Trace Hard Fault Cause

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	Find the sample code for the CPU to read the illegal address and cause the hardware hard fault
BSP Version	Nano100B Series BSP CMSIS v3.03.000
Hardware	NuTiny–EVB–NANO130_LQFP128 V1.2

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 Function Description

1.1 Introduction

When the microcontroller reads the memory through the 16-bit pointer, if the 16-bit pointer position does not meet the 2-byte alignment limit, the hardware will generate a fault and let the processor jump to the exception handler. It will demonstrate how to trace the cause of a hardware exception after a hard fault has occurred and jumped to hard fault handler.

1.2 Principle

In order to provide better access efficiency, the microprocessor adds a data alignment design. The development tool's linker usually uses Natural Alignment when arranging variables. The natural alignment is to align the size of the data type itself. Please refer to Figure 1-1, uint16_t aligns 2byte aligned, uint32_t aligns 4 byte aligned, uint8_t aligns byte aligned.

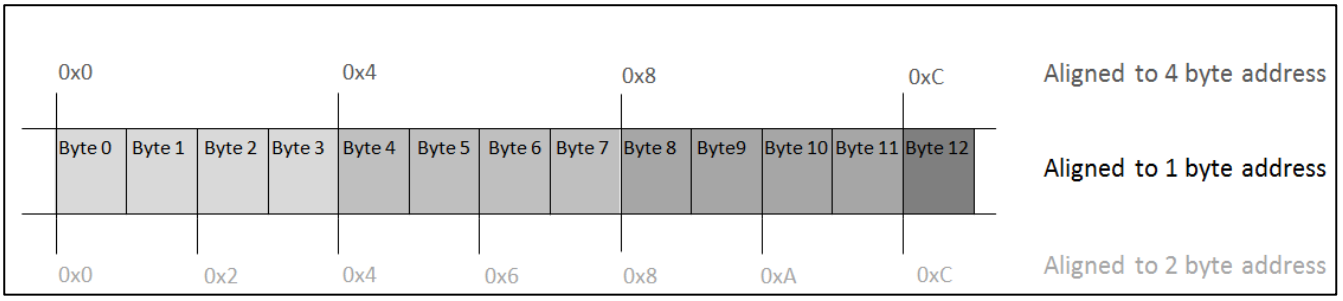

Figure 1-1 data natural alignment.

The Nuvoton NANO100 series is the core of the Arm® Cortex®–M0. Please refer to Figure 1-2 below. The Cortex®–M0 generic user guide provided by Arm® has special description for address alignment. Cortex®–M0 does not support non-aligned access. A non-aligned access method will cause a hardware exception to be interrupted.

Cortex-M0 Devices Generic User Guide

Home > The Cortex-M0 Instruction Set > About the instruction descriptions > Address alignment

3.3.4. Address alignment

An aligned access is an operation where a word-aligned address is used for a word, or multiple word access, or where a halfword-aligned address is used for a halfword access. Byte accesses are always aligned.

There is no support for unaligned accesses on the Cortex-M0 processor. Any attempt to perform an unaligned memory access operation results in a HardFault exception.

Figure 1-2 Arm® Cortex®-M0 Position Alignment Instructions (Ref. http://infocenter.arm.com)

When a hardware exception occurs, the scratchpad will be stored in the stack and go to the exception handler. At this point, you can observe the position of the stack indicator, and find out the value of the scratchpad when the hardware is abnormally interrupted by the position of the stack indicator. You can know the location of the hardware abnormality by referring to Figure 1-3, and query The location of the Disassembly PC is what instructions are executed. Equation 1-3 Temperature sensor voltage temperature conversion formula

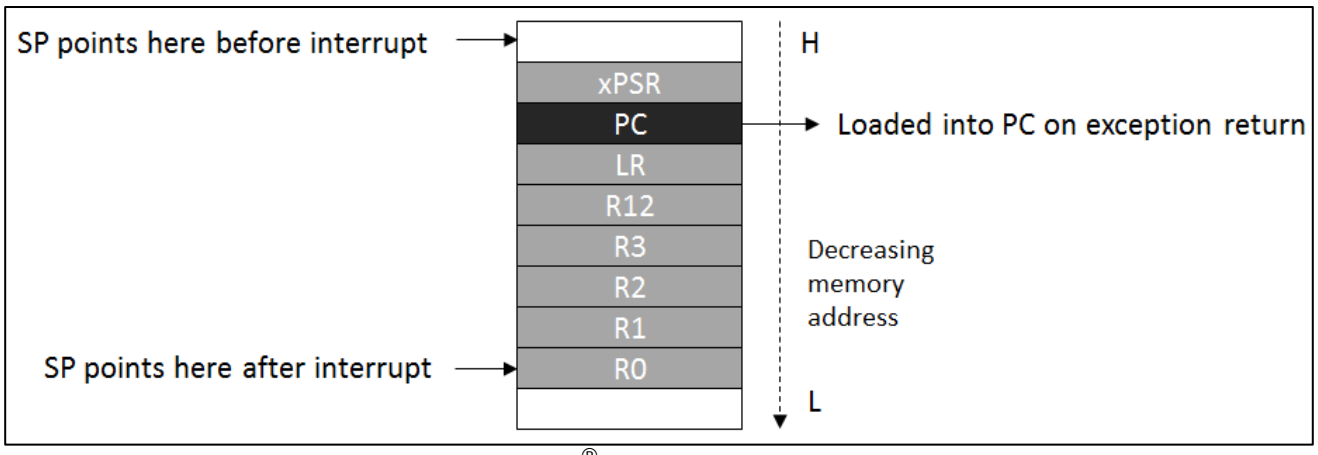

Figure 1-3 Cortex®-M0 stacking frame content

1.3 Demo Result

After hardware fault occurs, the data stored on the stack before the exception interrupt is showed, as shown in Figure 1-4. Next, we will briefly explain how to find out the cause from the debug mode.

```
In Hard Fault Handler
r0  = 0x20000001
r1  = 0x40050000
r2  = 0x400000
r3  = 0x791
r12 = 0x0
lr  = 0x775
pc  = 0x792
psr = 0x61000000
```

Figure 1-4 Example code result in UART

As shown in Figure 1-5 below,

1. After a hardware abnormal interrupt occurs, SP = 0x200004F0
2. Enter 0x200004F0 in the Memory window
3. According to the contents of the stacking frame, the PC is 0x78C.
4. In Disassembly, the 0x78C instruction can be found to read 16 bits for the 0x2000001D position.

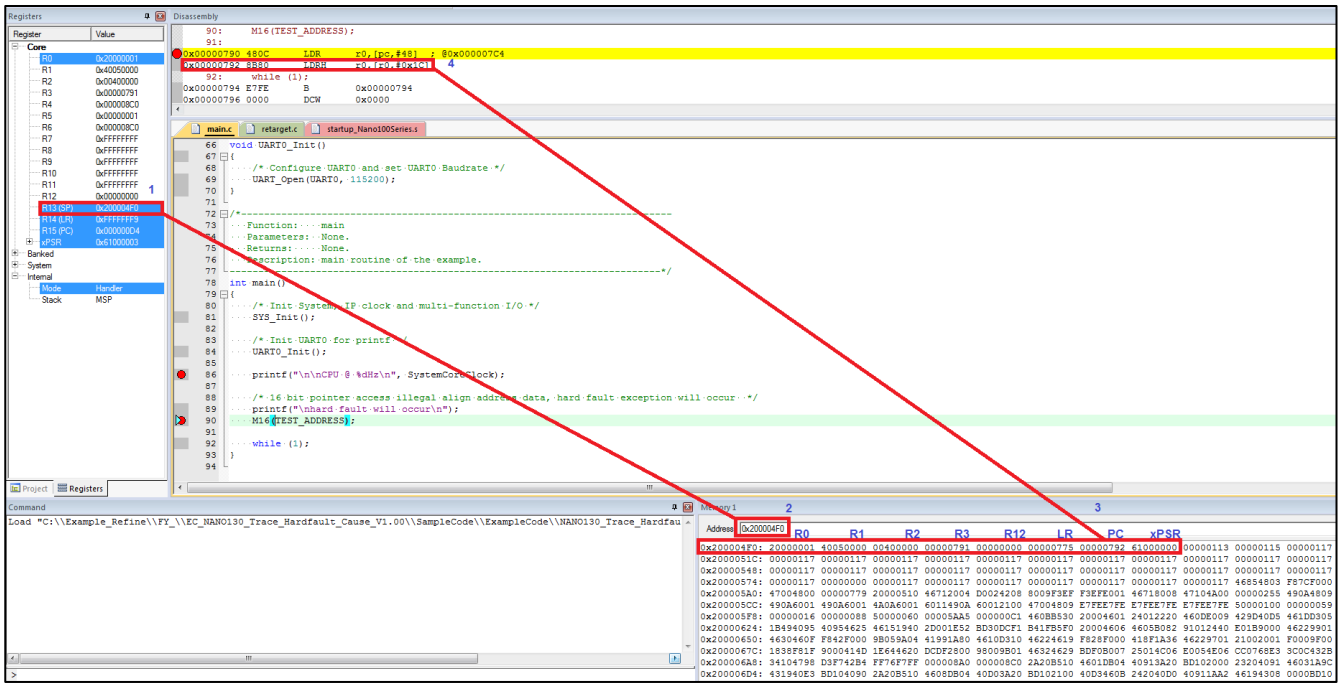

Figure 1-5 Example execution results in debug mode

2 Code Description

Define the memory location for the test access.

```
/* define pointer address – not follow data alignment */
#define TEST_ADDRESS                              0x2000001D
```

The test position is read using the 16 bits indicator and a hardware anomaly interrupt will be triggered after execution.

```
/* 16 bit pointer access illegal align address data */
    M16(TEST_ADDRESS);
```

3 Software and Hardware Environment

Software Environment

- BSP version
  - Nano100B Series BSP CMSIS v3.03.000
- IDE version
  - Keil uVersion 5.24

Hardware Environment

- Circuit components
  - NuTiny–EVB–NANO130_LQFP128 V1.2

4 Directory Information

- EC_NANO130_Trace_Hardfault_Cause_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code

5 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click Trace_Hardfault_Cause.uvproj.
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Jul. 9, 2019	1.00	1. Initially issued.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.