```
C2 06           USHORT usStructureSize;
0B              UCHAR  ucTableFormatRevision;   /*Change it when the Parser is not backward compatible
*/
00              UCHAR  ucTableContentRevision;  /*Change it only when the table needs to change but the
firmware */

02      UCHAR  ucTableRevision;
C2 06           USHORT usTableSize;
34 08 00 00     ULONG  ulGoldenPPID;
A4 35 00 00     ULONG  ulGoldenRevision;
7C 00           USHORT usFormatID;

09 00 00 00     ULONG  ulPlatformCaps;

1A      UCHAR  ucThermalControllerType;

FA 00           USHORT usSmallPowerLimit1;
FA 00           USHORT usSmallPowerLimit2;
FA 00           USHORT usBoostPowerLimit;
00 00           USHORT usODTurboPowerLimit;
00 00           USHORT usODPowerSavePowerLimit;
76 00           USHORT usSoftwareShutdownTemp;

01              UCHAR ucTableRevision;
0B 00 00 00     ULONG PowerSavingClockCount;              // Count of PowerSavingClock Mode
        ULONG PowerSavingClockMax  [16];             // PowerSavingClock Mode Clock Maximum array In
MHz
09 07 00 00
6E 04 00 00
CC 03 00 00
CC 03 00 00
CC 03 00 00
E8 03 00 00
C9 04 00 00
6E 04 00 00
6E 04 00 00
34 04 00 00
2A 03 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
        ULONG PowerSavingClockMin  [16];             // PowerSavingClock Mode Clock Minimum array In
MHz
BC 02 00 00
66 01 00 00
36 01 00 00
36 01 00 00
36 01 00 00
5E 01 00 00
26 02 00 00
66 01 00 00
66 01 00 00
93 00 00 00
0E 01 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00

01              UCHAR ucODTableRevision;
0E 00 00 00     ULONG ODFeatureCount;
        UCHAR ODFeatureCapabilities [32];   //OD feature support flags
01
01
01
01
01
01
01
01
```

```
01
00
01
01
01
01
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
00
1D 00 00 00     ULONG ODSettingCount;
        ULONG ODSettingsMax [32];           //Upper Limit for each OD Setting
98 08 00 00
98 08 00 00
98 08 00 00
C2 04 00 00
98 08 00 00
C2 04 00 00
98 08 00 00
C2 04 00 00
B0 04 00 00
14 00 00 00
0A 0F 00 00
0A 0F 00 00
5F 00 00 00
6E 00 00 00
02 00 00 00
00 00 00 00
01 00 00 00
01 00 00 00
01 00 00 00
5F 00 00 00
64 00 00 00
5F 00 00 00
64 00 00 00
5F 00 00 00
64 00 00 00
5F 00 00 00
64 00 00 00
5F 00 00 00
64 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
        ULONG ODSettingsMin [32];           //Lower Limit for each OD Setting
28 03 00 00
28 03 00 00
28 03 00 00
E2 02 00 00
28 03 00 00
E2 02 00 00
28 03 00 00
E2 02 00 00
5E 01 00 00
14 00 00 00
C2 01 00 00
C2 01 00 00
19 00 00 00
32 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
```

```
00 00 00 00
00 00 00 00
19 00 00 00
14 00 00 00
19 00 00 00
14 00 00 00
19 00 00 00
14 00 00 00
19 00 00 00
14 00 00 00
19 00 00 00
14 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
                USHORT usReserve[5];
00 00
00 00
00 00
00 00
00 00

03 00 00 00     uint32_t Version;
                uint32_t FeaturesToRun[2];
FF EF F4 39
00 00 00 00

FA 00           uint16_t SocketPowerLimitAc0;
00 00           uint16_t SocketPowerLimitAc0Tau;
00 00           uint16_t SocketPowerLimitAc1;
00 00           uint16_t SocketPowerLimitAc1Tau;
00 00           uint16_t SocketPowerLimitAc2;
00 00           uint16_t SocketPowerLimitAc2Tau;
00 00           uint16_t SocketPowerLimitAc3;
00 00           uint16_t SocketPowerLimitAc3Tau;
FA 00           uint16_t SocketPowerLimitDc;
00 00           uint16_t SocketPowerLimitDcTau;
32 00           uint16_t TdcLimitSoc;
00 00           uint16_t TdcLimitSocTau;
4A 01           uint16_t TdcLimitGfx;
00 00           uint16_t TdcLimitGfxTau;

64 00           uint16_t TedgeLimit;
6E 00           uint16_t ThotspotLimit;
5E 00           uint16_t ThbmLimit;
73 00           uint16_t Tvr_gfxLimit;
73 00           uint16_t Tvr_memLimit;
FF FF           uint16_t Tliquid1Limit;
FF FF           uint16_t Tliquid2Limit;
FF FF           uint16_t TplxLimit;
00 00 00 00     uint32_t FitLimit;

00 00           uint16_t PpmPowerLimit;
00 00           uint16_t PpmTemperatureThreshold;

01              uint8_t  MemoryOnPackage;
00              uint8_t  padding8_limits;
73 00           uint16_t Tvr_SocLimit;

00 00           uint16_t  UlvVoltageOffsetSoc;
00 00           uint16_t  UlvVoltageOffsetGfx;

00              uint8_t  UlvSmnclkDid;
00              uint8_t  UlvMp1clkDid;
00              uint8_t  UlvGfxclkBypass;
00              uint8_t  Padding234;

86 0B           uint16_t     MinVoltageGfx;
22 0B           uint16_t     MinVoltageSoc;
0B 13           uint16_t     MaxVoltageGfx;
43 12           uint16_t     MaxVoltageSoc;

26 00           uint16_t     LoadLineResistanceGfx;
00 00           uint16_t     LoadLineResistanceSoc;

                [PPCLK_GFXCLK]
```

```
01                          .VoltageMode          = UCHAR
01                          .SnapToDiscrete       = UCHAR
09                          .NumDiscreteLevels    = UCHAR
00                          .padding              = UCHAR
                            .ConversionToAvfsClk{m = ULONG b = ULONG}
00 00 00 00
00 00 00 00
                            .SsCurve            {a = ULONG b = ULONG c = ULONG}
5B B1 BF 3E
EC 51 F8 BE
65 19 52 3F
                    [PPCLK_VCLK]
00                          .VoltageMode          = UCHAR
01                          .SnapToDiscrete       = UCHAR
08                          .NumDiscreteLevels    = UCHAR
00                          .padding              = UCHAR
                            .ConversionToAvfsClk{m = ULONG b = ULONG}
64 3B 9F 3F
16 DE A5 BD
                            .SsCurve            {a = ULONG b = ULONG c = ULONG}
00 00 00 00
00 00 00 00
00 00 00 00
                    [PPCLK_DCLK]
00                          .VoltageMode          = UCHAR
01                          .SnapToDiscrete       = UCHAR
08                          .NumDiscreteLevels    = UCHAR
00                          .padding              = UCHAR
                            .ConversionToAvfsClk{m = ULONG b = ULONG}
35 5E 9A 3F
8F 36 2E 3E
                            .SsCurve            {a = ULONG b = ULONG c = ULONG}
00 00 00 00
00 00 00 00
00 00 00 00
                    [PPCLK_ECLK]
00                          .VoltageMode          = UCHAR
01                          .SnapToDiscrete       = UCHAR
08                          .NumDiscreteLevels    = UCHAR
00                          .padding              = UCHAR
                            .ConversionToAvfsClk{m = ULONG b = ULONG}
69 6F B0 3F
48 33 16 3D
                            .SsCurve            {a = ULONG b = ULONG c = ULONG}
00 00 00 00
00 00 00 00
00 00 00 00
                    [PPCLK_SOCCLK]
00                          .VoltageMode          = UCHAR
01                          .SnapToDiscrete       = UCHAR
08                          .NumDiscreteLevels    = UCHAR
00                          .padding              = UCHAR
                            .ConversionToAvfsClk{m = ULONG b = ULONG}
E5 61 A1 3F
12 F7 58 BD
                            .SsCurve            {a = ULONG b = ULONG c = ULONG}
00 00 00 00
00 00 00 00
00 00 00 00
                    [PPCLK_UCLK]
00                          .VoltageMode          = UCHAR
01                          .SnapToDiscrete       = UCHAR
03                          .NumDiscreteLevels    = UCHAR
00                          .padding              = UCHAR
                            .ConversionToAvfsClk{m = ULONG b = ULONG}
B5 15 83 3F
C2 DD 19 3E
                            .SsCurve            {a = ULONG b = ULONG c = ULONG}
00 00 00 00
00 00 00 00
00 00 00 00
                    [PPCLK_DCEFCLK]
00                          .VoltageMode          = UCHAR
01                          .SnapToDiscrete       = UCHAR
08                          .NumDiscreteLevels    = UCHAR
00                          .padding              = UCHAR
```

```
                        .ConversionToAvfsClk{m = ULONG b = ULONG}
86 38 86 3F
0E BE 30 3E
                        .SsCurve            {a = ULONG b = ULONG c = ULONG}
00 00 00 00
00 00 00 00
00 00 00 00
                [PPCLK_DISPCLK]
00                      .VoltageMode          = UCHAR
01                      .SnapToDiscrete       = UCHAR
08                      .NumDiscreteLevels    = UCHAR
00                      .padding              = UCHAR
                        .ConversionToAvfsClk{m = ULONG b = ULONG}
83 C0 5A 3F
4D A1 F3 3D
                        .SsCurve            {a = ULONG b = ULONG c = ULONG}
00 00 00 00
00 00 00 00
00 00 00 00
                [PPCLK_PIXCLK]
02                      .VoltageMode          = UCHAR
01                      .SnapToDiscrete       = UCHAR
08                      .NumDiscreteLevels    = UCHAR
00                      .padding              = UCHAR
                        .ConversionToAvfsClk{m = ULONG b = ULONG}
00 00 00 00
00 00 00 00
                        .SsCurve            {a = ULONG b = ULONG c = ULONG}
A4 70 8D 3F
E4 14 85 BF
D5 E7 32 3F
                [PPCLK_PHYCLK]
02                      .VoltageMode          = UCHAR
01                      .SnapToDiscrete       = UCHAR
03                      .NumDiscreteLevels    = UCHAR
00                      .padding              = UCHAR
                        .ConversionToAvfsClk{m = ULONG b = ULONG}
00 00 00 00
00 00 00 00
                        .SsCurve            {a = ULONG b = ULONG c = ULONG}
00 00 00 00
00 00 00 00
D5 E7 32 3F
                [PPCLK_FCLK]
01                      .VoltageMode          = UCHAR
01                      .SnapToDiscrete       = UCHAR
08                      .NumDiscreteLevels    = UCHAR
00                      .padding              = UCHAR
                        .ConversionToAvfsClk{m = ULONG b = ULONG}
00 00 80 3F
00 00 00 00
                        .SsCurve            {a = ULONG b = ULONG c = ULONG}
D1 91 FC 3E
1F 85 2B BF
32 77 75 3F
                uint16_t       FreqTableGfx[16];
BC 02
28 03
6E 04
5C 05
0A 06
93 06
D5 06
ED 06
09 07
00 00
00 00
00 00
00 00
00 00
00 00
00 00
                uint16_t       FreqTableVclk     [NUM_VCLK_DPM_LEVELS    8];
66 01
E6 01
6B 02
```

```
F4 02
52 03
CC 03
6E 04
6E 04
                uint16_t       FreqTableDclk     [NUM_DCLK_DPM_LEVELS    8];
36 01
90 01
0C 02
6B 02
A8 02
F4 02
52 03
CC 03
                uint16_t       FreqTableEclk     [NUM_ECLK_DPM_LEVELS    8];
36 01
90 01
0C 02
6B 02
A8 02
F4 02
52 03
CC 03
                uint16_t       FreqTableSocclk   [NUM_SOCCLK_DPM_LEVELS  8];
36 01
0C 02
37 02
6B 02
A8 02
F4 02
52 03
CC 03
                uint16_t       FreqTableUclk     [NUM_UCLK_DPM_LEVELS    4];
5E 01
20 03
E8 03
E8 03
                uint16_t       FreqTableFclk     [NUM_FCLK_DPM_LEVELS    8];
26 02
62 02
B2 02
F8 02
66 03
C0 03
38 04
C9 04
                uint16_t       FreqTableDcefclk  [NUM_DCEFCLK_DPM_LEVELS 8];
66 01
C6 01
37 02
A8 02
F4 02
52 03
CC 03
6E 04
                uint16_t       FreqTableDispclk  [NUM_DISPCLK_DPM_LEVELS 8];
66 01
C6 01
37 02
A8 02
F4 02
52 03
CC 03
6E 04
                uint16_t       FreqTablePixclk   [NUM_PIXCLK_DPM_LEVELS  8];
93 00
F2 00
58 01
E4 01
15 02
AA 03
13 04
34 04
                uint16_t       FreqTablePhyclk   [NUM_PHYCLK_DPM_LEVELS  8];
0E 01
1C 02
```

```
2A 03
00 00
00 00
00 00
00 00
00 00
                uint16_t       DcModeMaxFreq     [PPCLK_COUNT            11];
09 07
6E 04
CC 03
CC 03
CC 03
E8 03
6E 04
6E 04
34 04
2A 03
C9 04

00 00           uint16_t       Padding8_Clks;
                uint16_t       Mp0clkFreq        [NUM_MP0CLK_DPM_LEVELS 2];
C8 00
2C 01
                uint16_t       Mp0DpmVoltage     [NUM_MP0CLK_DPM_LEVELS 2];
60 09
F0 0A

28 03           uint16_t        GfxclkFidle;
00 00           uint16_t        GfxclkSlewRate;
00 00           uint16_t        CksEnableFreq;
00 00           uint16_t        Padding789;

                QuadraticInt_t  CksVoltageOffset;
00 00 00 00             uint32_t a;
00 00 00 00             uint32_t b;
00 00 00 00             uint32_t c;

                uint8_t         Padding567[4];
00
00
00
00

09 07           uint16_t        GfxclkDsMaxFreq;
01              uint8_t         GfxclkSource;
00              uint8_t         Padding456;

00              uint8_t      LowestUclkReservedForUlv;
                uint8_t      Padding8_Uclk[3];
00
00
00


                uint8_t      PcieGenSpeed[NUM_LINK_LEVELS 2];
00
02
                uint8_t      PcieLaneCount[NUM_LINK_LEVELS 2];
06
06
                uint16_t     LclkFreq[NUM_LINK_LEVELS 2];
50 00
34 01

00 00           uint16_t     EnableTdpm;
00 00           uint16_t     TdpmHighHystTemperature;
00 00           uint16_t     TdpmLowHystTemperature;
00 00           uint16_t     GfxclkFreqHighTempLimit;

00 00           uint16_t     FanStopTemp;
00 00           uint16_t     FanStartTemp;

90 01           uint16_t     FanGainEdge;
90 01           uint16_t     FanGainHotspot;
90 01           uint16_t     FanGainLiquid;
90 01           uint16_t     FanGainVrGfx;
90 01           uint16_t     FanGainVrSoc;
```

```
90 01           uint16_t     FanGainPlx;
90 01           uint16_t     FanGainHbm;
14 00           uint16_t     FanPwmMin;
54 0B           uint16_t     FanAcousticLimitRpm;
54 0B           uint16_t     FanThrottlingRpm;
0A 0F           uint16_t     FanMaximumRpm;
5F 00           uint16_t     FanTargetTemperature;
00 00           uint16_t     FanTargetGfxclk;
00              uint8_t      FanZeroRpmEnable;
02              uint8_t      FanTachEdgePerRev;

00 00           int16_t      FuzzyFan_ErrorSetDelta;
00 00           int16_t      FuzzyFan_ErrorRateSetDelta;
00 00           int16_t      FuzzyFan_PwmSetDelta;
00 00           uint16_t     FuzzyFan_Reserved;

                uint8_t           OverrideAvfsGb[AVFS_VOLTAGE_COUNT 2];
00
01
                uint8_t           Padding8_Avfs[2];
00
00


                QuadraticInt_t    qAvfsGb[AVFS_VOLTAGE_COUNT 2];
00 00 00 00             uint32_t a;
50 8D 97 3C             uint32_t b;
0A D7 A3 3B             uint32_t c;

00 00 00 00             uint32_t a;
EA B2 98 3C             uint32_t b;
87 A2 40 3D             uint32_t c;
                DroopInt_t        dBtcGbGfxCksOn;
00 00 00 00             uint32_t a;
00 00 00 00             uint32_t b;
00 00 00 00             uint32_t c;
                DroopInt_t        dBtcGbGfxCksOff;
00 00 00 00             uint32_t a;
00 00 00 00             uint32_t b;
00 00 00 00             uint32_t c;
                DroopInt_t        dBtcGbGfxAfll;
00 00 00 00             uint32_t a;
00 00 00 00             uint32_t b;
00 00 00 00             uint32_t c;
                DroopInt_t        dBtcGbSoc;
00 00 00 00             uint32_t a;
00 00 00 00             uint32_t b;
00 00 00 00             uint32_t c;
                LinearInt_t       qAgingGb[AVFS_VOLTAGE_COUNT 2];
00 00 00 00             uint32_t m;
00 00 00 00             uint32_t b;
                LinearInt_t;
00 00 00 00             uint32_t m;
00 00 00 00             uint32_t b;

                QuadraticInt_t    qStaticVoltageOffset[AVFS_VOLTAGE_COUNT 2];
00 00 00 00             uint32_t a;
00 00 00 00             uint32_t b;
00 00 00 00             uint32_t c;

00 00 00 00             uint32_t a;
00 00 00 00             uint32_t b;
00 00 00 00             uint32_t c;

                uint16_t          DcTol[AVFS_VOLTAGE_COUNT 2];
00 00
A0 00
                uint8_t           DcBtcEnabled[AVFS_VOLTAGE_COUNT 2];
01
00
                uint8_t           Padding8_GfxBtc[2];
00
00
                int16_t           DcBtcMin[AVFS_VOLTAGE_COUNT 2];
00 00
00 00
                uint16_t          DcBtcMax[AVFS_VOLTAGE_COUNT 2];
```

```
A0 00
00 00
                uint8_t           XgmiLinkSpeed   [NUM_XGMI_LEVELS 2];
08
10
                uint8_t           XgmiLinkWidth   [NUM_XGMI_LEVELS 2];
02
10
                uint16_t          XgmiFclkFreq    [NUM_XGMI_LEVELS 2];
1A 04
4C 04
                uint16_t          XgmiUclkFreq    [NUM_XGMI_LEVELS 2];
E8 03
E8 03
                uint16_t          XgmiSocclkFreq  [NUM_XGMI_LEVELS 2];
E8 03
E8 03
                uint16_t          XgmiSocVoltage  [NUM_XGMI_LEVELS 2];
00 00
00 00

00 00 00 00     uint32_t          DebugOverrides;

                QuadraticInt_t    ReservedEquation0;
00 00 00 00             uint32_t a;
00 00 00 00             uint32_t b;
00 00 00 00             uint32_t c;
                QuadraticInt_t    ReservedEquation1;
00 00 00 00             uint32_t a;
00 00 00 00             uint32_t b;
00 00 00 00             uint32_t c;
                QuadraticInt_t    ReservedEquation2;
00 00 00 00             uint32_t a;
00 00 00 00             uint32_t b;
00 00 00 00             uint32_t c;
                QuadraticInt_t    ReservedEquation3;
00 00 00 00             uint32_t a;
00 00 00 00             uint32_t b;
00 00 00 00             uint32_t c;
                QuadraticInt_t;

86 0B           uint16_t     MinVoltageUlvGfx;
22 0B           uint16_t     MinVoltageUlvSoc;

54 0B           uint16_t     MGpuFanBoostLimitRpm;
00 00           uint16_t     padding16_Fan;

90 01           uint16_t     FanGainVrMem0;
90 01           uint16_t     FanGainVrMem1;

                uint16_t     DcBtcGb[AVFS_VOLTAGE_COUNT 2];
38 00
00 00
                uint32_t     Reserved[11];
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
                uint32_t     Padding32[3];
00 00 00 00
00 00 00 00
00 00 00 00

00 00           uint16_t     MaxVoltageStepGfx;
00 00           uint16_t     MaxVoltageStepSoc;

00              uint8_t      VddGfxVrMapping;
00              uint8_t      VddSocVrMapping;
00              uint8_t      VddMem0VrMapping;
```

```
00              uint8_t      VddMem1VrMapping;

00              uint8_t      GfxUlvPhaseSheddingMask;
00              uint8_t      SocUlvPhaseSheddingMask;
00              uint8_t      ExternalSensorPresent;
00              uint8_t      Padding8_V;


00 00           uint16_t     GfxMaxCurrent;
00              int8_t       GfxOffset;
00              uint8_t      Padding_TelemetryGfx;

00 00           uint16_t     SocMaxCurrent;
00              int8_t       SocOffset;
00              uint8_t      Padding_TelemetrySoc;

00 00           uint16_t     Mem0MaxCurrent;
00              int8_t       Mem0Offset;
00              uint8_t      Padding_TelemetryMem0;

00 00           uint16_t     Mem1MaxCurrent;
00              int8_t       Mem1Offset;
00              uint8_t      Padding_TelemetryMem1;

00              uint8_t      AcDcGpio;
00              uint8_t      AcDcPolarity;
00              uint8_t      VR0HotGpio;
00              uint8_t      VR0HotPolarity;

00              uint8_t      VR1HotGpio;
00              uint8_t      VR1HotPolarity;
00              uint8_t      Padding1;
00              uint8_t      Padding2;

00              uint8_t      LedPin0;
00              uint8_t      LedPin1;
00              uint8_t      LedPin2;
00              uint8_t      padding8_4;

00              uint8_t      PllGfxclkSpreadEnabled;
00              uint8_t      PllGfxclkSpreadPercent;
00 00           uint16_t     PllGfxclkSpreadFreq;

00              uint8_t      UclkSpreadEnabled;
00              uint8_t      UclkSpreadPercent;
00 00           uint16_t     UclkSpreadFreq;

00              uint8_t      FclkSpreadEnabled;
00              uint8_t      FclkSpreadPercent;
00 00           uint16_t     FclkSpreadFreq;

00              uint8_t      FllGfxclkSpreadEnabled;
00              uint8_t      FllGfxclkSpreadPercent;
00 00           uint16_t     FllGfxclkSpreadFreq;

                I2cControllerConfig_t I2cControllers[I2C_CONTROLLER_NAME_COUNT 7];
00 00 00 00             uint32_t Enabled;
00 00 00 00             uint32_t SlaveAddress;
00 00 00 00             uint32_t ControllerPort;
00 00 00 00             uint32_t ControllerName;
00 00 00 00             uint32_t ThermalThrottler;
00 00 00 00             uint32_t I2cProtocol;
00 00 00 00             uint32_t I2cSpeed;

00 00 00 00             uint32_t Enabled;
00 00 00 00             uint32_t SlaveAddress;
00 00 00 00             uint32_t ControllerPort;
00 00 00 00             uint32_t ControllerName;
00 00 00 00             uint32_t ThermalThrottler;
00 00 00 00             uint32_t I2cProtocol;
00 00 00 00             uint32_t I2cSpeed;

00 00 00 00             uint32_t Enabled;
00 00 00 00             uint32_t SlaveAddress;
00 00 00 00             uint32_t ControllerPort;
00 00 00 00             uint32_t ControllerName;
```

```
00 00 00 00             uint32_t ThermalThrottler;
00 00 00 00             uint32_t I2cProtocol;
00 00 00 00             uint32_t I2cSpeed;

00 00 00 00             uint32_t Enabled;
00 00 00 00             uint32_t SlaveAddress;
00 00 00 00             uint32_t ControllerPort;
00 00 00 00             uint32_t ControllerName;
00 00 00 00             uint32_t ThermalThrottler;
00 00 00 00             uint32_t I2cProtocol;
00 00 00 00             uint32_t I2cSpeed;

00 00 00 00             uint32_t Enabled;
00 00 00 00             uint32_t SlaveAddress;
00 00 00 00             uint32_t ControllerPort;
00 00 00 00             uint32_t ControllerName;
00 00 00 00             uint32_t ThermalThrottler;
00 00 00 00             uint32_t I2cProtocol;
00 00 00 00             uint32_t I2cSpeed;

00 00 00 00             uint32_t Enabled;
00 00 00 00             uint32_t SlaveAddress;
00 00 00 00             uint32_t ControllerPort;
00 00 00 00             uint32_t ControllerName;
00 00 00 00             uint32_t ThermalThrottler;
00 00 00 00             uint32_t I2cProtocol;
00 00 00 00             uint32_t I2cSpeed;

00 00 00 00             uint32_t Enabled;
00 00 00 00             uint32_t SlaveAddress;
00 00 00 00             uint32_t ControllerPort;
00 00 00 00             uint32_t ControllerName;
00 00 00 00             uint32_t ThermalThrottler;
00 00 00 00             uint32_t I2cProtocol;
00 00 00 00             uint32_t I2cSpeed;

                uint32_t     BoardReserved[10];
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
                uint32_t     MmHubPadding[8];
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
00 00 00 00
```