BROADCOM BCM4350 CARDS UNDER HIGH SIERRA/MOJAVE/CATALINA

By Hervé, February 10, 2019 in Wireless and Bluetooth

Hervé

Brigadier General

Moderators

Admin

899

9151 posts

Posted February 10, 2019

Last update: ***06 Jan 2020***

Questions around this Broadcom BCM4350 chipset (in particular the Dell DW1820A) have resurfaced again so I digged into the matter since most people reported it did not work. Outside the model fitted to Apple MacBooks (subsystem id 106b:0131, rev. 05), Wikidevi lists a few cards cards for this chipset, including:

- Dell DW1820A (several models with different part numbers)
- Foxconn T77H649.00 (Lenovo part number 00JT494)
- Lite-on WCBN808B (Lenovo part number 00JT493)

All those cards carry PCI id 14e4:43a3 and normally offer high speed 802.11ac wireless + Bluetooth 4.1 services.

Broadcom BCM4350 chipset is supported since Yosemite 10.10 and its hardware id is listed in the Info.plist file of IO80211Family's PlugIn kext ***AirPortBrcm4360*** up to macOS Sierra 10.12, then ***AirPortBrcmNIC*** since macOS High Sierra 10.13:

```
        <key>Broadcom 802.11 PCI</k
        <dict>
            <key>CFBundleIdentifier
            <string>com.apple.drive
            <key>IOClass</key>
            <string>AirPort_BrcmNIC
            <key>IOMatchCategory</k
            <string>IODefaultMatchC
            <key>IONameMatch</key>
            <array>
                <string>pci14e4,43b
                <string>pci14e4,43a
                <string>pci14e4,43a
            </array>
            <key>IOProbeScore</key>
            <integer>1241</integer>
            <key>IOProviderClass</k
            <string>IOPCIDevice</st
            <key>TruePowerOff</key>
            <true/>
        </dict>
```

Edit: 06 Jan 2020

Removed the warning about DW1820/BCM4350 cards following tests confirming that disabling ASPM does fix the stability issues previously

encountered by most users.

1 1

Latitude 7490, BIOS 1.13.1, Core i7-8650U 1.9GHz, 16Go DDR4-2400, Intel UHD620, 1920x1080, Moj 10.14.6/Cat 10.15.3
Latitude E6230, BIOS A20, Core i7-3540M 3.0GHz,

Hervé
Brigadier General

Moderators
Admin
899
9151 posts

Posted April 8, 2019

Report post

***Last update:* 28 Jan 2020**

Dell DW1820A:

Ok

Ok Ok

DW1820A is a pretty good combo card providing 2.4/5GHz 802.11ac wireless at 867Mbps + Bluetooth 4.1. As stated at Wikidevi, there appears to be a few different models of the card. Whilst they all carry the same Broadcom id 14e4:43a3, they each carry a different subsystem id/part #:

- 1028:0021 (part # CN-0VW3T3) -> 100% Ok with ASPM disabled (originally tested 100% Ok on my Latitude 7490 without ASPM tuning)
- 1028:0021 (part # CN-096JNT) -> 100% Ok with APSM disabled
- 1028:0022 (part # CN-096JNT) -> 100% Ok with APSM disabled
- 1028:0023 (part # CN-0VW3T3) -> 100% Ok with APSM disabled
- 1028:0023 (part # CN-08PKF4) -> 100% Ok with APSM disabled

I acquired a Dell DW1820A a few months ago and was able to play with that card on a Latitude 7490 laptop (fitted with an M.2 2230 Key A+E WLAN slot) that I targeted for Mojave. The model I

received was CN-0VW3T3. Before it was later established that not all DW1820A were equal, I purchased a 2nd one and, as it turned out to be, was lucky to receive another 0VW3T3 model.

⚠ The 1st thing I want to report is that I encountered difficulties booting my Mojave USB installer and installing macOS with the card plugged in. **I had to disable wireless in BIOS to be able to install Mojave**. Once it was installed, booting Mojave with wireless enabled in BIOS would cause quite severe lag and performance degradation once at the desktop, as if the card just clogged up CPU ressources. This was because I had not applied any particular tuning for the card and, of course, this was resolved once I implemented the necessary fix detailed below. ⚠

Searching through the Web for that DW1820A/BCM4350, I came accross a few forum posts/threads that mentionned:

1. rolling back the Yosemite IO80211Family kext to get the card to work, although with instability and regular KPs
2. removing AirPortBrcmNIC plugin kext from IO80211Family kext, patching AirPortBrcm4360 plugin kext with the id of the DW1820A and installing AirportBrcmFixup kext with a couple of parameters (Credits to Hugotai, cf. his post @Voldemort's place, 2nd Dec 2018)

Whilst I did not really contemplate doing the 1st thing, I did envisage the 2nd one and started to look at the differences between Yosemite's version of IO80211Family kext and Mojave's. The main difference I had already noticed was that device id 14e4:43a3 was handled by AirPortBrcm4360 up to Sierra 10.12 and by AirPortBrcmNIC since High Sierra 10.13. Building on Hugotai's success, I seeked to work out an easier solution that would

not require kext removal and Info.plist patching but, instead, something that could be implemented through hardware properties injection, either through DSDT patching or Clover configuration.

Once Hugotai's solution was verified and confirmed, I worked out the following Clover-based solution for HighSierra/Mojave:

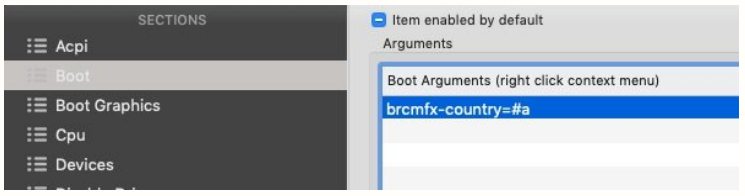
1. identify the IOReg/ACPI device to which the DW1820A card is attached (use **IORegistryExplorer** app to that effect). Eg: RP0*n*@*xx*,*yy*->PXSX@0.
2. in the absence of individual ACPI device entry under the PCI bridge for the card, select "FixAirport" ACPI Fix in Clover. That'll create a device "ARPT" @0 under the bridge and that's what you'll inject properties to. This may also require to select "AddDTGP" ACPI Clover fix if your DSDT does not possess any DTGP method. Use **Clover Configurator** app to that effect.
3. inject the following properties either in DSDT or through Clover (latter recommended):
    - compatibility of the card with Broadcom chips 14e4:4331 or 14e4:4353 that are handled by *IO80211Family*'s PlugIn kext *AirPortBrcm4360*
    - ASPM disabling (required for most cards to avoid CPU clogging and system freeze)
    - optionally, add SysProfiler's cosmetic info such as PCIe Slot, card's make and model, etc.

and that's it ! Nothing to do to IO80211Family kext or its PlugIns which all remain untouched/unmodified/full vanilla in /S/L/E. It really could not be simpler...

NB: if your card's country code absolutely requires

to be changed, say for full 5GHz/80MHz operations, add the following steps:

- install *AirportBrcmFixup* kext in /L/E + repair permissions + rebuild your cache or inject it through Clover's EFI/CLOVER/kexts/Other
- add boot argument ***brcmfx-country=XX*** (where XX is the target value, eg: US, FR, #a, etc.) to the ***Boot*** section of your Clover config

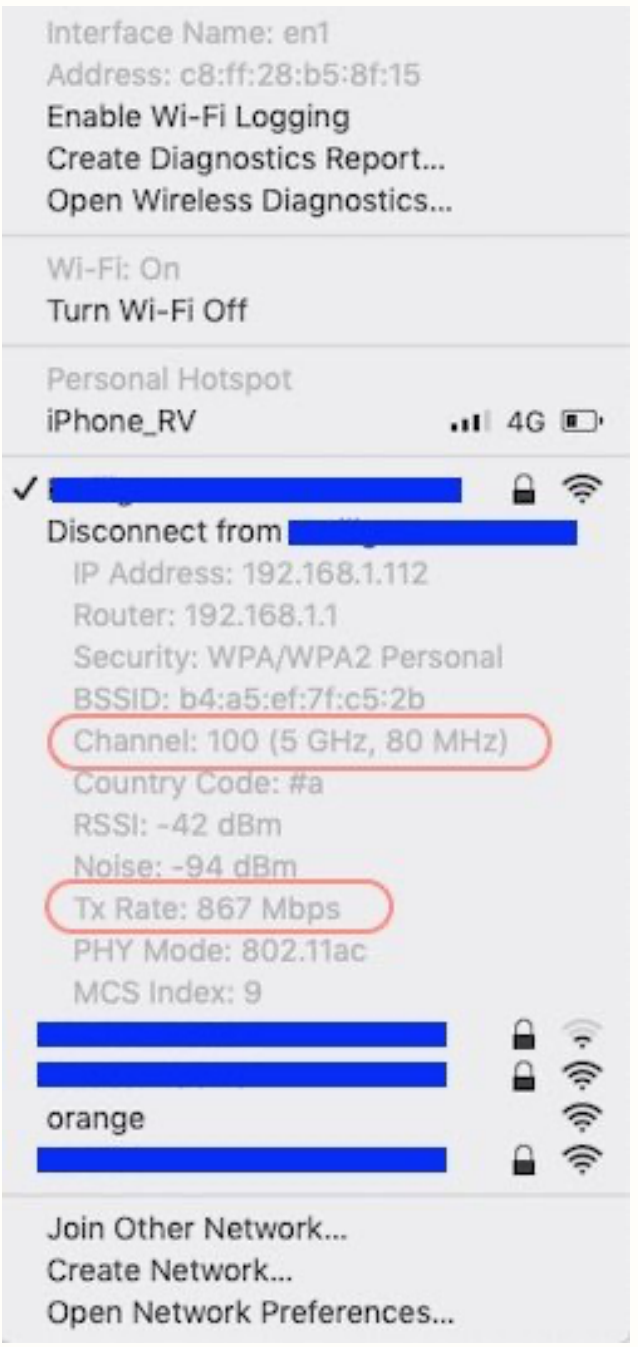


but, ***beware***, I found that using *AirportBrcmFixup* (v1.1.9 at time of writing) with country code other than default's US setting of the card impacted my wireless performance (fluctuating and reduced RX/TX rate). For instance,

- with country code set to FR (for France), I would not connect at 80MHz on my 5GHz Wireless network, only to 40MHz which resulted in a local wireless connection limited to 400Mbps (vs. 867Mbps) and reduced my overall Internet RX rate by about 60% and TX rate by about 10%:

- with country code set to #a (to get full

80MHz connection), I could not obtain DFS channel mode to my local router and I noticed fluctuating/unstable RX/TX rates:



- without AirportBrmfixup, my wireless connection runs solid at 867Mbps and SpeedTest returns a solid TX and RX rates at 300Mbps (i.e. the maximum speed of my Internet connection).

Properties injection through DSDT:

On my laptop, the DW1820A was found attached to device RP03.PXSX located at IO address 0x001C0002, i.e. 1C,2.

The DSDT patch required to inject properties could look like this (devices names will differ from one computer to another of course!):

```
Device (RP03)      // PCIe Root Bri
{
    name (_ADR, 0x001C0002)
    [...]
    [...]
    [...]
    Device (PXSX)       // DW1820A
    {
        Name (_ADR, Zero)
        [...]
        [...]
        [...]
        Method (_DSM, 4, NotSeriali
        {
            If (LEqual (Arg2, Zero)
            {
                Return (Buffer (One
                {
                    0x03
                })
            }
            Return (Package ()
            {
                "AAPL,slot-name",
                Buffer ()
                {
                    "WLAN"
                },
                "device_type",
                Buffer ()
                {
                    "Airport Extrem
```

```
                },
                "name",
                Buffer ()
                {
                    "Airport"
                },
                "model",
                Buffer ()
                {
                    "Dell DW1820A 8
                },
                "compatible",
                Buffer ()
                {
                    "pci14e4,4331"
                },
                "pci-aspm-default",
                Buffer (One)
                {
                    0x00
                }
            })
        }
    }
}
```

Properties injection through Clover config:

An easier alternative is to inject those properties in Clover via **Clover Configurator** app. This can be done within the ***Devices*** section by injecting the desired properties in the ***Properties*** sub-section:

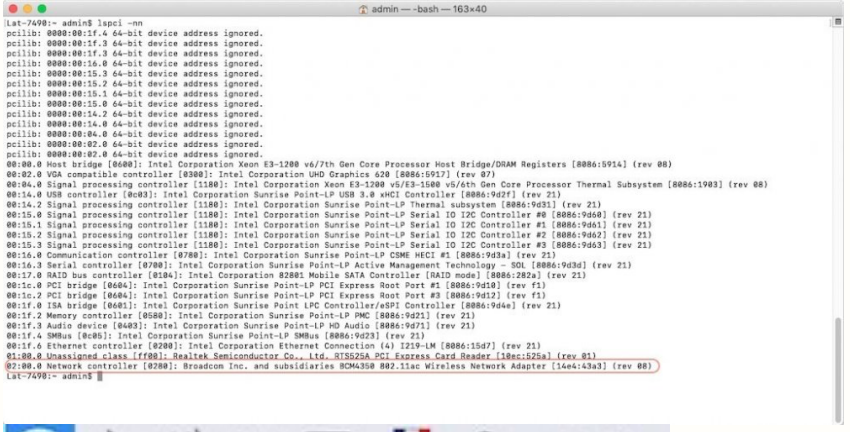
- In the left part, add the PCIe address of the targeted device in the form PciRoot(0x0)/Pci(<root device address>)/Pci(<actual device address>)
- In the right part, add the above properties in single lines and with the right types (String, Data, Number)

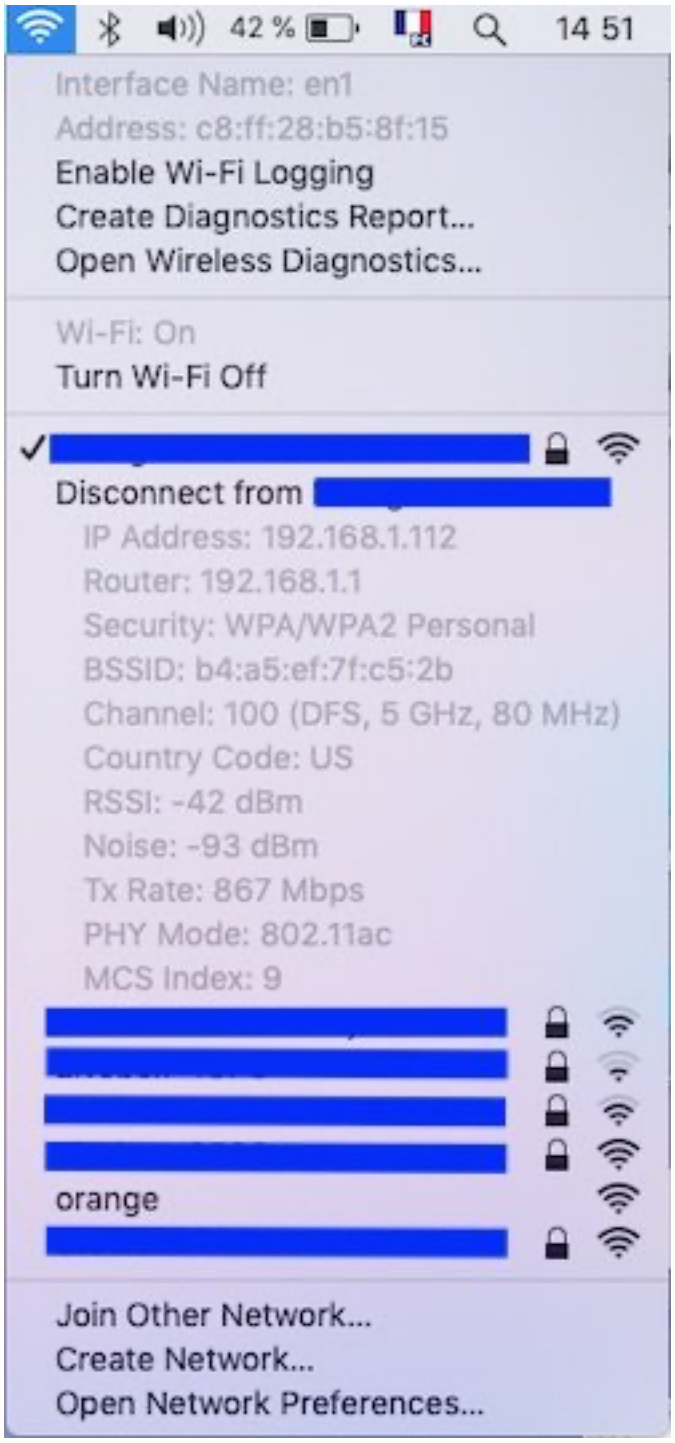
For instance, in the case of my laptop, the target device will be ***PciRoot (0x0)/Pci(0x1C,0x02)/Pci(0x0,0x0)*** for ***PCI0@0->RP03@1C,2->PXSX@0***. Then, properties will be injected as lines of keys of 3 x possible types: strings, hex data blocks or numbers. For instance, to declare ***compatibility with 14e4:4353***, the line will consist of Property Key set to ***compatible***, Key Value set to ***pci14e4,4353*** and Key Type set to ***STRING***. The complete properties injection will be:

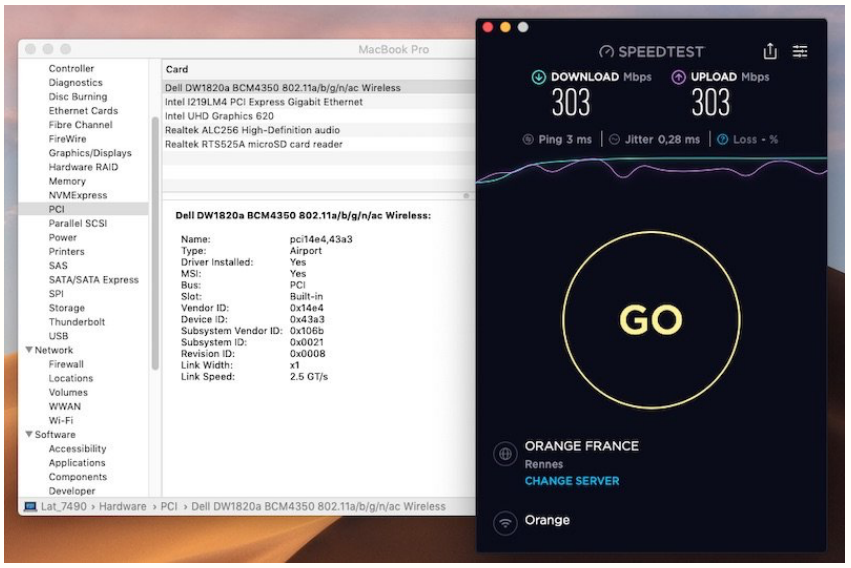
Device = PciRoot(0x0)/Pci(0x1c,0x02)/Pci(0x0,0x0)
Key = **compatible** | **Value** = **pci14e4,4353** | **Type** = **STRING**
Key = **pci-aspm-default** | **Value** = **0** | **Type** = **NUMBER**
Key = **AAPL,slot-name** | **Value** = **WLAN** | **Type** = **STRING** (optional)
Key = **device_type** | **Value** = **Airport Extreme** | **Type** = **STRING** (optional)
Key = **name** | **Value** = **Airport** | **Type** = **STRING** (optional)
Key = **model** | **Value** = **Dell DW1820 (BCM4350) 802.11ac wireless** | **Type** = **STRING** (optional)

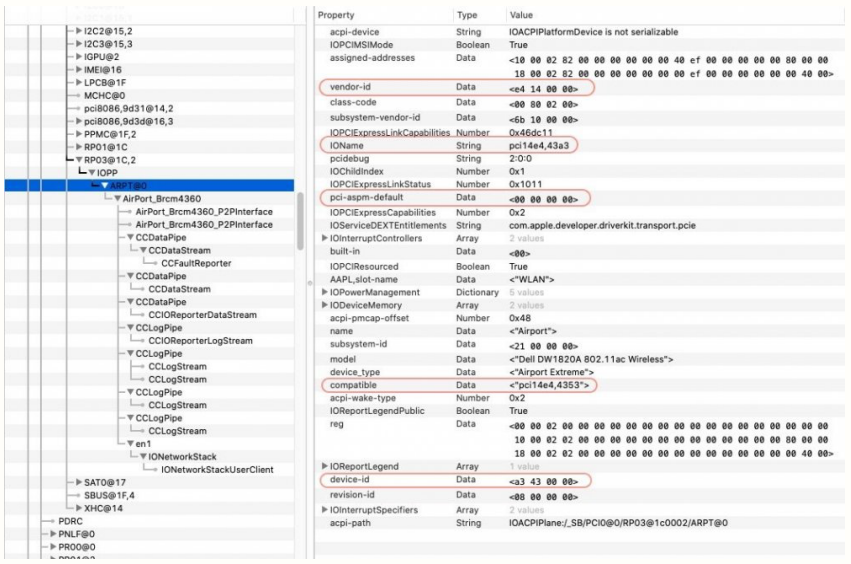
Once the device properties are injected in Clover or DSDT, there's nothing left to do but reboot the computer. The DW1820A card will then be fully active and capable to connect to 2.4/5GHz networks at full speed.

Results:


admin — -bash — 163×40
Lat-7490:~ admin$ lspci -nn
pcilib: 0000:00:1f.4 64-bit device address ignored.
pcilib: 0000:00:1f.3 64-bit device address ignored.
pcilib: 0000:00:1f.3 64-bit device address ignored.
pcilib: 0000:00:16.0 64-bit device address ignored.
pcilib: 0000:00:15.3 64-bit device address ignored.
pcilib: 0000:00:15.2 64-bit device address ignored.
pcilib: 0000:00:15.1 64-bit device address ignored.
pcilib: 0000:00:15.0 64-bit device address ignored.
pcilib: 0000:00:14.2 64-bit device address ignored.
pcilib: 0000:00:14.0 64-bit device address ignored.
pcilib: 0000:00:04.0 64-bit device address ignored.
pcilib: 0000:00:02.0 64-bit device address ignored.
pcilib: 0000:00:02.0 64-bit device address ignored.
00:00.0 Host bridge [0600]: Intel Corporation Xeon E3-1200 v6/7th Gen Core Processor Host Bridge/DRAM Registers [8086:5914] (rev 08)
00:02.0 VGA compatible controller [0300]: Intel Corporation UHD Graphics 620 [8086:5917] (rev 07)
00:04.0 Signal processing controller [1180]: Intel Corporation Xeon E3-1200 v5/E3-1500 v5/6th Gen Core Processor Thermal Subsystem [8086:1903] (rev 08)
00:14.0 USB controller [0c03]: Intel Corporation Sunrise Point-LP USB 3.0 xHCI Controller [8086:9d2f] (rev 21)
00:14.2 Signal processing controller [1180]: Intel Corporation Sunrise Point-LP Thermal subsystem [8086:9d31] (rev 21)
00:15.0 Signal processing controller [1180]: Intel Corporation Sunrise Point-LP Serial IO I2C Controller #0 [8086:9d60] (rev 21)
00:15.1 Signal processing controller [1180]: Intel Corporation Sunrise Point-LP Serial IO I2C Controller #1 [8086:9d61] (rev 21)
00:15.2 Signal processing controller [1180]: Intel Corporation Sunrise Point-LP Serial IO I2C Controller #2 [8086:9d62] (rev 21)
00:15.3 Signal processing controller [1180]: Intel Corporation Sunrise Point-LP Serial IO I2C Controller #3 [8086:9d63] (rev 21)
00:16.0 Communication controller [0780]: Intel Corporation Sunrise Point-LP CSME HECI #1 [8086:9d3a] (rev 21)
00:16.3 Serial controller [0700]: Intel Corporation Sunrise Point-LP Active Management Technology - SOL [8086:9d3d] (rev 21)
00:17.0 RAID bus controller [0104]: Intel Corporation 82801 Mobile SATA Controller [RAID mode] [8086:282a] (rev 21)
00:1c.0 PCI bridge [0604]: Intel Corporation Sunrise Point-LP PCI Express Root Port #1 [8086:9d10] (rev f1)
00:1c.2 PCI bridge [0604]: Intel Corporation Sunrise Point-LP PCI Express Root Port #3 [8086:9d12] (rev f1)
00:1f.0 ISA bridge [0601]: Intel Corporation Sunrise Point LPC Controller/eSPI Controller [8086:9d4e] (rev 21)
00:1f.2 Memory controller [0580]: Intel Corporation Sunrise Point-LP PMC [8086:9d21] (rev 21)
00:1f.3 Audio device [0403]: Intel Corporation Sunrise Point-LP HD Audio [8086:9d71] (rev 21)
00:1f.4 SMBus [0c05]: Intel Corporation Sunrise Point-LP SMBus [8086:9d23] (rev 21)
00:1f.6 Ethernet controller [0200]: Intel Corporation Ethernet Connection (4) I219-LM [8086:15d7] (rev 21)
01:00.0 Unassigned class [ff00]: Realtek Semiconductor Co., Ltd. RTS525A PCI Express Card Reader [10ec:525a] (rev 01)
02:00.0 Network controller [0280]: Broadcom Inc. and subsidiaries BCM4350 802.11ac Wireless Network Adapter [14e4:43a3] (rev 08)
Lat-7490:~ admin$


42 %
14 51
Interface Name: en1
Address: c8:ff:28:b5:8f:15
Enable Wi-Fi Logging
Create Diagnostics Report...
Open Wireless Diagnostics...
Wi-Fi: On
Turn Wi-Fi Off
Disconnect from
IP Address: 192.168.1.112
Router: 192.168.1.1
Security: WPA/WPA2 Personal
BSSID: b4:a5:ef:7f:c5:2b
Channel: 100 (DFS, 5 GHz, 80 MHz)
Country Code: US
RSSI: -42 dBm
Noise: -93 dBm
Tx Rate: 867 Mbps
PHY Mode: 802.11ac
MCS Index: 9
orange
Join Other Network...
Create Network...
Open Network Preferences...


MacBook Pro
Controller
Diagnostics
Disc Burning
Ethernet Cards
Fibre Channel
FireWire
Graphics/Displays
Hardware RAID
Memory
NVMExpress
PCI
Parallel SCSI
Power
Printers
SAS
SATA/SATA Express
SPI
Storage
Thunderbolt
USB
Network
Firewall
Locations
Volumes
WWAN
Wi-Fi
Software
Accessibility
Applications
Components
Developer
Card
Dell DW1820a BCM4350 802.11a/b/g/n/ac Wireless
Intel I219LM4 PCI Express Gigabit Ethernet
Intel UHD Graphics 620
Realtek ALC256 High-Definition audio
Realtek RTS525A microSD card reader
Dell DW1820a BCM4350 802.11a/b/g/n/ac Wireless:
Name: pci14e4,43a3
Type: Airport
Driver Installed: Yes
MSI: Yes
Bus: PCI
Slot: Built-in
Vendor ID: 0x14e4
Device ID: 0x43a3
Subsystem Vendor ID: 0x106b
Subsystem ID: 0x0021
Revision ID: 0x0008
Link Width: x1
Link Speed: 2.5 GT/s
Lat_7490 › Hardware › PCI › Dell DW1820a BCM4350 802.11a/b/g/n/ac Wireless
SPEEDTEST
DOWNLOAD Mbps
UPLOAD Mbps
303
303
Ping 3 ms
Jitter 0,28 ms
Loss - %
GO
ORANGE FRANCE
Rennes
CHANGE SERVER
Orange



On the Bluetooth side, once the usual Rehabman's Broadcom firmware patching kexts are installed (*BrcmFirmwareRepo* + *BrcmPatchRAM2*), the BT4.1 module will be fully operational and capable of supporting AirDrop and Handoff!

Links:

- Clover Configurator app
- Rehabman's Broadcom Firmware patching kexts
- Acidanthera AirportBrcmFixup kext

- - - - - - - - - -

Edit: 06 Jan 2020

Updated with freshly found ASPM disabling property injection that fixes all outstanding stability issues. DW1820A #096JNT and Foxconn T77H649 now tested 100% Ok. DW1820A #08PKF4 should be just the same.

- - - - - - - - - -

Edit: 28 Jan 2020

Finally got to test a DW1820A #08PKF4 and it's also 100% Ok with ASPM disabled.

6

Latitude 7490, BIOS 1.13.1, Core i7-8650U 1.9GHz, 16Go DDR4-2400, Intel UHD620, 1920x1080, Moj 10.14.6/Cat 10.15.3

Latitude E6230, BIOS A20, Core i7-3540M 3.0GHz,

Hervé

Brigadier General

Moderators

Admin

899

9151 posts

Posted May 3, 2019

Report post

Last Update: 06 Jan 2020

Foxconn T77H649.00:

Ok

Foxconn T77H649.00 appears mostly offered with Lenovo laptops and carries Lenovo FRU (i.e. part #) 00JT494. The cards carries the usual Broadcom PCI id 14e4:43a3 and subsystem id 17aa:075a.

- - - - - - - - - -

Edit: 06 Jan 2019

This Foxconn T77H649 actually works perfectly with ASPM disabled through additional Clover property injection under the associated device address:

Key = pci-aspm-default

| Value = 0 **| Type =** NUMBER

⚠ ASPM disabling appears mandatory with this

card to avoid CPU overloading and system freeze after a few minutes.

Latitude 7490, BIOS 1.13.1, Core i7-8650U 1.9GHz, 16Go DDR4-2400, Intel UHD620, 1920x1080, Moj 10.14.6/Cat 10.15.3
Latitude E6230, BIOS A20, Core i7-3540M 3.0GHz,

Hervé
Brigadier General

Moderators
Admin
899
9151 posts

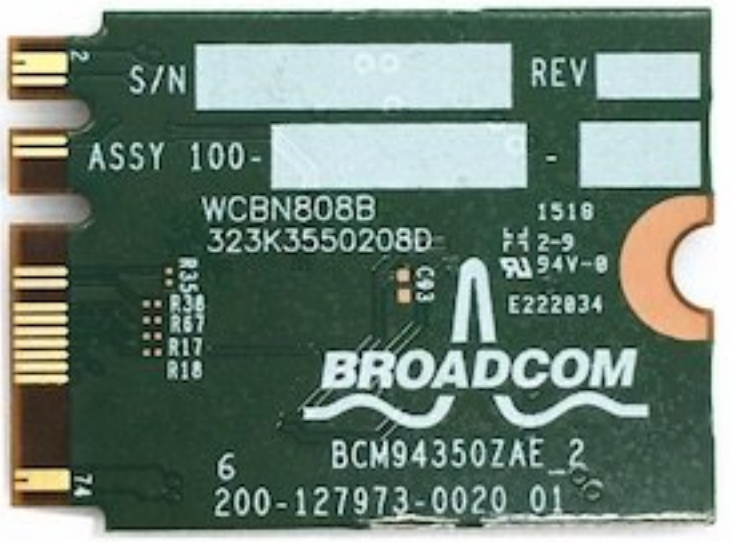
Posted May 3, 2019

Report post

Lite-On WCBN808B:

According to Wikidevi, DW1820A are manufactured by Lite-On with part # WCBN808B. I was not able to verify this for certain and could not find any specific info for a Lite-On WCBN808B card but this number is printed at the back of my Dell DW1820A 0VW3T3 cards.

I also found a Lenovo card (FRU 00JT493) carrying that WCBN808B reference on its front label. That card is definitely based on BCM4350 chipset. Afaik, it's not been tested yet but I'll try to get one too.



Untested

So, as it turns out so far, WCBN808B reference is printed on:

- Dell's CN-0VW3T3
- Dell's CN-096JNT
- Dell's CN-08PKF4
- Lenovo's 00JT493

It is not printed on:

- Foxconn's T77H649

Latitude 7490, BIOS 1.13.1, Core i7-8650U 1.9GHz, 16Go DDR4-2400, Intel UHD620, 1920x1080, Moj 10.14.6/Cat 10.15.3
Latitude E6230, BIOS A20, Core i7-3540M 3.0GHz,

Hervé
Brigadier General

Posted January 6

Report post

I was contacted by member @plastikman about a potential universal fix for the troublesome DW1820a cards. Hugotai posted again at Voldemort's place early december 2019 about a

Moderators

Admin

899

9151 posts

property injection that apparently fixed the stability issues encountered with so many DW1820a on so many laptops. Said property is **pci-aspm-default** and the value to inject is **0**.

Hugotai gave no explanation or description whatsoever so I cannot say whether the solution comes from him directly or not. A quick search on the Net revealed to me that ASPM means Active-State Power Management and relates to PCIe devices. Basically it's a protocol that offers dynamic power management through different possible states ranging from idle to powersave or performance. I also understand there is a default state which usually means power is controlled by BIOS, not the OS/kernel. This is totally unknown territory until I read more about it but setting ASPM default state to 0 probably means ASPM is disabled. On reading a little further, I came across writings that mentioned potential conflicts between idle timers or that ASPM compliance varied by device. So, the troubles experienced by so many people may indeed come from that.

Anyway, I tried the property injection posted by Hugotai on the other 2 x DW1820a/BCM4350 cards I still possess and that did not properly work in my Latitude 7490: the 096JNT model (subsystem 1028:0021) and the T77H649 (subsystem 17aa:075a). Results were as follows:
096JNT:

1. Without injection of ASPM property
    - CPU load up to the roof after a few minutes, system freeze as expected and experienced before.
    - IOReg reveals **pci-aspm-default** parameter natively set to **0x102** under the wireless card's ACPI device.
2. With injection of ASPM defaut set to 0
    - All Ok, was able to use the laptop without any issue, performance

degradation of wifi problems for well over 1hr.
- IOReg confirms **pci-aspm-default** parameter set to **0** under the wireless card's ACPI device.

T77H649:

1. Without injection of ASPM property
    - CPU load up to the roof after a few minutes, system freeze as expected and experienced before.
    - IOReg reveals **pci-aspm-default** parameter natively set to **0x102** under the wireless card's ACPI device.
2. With injection of ASPM defaut set to 0
    - All Ok, was able to use the laptop without any issue, performance degradation of wifi problems for well over 1hr.
    - IOReg confirms **pci-aspm-default** parameter set to **0** under the wireless card's ACPI device.

I remind everyone that all I use is the Clover property injection detailed in post #2, in order to declare compatibility with Broadcom chips 14e4:4331 or 14e4:4353; no AirportBrcmFixup plugin kext, no bcrmfx-driver boot argument and no pin masking.

So, it would seem we now have a definitive solution for DW1820A/BCM4350-based cards.

I've updated the above posts to reflect on the good news.

All credits to Hugotai and/or whoever identified this as a fix.

3

Latitude 7490, BIOS 1.13.1, Core i7-8650U 1.9GHz, 16Go DDR4-2400, Intel UHD620, 1920x1080, Moj 10.14.6/Cat 10.15.3
Latitude E6230, BIOS A20, Core i7-3540M 3.0GHz,

Hervé
Brigadier General

Moderators
Admin
899
9151 posts

Posted January 28

Report post

Disabling ASPM does not prevent from injecting compatibility property to bypass AirPortBrcmNIC; whilst running off the kext is a lot better than before, it still causes occasional glitches/freezes of the wireless service. So stick to the guidance posted above.

1

Latitude 7490, BIOS 1.13.1, Core i7-8650U 1.9GHz, 16Go DDR4-2400, Intel UHD620, 1920x1080, Moj 10.14.6/Cat 10.15.3
Latitude E6230, BIOS A20, Core i7-3540M 3.0GHz,

S

⚠ This topic is now closed to further replies.

GO TO TOPIC LISTING

NEXT UNREAD TOPIC

Home > Forums >
Technical Information, Projects and other Research & Development, >

Unread Content
Mark site read