

Re: arm-elf-gcc building erroneous code for ISR (long posting)

Source: <http://coding.derkeiler.com/Archive/General/comp.arch.embedded/2004-06/1090.html>

From: Peter Dickerson (*first{dot}surname_at_ukonline.co.uk*)

Date: 06/17/04

Date: Thu, 17 Jun 2004 17:51:08 +0100

"Jens Hildebrandt" <jens.hildebrandt@remove_this.etechnik.uni-rostock.de> wrote in message news:40cd60da@news.uni-rostock.de...
> *Hello group,*
>
> *I'm curenly trying to get familiar with the ARM7 by building some*
> *example-projects for a LPC2106-controller. One such project is the*
blinky_irq
> *example from Keil which should demonstrate the use of interrupts and of*
the
> *LPC2xxx vectored interrupt controller. I'm using gcc-3.2.1 with binutils*
3.13.1
> *and newlib 1.11.0 . So far I had no problems with these tools and I was*
able to
> *build and succesfully run some small demonstrator programs (without*
interrupts)
> *on my LPC2106 board. For the blinky_irq example program I use the linker*
script
> *and startup.s files (with adjustment for the Flash and RAM sizes of the*
LPC2106)
> *that came with the source code from Keil's download area. Since the*
program
> *didn't work I stripped it from all unnecessary functionality (main() is*
> *essentially an empty loop now) and simply let the timer ISR toggle a LED*
on
> *GPIO24. That worked initially, but after some time (~ 3 minutes) the LED*
stopped
> *toggling. I suspected that somewhere I'm loosing memory such that after*
some time
> *I'm out of RAM. Hence, I took a look at the disassembled code and saw that*
the
> *ISR is messing up the IRQ-mode stack pointer, loosing seven words each*
time the
> *ISR is invoked.*
> *Since the ISR entry and exit codes are automatically generated by the*
compiler I

comp.arch.embedded: Re: arm-elf-gcc building erroneous code for ISR (long posting)

> *suspect a bug in gcc. On the other hand, since this would be a very severe*
> *failure other users should have come across it, too. Does anybody of you*
know
> *that error (i.e. should I change to another gcc version) or am I missing*
> *something (e.g. gcc options)?*
> *Below you'll find the build commands used, the sources of the program and*
the
> *disassembled ISR code.*
>
> *TIA,*
> *Jens*
[snip]

using arm-elf-gcc 3.3.1 I don't have this problem. Not that the interrupt attribute needs to specify the type of interrupt ("IRQ", "FIQ", "SWI" etc).

```
void __attribute__((interrupt("IRQ"))) SYS_10ms_irq_handler(void)
{
    TIMER2->status = 4; // clear timer int
    VIC->VectAddr = 0; // clear VIC
    tick_count++;
}
```

```
arm-elf-gcc -S -O -DUVDAS -mcpu=arm7tdmi sys_irq.c
```

SYS_10ms_irq_handler:

```
@ Interrupt Service Routine.
@ args = 0, pretend = 0, frame = 0
@ frame_needed = 0, uses_anonymous_args = 0
@ link register save eliminated.
stmfd sp!, {r2, r3}
@ lr needed for prologue
mvn r3, #244736
sub r3, r3, #940
sub r3, r3, #3
mov r2, #4
str r2, [r3, #8]
mov r3, #-2147483648
mov r3, r3, asr #19
mov r2, #0
str r2, [r3, #48]
ldr r2, .L3
ldr r3, [r2, #0]
add r3, r3, #1
str r3, [r2, #0]
ldmfd sp!, {r2, r3}
subs pc, lr, #4
.L4:
.align 2
.L3:
.word tick_count
```

Re: arm-elf-gcc building erroneous code for ISR (long posting)

comp.arch.embedded: Re: arm-elf-gcc building erroneous code for ISR (long posting)

this include the correct subs pc,lr,#4 that performs a return from interrupt
(swapping back to old stack).

regards
Peter