) {

y = (((P1 & 0xf0) >> 4) \* 10 + (P1 & 0xf) << 8 );

y = (y % 100 > 50)? y / 100 + 1 : y / 100;

P2 = y;

y \*= 10;

P3 = ((y & 0xf00) >> 4);

P3 += (((y & 0xff) \* 10) & 0xf00) >> 8;

}

}

**Листинг:**

C51 COMPILER V9.60.0.0 IOPROGRAMC 10/08/2020 04:21:12 PAGE 1

C51 COMPILER V9.60.0.0, COMPILATION OF MODULE IOPROGRAMC

OBJECT MODULE PLACED IN .\Objects\IOProgramC.obj

COMPILER INVOKED BY: C:\Keil\_v5\C51\BIN\C51.EXE ..\LAB2\IOProgramC.c ROM(SMALL) OPTIMIZE(1,SPEED) BROWSE DEBUG OBJECTEXT

-END CODE PRINT(.\Listings\IOProgramC.lst) TABS(2) OBJECT(.\Objects\IOProgramC.obj)

line level source

1 #include <reg51.h>

2 int y;

3

4 main() {

5 1 while(1) {

6 2 y = (((P1 & 0xf0) >> 4) \* 10 + (P1 & 0xf) << 8 );

7 2 y = (y % 100 > 50)? y / 100 + 1 : y / 100;

8 2 P2 = y;

9 2 y \*= 10;

10 2 P3 = ((y & 0xf00) >> 4);

11 2 P3 += (((y & 0xff) \* 10) & 0xf00) >> 8;

12 2 }

13 1 }

C51 COMPILER V9.60.0.0 IOPROGRAMC 10/08/2020 04:21:12 PAGE 2

ASSEMBLY LISTING OF GENERATED OBJECT CODE

; FUNCTION main (BEGIN)

; SOURCE LINE # 4

0000 ?C0001:

; SOURCE LINE # 5

; SOURCE LINE # 6

0000 E590 MOV A,P1

0002 54F0 ANL A,#0F0H

0004 FF MOV R7,A

0005 EF MOV A,R7

0006 C4 SWAP A

0007 540F ANL A,#0FH

0009 FF MOV R7,A

000A 7E0A MOV R6,#0AH

000C EF MOV A,R7

000D 8EF0 MOV B,R6

000F A4 MUL AB

0010 FF MOV R7,A

0011 AEF0 MOV R6,B

0013 E590 MOV A,P1

0015 540F ANL A,#0FH

0017 FD MOV R5,A

0018 7C00 MOV R4,#00H

001A EF MOV A,R7

001B 2D ADD A,R5

001C FF MOV R7,A

001D EE MOV A,R6

001E 3C ADDC A,R4

001F FE MOV R6,A

0020 EF MOV A,R7

0021 750000 R MOV y+01H,#00H

0024 F500 R MOV y,A

; SOURCE LINE # 7

0026 AE00 R MOV R6,y

0028 AF00 R MOV R7,y+01H

002A 7C00 MOV R4,#00H

002C 7D64 MOV R5,#064H

002E 1100 E ACALL ?C?SIDIV

0030 D3 SETB C

0031 ED MOV A,R5

0032 9432 SUBB A,#032H

0034 EC MOV A,R4

0035 6480 XRL A,#080H

0037 9480 SUBB A,#080H

0039 4014 JC ?C0003

003B AE00 R MOV R6,y

003D AF00 R MOV R7,y+01H

003F 7C00 MOV R4,#00H

0041 7D64 MOV R5,#064H

0043 1100 E ACALL ?C?SIDIV

0045 EF MOV A,R7

0046 2401 ADD A,#01H

0048 FF MOV R7,A

0049 EE MOV A,R6

004A 3400 ADDC A,#00H

004C FE MOV R6,A

004D 800A SJMP ?C0004

004F ?C0003:

004F AE00 R MOV R6,y

0051 AF00 R MOV R7,y+01H

C51 COMPILER V9.60.0.0 IOPROGRAMC 10/08/2020 04:21:12 PAGE 3

0053 7C00 MOV R4,#00H

0055 7D64 MOV R5,#064H

0057 1100 E ACALL ?C?SIDIV

0059 ?C0004:

0059 8E00 R MOV y,R6

005B 8F00 R MOV y+01H,R7

; SOURCE LINE # 8

005D 8500A0 R MOV P2,y+01H

; SOURCE LINE # 9

0060 AE00 R MOV R6,y

0062 AF00 R MOV R7,y+01H

0064 7C00 MOV R4,#00H

0066 7D0A MOV R5,#0AH

0068 1100 E ACALL ?C?IMUL

006A 8E00 R MOV y,R6

006C 8F00 R MOV y+01H,R7

; SOURCE LINE # 10

006E E500 R MOV A,y

0070 540F ANL A,#0FH

0072 FE MOV R6,A

0073 7F00 MOV R7,#00H

0075 EF MOV A,R7

0076 7804 MOV R0,#04H

0078 ?C0006:

0078 CE XCH A,R6

0079 A2E7 MOV C,ACC.7

007B 13 RRC A

007C CE XCH A,R6

007D 13 RRC A

007E D8F8 DJNZ R0,?C0006

0080 FF MOV R7,A

0081 8FB0 MOV P3,R7

; SOURCE LINE # 11

0083 7E00 MOV R6,#00H

0085 AF00 R MOV R7,y+01H

0087 7C00 MOV R4,#00H

0089 7D0A MOV R5,#0AH

008B 1100 E ACALL ?C?IMUL

008D EE MOV A,R6

008E 540F ANL A,#0FH

0090 FE MOV R6,A

0091 7F00 MOV R7,#00H

0093 EE MOV A,R6

0094 FF MOV R7,A

0095 33 RLC A

0096 95E0 SUBB A,ACC

0098 FE MOV R6,A

0099 EF MOV A,R7

009A 25B0 ADD A,P3

009C F5B0 MOV P3,A

; SOURCE LINE # 12

009E 0100 R AJMP ?C0001

00A0 22 RET

; FUNCTION main (END)

MODULE INFORMATION: STATIC OVERLAYABLE

CODE SIZE = 161 ----

CONSTANT SIZE = ---- ----

XDATA SIZE = ---- ----

PDATA SIZE = ---- ----

C51 COMPILER V9.60.0.0 IOPROGRAMC 10/08/2020 04:21:12 PAGE 4

DATA SIZE = 2 ----

IDATA SIZE = ---- ----

BIT SIZE = ---- ----

END OF MODULE INFORMATION.

C51 COMPILATION COMPLETE. 0 WARNING(S), 0 ERROR(S)

**Программы ввода-вывода двузначного числа:**

(Си)

#include <reg51.h>

main() {

P2 = P1;

return 0;

}

Листинг:

C51 COMPILER V9.60.0.0 IOCONTROLLER 10/08/2020 04:14:11 PAGE 1

C51 COMPILER V9.60.0.0, COMPILATION OF MODULE IOCONTROLLER

OBJECT MODULE PLACED IN .\Objects\IOController.obj

COMPILER INVOKED BY: C:\Keil\_v5\C51\BIN\C51.EXE IOController.c OPTIMIZE(1,SPEED) BROWSE DEBUG OBJECTEXTEND CODE PRINT(.\

-Listings\IOController.lst) TABS(2) OBJECT(.\Objects\IOController.obj)

line level source

1 #include <reg51.h>

2

3 main() {

4 1 P2 = P1;

5 1 return 0;

6 1 }

C51 COMPILER V9.60.0.0 IOCONTROLLER 10/08/2020 04:14:11 PAGE 2

ASSEMBLY LISTING OF GENERATED OBJECT CODE

; FUNCTION main (BEGIN)

; SOURCE LINE # 3

; SOURCE LINE # 4

0000 8590A0 MOV P2,P1

; SOURCE LINE # 5

0003 E4 CLR A

0004 7E00 MOV R6,#00H

0006 7F00 MOV R7,#00H

; SOURCE LINE # 6

0008 22 RET

; FUNCTION main (END)

MODULE INFORMATION: STATIC OVERLAYABLE

CODE SIZE = 9 ----

CONSTANT SIZE = ---- ----

XDATA SIZE = ---- ----

PDATA SIZE = ---- ----

DATA SIZE = ---- ----

IDATA SIZE = ---- ----

BIT SIZE = ---- ----

END OF MODULE INFORMATION.

C51 COMPILATION COMPLETE. 0 WARNING(S), 0 ERROR(S)

(Ассемблер)

cseg at 0

mov a, P1

mov P2, a

end

Листинг:

A51 MACRO ASSEMBLER IOCONTROLLER 10/08/2020 04:16:16 PAGE 1

MACRO ASSEMBLER A51 V8.2.7.0

OBJECT MODULE PLACED IN .\Objects\IOController.obj

ASSEMBLER INVOKED BY: C:\Keil\_v5\C51\BIN\A51.EXE IOController.a51 SET(SMALL) DEBUG PRINT(.\Listings\IOController.lst) OB

JECT(.\Objects\IOController.obj) EP

LOC OBJ LINE SOURCE

---- 1 cseg at 0

2

0000 E590 3 mov a, P1

0002 F5A0 4 mov P2, a

5

6 end

A51 MACRO ASSEMBLER IOCONTROLLER 10/08/2020 04:16:16 PAGE 2

SYMBOL TABLE LISTING

------ ----- -------

N A M E T Y P E V A L U E ATTRIBUTES

P1 . . . . . . . . D ADDR 0090H A

P2 . . . . . . . . D ADDR 00A0H A

REGISTER BANK(S) USED: 0

ASSEMBLY COMPLETE. 0 WARNING(S), 0 ERROR(S)

**Вывод:**

В процессе выполнения работы изучены основные конструкции использования аппаратных средств MCS-51 языков Си и Ассемблер и принципы программного перевода чисел между двоично-десятичной и двоичной системами счисления, реализованы соответствующие программы, приведены листинги.

Сравнение программ показало, что простые программы на Си могут быть меньше по размеру, а соответственно более быстрыми в исполнении ввиду оптимизационных замен в сгенерированном коде на ассемблере.