# Decoding configuration space of a PCI device

Command:

# lspci -xx

WLAN1: -

----------

0000:01:00.0 Class 0280: Device 12be:bd31 (rev 01)

00: be 12 31 bd 00 00 10 00 01 00 80 02 00 00 00 00

10: 04 00 80 60 00 00 00 00 04 00 00 60 00 00 00 00

20: 00 00 00 00 00 00 00 00 00 00 00 00 be 12 31 bd

30: 00 00 00 00 48 00 00 00 00 00 00 00 00 01 00 00

40: 00 00 00 00 00 00 00 00 01 58 03 06 08 41 00 00

50: 00 00 00 00 00 00 00 00 05 68 8a 00 00 00 00 00

60: 00 00 00 00 00 00 00 00 09 a0 38 00 46 01 00 80

70: 00 00 10 18 00 00 00 00 00 00 00 00 00 00 00 00

80: 00 00 00 18 00 00 00 00 80 80 00 00 00 00 00 00

90: 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00

a0: 11 ac 3f 00 00 80 00 00 00 88 00 00 10 00 02 00

b0: 82 8f 00 10 10 2c 10 00 12 dc 46 00 40 00 12 10

c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

d0: 1f 08 08 00 00 04 00 00 06 00 00 00 01 00 00 00

e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Decoding: -

---------------

Note - lspci data is stored in Little endian.

Vendor ID - 0x12be

Product ID - 0xbd31

Status register (Offset 0x06) - 0x0010 (Bit-3 set to 1, indicates that device has a pointer to the capabilities list ).

Capability pointer (offset 0x34) - 0x00000048

Next capability (offset 0x48) - 0x06035801 ( Capability ID - 0x01, Next Capability pointer - 0x58)

Next capability (offset 0x58) - 0x008a6805 ( Capability ID - 0x05, Next Capability pointer - 0x68)

So, MSI Capability register value - 0x008a6805

MSI Message Control field - 0x008a (As Bit-0 is 0, so MSI is disable,

Bit-23 is 1, so 64bit addressing capable)

MSI Message address - 0x00000000

0001:01:00.0 Class 0280: Device 12be:bd31 (rev 01)

00: be 12 31 bd 00 00 10 00 01 00 80 02 00 00 00 00

10: 04 00 80 70 00 00 00 00 04 00 00 70 00 00 00 00

20: 00 00 00 00 00 00 00 00 00 00 00 00 be 12 31 bd

30: 00 00 00 00 48 00 00 00 00 00 00 00 00 01 00 00

40: 00 00 00 00 00 00 00 00 01 58 03 06 08 41 00 00

50: 00 00 00 00 00 00 00 00 05 68 8a 00 00 00 00 00

60: 00 00 00 00 00 00 00 00 09 a0 38 00 46 01 00 80

70: 00 00 10 18 00 00 00 00 00 00 00 00 00 00 00 00

80: 00 00 00 18 00 00 00 00 80 80 00 00 00 00 00 00

90: 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00

a0: 11 ac 3f 00 00 80 00 00 00 88 00 00 10 00 02 00

b0: 82 8f 00 10 10 2c 10 00 12 dc 46 00 40 00 12 10

c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

d0: 1f 08 08 00 00 04 00 00 06 00 00 00 01 00 00 00

e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Capability pointer (offset 0x34) - 0x00000048

Next capability (offset 0x48) - 0x06035801 ( Capability ID - 0x01, Next Capability pointer - 0x58)

Next capability (offset 0x58) - 0x008a6805 ( Capability ID - 0x05, Next Capability pointer - 0x68)

Note – Power Managemnet Capability ID – 0x01

MSI Capability ID – 0x05

MSIX Capability ID – 0x05

Note - After WLAN-1 driver is loaded, MSI interrupt address is assigned, as shown below.

0000:01:00.0 Class 0280: Device 12be:bd31 (rev 01)

00: be 12 31 bd 06 04 10 08 01 00 80 02 00 00 00 00

10: 04 00 80 60 00 00 00 00 04 00 00 60 00 00 00 00

20: 00 00 00 00 00 00 00 00 00 00 00 00 be 12 31 bd

30: 00 00 00 00 48 00 00 00 00 00 00 00 8d 01 00 00

40: 00 00 00 00 00 00 00 00 01 58 03 06 08 41 00 00

50: 00 00 00 00 00 00 00 00 05 68 8b 00 00 00 00 b6

60: 00 00 00 00 01 00 00 00 09 a0 38 00 46 01 00 80

70: 00 30 10 18 00 00 00 00 00 00 00 00 00 00 00 00

80: 00 30 00 18 00 00 00 00 80 80 00 00 00 00 00 00

90: 04 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00

a0: 11 ac 3f 00 00 80 00 00 00 88 00 00 10 00 02 00

b0: 82 8f 00 10 10 2c 11 00 12 dc 46 00 40 01 12 10

c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

d0: 1f 08 08 00 00 04 00 00 06 00 00 00 01 00 00 00

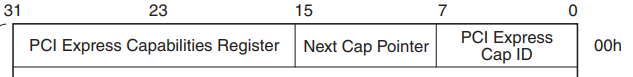
e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

We can see 0xb6 is added.

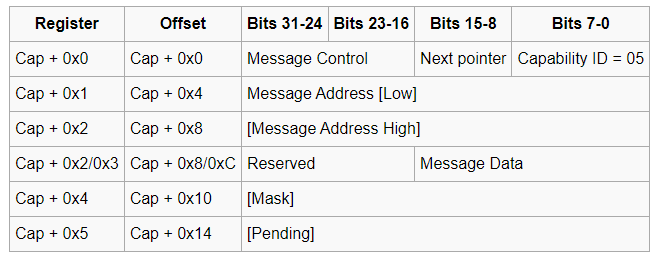
## Decoding Capability list

First, check that the device has a pointer to the capabilities list (status register bit 4 set to 1). Then, traverse the capabilities list. The low 8 bits of a capability register are the ID - 0x5 for MSI. The next 8 bits are the offset (in [PCI Configuration Space](https://wiki.osdev.org/PCI#Configuration_Space)) of the next capability.



First capability pointer of 1 byte is at offset 0x34. From this onwards, all capabilities are of 4 bytes each. The least significant byte gives “Capability ID” and next byte to LSB gives pointer to next capability regitser in configuration space.

The MSI capability is as follows:



## FCA\_R1 bridge device config space decoding: -

00:13.0 Class 0604: Device 8086:5ad8 (rev fd)

PCI Configuration header (PCI-to-PCI-Bridge, Header type - 0x01)

00: 86 80 d8 5a 07 04 10 00 fd 00 04 06 00 00 81 00

10: 00 00 00 00 00 00 00 00 00 01 01 00 30 30 00 20

20: 20 b3 30 b3 81 b3 91 b3 00 00 00 00 00 00 00 00

30: 00 00 00 00 40 00 00 00 00 00 00 00 16 00 00 00

40: 10 80 42 01 01 80 00 00 0f 00 10 00 12 00 71 03

50: c0 00 12 70 60 b2 14 00 28 10 40 00 08 00 00 00

60: 00 00 00 00 37 08 08 00 00 04 00 00 06 00 00 00

70: 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

80: 05 90 01 00 0c f0 e0 fe 92 41 00 00 00 00 00 00

90: 0d a0 00 00 00 00 00 00 00 00 00 00 00 00 00 00

a0: 01 00 03 c8 00 00 00 00 00 00 00 00 00 00 00 00

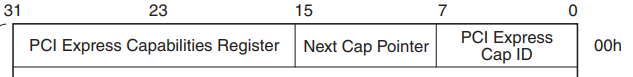
b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

d0: 77 00 00 00 42 1a 00 00 09 00 10 01 00 00 00 00

e0: 00 00 e3 00 00 00 00 00 06 00 01 00 00 00 00 00

f0: 50 01 00 00 00 04 00 0c 1c 0f 00 01 04 00 00 00



Capability pointer tracing: -

===============================

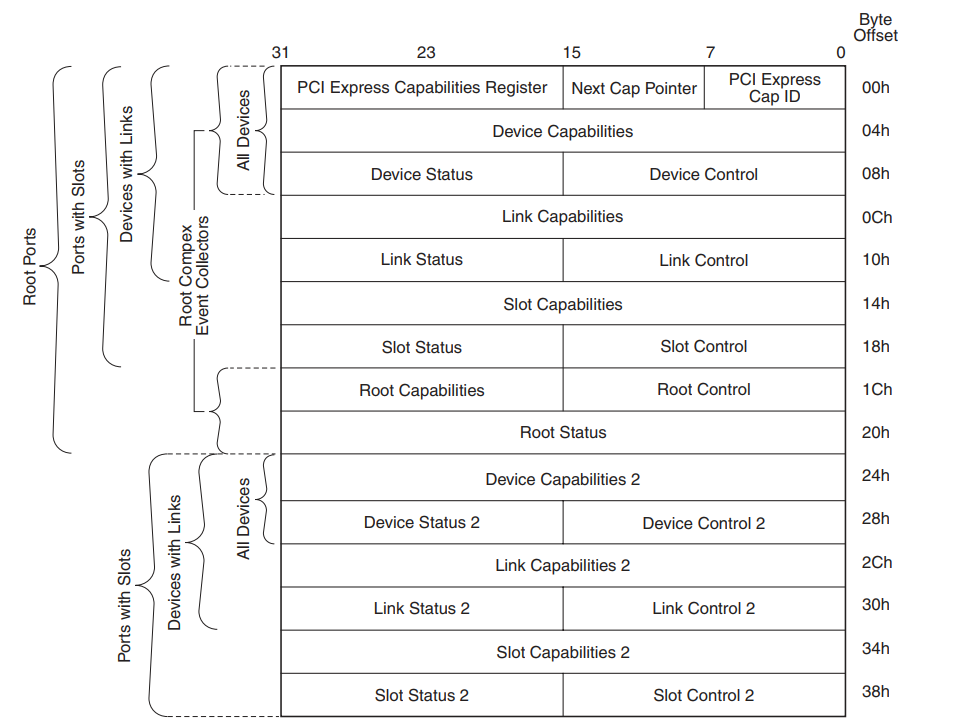
0x34 -> 0x40

0x40 -> 0x01 42 80 10 (Capability ID - 0x10) PCI Xpress

0x80 -> 0x00 01 90 05 (Capability ID - 0x05) MSI Capability

0x90 -> 0x00 00 a0 0d (Capability ID - 0x0d) PCI Bridge Subsystem Vendor ID

0xa0 -> 0xc8 03 00 01 (Capability ID - 0x01) Power Management Capability



0x40 + 0-> PCI Express Cap ID

Offset: 0x04 -> (PCI Express Device Capabilities Register)

00 00 80 01 [01 -> Max Payload size – 256 bytes]

Offset: 0x08 -> PCI Express Device Control and Status Register

Offset: 0x0C -> Link Capabilities Register

Offset: 0x10-> Link Control and Status Register

Offset: 0x14 -> Slot Capabilities Register

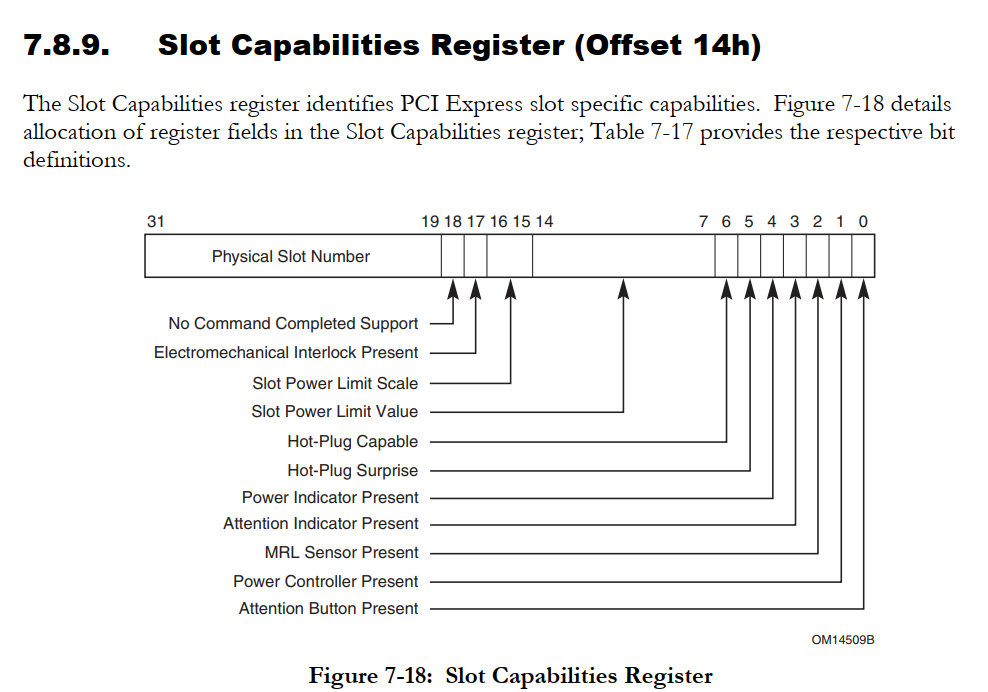
Address: 0x54 -> 0x00 14 b2 60 => Hot Plug Capable and Hot Plug Surprise

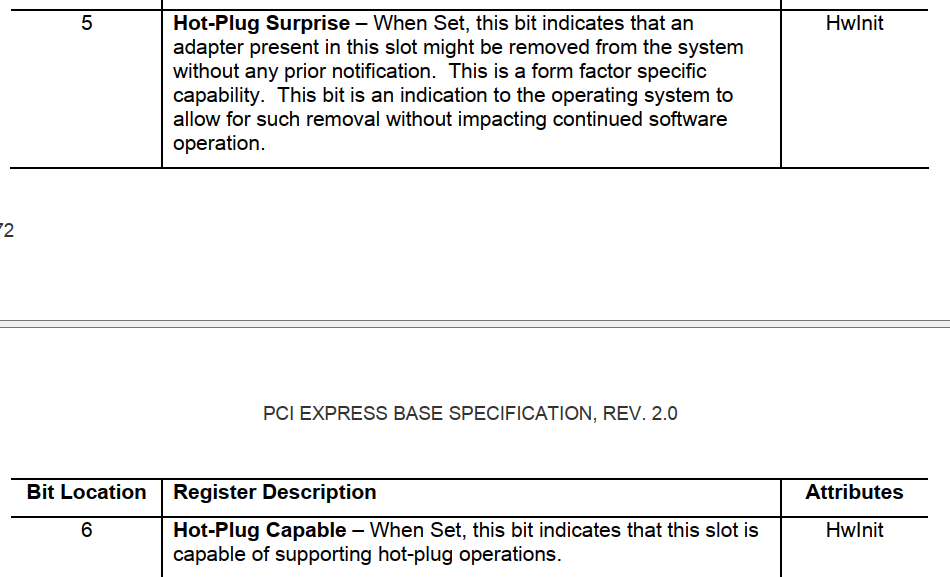
Physical Slot No = 2

Bit[6] = 1 -> Hot Plug Capable

Bit[5] = 1 -> Hot Plug Surprise

Bit[31:19] = Physical Slot No



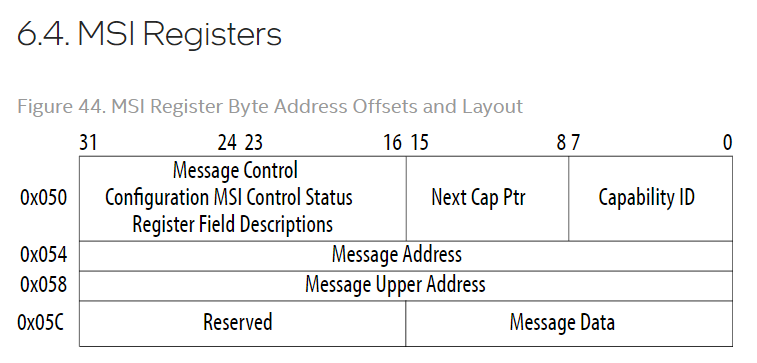


0x18 -> Slot Control and Status Register

0x1C -> Root Control and Capabilities Register

0x20 -> Root Status Register

**Note : - Changing MSI REGISTERS**

****

Follow this URL for MSI registers in detail –

<https://www.intel.com/content/www/us/en/docs/programmable/683686/20-4/msi-registers.html>

# ./setpci -s 00:13.0 VENDOR\_ID.W

8086

fca\_dv2:/data #

fca\_dv2:/data # ./setpci -s 00:13.0 VENDOR\_ID.W=8087

fca\_dv2:/data # ./setpci -s 00:13.0 VENDOR\_ID.W

8086

fca\_dv2:/data # ./setpci -s 00:13.0 84.W

f00c

fca\_dv2:/data # ./setpci -s 00:13.0 84.W=f00d

fca\_dv2:/data # ./setpci -s 00:13.0 84.W

f00c

fca\_dv2:/data # ./setpci -s 00:13.0 84.W=f00d

fca\_dv2:/data #

fca\_dv2:/data #

fca\_dv2:/data # ./setpci -s 00:13.0 84.W

f00c

fca\_dv2:/data # ./setpci -s 00:13.0 04.W

0407

fca\_dv2:/data # ./setpci -s 00:13.0 82.W

0001

fca\_dv2:/data # ./setpci -s 00:13.0 82.W=0000

fca\_dv2:/data # ./setpci -s 00:13.0 82.W

0000

fca\_dv2:/data # ./setpci -s 00:13.0 88.l

00004192

fca\_dv2:/data # ./setpci -s 00:13.0 88.l=4190

fca\_dv2:/data # ./setpci -s 00:13.0 88.l

00004190

fca\_dv2:/data #

## Reference: -

https://www.intel.com/content/www/us/en/docs/programmable/683686/20-4/pci-express-capability-structure.html

https://support.xilinx.com/s/article/1148199?language=en\_US

https://forums.developer.nvidia.com/t/i-unable-to-access-the-pci-registers-when-my-pci-is-connected-to-jetson-i-am-getting-these-errors/73474/8

http://ccrma.stanford.edu/planetccrma/man/man8/setpci.8.html

https://unix.stackexchange.com/questions/724953/why-setpci-failed

[Date – 28-08-2023]

# Decoding PCI config space especially for Power Mgmt register GM project: -

There is a Capability Pointer register at offset 0x34 of size 1 byte. The capability pointer register gives the offset for the first capability structure.

Each Capability structure must have a Capability ID assigned by the PCI-SIG.

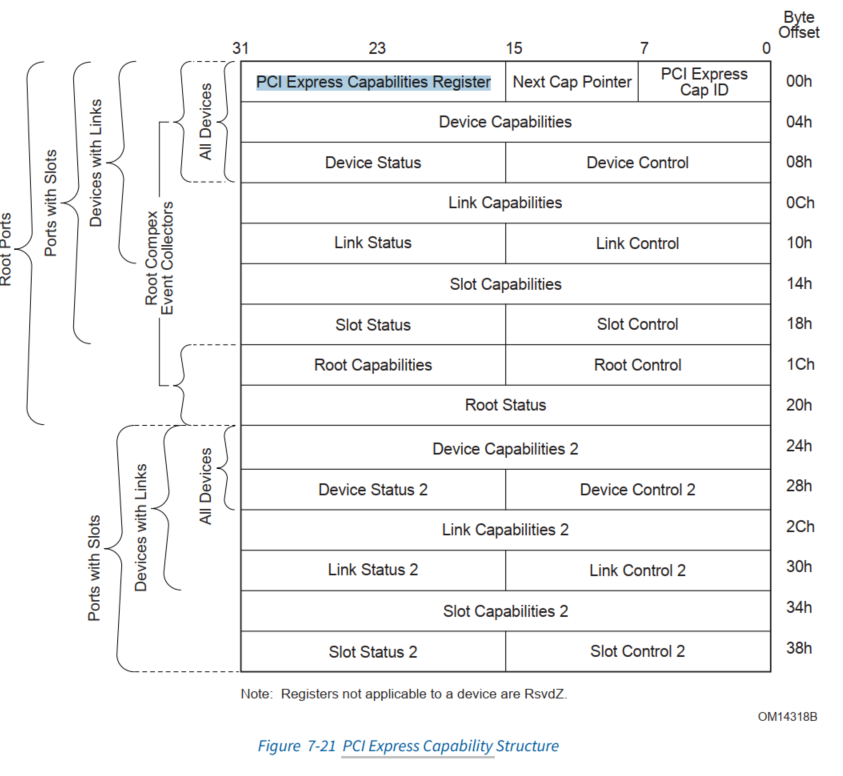
|  |  |
| --- | --- |
| **ID** | **Capability** |
| 00h | Null capability |
| 01h | PCI Power Management Interfa |
| 02h | AGP, Accelerated Graphics Port |
| 03h | VPD |
| 04h | Slot Identification |
| 05h | MSI |
| 06 | CompactPCI Hot Swap |
| 07 | PCI-X |
| 08 | HyperTransport |
| 09 | Vendor Specific |
| 0A | Debug port |
| 0B | CompactPCI central resource control |
| 0C | PCI Hot-Plug |
| 0D | PCI Bridge Subsystem Vendor ID |
| 0E | AGP 8x |
| 0F | Secure Device |
| 10 | PCI Express |
|  |  |

## Reference for Capability IDs in different Capability structure: -

<https://pcisig.com/sites/default/files/files/PCI_Code-ID_r_1_11__v24_Jan_2019.pdf>

## Various capability structures are mentioned below: -

1. PCI express capability structure



1. Power management capability structure

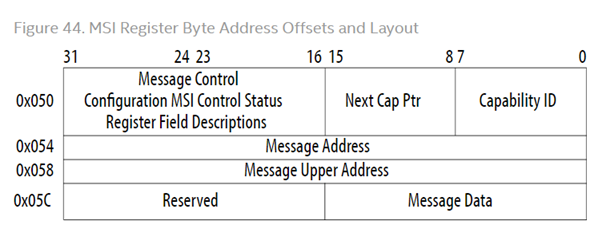
To decode Power Management capability structure, first using Capability Pointer (0x34) we have to first find Power Mgmt Cap. Structure using first out of 4 bytes to identify Capability ID as 0x01, we confirm it to be as “Power Management capability structure”. Then we use offset 4 to read PMCSR register.

A diagram of power management

Description automatically generated

Refer section - 7.5.2.1 (Page No – 712) for Power Mgmt Cap. structure and bit fields.

1. MSI capability structure



Example: -

02:00.0 Ethernet controller: Marvell Technology Group Ltd. Device 2b43 (rev 03)

[[2023-08-28 12:51:36.288] Kernel driver in use: wlan\_pcie

[2023-08-28 12:51:36.288] 00: 4b 1b 43 2b 06 04 10 00 03 00 00 02 00 00 80 00

[2023-08-28 12:51:36.288] 10: 0c 00 30 b3 00 00 00 00 0c 00 40 b3 00 00 00 00

[2023-08-28 12:51:36.288] 20: 0c 40 42 b3 00 00 00 00 00 00 00 00 4b 1b 43 2b

[2023-08-28 12:51:36.288] 30: 00 00 00 00 40 00 00 00 00 00 00 00 14 01 00 00

[2023-08-28 12:51:36.288] 40: 01 50 c3 db 08 00 00 00 00 00 00 00 00 00 00 00

[2023-08-28 12:51:36.288] 50: 05 70 8b 01 0c f0 e0 fe 00 00 00 00 d3 41 00 00

[2023-08-28 12:51:36.288] 60: fe ff ff ff 00 00 00 00 00 00 00 00 00 00 00 00

[2023-08-28 12:51:36.288] 70: 10 b0 02 00 c0 8f 90 15 10 28 19 00 12 cc 47 00

[2023-08-28 12:51:36.288] 80: 40 01 11 10 00 00 00 00 00 00 00 00 00 00 00 00

[2023-08-28 12:51:36.288] 90: 00 00 00 00 1f 08 0c 00 00 00 00 00 06 00 00 00

[2023-08-28 12:51:36.288] a0: 02 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00

[2023-08-28 12:51:36.297] b0: 11 00 1f 00 04 20 00 00 04 30 00 00 00 00 00 00

[2023-08-28 12:51:36.297] c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

[2023-08-28 12:51:36.297] d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

[2023-08-28 12:51:36.297] e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

[2023-08-28 12:51:36.297] f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

1. Capability pointer Register – 0x34h
2. 0x34 is a 1-byte regsiter, which points to 0x40.
3. 0x40 points to – “ db c3 50 01”

01 50 c3 db