Using the ST10 MAC Unit (on the ST10-272) with Keil C166 Development Tools

Application Note 140

July 26, 1999, Munich, Germany

by Reinhard Keil, Keil Elektronik GmbH rk@keil.com ++49 89 456040-13

The Keil C166 ***Version 4.03 or higher*** fully supports the ST10 MAC unit that is available in several ST10 derivatives. The MAC unit is a DSP co-processor that is integrated into the ST10 device and boosts the performance of multiplication, mul/add combinations and memory moves. The MAC unit is used by the software tools as follows:

- **C166 Compiler:** the MAC directive enables the usage of the CoMOV, CoMUL, CoMAC, CoMAC- and CoMACR instruction. With specific intrinsic functions it is possible to write C macros that are using almost all MAC instructions.
- **A166 Macro Assembler:** the EXTMAC directive enables all MAC instructions
- **µVision2 Debugger/Simulator:** choosing a device with MAC unit enables the debugging for ST10 MAC instructions; all MAC instructions are fully simulated including timing and MSW flags.
- **Monitor-166:** supports fully the ST10 MAC instruction set; no special version or directives required.

This application note describes how to use the Keil C16x/ST10 development tools and the L166 Linker/Locater for flash programming.

C166 Compiler

The following MAC extensions are implemented:

- **MAC** directive: C166 uses CoMUL/CoMAC/CoMACR/CoMAC- instead of MUL; CoMOV for word-aligned struct copy in near memory.
- **ST10MAC.H** header file: defines intrinsic functions that allow you to use virtually all MAC instructions.

MAC Directive

Description: ST10 MAC instructions replace MUL and MUL/ADD combinations; CoMOV is used instead of the struct copy instrinis function. Interrupts functions save the required MAC registers. The MAC directive gives detailed control to the usage of the ST10 MAC unit and can be used selectively within the source code. Therefore the MAC directive has the following sub-options

Directive	Description
MAC	Enables complete support for the ST10 MAC instructions.
MAC (MUL)	Enable CoMUL, CoMAC, CoMACR and CoMAC- instructions.
MAC (MOV)	Enable CoMOV instruction.
MAC (INTR)	Enable save/restore of MAC registers within interrupt functions.
MAC ()	Disable the usage of MAC instructions.

The sub-options of the MAC directive can be combined.
Example: MAC (MOV, INTR) will enable the usage of CoMOV and the save/restore of MAC registers in interrupt functions.

Example:

```
C166 SAMPLE.C MAC

#pragma MAC (MUL, MOV)
```

The following example shows the generated code using the MAC directive:

```
stmt lvl     source

    1         #pragma MAC
    2
    3                 int  i1, i2, i3, i4, c1;
    4         unsigned int  u1, u2, u3;
    5         unsigned long l1, l2, l3;
    6
    7         struct h {
    8           int a[10];
    9           int b[10];
   10         } h1, h2;
   11
   12
   13         long mac_func (void)  {
   14  1        long l;
   15  1        l  = ((long) u1 * 2230) - ((long) u2 * 0x4442);
   16  1        return (((long)i1*c1) + ((long)i2*c1) - ((long)i3*c1) + ((long)i4*c1) + l);
   17  1      }
   18
   19
   20         void intr (void) interrupt 20  {
   21  1        h1 = h2;
   22  1      }
```

```
ASSEMBLY LISTING OF GENERATED OBJECT CODE

            ; FUNCTION mac_func (BEGIN  RMASK = @0x47F0)
                                          ; SOURCE LINE # 13
                                          ; SOURCE LINE # 15
0000 F2F50200 R    MOV       R5,u2
0004 E6F44244      MOV       R4,#04442H
0008 A3540000      CoMULu    R5,R4
000C F2F70400 R    MOV       R7,u1
0010 E6F6B608      MOV       R6,#08B6H
0014 A3763000      CoMACRu   R7,R6
0018 C3550800      CoSTORE   R5,MAH
001C C3442000      CoSTORE   R4,MAL
0020 F084          MOV       R8,R4
0022 F095          MOV       R9,R5
;---- Variable 'l' assigned to Register 'R8/R9' ----
                                          ; SOURCE LINE # 16
0024 F2FA6A00 R    MOV       R10,c1
0028 F2F41600 R    MOV       R4,i2
002C A34AC000      CoMUL     R4,R10
0030 F2F61800 R    MOV       R6,i1
0034 A36AD000      CoMAC     R6,R10
0038 F2F41400 R    MOV       R4,i3
003C A34AE000      CoMAC-    R4,R10
0040 F2F40E00 R    MOV       R4,i4
0044 A34AD000      CoMAC     R4,R10
0048 C3550800      CoSTORE   R5,MAH
004C C3442000      CoSTORE   R4,MAL
0050 0048          ADD       R4,R8
0052 1059          ADDC      R5,R9
                                          ; SOURCE LINE # 17
0054 CB00          RET
            ; FUNCTION mac_func (END    RMASK = @0x47F0)

            ; FUNCTION intr (BEGIN  RMASK = @0x4050)
                                          ; SOURCE LINE # 20
0056 ECEF          PUSH      MSW
0058 EC2F          PUSH      MAH
005A EC2E          PUSH      MAL
005C C6ED0000      SCXT      MRW,#00H
0060 EC84          PUSH      IDX0
0062 C6030300      SCXT      DPP3,#03H
0066 ECF4          PUSH      R4
0068 ECF6          PUSH      R6
                                          ; SOURCE LINE # 21
006A E6F64200 R    MOV       R6,#h1
006E E6F41A00 R    MOV       R4,#h2
0072 F6F408FF      MOV       IDX0,R4
0076 D326009A   Repeat #20 times CoMOV [IDX0+],[R6+]
                                          ; SOURCE LINE # 22
007A FCF6          POP       R6
007C FCF4          POP       R4
007E FC03          POP       DPP3
0080 FC84          POP       IDX0
0082 FCED          POP       MRW
0084 FC2E          POP       MAL
0086 FC2F          POP       MAH
0088 FCEF          POP       MSW
008A FB88          RETI
            ; FUNCTION intr (END    RMASK = @0x4050)
```

ST10MAC.H Header File

The C166 Compiler supports several intrinsic functions that allow you to generate virtually all MAC instructions. These intrinsic functions are defined in **ST10MAC.H**. The following table provides a brief overview of the available intrinsic functions.

Routine	Attributes	Generates ST10 instructions
mac	intrinsic, reentrant	with the form Co*XXX* Rn, Rm or Co*XXX* Rn, [Rm]. The function accepts two int arguments.
macl	intrinsic, reentrant	with the form Co*XXX* Rn, Rm or Co*XXX* Rn, [Rm]. The function accepts one long argument.
macv	intrinsic, reentrant	with the form Co*XXX* with no further arguments; used to generate CoABS and CoRND.
maci	intrinsic, reentrant	with the form Co*XXX* [IDX@], [Rm].
lmac	intrinsic, reentrant	returns MAH, MAL with CoSTORE.
_lmac_sat	intrinsic, reentrant	returns MAS, MAL with CoSTORE.
_imac_sat	intrinsic, reentrant	returns MAS with CoSTORE.
mmov	intrinsic, reentrant	parameters indentical to memcpy, but uses CoMOV and requires word-aligned objects.

The following example shows the usage of the ST10MAC header file:

```
stmt lvl     source

    1
    2         #include <st10mac.h>
    3         #include <reg272.h>
    4
    5         unsigned int u1;
    6
    7         long mac_ifunc (long l1)  {
    8  1        _macl_ (CoLOAD, l1);
    9  1        _mac_  (CoMACu, u1, 0x1234);
   10  1        return (_lmac_ ());
   11  1      }
   12
   13
   14         int idata arr[5];
   15
   16         long mac_ifunc2 (unsigned int idata *p)  {
   17  1        IDX0 = (unsigned int) arr;
   18  1        _maci_ (CoMUL,  _IDX0p_, p);
   19  1        _maci_ (CoMACu, _IDX0p_, p);
   20  1        return (_lmac_ ());
   21  1      }


ASSEMBLY LISTING OF GENERATED OBJECT CODE


             ; FUNCTION mac_ifunc (BEGIN  RMASK = @0x4030)
                                           ; SOURCE LINE # 7
;---- Variable 'l1' assigned to Register 'R8/R9' ----
                                           ; SOURCE LINE # 8
0000 A3892200      CoLOAD    R8,R9
                                           ; SOURCE LINE # 9
0004 F2F50000 R    MOV       R5,u1
0008 E6F43412      MOV       R4,#01234H
000C A3541000      CoMACu    R5,R4
                                           ; SOURCE LINE # 10
0010 C3442000      CoSTORE   R4,MAL
0014 C3550800      CoSTORE   R5,MAH
                                           ; SOURCE LINE # 11
0018 CB00          RET
             ; FUNCTION mac_ifunc (END    RMASK = @0x4030)
```

```
            ; FUNCTION mac_ifunc2 (BEGIN  RMASK = @0x4010)
                                          ; SOURCE LINE # 16
;---- Variable 'p' assigned to Register 'R8' ----
                                          ; SOURCE LINE # 17
001A E6F40000 R     MOV       R4,#arr
001E F6F408FF       MOV       IDX0,R4
                                          ; SOURCE LINE # 18
0022 9328C001       CoMUL     [IDX0+],[R8]
                                          ; SOURCE LINE # 19
0026 93281001       CoMACu    [IDX0+],[R8]
                                          ; SOURCE LINE # 20
002A C3442000       CoSTORE   R4,MAL
002E C3550800       CoSTORE   R5,MAH
                                          ; SOURCE LINE # 21
0032 CB00           RET
            ; FUNCTION mac_ifunc2 (END    RMASK = @0x4010)
```

FIX272 Directive

Description: The FIX272 directive allows you to generate save code for the ST10R272L device. It bypasses the chip problems Kfm_BR04 and Kfm_BR05 described in the Errata Sheet dated 15. Sep. 1998

Bypassing 272 Problem BR06

To bypass the Kfm_BR06 problem of the chip (JMPS/PEC) broken use the following work-around:

- Do not enable Optimize (7)
- Change the instruction "JMP FAR main" at the end of the START167.A66 file to "CALL FAR main"