# 第7章 STC单片机汇编语言 编程模型

何宾 2018.03

#### Keil µVision汇编语言设计流程 --建立新的设计工程

- 打开µVision5集成开发环境。
- 在μVision5集成开发环境主界面主菜单下,选择Project->New μVision Project...。
- 出现Create New Project对话框界面。在文件名右侧的文本框中输入top。

注:表示该工程的名字是top.uvproj。

#### Keil µVision汇编语言设计流程 --建立新的设计工程

- 单击OK按钮。
- 出现Select a CPU Data Base File对话框界面。在该界面中的下拉框中,选择STC MCU Database选项。
- 单击OK按钮。
- 出现Select Device for Target' Target 1' …对话框界面。在该界面中左侧的窗口中,找到并展开STC前面的'+'。在展开场中,找到并选择STC15W4K32S4。

### Keil µVision汇编语言设计流程 --建立新的设计工程

- 单击OK按钮。
- 出现Copy'STARTUP.A51' to Project Folder and Add File to Project?对话框界面。该界面提示是不是在当前设计工程中添加STARTUP.A51文件。

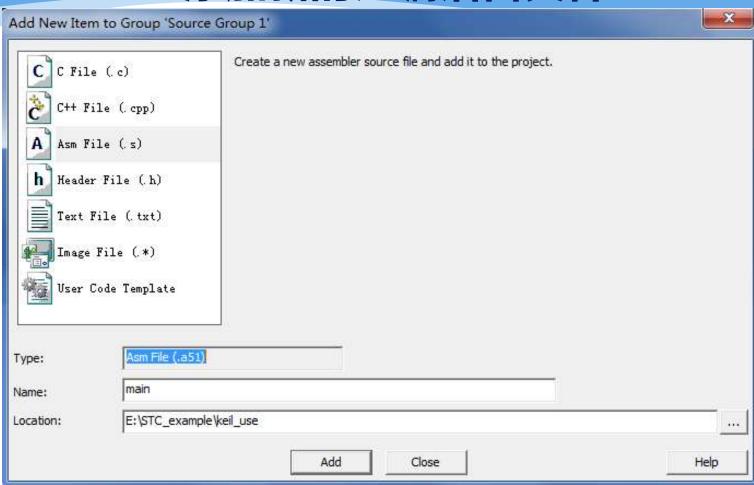
注:在汇编语言程序设计中,不需要添加该文件。在C语言程序设计中,也不需要添加该文件。

- 单击 "否 (N) " 按钮。
- 在主界面左侧窗口中,选择Project标签。在该标签窗口下,给出了工程信息。

### Keil µVision汇编语言设计流程 --添加新的汇编语言文件

- 在Project窗口界面下,选择Source Group 1,单击右键,出现浮动菜单。在浮动菜单内,选择Add New Item to Group 'Source Group 1' 选项。
- 出现Add New Item to Group' Source Group 1' 对话框界面,按下面设置参数:
  - □ 在该界面左侧窗口中,选中Asm File(.s)。
  - □ 在Name右侧的文本框中输入main。

### Keil µVision汇编语言设计流程 --添加新的汇编语言文件



注:该汇编语言的文件名字为main.a51。

■ 点击Add按钮。

#### Keil µVision汇编语言设计流程 --添加新的汇编语言文件

- 在右图所示的Project窗口中,在Source Group 1子目录下添加名字为main.a51的汇编语言文件。
- 在右侧窗口中,自动打开了main.a51文件。
- 输入代码, 如后图所示。
- 保存设计代码。



```
1 NAME main
   seg 1 SEGMENT CODE
                      seg 1
 3
               RSEG
               DB
   TABLE:
                       3,2,5,0x0EE
    seg 2 SEGMENT IDATA
 6
               RSEG
                       seg 2
    BUFFER:
               DS
                       19
 8
    seg 3 SEGMENT XDATA
 9
               RSEG
                       seg 3
10
    XBUFFER:
               Ds
                       100
11
12
   myprog SEGMENT CODE
13
               RSEG
                       myprog
               USING 0
14
               LJMP
15
                      main
16
                      0x100
               org
17
   main:
               MOV
                      DPTR, #TABLE
                                       ;access code segment
18
                      A, #3
               MOV
              MOVC
                     A, @A+DPTR
19
20
               MOV
                     P1,#0
21
               MOV
                      P1, A
22
                       RO, #BUFFER
23
               MOV
                                   ;access IDATA segment
24
               MOV
                       @RO, A
25
               INC
                       A
26
               INC
                       R0
27
               MOV
                       @RO, A
28
29
               MOV
                       DPTR, #XBUFFER ; access XDATA segment
30
               INC
                       @DPTR, A
31
               MOVX
32
                       DPTR
               INC
33
               CLR
                       A
34
               MOVX
                       @DPTR, A
35
36 END
37
```

#### Keil µVision5汇编语言设计流程 --设置建立参数

■ 在如右图所示的窗口中,选中Target 1文件夹,并单击右键, 出现浮动菜单。在浮动菜单内,选中Options for Target 'Target 1'…选项。

1 单击Output wice   Target Output   Listing		1 Locate   BL51 Misc   Debug   Utilities
Select Folder for Objects	Name of Executable:	top
		i
Greate Exceptable top		
Debug Information	■ Browse Information	
✓ Debug Information	at: HEX-80 ▼	

## Keil µVision5汇编语言设计流程 --设置建立参数

Device   Target   Output   Listing   User	C51   A5	1   BL51 Locate	BL51 Misc Debug   Utilities
Use Simulator  Limit Speed to Real-Time  2 选中Use Simulator	Settings	C Use:	1 单击 <b>Deb⊌g标</b> 签
✓ Load Application at Startup     ✓ Run to r	main()	✓ Load Application at	t Startup Run to main()
Initialization File:	Edit	nitialization File:	Edit

3 单击OK按钮,退出Options for Target 对话框界面

### Keil µVision5汇编语言设计流程 --建立设计

在主界面主菜单下,选择Project->Build target。开始对设计进行建立的过程。

注:该过程对汇编文件,进行汇编和链接,最后生成可执行二进制文件和HEX文件。

## Keil µVision5汇编语言设计流程 --分析.m51文件

在当前设计工程的目录中,找到并用写字板打开top.m51文件。在该文件中,LINK MAP OF MODULE:标题下给出了该设计中,不同段在存储器中的空间分配情况,如下图所示。

LINK MAP OF MODULE: top (MAIN)

TYPE	BASE	LENGTH	RELOCATION	SEGMENT NAME
* * * *	* * *	D A T A	M E M O R Y	* * * * * * *
REG	0000H	0008H	ABSOLUTE	"REG BANK 0"
IDATA	0008H	0013H	UNIT	SEG_2
* * * *	* * *	X D A T A	M E M O R Y	* * * * * * *
XDATA	0000H	0064H	UNIT	SEG_3
* * * * CODE CODE	* * * 0000H 0119H	C O D E 0119H 0004H	M E M O R Y UNIT UNIT	* * * * * * * MYPROG SEG_1

#### Keil µVision5汇编语言设计流程 --分析.m51文件

SYMBOL TABLE OF MODULE: top (MAIN)

■ 继续浏览该文件,在该文件

SYMBOL TABLE OF MODULE

程序代码中,所有变量和端口

在存储器中的空间分配。

■ 关闭该文件。

VALUE	TYPE	NAME
	MODULE	MAIN
C:0119H	SEGMENT	SEG_1
I:0008H	SEGMENT	SEG_2
X:0000H	SEGMENT	SEG_3
C:0000H	SEGMENT	MYPROG
I:0008H	SYMBOL	BUFFER
C:0100H	SYMBOL	MAIN
D:0090H	SYMBOL	P1
C:0119H	SYMBOL	TABLE
X:0000H	SYMBOL	XBUFFER
C:0000H	LINE#	15
C:0100H	LINE#	17
C:0103H	LINE#	18
C:0105H	LINE#	19
C:0106H	LINE#	20
C:0109H	LINE#	21
C:010BH	LINE#	23
C:010DH	LINE#	24
C:010EH	LINE#	25
C:010FH	LINE#	26
C:0110H	LINE#	27
C:0111H	LINE#	29
C:0114H	LINE#	30
C:0115H	LINE#	31
C:0116H	LINE#	32
C:0117H	LINE#	33

#### Keil µVision5汇编语言设计流程 --分析.lst文件

在当前设计工程的目录中,找到并用写字板打开main.lst文件。在该文件中,LINK MAP OF MODULE:标题下给出了该设计中,不同段在存储器中的空间分配情况,如下图所示。

注: (1)在该文件中,给出了每条汇编语言助记符机器指令在Flash程序存储器中的相对位置。

- (2)在该文件中,给出了每条汇编语言助记符的机器指令的机器码。
- 关闭该文件。

#### 程序代码在 STC 单片机程序 Flash 的位置分配↔

#### 汇编语言助记符对应的机器指令≠

LOG	0BJ		LINE	SOURCE				D D
	4		1	NAME main				
			2	seg_1 SEG		Œ		
			3	200_1 220	RSEG	seg_1		
0000	030205EB		4	TABLE:	DB	3, 2, 5, 0x0EE		
				seg_2 SEG		Control of the Contro		
			5 6 7		RSEG	seg_2		
0000			7	BUFFER:	DS	19		
7.77.70			8	seg_3 SEG				
			9		RSEG	seg_3		
0000			10	XBUFFER:	Ds	100		
			11		:T.F.			
			12	myprog SE	GMENT CO	DDE		
			13		RSEG	myprog		
			14		USING			
0000	020000	F	15		LJMP	main		
0100	DATES AND AND THE RESIDENCE OF THE PARTY OF		16		org	0x100		
0100	900000	F	17	main:	MOV	DPTR, #TABLE	access	code segment
0103	7403		18		MOV	A,#3	Tryd, to our tracery	ic and the store Transcribed
0105	93		19		MOVC	A, @A+DPTR		
0106	759000		20		MOV	P1,#0		
0109	F590		21		MOV	P1, A		
	5350000		22			00007000		
010B	7800	F	23		MOA	RO, #BUFFER	access	IDATA segment
010D	F6		24		MOA	@RO, A	Westerner.	
010E	04		25		INC	A		
010F	08		26		INC	RO		
0110	F6		27		MOV	@RO, A		
			28					
0111	900000	F	29		MOV	DPTR, #XBUFFER	;access	XDATA segment
0114	04		30		INC	A	MI	5
0115	FO		31		MOVX	@DPTR, A		

### Keil µVision5汇编语言设计流程 --分析HEX文件

■ 在当前设计工程的目录中,找到并用写字板打开top.hex文件, 如下图所示。

:04011900030205EEEA

:03000000020100FA

:10010000900119740393759000F5907808F604082F

:09011000F690000004F0A3E4F0F5

:0000001FF

#### Keil µVision5汇编语言设计流程 --分析HEX文件

许多Flash编程器都要求输入文件具有Intel HEX格式,一个Intel HEX文件的一行称为一个记录,每个记录都由16进制字符构成,两个字符表示一个字节的值。Intel HEX文件通常由若干记录组成,每个记录具有如下的格式:

: Il aaaa tt dd...dd cc

#### 其中:

□ ":" 表示记录起始的标志。Intel HEX文件的每一行都是以 ":" 开 头。

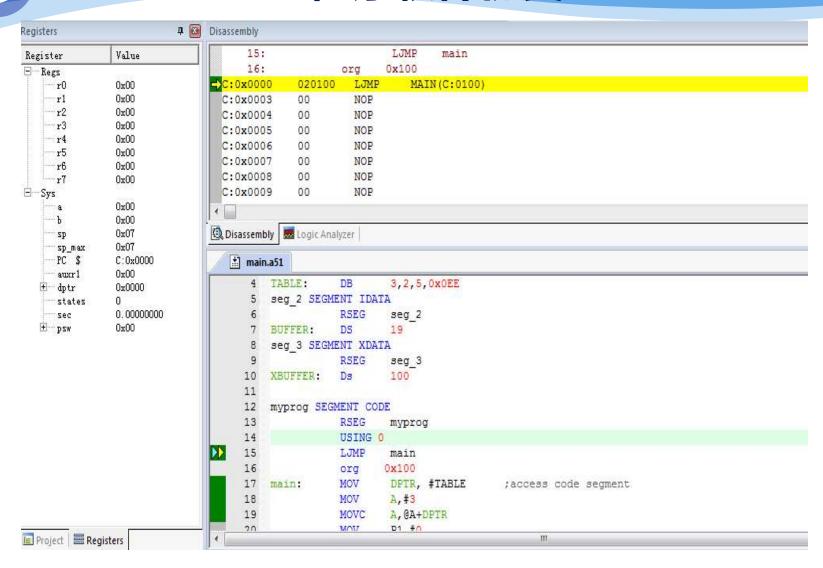
### Keil µVision5汇编语言设计流程 --分析HEX文件

- □ "Ⅱ"表示记录的长度。用来标识该记录的数据字节数。
- □ "aaaa" 装入地址。他是该记录中第一个数据字节的16位地址值,用来表示该记录在EPROM存储器中的绝对地址。
- □ "tt" 记录类型。00表示数据记录,01表示文件结束。
- □ "dddd" 记录的实际字节数据值。每一个记录都由II个字节的数据值。
- □ "cc" 校验和。将它的值与记录中所有字节(包括记录长度字节)内容相加,其结果应该为0,如果为其它数值为表明该记录有错。
- 关闭该文件。

本节将对程序进行软件仿真。软件仿真是指,在Keil µVision5集成开发环境中运行设计程序代码。

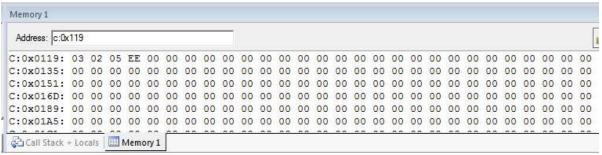
- 这个运行过程不需要真实STC单片机硬件平台。当程序设计者在没有实际的STC单片机开发平台时,可以借助于集成开发环境提供的各种调试工具,初步判断一下所设计的软件代码是否有缺陷,这样就能及时发现程序设计中的问题。
- 程序软件仿真也称为脱机仿真,也就是脱离基于STC单片机的 具体硬件平台的仿真。

- 在Keil μVision主界面主菜单下,选择Debug->Start/Stop Debug Session,进入调试器模式。
- 出现调试器界面。在该调试器左边出现Registers标签窗口。在该界面的上方出现Disassembly窗口,该窗口时程序代码的反汇编程序。在该窗口下方是汇编语言源程序界面。



- 在当前调试界面工具栏内,选择并单击按钮 , 对程序代码进行单步运行, 然后观察寄存器界面内的寄存器内容的变化情况。然后, 再单击按钮 , 再次观察寄存器内容的变化, 一直运行程序直到单步运行到END为止结束。
- 在当前调试模式主界面主菜单下,选择Debug->Reset CPU, 准备重新运行程序。
- 在当前调试模式主界面主菜单下,选择View->Memory Windows->Memory1;或者在当前调试模式主界面工具栏内单击按钮 □ , 出现浮动菜单。在浮动菜单内,选择Memory1。

■ 在当前调试模式主界面右下角出现Memory1界面,如下图所示。 在Address:右侧输入c:0x0119。



在当前调试界面工具栏内,单击按钮 ,对程序代码连续运行单步调试,一直到运行完第21行程序 如下图所示,然后观察图1内的存储器内容的变化情况。

17	main:	MOV	DPTR, #TABLE	;access	code segment
18		MOV	A,#3		
19		MOVC	A, @A+DPTR		
20		MOV	P1,#0		
21		MOV	P1, A		
22					

- 当前调试模式主界面主菜单下,选择View->Memory Windows->Memory2。 或者在当前调试模式主界面工具栏内单击按钮 💷 , 出现浮动菜单。在浮动菜单内,选择Memory2。
- 在当前调试模式主界面右下角出现Memory2界面,如图所示。在Address: 右侧输入d:0x00。

Address: d:0	c00																											
D:0x00:	09	00	00	00	00	00	00	00	EE	EF	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0
D:0x1C:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0
D:0x38:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0
D:0x54:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0
D:0x70:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	FF	07	19	08	00	00	04	00	00	00	00	0
D:0x8C:	00	00	01	00	EE	08	00	00	00	00	00	00	00	00	00	FC	00	00	00	00	FF	00	00	00	00	00	00	0
D:0xA8:	00	00	00	00	00	00	00	00	FF	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0
D:0xC4:	83	00	10	FF	00	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00	40	00	00	00	00	00	00	0
D: 0xE0:	EF	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	0
D.OxEC.	00	nn	nn	00	09	00	00	00	nn	nn	nn	00	ਸਸ	FF	nn	00	nn	00	00	nn	00	00	00	00	nn	00	00	0

在当前调试界面工具栏内,单击按钮 ,对程序代码连续运行单步调试,一直到运行完第28行程序,如下图所示,然后观察上图内的存储器内容的变化情况。

```
23 MOV RO, #BUFFER
24 MOV @RO, A
25 INC A
26 INC RO
27 MOV @RO, A
28
```

- 当前调试模式主界面主菜单下,选择View->Memory Windows->Memory3。或者在当前调试模式主界面工具栏内单击按钮 ,出现浮动菜单。在浮动菜单内,选择Memory3。
- 在当前调试模式主界面右下角出现Memory3界面,如图所示。 在Address:右侧输入x:0x0000。

Address: X:0X00	000																						
X:0x000000:	F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
X:0x000017:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
X:0x00002E:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
X:0x000045:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
X:0x00005C:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
X:0x000073:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
V.0	0.0	രവ	രവ	രവ	0.0	രവ	രവ	0.0	0.0	രവ	00	00	0.0	0.0	0.0	രവ	രാ						

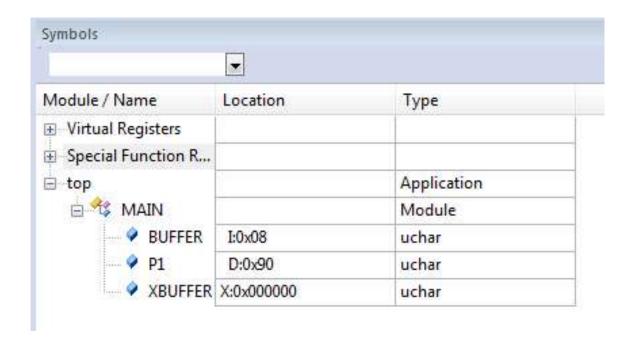
在当前调试界面工具栏内,单击按钮 ,对程序代码连续运行单步调试,一直到运行完第34行程序,如图所示,然后观察上图内的存储器内容的变化情况。

26		INC	RO			
27		MOV	@RO,A			
28						
29		MOV	DPTR, #XBUFFER	;access	XDATA	segment
30		INC	A			
31		MOVX	@DPTR, A			
32		INC	DPTR			
33		CLR	A			
34		MOVX	@DPTR, A			
35						
36	END					
37						

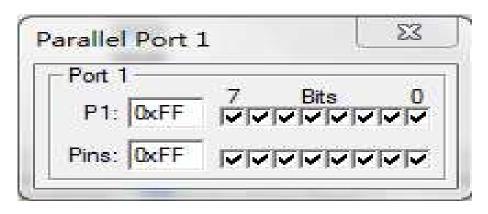
- 在当前调试界面主菜单,选择Debug->Reset CPU,准备重新运行程序。
- 在当前调试界面主菜单下,选择View->Trace->Enable Trace Recording 前面的复选框。然后,再次选择View->Trace->Instruction Trace; 或者在 当前调试主界面工具栏内,单击按钮 □ , 出现浮动菜单。在浮动菜单内,选择Enable Trace Recording前面的复选框,然后,再次单击按钮 □ , 出现浮动菜单。在浮动菜单内,选择Instruction Trace选项。
- 出现指令Instruction Trace (跟踪调试) 窗口界面。
- 再次连续单步运行程序,一直到程序结束为止。在每运行一步单步调试时, 在跟踪调试窗口界面中,可以看到所执行的指令,以及该指令在程序存储器 内所分配的地址,以及该指令的机器码,如后图所示。

ilter:	Execution-All	_	]						
Nr.	Address	Opcode	Instruct	tion					
1	C:0x0000	020100	LJMP	MAIN (C: 01	(00)				
2	C:0x0000		15:		0-000-001-	LJMP	main		
3	C:0x0100	900119	MOV	DPTR, #0x0	119				
4	C:0x0100		17:	main:	MOV	DPTR,	#TABLE	; access	code segment
5	C:0x0103	7403	MOV	A, #0x03					
6	C:0x0103		18:		MOV	A, #3			
7	C:0x0105	93	MOVC	A, @A+DPTE	1				
8	C:0x0105		19:		MOVC	A, @A+I	PTR		
9	C:0x0106	759000	MOV	P1 (0x90),	#0x00				
10	C:0x0106	4000000	20:			MOV	P1,#0		
11	C:0x0109	F590	MOV	P1 (0x90),	A				
12	C:0x0109		21:			MOV	P1, A		
13	C:0x010B	7808	MOV	RO, #0x08					
14	C:0x010B		23:			MOV	RO, #BUFFEE	1	; access IDATA segment
15	C:0x010B	F6	MOV	@RO, A					
16	C:0x010D		24:		MOV	@RO, A			
17	C:0x010E	04	INC	A					
18	C:0x010E		25:		INC	À			
19	C:0x010F	08	INC	RO					
20	C:0x010F		26:		INC	RO			
21	C:0x0110	F6	MOV	@RO, A					
22	C:0x0110		27:		MOV	@RO, A			
23	C:0x0111	900000	MOV	DPTR, #0x0	0000				
24	C:0x0111		29:		MOV	DPTR, #	#XBUFFER	access	XDATA segment
25	C:0x0114	04	INC	A					
26	C:0x0114	200	30:		INC	A			
27	C:0x0115	FO FO	MOVX	@DPTR, A					
28	C:0x0115		31:			MOVX	@DPTR, A		
29	C:0x0116	A3	INC	DPTR					
30	C:0x0116	200	32:		INC	DPTR			
31	C:0x0117	E4	CLR	A					
32	C:0x0117		33:		CLR	A			
33	C:0x0118	FO FO	MOVX	@DPTR, A					
34	C:0x0118	400	34:		MOVX	@DPTR,	A		
35	C:0x0119	03	RR	A					
36	C:0x011A	0205 <b>EE</b>	LJMP	C:05EE					

- 在当前调试模式主界面主菜单下,选择View->Symbols Window;或者在当前调试模式主界面工具栏内,单击按钮 🗓 。
- 在调试界面右侧,出现Symbols窗口界面,如下图所示。在该界面下,给出SFR寄存器的地址,以及程序代码中所用到变量所分配的段以及地址信息。



- 在当前调试界面主菜单,选择Debug->Reset CPU,准备重新运行程序。
- 在当前调试界面主菜单下,选择Peripherals->I/O-Ports->Port 1。
- 弹出Parallel Port 1端口界面,如图所示。在该界面中,给出了端口1各个引脚当前的状态。
- 再次单步运行程序,一直运行到程序代码的第23行为止。



- 在当前调试界面主菜单,选择Debug->Reset CPU,准备重新运行程序。
- 在当前调试主界面主菜单下,选择View->Analysis Windows->Code Coverage选项;或者在当前调试主界面工具栏内,单击按钮 🐷 、出现浮动菜单。在浮动菜单内,选择Code Coverage选项。
- 在调试主界面内,出现Code Coverage窗口界面,如图所示。
- 在Code Coverage窗口界面中Module:右侧的下拉框中,选择 MAIN。
- 运行单步调试,可以看到代码覆盖率随程序的单步执行而不断的增加。

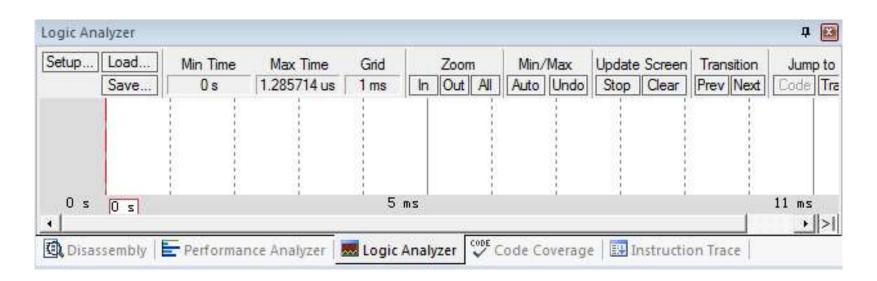
■ 在当前调试界面主菜单下,选择Debug->Start/Stop Debug Session,退出调试模式主界面。

注:该步操作的目的是为了清除前面所做操作,对存储空间内容的影响。因此, 下面的步骤会调用软件逻辑分析工具,调试信号的逻辑状态变化。

#### 对该段代码使用逻辑分析仪进行分析的步骤主要包括:

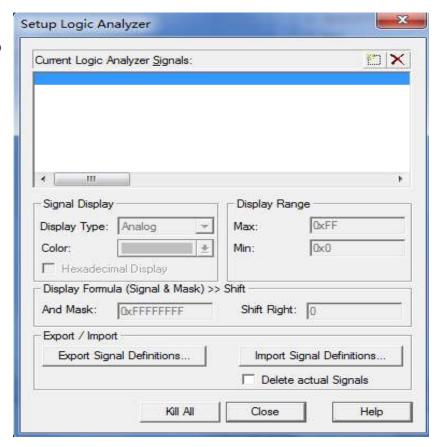
- 在Keil μVision主界面主菜单下,选择Debug->Start/Stop
   Debug Session,进入调试模式主界面。
- 在当前调试模式主界面主菜单下,选择View->Analysis Windows->Logic Analysis; 或者在当前调试主界面工具栏中,单击按钮 ,出现浮动菜单。在浮动菜单内,选择Logic Analyzer。

■ 出现Logic Analyszer (逻辑分析仪) 窗口界面,如下图所示。



- 在该窗口界面中,单击Setup...按钮。
- 出现Setup Logic Analyzer界面,如下图所示。在该界面内,

单击 🍱 按钮。



- 在Current Logic Analyzer Signals:窗口下新添加了一个空白行。在该空白行中输入'P1'。然后,选中该行。
- 在该窗口下面的And Mask: 右侧的文本框中输入0xFFFFFFFF。 其余参数保持不变,如下图所示。



■ 单击Close按钮,退出配置逻辑分析仪选项界面。

■ 在当前调试模式主界面下面的Command窗口内的命令行中, 输入la buffer命令,如下图所示。

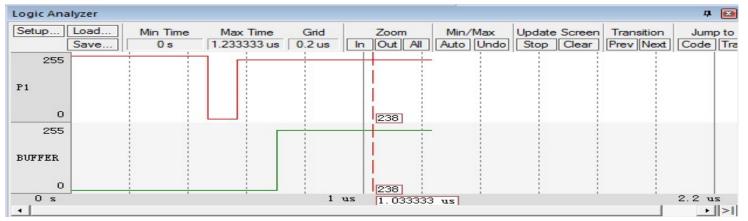
Command	T 🔀
	A
	_
4	h.
>la buffer	
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList Break	akSet

■ 按回车键,可以看到在在Logic Analyzer窗口界面内,新添加了P1和buffer两个逻辑信号,如图所示。

Setup	Load	Min Time	Max Time	Grid		Zoom		Min/	Max	Update	Screen	Trans	sition	Jum	p to
	Save	0 s	0.285714 us	0.2 us	In	Out	All	Auto	Undo	Stop	Clear	Prev	Next	Code	Tra
255	1			3											
P1															
0	0														
255	Part Co						-						1		
buffer							1		1				1		
0	238		1	1					1						
0 s	0 s	to to	74.	0.	97142	29 us			W.	- 1	,		2.1	71429	us
4	150 001														>

■ 单步运行程序,一直运行到程序代码的第29行。

- 在Logic Analyzer窗口界面中,连续单击按钮多次 [m],用于放大窗口内的信号。
- 当调整到观察范围内时,看到信号的变化过程,如图所示。



■ 在当前调试主界面主菜单下,选择Debug->Start/Stop Debug Session,退出调试模式主界面。

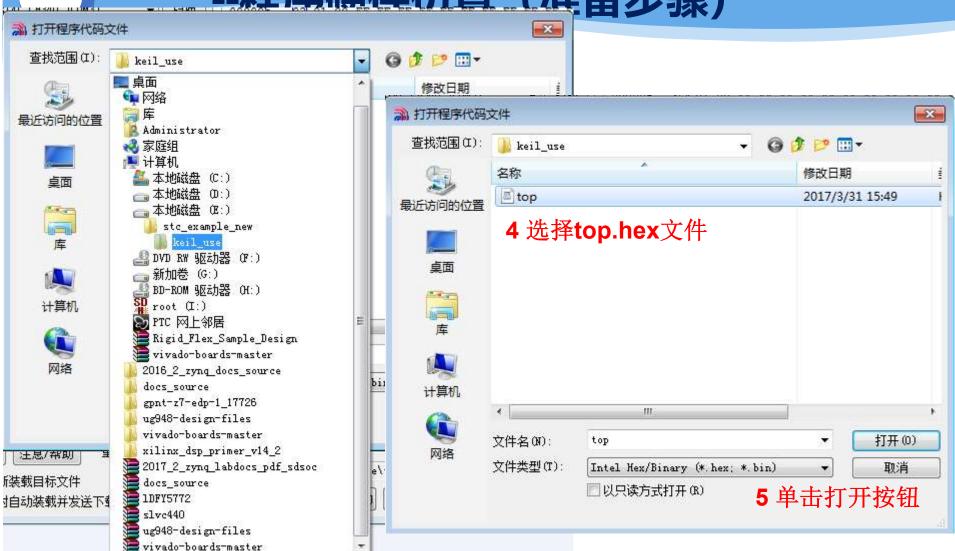
- 打开STC学习板,在该开发板左侧找到标识为CON5 mini USB接口。将STC提供USB数据电缆的两端,分别和开发板上标识为CON5 的USB插座和PC机/笔记本电脑上的USB插座进行连接。
- 打开本书所提供资料下的STC-ISP软件(操作步骤如下)。

单击打开程序文件按钮

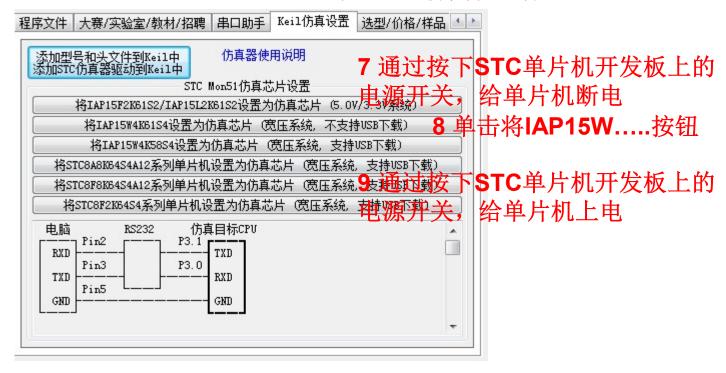
輸入用户程序运行时的IRC频率 11.0582 ▼ MHz ▼ 振荡器放大增益(12M以上建议选择)	E
輸入用户程序运行时的IRC频率 11.0582 ▼ MHz ▼ 振荡器放大增益(12M以上建议选择)	<b>=</b>
☑ 选择使用内部IRC时钟(不选为外部时钟)	_
硬件选项 脱机下载/U8/U7 程序加密后传输 ID4	7,000
<ul><li>記始地址</li><li>bx0000 ▼ 清除代码缓冲区 打开程序文件</li><li>bx0000 ▼ 清除EEPROM缓冲区 はTHEEPROM文</li></ul>	一〇千
贵低波特率 2400 ▼ 最高波特率 18280 □	号号

# Keil µVision5汇编语言设计流程

--程序碩件仿直 (准备步骤)



#### 6 单击Keil仿真设置标签

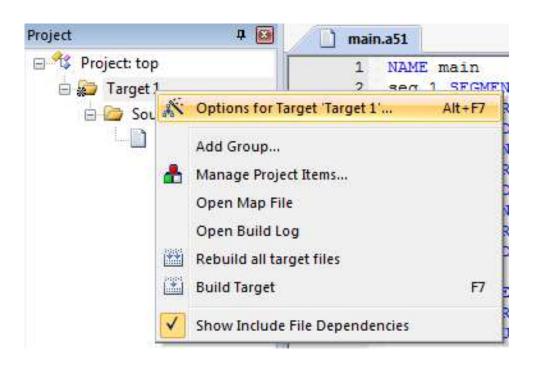


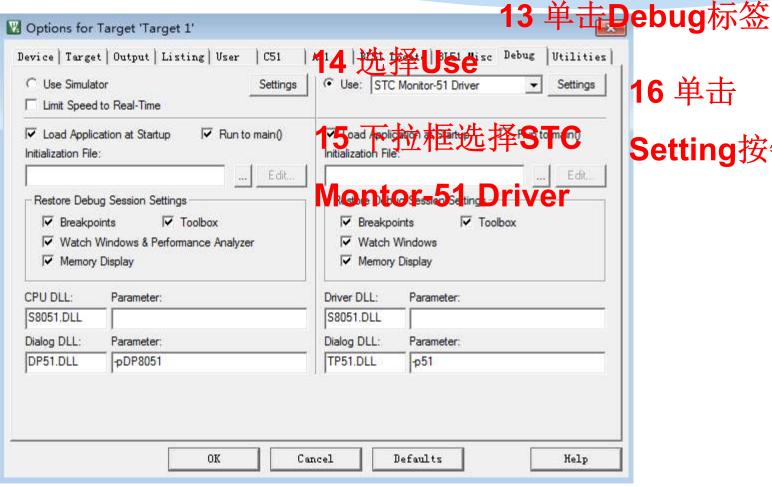
10 在STC-ISP软件右下方窗口中,出现下载代码的相关信息。

```
开始调节频率 ... [0.889*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*] [0.125*]
```

11 定位到\stc\_example\keil\use目录下,打开top.uvproj工程

### 12 准备打开Options for Target对话框界面





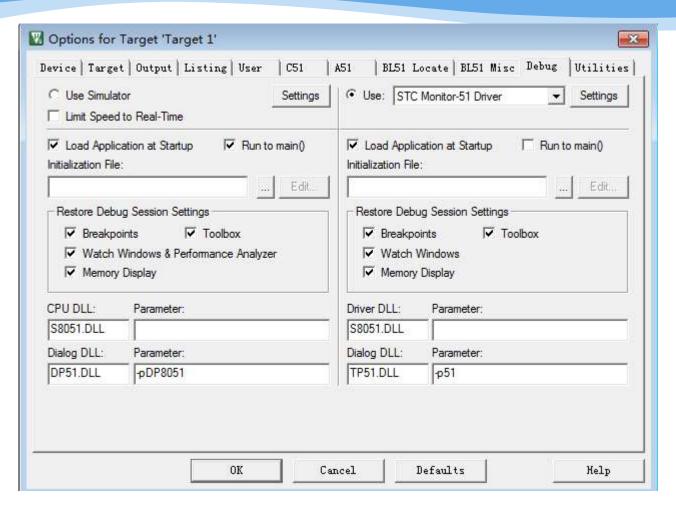
16 单击

Setting按钮



17 正确设置Com Port,与STC-ISP中的串口号相同

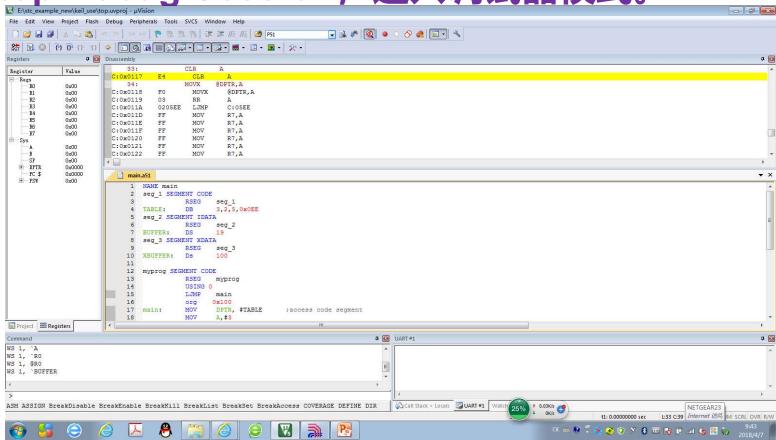
18 单击OK按钮,退出Target Setup



19 单击OK按钮,退出Options for Target界面

在Keil µVision主界面主菜单下,选择Debug->Start

/Stop Debug Session, 进入调试器模式。



■ 运行单步调试, 先单步运行完第20行代码。即:

### **MOV P1,#0**

15		LJMP	main
16		org	0x100
17	main:	MOV	DPTR, #TABLE
18		MOV	A,#3
19		MOVC	A, @A+DPTR
20		MOV	P1,#0
21		MOV	P1, A
22			

注:该指令是使得P1端口置零,其中就包括让P1.6和P1.7端口置零。因此,看到LED6和LED7灯亮(STC官方开发板)。

■ 运行单步调试, 先单步运行完第21行代码。即:

MOV P1, A

注:该指令将累加器A的内容送给P1端口。由于此时累加器A的内容是0xEE,也就是P1.7和P1.6置1。因此,看到LED6和LED7灯灭。

思考与练习:在执行指令的时候,观察寄存器的变化情况、存储器内容和端口1的变化情况。

- 在调试主界面下,选择Debug->Reset CPU选项,准备重新执行程序。
- 在程序代码行号前单击左键,分别在第20行、第24行和第30行 添加断点,如图所示。

■ 在程序代码行号前单击左键,分别在第20行、第24行和第30行 添加断点,如图所示。

```
myprog SEGMENT CODE
                RSEG
                        myprog
14
                USING 0
15
                LJMP
                        main
16
                       0x100
                org
17 main:
                MOV
                       DPTR, #TABLE
                                           ;access code segment
18
                       A, #3
                MOV
19
                MOVC
                       A, @A+DPTR
                        P1,#0
20
                MOV
21
                MOV
                        P1, A
22
23
                        RO, #BUFFER
                MOV
                                           ;access IDATA segment
24
                        @RO.A
                MOV
25
                INC
                        A
26
                INC
                        RO
27
                        @RO.A
                MOV
28
29
                MOV
                        DPTR, #XBUFFER
                                           ;access XDATA segment
30
                INC
31
                MOVX
                        @DPTR, A
32
                         DPTR
                INC
33
                CLR
                         A
34
                MOVX
                        @DPTR, A
35
36 END
```

- 在当前调试模式主界面主菜单下,选择Debug->Run;或者直接按键F5,运行断点调试功能。
- 在当前调试模式主界面主菜单下,选择Debug->Start/Stop Debug Session,退出调试器模式。
- 在Keil µVision主界面主菜单下,选择Project->Close Project, 退出当前工程。