



VFP Porting Guide for SMDK6410 (Windows Embedded CE 6.0)

S3C64x0

Feb. 11, 2009

(Preliminary) REV 0.3

Important Notice

The information in this publication has been carefully checked and is believed to be entirely accurate at the time of publication. Samsung assumes no responsibility, however, for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

Samsung reserves the right to make changes in its products or product specifications with the intent to improve function or design at any time and without notice and is not required to update this documentation to reflect such changes.

This publication does not convey to a purchaser of semiconductor devices described herein any license under the patent rights of Samsung or others.

Samsung makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Samsung assume any liability arising out of the application or use of any product or circuit and specifically disclaims any and all liability, including without limitation any consequential or incidental damages.

"Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by the customer's technical experts.

Samsung products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, for other applications intended to support or sustain life, or for any other application in which the failure of the Samsung product could create a situation where personal injury or death may occur.

Should the Buyer purchase or use a Samsung product for any such unintended or unauthorized application, the Buyer shall indemnify and hold Samsung and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim of personal injury or death that may be associated with such unintended or unauthorized use, even if such claim alleges that Samsung was negligent regarding the design or manufacture of said product

S3C64x0 RISC Microprocessor VFP Porting Guide

Copyright © 2008-2009 Samsung Electronics Co., Ltd.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electric or mechanical, by photocopying, recording, or otherwise, without the prior written consent of Samsung Electronics Co., Ltd.

Samsung Electronics Co., Ltd.
San #24 Nongseo-Dong, Giheung-Gu
Yongin-City Gyeonggi-Do, Korea
446-711

Home Page: <http://www.samsungsemi.com/>

E-Mail: mobilesol.cs@samsung.com



Preliminary product information describe products that are in development, for which full characterization data and associated errata are not yet available. Specifications and information herein are subject to change without notice.

Revision History

Revision No	Description of Change	Refer to	Author(s)	Date
0.1	Preliminary draft	-	WINCE TEAM	2008-7-17
0.2	Review Comments Updated		WINCE TEAM	2008-7-18
0.3	Correction for Directory Structure of BSP		WINCE TEAM	2009-2-11

NOTE: REVISED PARTS ARE WRITTEN IN BLUE.

Contents

1	INTRODUCTION.....	5
1.1	REFERENCE DOCUMENT	5
1.2	VFP LIBRARY INSTALLATION AND BUILD MECHANISM	5
1.3	CODE CHANGES FOR VFP INITIALIZATION IN STARTUP.S IN OAL	5
1.4	TEST APPLICATION INSTALLATION.....	9
1.5	BUILD METHODOLOGY FOR OS IMAGE WITH VFP SUPPORT.....	10
1.6	TEST PROCEDURE.....	10

1 Introduction

This document is intended to provide guideline to OEM developers for enabling the Samsung SMDK64x0's VFP Unit in Microsoft Windows Embedded CE Version 6.0.

1.1 Reference Document

The following additional documentation will help you to work easily.

- Samsung User's Manual of S3C64x0X Application Processor
- ARM1176JZFS TRM(VFP)

1.2 VFP library installation and build mechanism

- Follow all steps described in document SMDK64x0_WinCE6.0_VFP_PortingGuide_HQ.doc except following steps

1. Step 5 ("5. VFP build phase is SYSGEN (remap FPCRT functions to OEM FPCR functions). So if you want to apply VFP to current proceeding project, you must do SYSGEN or Rebuild Solution".) in Section "2.7 Build Image with FPCRT using VFP library point".
2. Add VFP register backup & restore code to startup.s in oal



SMDK64x0_WinCE6.
0_VFP_PortingGuide_

1.3 Code Changes for VFP initialization in startup.s in oal

- Add following macro after "Macro for Sleep Code" definition.
- SMDK6410BSP has already these codes. But, if you have used old BSP then apply all these codes to use VFP. you can check VFP running or not with some test application that described in Section 1.4 for TEST

```
-----  
; Macro For VFP  
-----
```

```
VFPEnable EQU (0x40000000)
```

- In **ResetHandler** label, add code block below, after following code block.

Check comment (; **Add following: SISO added**) below.

```

-----
; Enable VFP via Coprocessor Access Control Register
;-----
        mrc          p15, 0, r0, c1, c0, 2
        orr          r0, r0, #0x00F00000
        mcr          p15, 0, r0, c1, c0, 2

;-----
; Add following: SISO added
; Enable FPEXC enable bit to enable VFP
;-----
        MOV         r1, #0
        MCR         p15, 0, r1, c7, c5, 4
        MOV         r0, #VFPEnable
        FMXR        FPEXC, r0      ; FPEXC = r0
        nop
        nop
        nop
        nop
        nop

```

- In Store and Restore Code Section add following changes
 1. Insert register backup code in "3. Save CPU Register into Sleep Data Area in DRAM" in **OALCPUPowerOff** label after following code block.
(Check comment:; **Add following : SISO added**)

```
;-----
```

```
; System (User) mode CPU Registers
```

```
mov      r1, #Mode_SYS | NOINT ; Enter SYS mode, no interrupts
msr      cpsr, r1
stmia    r3!, {sp, lr} ; Store the SYS mode Registers [SleepState_SYS_SP,
SleepState_SYS_LR]
```

```
;-----
```

```
;Add following : SISO added
```

```
; 3-1. Save VFP Register into Sleep Data Area in DRAM
```

```
;-----
```

```
;-----
```

```
; Floating Point Status and Control Register using FMRX
```

```
; FMRX{cond} Rd, VFPsysreg VFPsysreg -> Rd
```

```
; FMXR{cond} VFPsysreg, Rd Rd -> VFPsysreg
```

```
; FPSCR
```

```
fmrx     r2, fpscr
```

```
str      r2, [r3], #4 ; [SleepState_VFP_FPSCR]
```

```
;-----
```

```
; Floating Point Exception Register
```

```
fmrx     r2, fpexc
```

```
str      r2, [r3], #4 ; [SleepState_VFP_FPEXC]
```

```
;-----
```

```
;VFP Register File (using FLDMX, FSTMX)
```

```
; FLDM<addressmode><precision>{cond} Rn,{!} VFPregisters
```

```

;FSTM<addressmode><precision>{cond} Rn,{!} VFPregisters
; <addressmode>
;IA : Incremental address After each transfer
;DB : Decremental address Before each transfer
;EA : Empty Ascending stack operation, this is the same as DB for loads, and the same
as IA for saves.

```

```

;FD : Full Descending stack operation, this is the same as IA for loads, and the same
as DB for saves.

```

```

; <precision>
;      S : for single-precision
;      D : for double-precision
;      X : for unspecified precision
;      example::
;      FLDMIAS      r2, {s1-s5}
;      FSTMFDD      r13!, {d3-d6}
;      FSTMFDX      r13!, {d0-d3}
;      FLDMFDX      r13!, {d0-d3}
fstmiax r3!,    {d0-d15}

```

2. Insert VFP register restore code between "1. Restore CPU Register from Sleep Data Area in DRAM" and "2. Pop SVC Register from our Stack" in WakeUp_Address label after following code block (Check comment;; **Add following : SISO added**)

```

;-----
; Supervisor mode CPU Registers

mov      r1, #Mode_SVC | I_Bit ; Enter SVC mode, no IRQ - FIQ is available
msr      cpsr, r1

ldr      r0, [r3, #SleepState_SVC_SPSR]
msr      spsr, r0
ldr      sp, [r3, #SleepState_SVC_SP]

```



```
;-----  
; Add following: SISO added  
; 1-1 Restore VFP system control registers  
;-----  
  
;-----  
;FMRX{cond} Rd, VFPsysreg    VFPsysreg -> Rd  
; FMXR{cond} VFPsysreg, Rd      Rd -> VFPsysreg  
;  
;      FPSCR  
ldr      r0, [r3, #SleepState_VFP_FPSCR]  
fmxr     fpscr, r0  
  
;-----  
;  
;      Floating Point Exception Register  
ldr      r0, [r3, #SleepState_VFP_FPEXC]  
fmxr     fpexc, r0
```

1.4 Test application installation.

- Create Platform\<BSP_NAME>\Src\Drivers\VFP_TEST folder.
- Unzip following test application folder in Platform\<BSP_NAME>\Src\Drivers\VFP_TEST



VFP_TEST.zip

- Add following in Platform\<BSP_NAME>\Src\Drivers\dir in Dir = \ at the end of directory list.
VFP_TEST \
- In Platform.bib file add following in FILES section.

VFP_DIVTEST.exe	\$(_FLATRELEASEDIR)\VFP_DIVTEST.exe	NK	U
VFP_MUTTEST.exe	\$(_FLATRELEASEDIR)\VFP_MUTTEST.exe	NK	U
VFP_ADDTEST.exe	\$(_FLATRELEASEDIR)\VFP_ADDTEST.exe	NK	U

1.5 Build methodology for OS image with VFP support.

- VFP build phase is SYSGEN (remap FPCRT functions to OEM FPCRT functions).
So to apply VFP to current proceeding project, you must do SYSGEN or Rebuild Solution"

1.6 Test Procedure.

- Flash Os image to target, refer document
"SMDK6410_WinCE6.0_FMD_InstallationManual.doc" for flashing OS image.
Open test applications VFP_DIVTEST.exe VFP_MUTTEST.exe and VFP_ADDTEST.exe, these applications are located in "My device\windows" folder on the wince target.
- Following results can be verified.
VFP_DIVTEST.exe : Division of two double numbers and Division by Zero Exception using VFP.
VFP_MUTTEST.exe : Multiplication of two double numbers.
VFP_ADDTEST.exe : Addition of two double numbers.
- Power down the SMDK6410 board by pressing external interrupt EINT11 switch and power up board again by pressing EINT11 again.
- Above application will continue to run and give same result as above.
- Result can be matched with the result in following result File.



VFP_TEST_RESULT.t
xt