

# NEON/VFP Application in StarterWare

Gary Wu

# Agenda

- NEON/VFP Overview
- NEON/VFP Enable
- NEON/VFP application in StarterWare
- Solution

# NEON/VFP Overview

- What is NEON?
  - a SIMD (Single Instruction Multiple Data) accelerator processor integrated in as part of the [ARM Cortex-A8](#)
  - Neon instructions are mainly for numerical, load/store, and some logical operations
- What is VFP(Vector Floating Point) ?
  - a floating point hardware accelerator
  - Function: speed up floating point calculations
  - supports both single and double precision floating point calculations compliant with IEEE754

# NEON/VFP Enable

- Adding the assembly code into the init.asm as a function:

; Initialize NEON and VFP

```
MRC    p15, #0, r3, c1, c0, #2    ; Read CPACR
ORR    r3, r3, #0x00F00000        ; Enable access to CP10 and CP11
MCR    p15, #0, r3, c1, c0, #2    ; Write CPACR
```

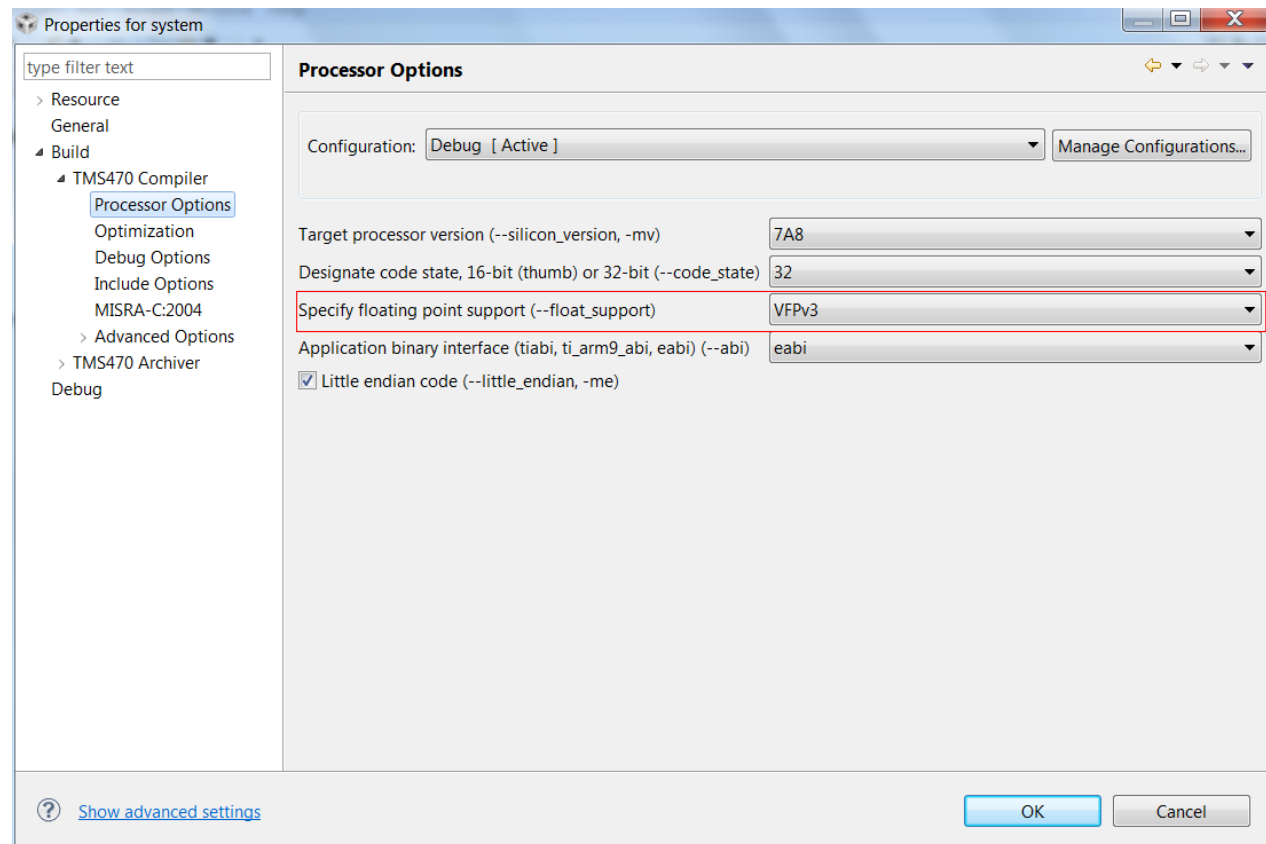
; Enable NEON and VFP

```
MOV r0, #0x40000000
VMSR FPEXC, r0
VMRS R1, FPSID
```

Pls refer to [http://processors.wiki.ti.com/index.php/Cortex-A8#How\\_to\\_enable\\_NEON](http://processors.wiki.ti.com/index.php/Cortex-A8#How_to_enable_NEON)

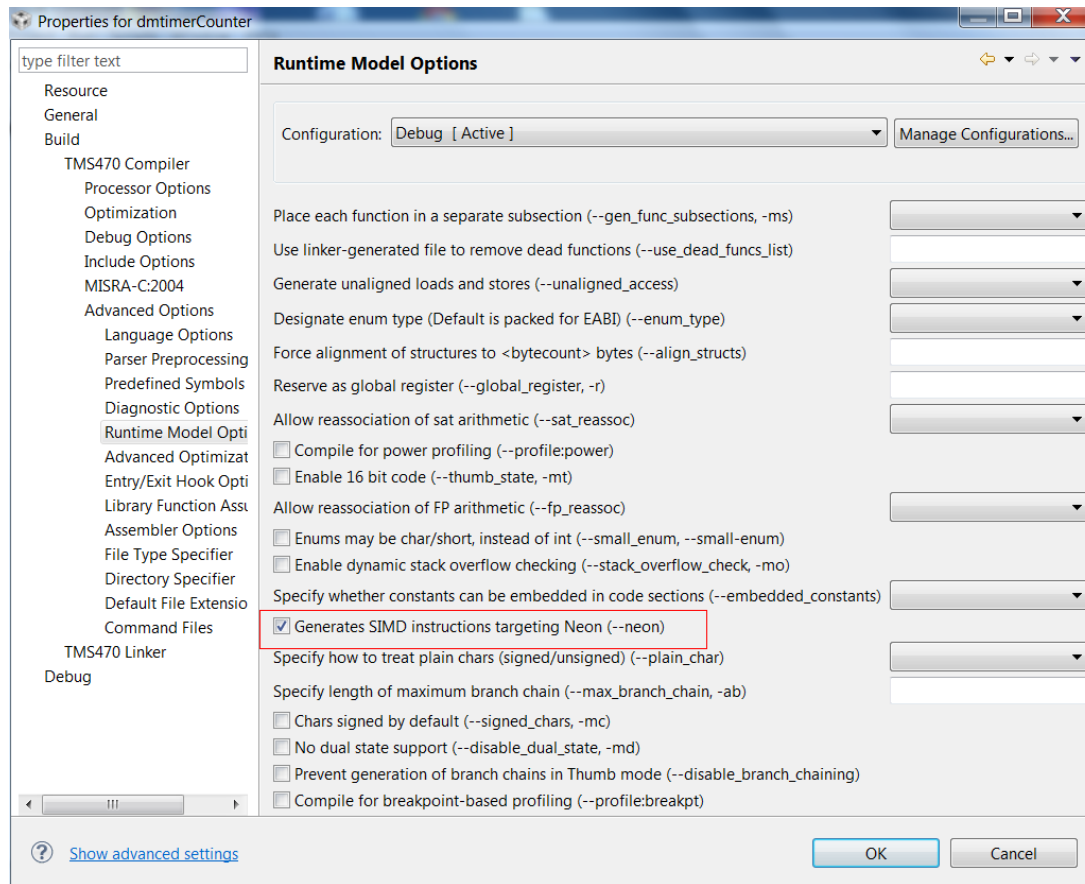
# VFP Enable

- Configure VFP support in build option
  - From CCS Project options: Build->TMS470 Compiler->Processor Options->Specify floating point support(--float\_support)



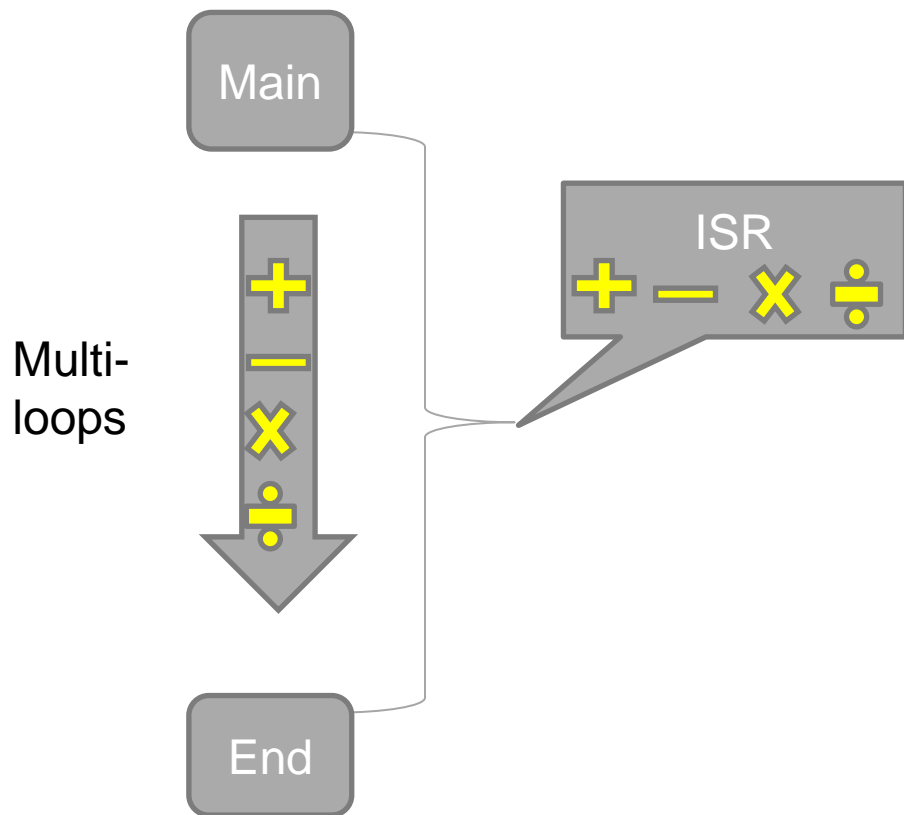
# Neon Enable

- From CCS Project options: Build->TMS470 Compiler->Advanced Options->Runtime Model Options->Generates SIMD instructions (--neon)



“TI Proprietary Information - Strictly Private” or similar placed here if applicable

# NEON/VFP Application in StarterWare

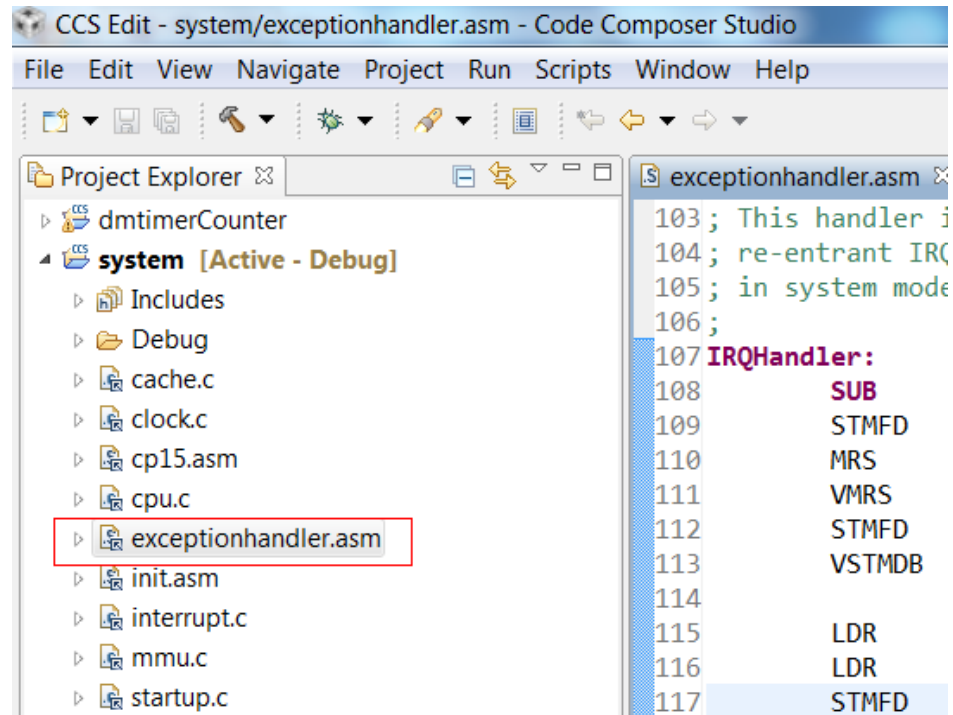


- StarterWare does not save NEON/VFP registers to the stack during an IRQ.
- Issue description:
  - The computing with in main function will be wrong sometimes if there is some computing in ISR when NEON/VFP enabled.

Note: This issue can be found in StarterWare\_02\_00\_00\_07 and previous versions.

# Solution

- Save off the NEON/VFP registers and restore them on interrupt handling completion
  - Modify “system” project in StarterWare and rebuild it.
    - "exceptionhandler.asm" needs to be modified to save and restore the NEON/VFP context.



The screenshot shows the Code Composer Studio interface. The title bar reads "CCS Edit - system/exceptionhandler.asm - Code Composer Studio". The menu bar includes File, Edit, View, Navigate, Project, Run, Scripts, Window, and Help. The Project Explorer on the left shows a tree view with the "system" project selected and expanded, listing files like cache.c, clock.c, cp15.asm, cpu.c, exceptionhandler.asm (highlighted with a red box), init.asm, interrupt.c, mmu.c, and startup.c. The main editor window displays the assembly code for exceptionhandler.asm, starting with line 103: "103; This handler i" and ending with line 117: "117 STMFD". The code includes comments and assembly instructions like SUB, STMFD, MRS, VMRS, VSTMDB, and LDR.

```
103; This handler i
104; re-entrant IRQ
105; in system mode
106;
107 IRQHandler:
108     SUB
109     STMFD
110     MRS
111     VMRS
112     STMFD
113     VSTMDB
114
115     LDR
116     LDR
117     STMFD
```



# Solution—con't

- Modification for exceptionhandler.asm

IRQHandler:

SUB r14, r14, #4 ; Apply lr correction

STMFD r13!, {r0-r3, r12, r14} ; Save context in IRQ stack

MRS r3, spsr ; Copy spsr

VMRS r1, FPSCR ; Copy fpscr

STMFD r13!, {r1, r3} ; Save fpcr and spsr

VSTMDB [r13]!, {d0-d7} ; Save NEON/VFP scratch registers

LDR r0, ADDR\_THRESHOLD ; Get the IRQ Threshold

. (Omit)

.

VLDMIA [r13]!, {d0-d7} ; Restore NEON/VFP scratch registers

LDMFD r13!, {r1, r3} ; get fpscr and spsr from stack

MSR spsr\_cxsf, r3 ; Restore spsr

VMSR FPSCR, r1 ; Restore fpscr

LDMFD r13!, {r0-r3, r12, pc}^ ; Restore the context and return

# Useful links on TI wiki

- <http://processors.wiki.ti.com/index.php/Cortex-A8>
- [http://processors.wiki.ti.com/index.php/Cortex-A8 Features](http://processors.wiki.ti.com/index.php/Cortex-A8_Features)
- [http://processors.wiki.ti.com/index.php/Using NEON and VFPv3 on Cortex-A8](http://processors.wiki.ti.com/index.php/Using_NEON_and_VFPv3_on_Cortex-A8)