JEFI Basic Tutorial (XIX) - Simple Use of PCIe



This article shows how to locate the PCI root bridge and enumerate PCI devices in UEFI applications. Through the `LocatePciRootBridge Io` and `PciDevicePresent` functions, the program looks for `gPciRootBridgeIoProtocol`, reads the device information, and uses `PciIoProtocol` to acce ss the device. The code implements the PCI device search process starting from the EFI entry point.

The summary is generated in C Know, supported by DeepSeek-R1 full version, go to experience>

.. Write source code

Al generated projects

登录复制

```
//OvmfPkg/HelloWorldPci/HelloWorldPci.c
  2
     #include <Uefi.h>
  3
     #include <Library/UefiBootServicesTableLib.h>
     #include <Library/ShellCEntryLib.h>
  5
     #include <Library/DebugLib.h>
  6
  7
     #include <Protocol/PciIo.h>
  8
     #include <Protocol/PciRootBridgeIo.h>
  9
     #include <IndustryStandard/Pci.h>
 10
 11
 12
     EFI_PCI_ROOT_BRIDGE_IO_PROTOCOL *gPciRootBridgeIo;
 13
     EFI STATUS LocatePciRootBridgeIo(void);
 14
 15
 16
     EFI STATUS PciDevicePresent(
 17
       IN EFI PCI ROOT BRIDGE IO PROTOCOL * PciRootBridgeIo,
 18
       OUT PCI TYPE00 * Pci,
 19
       IN UINT8 Bus,
 20
       IN UINT8 Device,
twen
       IN UINT8 Func
twen
       );
twen
twen EFI_STATUS ListPciInformation(void);
 25
 26
 27
     EFI STATUS
 28
     EFIAPI
 29
     UefiMain (
 30
       IN EFI HANDLE
                             ImageHandle,
 31
       IN EFI_SYSTEM_TABLE *SystemTable
 32
 33
 34
       EFI STATUS Status;
 35
 36
 37
       Status = LocatePciRootBridgeIo();
 38
       if(EFI ERROR(Status))
 39
 40
         DEBUG((DEBUG_ERROR, "[CSDN]Call LocatePciRootBridgeIo failed,Can't find protocol!\n"));
 41
```

```
42
       else
 43
 44
        DEBUG((DEBUG_ERROR, "[CSDN]Call LocatePciRootBridgeIo successed,Find protocol!\n"));
 45
 46
 47
 48
       ListPciInformation();
 49
 50
      return EFI SUCCESS;
51
    }
52
53
    EFI STATUS LocatePciRootBridgeIo()
54
55
      EFI_STATUS Status;
56
       EFI_HANDLE *PciHandleBuffer = NULL;
 57
              HandleIndex = 0;
 58
      UINTN
                HandleCount = 0;
 59
 60
      Status = gBS->LocateHandleBuffer(
 61
        ByProtocol,
 62
        &gEfiPciRootBridgeIoProtocolGuid,
 63
        NULL.
 64
        &HandleCount,
 65
        &PciHandleBuffer
 66
        );
 67
       if(EFI_ERROR(Status)) return Status;
 68
69
       for(HandleIndex = 0; HandleIndex < HandleCount; HandleIndex++)</pre>
 70
 71
        Status = gBS->HandleProtocol(
 72
          PciHandleBuffer[HandleIndex],
73
          &gEfiPciRootBridgeIoProtocolGuid,
 74
          (VOID **)&gPciRootBridgeIo
 75
          );
 76
        if(EFI_ERROR(Status)) continue;
 77
                              return EFI_SUCCESS;
 78
       }
 79
 80
      return Status;
81
82
83
84
    EFI STATUS ListPciInformation()
85
 86
      EFI_STATUS Status = EFI_SUCCESS;
 87
       PCI TYPE00 Pci;
 88
      UINT16 Dev,Func,Bus,PciDevicecount = 0;
 89
 90
       DEBUG((DEBUG_ERROR, "[CSDN] PciDeviceNo\tBus\tDev\tFunc | Vendor.Device.ClassCode\n"));
 91
       for(Bus=0; Bus<256; Bus++) {</pre>
 92
        for(Dev=0; Dev<= PCI MAX DEVICE; Dev++)</pre>
 93
          for(Func=0; Func<=PCI MAX FUNC; Func++)</pre>
 94
 95
            Status = PciDevicePresent(gPciRootBridgeIo,&Pci,(UINT8)Bus,(UINT8)Dev,(UINT8)Func);
 96
              if(Status == EFI_SUCCESS)
 97
 98
                PciDevicecount++;
99
                100
                \label{lem:debug} DEBUG((DEBUG\_ERROR, "%x\t%x\n", Pci.Hdr.VendorId, Pci.Hdr.DeviceId, Pci.Hdr.ClassCode[0])); \\
101
102
          }
103
104
       return EFI_SUCCESS;
105
106
107
```

```
108 EFI_STATUS
109 | PciDevicePresent (
110
     IN EFI_PCI_ROOT_BRIDGE_IO_PROTOCOL
                                               *PciRootBridgeIo,
      OUT PCI TYPE00
111
                                               *Pci,
112
      IN UINT8
                                               Bus,
       IN UINT8
                                               Device,
113
       IN UINT8
114
                                               Func
115
116 | {
       UINT64
                   Address;
117
118
       EFI_STATUS Status;
119
120
121
       // Create PCI address map in terms of Bus, Device and Func
122
123
       Address = EFI_PCI_ADDRESS (Bus, Device, Func, 0);
124
125
       //
126
       // Read the Vendor ID register
127
       //
128
       Status = PciRootBridgeIo->Pci.Read (
129
                                       PciRootBridgeIo,
130
                                       EfiPciWidthUint32,
131
                                       Address,
132
                                       1,
133
                                       Pci
134
                                       );
135
       if (!EFI ERROR (Status) && (Pci->Hdr).VendorId != 0xffff) {
136
137
         // Read the entire config header for the device
138
139
140
         Status = PciRootBridgeIo->Pci.Read (
141
                                         PciRootBridgeIo,
                                         EfiPciWidthUint32,
142
143
144
                                         sizeof (PCI_TYPE00) / sizeof (UINT32),
145
                                         Pci
146
                                         );
147
148
         return EFI SUCCESS;
149
150
151
       return EFI_NOT_FOUND;
( • ) }
```

收起 へ

Al generated projects

登录复制

```
1 //OvmfPkg/HelloWorldPci/HelloWorldPci.inf
2
   [Defines]
3
    INF_VERSION
                                    = 0 \times 00010005
4
    BASE_NAME
                                    = HelloWorldPci
5
    FILE_GUID
                                    = 6987935E-ED34-44db-AE97-1FA5E4ED2116
6
     MODULE_TYPE
                                    = UEFI APPLICATION
7
     VERSION_STRING
                                     = 1.0
8
      ENTRY_POINT
                                     = UefiMain
9
10
11
   # This flag specifies whether HII resource section is generated into PE image.
12
13
     UEFI HII RESOURCE SECTION
14
15
16
   # The following information is for reference only and not required by the build tools.
17
```

```
1/
 18
        VALID ARCHITECTURES
     #
                                        = IA32 X64 IPF EBC
 19
     #
 20
twen
      [Sources]
twen
       HelloWorldPci.c
twen
twen
 25
      [Packages]
 26
       MdePkg/MdePkg.dec
 27
       MdeModulePkg/MdeModulePkg.dec
 28
       ShellPkg/ShellPkg.dec
 29
 30
      [LibraryClasses]
 31
       UefiApplicationEntryPoint
 32
       UefiShellCEntryLib
 33
       BaseLib
 34
       BaseMemoryLib
 35
       DebugLib
 36
       UefiBootServicesTableLib
 37
       MemoryAllocationLib
 38
       Hefil ib
 39
       UefiLib
 40
       PcdLib
-
```

收起 へ

L. Compile and generate EFI files & run

```
FS0:\> HelloWorldPci.efi' Successs
 S0:\> Open '\HelloWorldPci.efi' Success
FSOpen: Open '\HelloWorldPci.efi' Success
FSOpen: Open '\HelloWorldPci.efi' Success
SOpen: Open '\HelloWorldPci.efi' Success
[Security] 3rd party image[0] can be loaded after EndOfDxe: PciRoot(0x0)/Pci(0x1,0x2)/U
SB(0x0,0x0)/\HelloWorldPci.efi.
InstallProtocolInterface: 5B1B31A1-9562-11D2-8E3F-00A0C969723B 6C86040
Loading driver at 0x00006221000 EntryPoint=0x00006222A39 HelloWorldPci.efi
InstallProtocolInterface: BC62157E-3E33-4FEC-9920-2D3B36D750DF 6C86B98
ProtectUefiImageCommon - 0x6C86040
 - 0x0000000006221000 - 0x000000000003500
InstallProtocolInterface: 752F3136-4E16-4FDC-A22A-E5F46812F4CA 7E91328
CSDNlCall LocatePciRootBridgeIo successed.Find protocol!
CSDN] PciDeviceNo
                         Bus
                                 Dev
                                          Func | Vendor.Device.ClassCode
CSDN]
                                                   8086
CSDN1
                                                   8086
                                                           7000
CSDN]
                                                   8086
                                                           7010
                                                                    80
CSDN] 4
                         0
                                                   8086
                                                           7020
                                  1
CSDN] 5
                         0
                                  1
                                                   8086
                                                           7113
                                                                    0
CSDN1
                                                   1013
                                                           В8
                                                                    0
CSDN]
                         0
                                  3
                                          0
                                                   8086
                                                           100E
                                                                    0
Sopen: Open '\ Success
                                                                    CSDN @xiaopangzi313
```

3. Summary

JEFI The enumeration process in PCIe device mainly creates two Protocol, namely gPciRootBridgeIoProtocol and PciIoProtocol. mong them, gPciRootBridgeIoProtocol corresponds to a RootBridge (RootCompex), which implements RootBridge all PCIe read and write mplementations under. PciIoProtocol Corresponding to each enumerated device. If you want to UEFI access the device under PCIe, you an use these two Protocol.

E绝无效搬运!Alex重构流量引擎

见场拆解:TikTok→独立站 双引擎→ 技术人优先预约通道

about Us Careers Business Seeking Cooperation Coverage 2400-660- kefu@csdn.net Customer Service 8:30-22:00

Public Security Registration Number 11010502030143 Beijing ICP No. 19004658 Beijing Internet Publishing House [2020] No. 1039-165

Commercial website registration information Beijing Internet Illegal and Harmful Information Reporting Center Parental Control
Online 110 Alarm Service China Internet Reporting Center Chrome Store Download Account Management Specifications

广告

Copyright and Disclaimer Copyright Complaints Publication License Business license ©1999-2025 Beijing Innovation Lezhi Network Technology Co., Ltd.