gcc inline assembly的format （ on ARM platform ）

```
asm(
    code
    : 输出运算符列表
    : 输入运算符列表
     : 被修改register列表
);
```

| 操作符 | 含义 |
| --- | --- |
| r | r0-r15 |
| m | memory address |
| I | 立即数 |
| X | 被修饰的操作符只能作为输出 |

| 修饰符 | 含义 |
| --- | --- |
| 无 | readony |
| = | writeonly |
| + | read-write |
| & | output only |
| | |

---

```
walterzh$ cat test.c
#include <stdio.h>

int main(int argc, char *argv[])
{
    unsigned int lr_register = 100;
    unsigned int pc_register = 200;

    int ret_val = 5;

    asm (
        "mov %0, lr\n"
        "mov %1, pc\n"
        : "=r" (lr_register), "=r" (pc_register)
        :
        :
    );

    printf("%08x %08x\n", lr_register, pc_register);  ①

    asm (
        "mov r0, %0\n"
        :
        : "r" (ret_val)
        :                                   ②
```

```
        );


//      return ret_val;

}


root@granite2:~# ./test

469396fc 000083ae

root@granite2:~# echo $?

5
```

①

输出lr and pc register。

②

由于“被修改register列表”为空（实际上该指令修改了r0），所以gcc没有意识到r0被修改了。而r0又作为main()的返回值return了。

```
walterzh$ arm-linux-gnueabi-objdump -d -S test


0000838c <main>:

#include <stdio.h>


int main(int argc, char *argv[])

{

   838c:       b5b0            push    {r4, r5, r7, lr}
```

```
838e:       b086            sub     sp, #24
8390:       af00            add     r7, sp, #0
8392:       6078            str     r0, [r7, #4]
8394:       6039            str     r1, [r7, #0]
   unsigned int lr_register = 100;
8396:       f04f 0364       mov.w   r3, #100        ; 0x64
839a:       60fb            str     r3, [r7, #12]
   unsigned int pc_register = 200;
839c:       f04f 03c8       mov.w   r3, #200        ; 0xc8
83a0:       613b            str     r3, [r7, #16]

   int ret_val = 5;
83a2:       f04f 0305       mov.w   r3, #5
83a6:       617b            str     r3, [r7, #20]

   asm (
83a8:       4675            mov     r5, lr
83aa:       467c            mov     r4, pc
83ac:       60fd            str     r5, [r7, #12]
83ae:       613c            str     r4, [r7, #16]
          : "=r" (lr_register), "=r" (pc_register)
          :
          :
   );
```

```
    printf("%08x %08x\n", lr_register, pc_register);
   83b0:    f248 4324     movw    r3, #33828    ; 0x8424
   83b4:    f2c0 0300     movt    r3, #0
   83b8:    4618          mov     r0, r3
   83ba:    68f9          ldr     r1, [r7, #12]
   83bc:    693a          ldr     r2, [r7, #16]
   83be:    f7ff ef90     blx     82e0 <_init+0x20>

    asm (
   83c2:    697b          ldr     r3, [r7, #20]
   83c4:    4618          mov     r0, r3
            : "r" (ret_val)
            :
    );

//  return ret_val;
}
   83c6:    4618          mov     r0, r3
   83c8:    f107 0718     add.w   r7, r7, #24
   83cc:    46bd          mov     sp, r7
   83ce:    bdb0          pop     {r4, r5, r7, pc}
```

---

```
   83aa:    467c          mov     r4, pc
   83ac:    60fd          str     r5, [r7, #12]
```

83ae:	613c	str	r4, [r7, #16]

从上可看出pc比当前executing的instruction addess超前 2 instructions。

当前正在执行address为88ae的instruction，但pc值确是82ae。这里由于是Thumb指令，所以是超前4.

为什么生成的是混合指令???

注：

walterzh$ arm-linux-gnueabi-gcc -fverbose-asm -g -o test_arm test.c

生成的是Thumb instruction的test_arm.

Ubuntu/Linaro 4.6.3-1ubuntu5

walterzh$ arm-linux-gnueabi-gcc -fverbose-asm -g -o test_arm -march=armv7 test.c

生成的是ARM instruction的test_arm (ARMv7)

000083bc <main>:

#include <stdio.h>

#include <unistd.h>

int main(int argc, char *argv[])

{

83bc:	b5b0	push	{r4, r5, r7, lr}

```
83be:       b086            sub     sp, #24
83c0:       af00            add     r7, sp, #0
83c2:       6078            str     r0, [r7, #4]
83c4:       6039            str     r1, [r7, #0]
    unsigned int lr_register = 100;
83c6:       f04f 0364       mov.w   r3, #100        ; 0x64
83ca:       60fb            str     r3, [r7, #12]
    unsigned int pc_register = 200;
83cc:       f04f 03c8       mov.w   r3, #200        ; 0xc8
83d0:       613b            str     r3, [r7, #16]


    int ret_val = 5;
83d2:       f04f 0305       mov.w   r3, #5
83d6:       617b            str     r3, [r7, #20]


    asm (
83d8:       4675            mov     r5, lr
83da:       467c            mov     r4, pc
83dc:       60fd            str     r5, [r7, #12]
83de:       613c            str     r4, [r7, #16]
            : "=r" (lr_register), "=r" (pc_register)
            :
            :
    );
```

```
    printf("%08x %08x\n", lr_register, pc_register);
83e0:       f248 4354       movw    r3, #33876      ; 0x8454
83e4:       f2c0 0300       movt    r3, #0
83e8:       4618            mov     r0, r3
83ea:       68f9            ldr     r1, [r7, #12]
83ec:       693a            ldr     r2, [r7, #16]
83ee:       f7ff ef8a       blx     8304 <_init+0x20>

    asm (
83f2:       697b            ldr     r3, [r7, #20]
83f4:       4618            mov     r0, r3
            :
    );

    while(1)
    {
            sleep(300);
83f6:       f44f 7096       mov.w   r0, #300        ; 0x12c
83fa:       f7ff ef8a       blx     8310 <_init+0x2c>
    }
83fe:       e7fa            b.n     83f6 <main+0x3a>
```

生成的code竟然是ARM与Thumb混合的???

---

使用Marvell toolchain生成elf

/opt/armv7-marvell-linux-gnueabi-hard-4.6.4_x86_64_20140402/bin/arm-marvell-linux-gnueabi-gcc -fverbose-asm -g -o test_arm test.c

0000848c <main>:

```
#include <stdio.h>
#include <unistd.h>

int main(int argc, char *argv[])
{
    848c:       e92d4830        push    {r4, r5, fp, lr}
    8490:       e28db00c        add     fp, sp, #12
    8494:       e24dd018        sub     sp, sp, #24
    8498:       e50b0020        str     r0, [fp, #-32]
    849c:       e50b1024        str     r1, [fp, #-36]  ; 0x24
        unsigned int lr_register = 100;
    84a0:       e3a03064        mov     r3, #100        ; 0x64
    84a4:       e50b3010        str     r3, [fp, #-16]
        unsigned int pc_register = 200;
    84a8:       e3a030c8        mov     r3, #200        ; 0xc8
    84ac:       e50b3014        str     r3, [fp, #-20]

        int ret_val = 5;
    84b0:       e3a03005        mov     r3, #5
    84b4:       e50b3018        str     r3, [fp, #-24]
```

```
    asm (
84b8:       e1a0500e        mov     r5, lr
84bc:       e1a0400f        mov     r4, pc
84c0:       e50b5010        str     r5, [fp, #-16]
84c4:       e50b4014        str     r4, [fp, #-20]
            : "=r" (lr_register), "=r" (pc_register)
            :
            :
    );

    printf("%08x %08x\n", lr_register, pc_register);
84c8:       e59f3020        ldr     r3, [pc, #32]   ; 84f0 <main+0x64>
84cc:       e1a00003        mov     r0, r3
84d0:       e51b1010        ldr     r1, [fp, #-16]
84d4:       e51b2014        ldr     r2, [fp, #-20]
84d8:       ebffffaf        bl      839c <_init+0x24>

    asm (
84dc:       e51b3018        ldr     r3, [fp, #-24]
84e0:       e1a00003        mov     r0, r3
            :
    );

    while(1)
```

```
    {
            sleep(300);
   84e4:    e3a00f4b    mov    r0, #300    ; 0x12c
   84e8:    ebffffae    bl    83a8 <_init+0x30>
    }
   84ec:    eafffffc    b    84e4 <main+0x58>
   84f0:    00008570    .word    0x00008570
```

生成的code完全是ARM instruction。

---

使用yocto build toolchain生成elf

walterzh$ /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-gnueabi/arm-poky-linux-gnueabi-gcc -fverbose-asm -g -o test_arm test.c

walterzh$ /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-gnueabi/arm-poky-linux-gnueabi-objdump -dS hello

```
000083ec <main>:
   83ec:    e52db004    push    {fp}        ; (str fp, [sp, #-4]!)
   83f0:    e28db000    add    fp, sp, #0
   83f4:    e3a03000    mov    r3, #0
   83f8:    e1a00003    mov    r0, r3
   83fc:    e24bd000    sub    sp, fp, #0
```

```
    8400:       e49db004        pop     {fp}            ; (ldr fp, [sp], #4)
    8404:       e12fff1e        bx      lr
```

生成的code完全是ARM instruction。

---

没有inline arm 汇编码

```
walterzh$ cat hello.c
int main()
{
        return 0;
}
```

① 用arm-linux-gnueabi-gcc编译

```
$ arm-linux-gnueabi-gcc -o hello hello.c
```

生成ARM与Thumb混合instruction.

```
00008354 <main>:
    8354:       b480            push    {r7}
    8356:       af00            add     r7, sp, #0
    8358:       f04f 0300       mov.w   r3, #0
    835c:       4618            mov     r0, r3
    835e:       46bd            mov     sp, r7
```

8360: bc80 pop {r7}

8362: 4770 bx lr

② 使用Marvell提供的toolchain

walterzh$ /opt/armv7-marvell-linux-gnueabi-hard-4.6.4_x86_64_20140402/bin/arm-marvell-linux-gnueabi-gcc -o hello hello.c

0000842c <main>:

842c: e52db004 push {fp} ; (str fp, [sp, #-4]!)

8430: e28db000 add fp, sp, #0

8434: e3a03000 mov r3, #0

8438: e1a00003 mov r0, r3

843c: e28bd000 add sp, fp, #0

8440: e8bd0800 ldmfd sp!, {fp}

8444: e12fff1e bx lr

生成的是完全的ARM指令

③ 使用yocto build出来的toolchain

/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-gnueabi/arm-poky-linux-gnueabi-gcc -o hello hello.c

000083ec <main>:

83ec: e52db004 push {fp} ; (str fp, [sp, #-4]!)

83f0: e28db000 add fp, sp, #0

83f4: e3a03000 mov r3, #0

```
83f8:	e1a00003	mov	r0, r3
83fc:	e24bd000	sub	sp, fp, #0
8400:	e49db004	pop	{fp}		; (ldr fp, [sp], #4)
8404:	e12fff1e	bx	lr
```

生成的是完全的ARM指令

唯一的区别是使用了不同的toolchain。

---

walterzh$ arm-linux-gnueabi-gcc -fverbose-asm -v -g -o test_arm test.c

Using built-in specs.

COLLECT_GCC=arm-linux-gnueabi-gcc

COLLECT_LTO_WRAPPER=/usr/lib/gcc/arm-linux-gnueabi/4.6/lto-wrapper

Target: arm-linux-gnueabi

Configured with: ../src/configure -v --with-pkgversion='Ubuntu/Linaro 4.6.3-1ubuntu5' --with-bugurl=file:///usr/share/doc/gcc-4.6/README.Bugs --enable-languages=c,c++,fortran,objc,obj-c++ --prefix=/usr --program-suffix=-4.6 --enable-shared --enable-linker-build-id --with-system-zlib --libexecdir=/usr/lib --without-included-gettext --enable-threads=posix --with-gxx-include-dir=/usr/arm-linux-gnueabi/include/c++/4.6.3 --libdir=/usr/lib --enable-nls --enable-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-time=yes --enable-gnu-unique-object --enable-plugin --enable-objc-gc --enable-multilib --disable-sjlj-exceptions --with-arch=armv7-a --with-float=softfp --with-fpu=vfpv3-d16 --with-mode=thumb --disable-werror --enable-checking=release --build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=arm-linux-gnueabi --program-prefix=arm-linux-gnueabi- --includedir=/usr/arm-linux-gnueabi/include --with-headers=/usr/arm-linux-gnueabi/include --with-libs=/usr/arm-linux-gnueabi/lib

Thread model: posix

gcc version 4.6.3 (Ubuntu/Linaro 4.6.3-1ubuntu5)

**COLLECT_GCC_OPTIONS='-fverbose-asm' '-v' '-g' '-o' 'test_arm' '-march=armv7-a' '-mfloat-abi=softfp' '-mfpu=vfpv3-d16' '-mthumb'**

**/usr/lib/gcc/arm-linux-gnueabi/4.6/cc1 -quiet -v -imultilib . -imultiarch arm-linux-gnueabi test.c -quiet -dumpbase test.c -march=armv7-a -mfloat-abi=softfp -mfpu=vfpv3-d16 -mthumb -auxbase test -g -version -fverbose-asm -fstack-protector -o /tmp/ccngWr0p.s**

GNU C (Ubuntu/Linaro 4.6.3-1ubuntu5) version 4.6.3 (arm-linux-gnueabi)

compiled by GNU C version 4.6.3, GMP version 5.0.2, MPFR version 3.1.0-p3, MPC version 0.9

GGC heuristics: --param ggc-min-expand=100 --param ggc-min-heapsize=131072

ignoring duplicate directory "/usr/lib/gcc/arm-linux-gnueabi/4.6/../../../../arm-linux-gnueabi/include"

ignoring nonexistent directory "/usr/include/arm-linux-gnueabi"

#include "..." search starts here:

#include <...> search starts here:

/usr/lib/gcc/arm-linux-gnueabi/4.6/include

/usr/lib/gcc/arm-linux-gnueabi/4.6/include-fixed

/usr/arm-linux-gnueabi/include

/usr/include

End of search list.

GNU C (Ubuntu/Linaro 4.6.3-1ubuntu5) version 4.6.3 (arm-linux-gnueabi)

compiled by GNU C version 4.6.3, GMP version 5.0.2, MPFR version 3.1.0-p3, MPC version 0.9

GGC heuristics: --param ggc-min-expand=100 --param ggc-min-heapsize=131072

Compiler executable checksum: c2a3eee802c1f4bc82a4d015bd8c8d6f

**COLLECT_GCC_OPTIONS='-fverbose-asm' '-v' '-g' '-o' 'test_arm' '-march=armv7-a' '-mfloat-abi=softfp' '-mfpu=vfpv3-d16' '-mthumb'**

**/usr/lib/gcc/arm-linux-gnueabi/4.6/../../../../arm-linux-gnueabi/bin/as -march=armv7-a -mfloat-abi=softfp -mfpu=vfpv3-d16 -meabi=5 -o /tmp/ccteDbiT.o /tmp/ccngWr0p.s**

COMPILER_PATH=/usr/lib/gcc/arm-linux-gnueabi/4.6/:/usr/lib/gcc/arm-linux-

gnueabi/4.6/:/usr/lib/gcc/arm-linux-gnueabi/:/usr/lib/gcc/arm-linux-gnueabi/4.6/:/usr/lib/gcc/arm-linux-gnueabi/:/usr/lib/gcc/arm-linux-gnueabi/4.6/../../../../arm-linux-gnueabi/bin/

LIBRARY_PATH=/usr/lib/gcc/arm-linux-gnueabi/4.6/:/usr/lib/gcc/arm-linux-gnueabi/4.6/../../../../arm-linux-gnueabi/lib/../lib/:/usr/lib/gcc/arm-linux-gnueabi/4.6/../../../../arm-linux-gnueabi/lib/

COLLECT_GCC_OPTIONS='-fverbose-asm' '-v' '-g' '-o' 'test_arm' '-march=armv7-a' '-mfloat-abi=softfp' '-mfpu=vfpv3-d16' '-mthumb'

/usr/lib/gcc/arm-linux-gnueabi/4.6/collect2 --build-id --no-add-needed --as-needed --eh-frame-hdr --dynamic-linker /lib/ld-linux.so.3 -X --hash-style=gnu -m armelf_linux_eabi -z relro -o test_arm /usr/lib/gcc/arm-linux-gnueabi/4.6/../../../../arm-linux-gnueabi/lib/../lib/crt1.o /usr/lib/gcc/arm-linux-gnueabi/4.6/../../../../arm-linux-gnueabi/lib/../lib/crti.o /usr/lib/gcc/arm-linux-gnueabi/4.6/crtbegin.o -L/usr/lib/gcc/arm-linux-gnueabi/4.6 -L/usr/lib/gcc/arm-linux-gnueabi/4.6/../../../../arm-linux-gnueabi/lib/../lib -L/usr/lib/gcc/arm-linux-gnueabi/4.6/../../../../arm-linux-gnueabi/lib -L/lib/arm-linux-gnueabi -L/usr/lib/arm-linux-gnueabi -L/usr/lib -L/lib /tmp/ccteDbiT.o -lgcc --as-needed -lgcc_s --no-as-needed -lc -lgcc --as-needed -lgcc_s --no-as-needed /usr/lib/gcc/arm-linux-gnueabi/4.6/crtend.o /usr/lib/gcc/arm-linux-gnueabi/4.6/../../../../arm-linux-gnueabi/lib/../lib/crtn.o

---

Toolchain from yocto

walterzh$ /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-gnueabi/arm-poky-linux-gnueabi-gcc -v -fverbose-asm -mthumb-interwork -g -o test_arm test.c

Using built-in specs.

COLLECT_GCC=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-gnueabi/arm-poky-linux-gnueabi-gcc

COLLECT_LTO_WRAPPER=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/libexec/armv7a-vfp-neon-poky-linux-gnueabi/gcc/arm-poky-linux-gnueabi/4.8.1/lto-wrapper

Target: arm-poky-linux-gnueabi

Configured with: /home/walterzh/gerrit/build-bundle/poky/build/tmp/work-shared/gcc-4.8.1-r0/gcc-4.8.1/configure --build=x86_64-linux --host=x86_64-linux --target=arm-poky-linux-gnueabi --prefix=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr --exec_prefix=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr --

bindir=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-gnueabi --sbindir=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-gnueabi --libexecdir=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/libexec/armv7a-vfp-neon-poky-linux-gnueabi --datadir=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/share --sysconfdir=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/etc --sharedstatedir=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/com --localstatedir=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/var --libdir=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/lib/armv7a-vfp-neon-poky-linux-gnueabi --includedir=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/include --oldincludedir=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/include --infodir=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/share/info --mandir=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/share/man --disable-silent-rules --disable-dependency-tracking --with-libtool-sysroot=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux --enable-clocale=generic --with-gnu-ld --enable-shared --enable-languages=c,c++ --enable-threads=posix --disable-multilib --enable-c99 --enable-long-long --enable-symvers=gnu --enable-libstdcxx-pch --program-prefix=arm-poky-linux-gnueabi- --without-local-prefix --enable-target-optspace --enable-lto --enable-libssp --disable-bootstrap --disable-libmudflap --with-system-zlib --with-linker-hash-style=gnu --enable-linker-build-id --with-ppl=no --with-cloog=no --enable-checking=release --enable-cheaders=c_global --with-gxx-include-dir=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2/usr/include/c++ --with-sysroot=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2 --with-build-sysroot=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2 --enable-poison-system-directories --disable-libunwind-exceptions --with-mpfr=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr --with-system-zlib --disable-nls

Thread model: posix

gcc version 4.8.1 (GCC)

**COLLECT_GCC_OPTIONS='-v' '-fverbose-asm' '-mthumb-interwork' '-g' '-o' 'test_arm' '-mtls-dialect=gnu'**

**/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/libexec/armv7a-vfp-neon-poky-linux-gnueabi/gcc/arm-poky-linux-gnueabi/4.8.1/cc1 -quiet -v test.c -quiet -dumpbase test.c -mthumb-interwork -mtls-dialect=gnu -auxbase test -g -version -fverbose-asm -o /tmp/cc9MXXTe.s**

GNU C (GCC) version 4.8.1 (arm-poky-linux-gnueabi)

compiled by GNU C version 4.6.3, GMP version 5.1.1, MPFR version 3.1.2, MPC version 1.0.1

GGC heuristics: --param ggc-min-expand=100 --param ggc-min-heapsize=131072

ignoring nonexistent directory "/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2/usr/local/include"

ignoring nonexistent directory "/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/lib/armv7a-vfp-neon-poky-linux-gnueabi/gcc/arm-poky-linux-gnueabi/4.8.1/../../../../../arm-poky-linux-gnueabi/include"

#include "..." search starts here:

#include <...> search starts here:

/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/lib/armv7a-vfp-neon-poky-linux-gnueabi/gcc/arm-poky-linux-gnueabi/4.8.1/include

/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/lib/armv7a-vfp-neon-poky-linux-gnueabi/gcc/arm-poky-linux-gnueabi/4.8.1/include-fixed

/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2/usr/include

End of search list.

GNU C (GCC) version 4.8.1 (arm-poky-linux-gnueabi)

compiled by GNU C version 4.6.3, GMP version 5.1.1, MPFR version 3.1.2, MPC version 1.0.1

GGC heuristics: --param ggc-min-expand=100 --param ggc-min-heapsize=131072

Compiler executable checksum: 75333558048c4b590920e22e3ba753ba

COLLECT_GCC_OPTIONS='-v' '-fverbose-asm' '-mthumb-interwork' '-g' '-o' 'test_arm' '-mtls-dialect=gnu'

/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/libexec/armv7a-vfp-neon-poky-linux-gnueabi/gcc/arm-poky-linux-gnueabi/4.8.1/as -v -mthumb-interwork -meabi=5 -o /tmp/cch7OYoi.o /tmp/cc9MXXTe.s

GNU assembler version 2.23.2 (arm-poky-linux-gnueabi) using BFD version (GNU Binutils) 2.23.2

COMPILER_PATH=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/libexec/armv7a-vfp-neon-poky-linux-gnueabi/gcc/arm-poky-linux-gnueabi/4.8.1/:/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/libexec/armv7a-vfp-neon-poky-linux-gnueabi/gcc/arm-poky-linux-gnueabi/4.8.1/:/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/libexec/armv7a-vfp-neon-poky-linux-gnueabi/gcc/arm-poky-linux-

gnueabi/:/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/lib/armv7a-vfp-neon-poky-linux-gnueabi/gcc/arm-poky-linux-gnueabi/4.8.1/:/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/lib/armv7a-vfp-neon-poky-linux-gnueabi/gcc/arm-poky-linux-gnueabi/

LIBRARY_PATH=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/lib/armv7a-vfp-neon-poky-linux-gnueabi/gcc/arm-poky-linux-gnueabi/4.8.1/:/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2/lib/:/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2/usr/lib/arm-poky-linux-gnueabi/4.8.1/:/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2/usr/lib/

COLLECT_GCC_OPTIONS='-v' '-fverbose-asm' '-mthumb-interwork' '-g' '-o' 'test_arm' '-mtls-dialect=gnu'

/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/libexec/armv7a-vfp-neon-poky-linux-gnueabi/gcc/arm-poky-linux-gnueabi/4.8.1/collect2 --sysroot=/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2 --build-id --eh-frame-hdr --hash-style=gnu -dynamic-linker /lib/ld-linux.so.3 -X -m armelf_linux_eabi -o test_arm /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2/usr/lib/crt1.o /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2/usr/lib/crti.o /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2/usr/lib/arm-poky-linux-gnueabi/4.8.1/crtbegin.o -L/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/lib/armv7a-vfp-neon-poky-linux-gnueabi/gcc/arm-poky-linux-gnueabi/4.8.1 -L/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2/lib -L/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2/usr/lib/arm-poky-linux-gnueabi/4.8.1 -L/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2/usr/lib /tmp/cch7OYoi.o -lgcc --as-needed -lgcc_s --no-as-needed -lc -lgcc --as-needed -lgcc_s --no-as-needed /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2/usr/lib/arm-poky-linux-gnueabi/4.8.1/crtend.o /home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/granite2/usr/lib/crtn.o

---

之所以不同gcc compiler生成instruction不同，是因为默认的编译选项不同。

arm-gcc from Ubuntu/Linaro 4.6.3-1ubuntu5 has -mthumb option, and arm-gcc from yocto has not the option.

果然，给arm-gcc from yocto加上"-mthumb" option

/home/walterzh/gerrit/build-bundle/poky/build/tmp/sysroots/x86_64-linux/usr/bin/armv7a-vfp-neon-poky-linux-gnueabi/arm-poky-linux-gnueabi-gcc **-mthumb** -fverbose-asm -mthumb-interwork -g -o test_arm test.c

00008454 <main>:

```
    8454:       b580            push    {r7, lr}
    8456:       b086            sub     sp, #24
    8458:       af00            add     r7, sp, #0
    845a:       6078            str     r0, [r7, #4]
    845c:       6039            str     r1, [r7, #0]
    845e:       2364            movs    r3, #100        ; 0x64
    8460:       617b            str     r3, [r7, #20]
    8462:       23c8            movs    r3, #200        ; 0xc8
    8464:       613b            str     r3, [r7, #16]
    8466:       2305            movs    r3, #5
    8468:       60fb            str     r3, [r7, #12]
    846a:       0000            movs    r0, r0
    846c:       e1a0200e        mov     r2, lr
    8470:       e1a0300f        mov     r3, pc
    8474:       617a            str     r2, [r7, #20]
    8476:       613b            str     r3, [r7, #16]
    8478:       4907            ldr     r1, [pc, #28]   ; (8498 <main+0x44>)
    847a:       697a            ldr     r2, [r7, #20]
```

```
    847c:       693b            ldr     r3, [r7, #16]
    847e:       1c08            adds    r0, r1, #0
    8480:       1c11            adds    r1, r2, #0
    8482:       1c1a            adds    r2, r3, #0
    8484:       f7ff ef2e       blx     82e4 <_init+0x20>
    8488:       68fb            ldr     r3, [r7, #12]
    848a:       1c18            adds    r0, r3, #0
    848c:       2396            movs    r3, #150        ; 0x96
    848e:       005b            lsls    r3, r3, #1
    8490:       1c18            adds    r0, r3, #0
    8492:       f7ff ef2e       blx     82f0 <_init+0x2c>
    8496:       e7f9            b.n     848c <main+0x38>
    8498:       00008510        .word   0x00008510
```

而给arm-gcc from Ubuntu/Linaro 4.6.3-1ubuntu5加上"-marm" option

```
$ arm-linux-gnueabi-gcc -fverbose-asm -marm -g -o test_arm test.c
```

```
000083bc <main>:
    83bc:       e92d4830        push    {r4, r5, fp, lr}
    83c0:       e28db00c        add     fp, sp, #12
    83c4:       e24dd018        sub     sp, sp, #24
    83c8:       e50b0020        str     r0, [fp, #-32]
    83cc:       e50b1024        str     r1, [fp, #-36]  ; 0x24
    83d0:       e3a03064        mov     r3, #100        ; 0x64
```

```
83d4:	e50b3018	str	r3, [fp, #-24]
83d8:	e3a030c8	mov	r3, #200	; 0xc8
83dc:	e50b3014	str	r3, [fp, #-20]
83e0:	e3a03005	mov	r3, #5
83e4:	e50b3010	str	r3, [fp, #-16]
83e8:	e1a0500e	mov	r5, lr
83ec:	e1a0400f	mov	r4, pc
83f0:	e50b5018	str	r5, [fp, #-24]
83f4:	e50b4014	str	r4, [fp, #-20]
83f8:	e3083478	movw	r3, #33912	; 0x8478
83fc:	e3403000	movt	r3, #0
8400:	e1a00003	mov	r0, r3
8404:	e51b1018	ldr	r1, [fp, #-24]
8408:	e51b2014	ldr	r2, [fp, #-20]
840c:	ebffffbc	bl	8304 <_init+0x20>
8410:	e51b3010	ldr	r3, [fp, #-16]
8414:	e1a00003	mov	r0, r3
8418:	e3a00f4b	mov	r0, #300	; 0x12c
841c:	ebffffbb	bl	8310 <_init+0x2c>
8420:	eafffffc	b	8418 <main+0x5c>
```