**Modern Standby Spec**

**Version 0.9**

Contents

[1 What is Modern Standby? 4](#_Toc9266452)

[1.1 Microsoft’s Modern Standby Overview 4](#_Toc9266453)

[1.2 Microsoft defines six key area of user experience: 4](#_Toc9266454)

[2 How to enable MSC? 6](#_Toc9266455)

[2.1 Microsoft’s CS platform requirement is as follows: 6](#_Toc9266456)

[3 How to optimize power consumption during MSC? 7](#_Toc9266457)

[3.1 System Power Consumption Consideration 7](#_Toc9266458)

[3.2 Design Consideration during Each Development Phase 7](#_Toc9266459)

[3.2.1 Pre-DB - Planning 7](#_Toc9266460)

[3.2.2 DB Phase – enablement and assessment 10](#_Toc9266461)

[3.2.3 SI Phase 13](#_Toc9266462)

[3.2.4 PV Phase 14](#_Toc9266463)

[3.2.5 MV Phase 15](#_Toc9266464)

[3.2.6 Adaptive Hibernation 15](#_Toc9266465)

[3.2.6 “Hot Bag” Effect 15](#_Toc9266466)

[How do we validate and debug MSC? 16](#_Toc9266467)

**Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author** | **Changes** |
| 0.8 | 03/26/2019 | Robert Sullivan | 1. Initial Release |
| 0.9 | 05/20/2019 | Ocean Wu | 1. Add Index. |
|  |  |  |  |
|  |  |  |  |

# 1 What is Modern Standby?

## 1.1 Microsoft’s Modern Standby Overview

* Microsoft’s Modern Standby has changed names many times since its inception … i.e. – Always On / Always Connected, Instant Go, Connected Standby, etc.
* Modern Standby can be enabled with two capabilities … Connected and Disconnected. Both offer the same functionality, except that Disconnected does not connect to the Internet while in Modern Standby. POR at HP is to only support Modern Standby Connected (MSC) … starting with the 2018 BNB launch cycle.
* Modern Standby Connected (MSC) brings the smartphone power model to Windows PCs. An MSC PC can instantly resume from sleep and is always connected to the Internet. UWP Apps on an MSC PC are automatically updated while the system is in MSC so that critical information—including email and messages—are already synced to the PC when the user turns it on.
* Modern Standby is a screen-off sleep state. Any time the system has the screen off, it is said to be in modern standby.
* While the system is in MSC, it can pass through various hardware and software operating modes. For most of the time spent in MSC, the hardware is in a low-power state and the software is paused or stopped. However, the system intermittently powers up to process an incoming email, alert the user to an incoming Skype call, or perform other app-related background activities.

## 1.2 Microsoft defines six key area of user experience:

|  |  |
| --- | --- |
| **Area of User Experience** | **Design Guideline** |
| Turning off | Must be instant.  Could be via button, lid/cover, menu, or timeout. |
| Turning on | Must be instant.  Wake source is defined by Microsoft [here](https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby-wake-sources). Note: HP may not enable all recommended wake sources. Refer to the platform PDD/MRD for guidance. |
| Simple app and tile updates | Metro (UWP) apps have limited runtime and network access.  Desktop apps cannot run and cannot access network and are suspended during connected standby.  System services have limited runtime, but no network access. |
| Lock screen apps and push notifications | Lock screen apps are allowed extra time to run in the background.  Allows audio alert but screen stays off, except…  Voice-over-IP (VOIP) (e.g. Lync and Skype), generates audio and also turn on the screen for the duration of the ringtone. |
| Audio notifications | refer to “lock screen apps and push notifications” |
| Quiet hours | Setting to suppress notifications and save battery power. |

Please check with Microsoft on the latest changes.

<https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/modern-standby>

# 2 How to enable MSC?

## 2.1 Microsoft’s CS platform requirement is as follows:

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Motivation** | **Who’s responsible** |
| The system ACPI firmware must set the ACPI\_S0\_LOW\_POWER\_IDLE FADT flag. | Indicates that the hardware platform supports the low-power idle mode for MSC. | System firmware developer  (Core silicon or SoC must be capable of low-power idle.) |
| The system ACPI firmware must not provide an S3 object in the root of the namespace. | Windows supports a platform that exposes either the S3 object or the ACPI\_S0\_LOW\_POWER\_IDLE FADT flag, but not both at the same time. | System firmware developer |
| The core silicon or SoC manufacturer must provide a power engine plug-in (PEP). | The PEP coordinates device state and processor idle state dependencies. A minimal PEP is required to communicate to Windows when the device power state dependencies have been achieved for the lowest SoC idle power mode. | Core silicon or SoC provider |
| x86/x64-based connected standby PCs must also support Hibernate. | Hibernate is used to save the state of desktop/Win32 applications when critical-low battery capacity is reached. | System firmware developer |
| All built-in networking devices must be compatible with NDIS 6.3 (specifically WoL patterns, protocol offloads, and D0 packet coalescing). | Enables the SoC to enter low-power modes while the networking device maintains connectivity. | System designer (OEM/ODM) |
| The primary storage volume must be solid-state and have a 0-millisecond seek penalty. | Ensures that Windows can safely access the storage device any time during MSC. | System designer (OEM/ODM) |

This enables MSC, but then we need to optimize for power consumption.

# 3 How to optimize power consumption during MSC?

## 3.1 System Power Consumption Consideration

During MSC, none of the power rails are forced to turn off except for PVCore when processor enters C10. So, it is entirely up to each component of the system to achieve its lowest power consumption and state; therefore, enable the system to achieve its lowest power consumption and state. This gives us the *floor power*.

The other aspect of power consumption is *average power*. During MSC, the system can wake up by the OS for maintenance, we need to study the system over time and under difference scenarios for average power consumption. This proves to be challenging since it depends on software/firmware behavior.

## 3.2 Design Consideration during Each Development Phase

### 3.2.1 Pre-DB - Planning

Although the same software, BIOS and hardware can be shared to enable MSC on a non-MSC platform, it does not guarantee the machine can pass MSC power consumption requirements and functional tests. Both hardware and software teams must work together to ensure their component is prepared for MSC.

#### 3.2.1.1 Component Selection

Marketing must provide target battery life for MSC.

Hardware architecture team should select components with the proper power consumption at low power idle state. Based on these numbers, we can have an estimate for power consumption during MSC.

MSC power consumption should be estimated prior to DB. As an example, please see the Power Estimate spreadsheet below

<https://hp.sharepoint.com/:x:/r/teams/CMITSE/Architecture/Modern_Standby/Modern%20Standby%20and%20Low%20Power%20Enabling%20BIOS%20Checklist/MS_Power_Breakdown_Spec_V0.3.xlsx?d=wb77e86237dee4f6692aa7b301c01b3d8&csf=1&e=ZMKQPw>

The MSC power map should be done prior to DB and should be validated by using a multimeter or DAQ for each phase (DB, SI, PV) of the platform.

***Lesson learned since Broadwell***

* Ensure component support low power mode – such as hubs/switches. It may block upstream component from entering low power state
* Power consumption on Ellis - SIO1086332 - Power usage increases when Elis jacket is used with Olympia PrePV during CS (root cause is the Realtek RT7285C to Texas Instruments TPS62122DRVR due inefficiency at low currents (<25% during CS).
* Wacom Touch stays in D0 during CS, should be D3
* Some memory vendors consume more power than others (i.e. – Samsung 16GB DDR4 consumes 30 mW more than 16GB of Hynix DDR4)
* Optane NVMe storage requires different BIOS constraints than other NVMe drives in order to allow SLP\_S0# assertion

#### 3.2.1.2 Microsoft Requirement on device drivers

Make sure the component that requires device driver is aware the platform supports MSC. Review the Microsoft [Device-specific power management for connected standby](https://docs.microsoft.com/en-us/windows-hardware/design/device-experiences/device-specific-power-management-for-modern-standby) .

Only UWP Apps are allowed to run during MSC. Legacy Apps will be suspended during MSC. For example, Windows Media Player is a Legacy App and will not run during MSC. Windows Groove Player is a UWP App and will play locally stored mp3’s while in MSC. Also, Microsoft Outlook is a Legacy App that is suspended during MSC, but using Mail (the UWP App in the Windows Store) will allow the user to receive POP Mail while in MSC. Reference the Microsoft website for more information on UWP requirements <https://docs.microsoft.com/en-us/windows/uwp/>.

#### 3.2.1.3 Microsoft Requirement for BIOS

BIOS needs to adhere to [Microsoft ACPI design guide for SOC platforms](https://msdn.microsoft.com/en-us/library/windows/hardware/dn495676(v=vs.85).aspx).

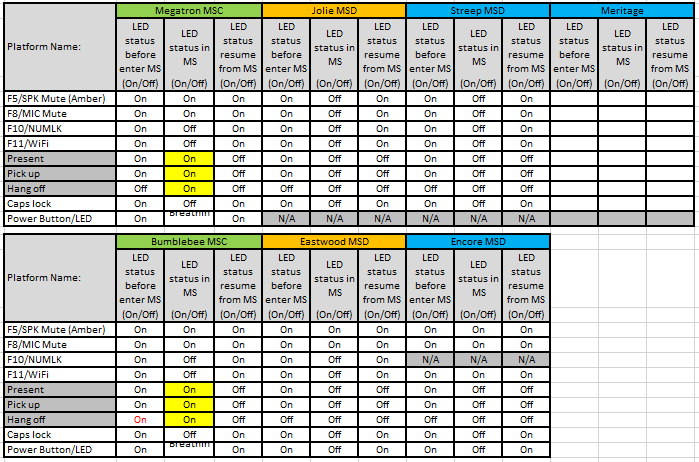
***Lesson learned from Broadwell***

SIO1120355 - Potentially root cause for 4-hour hibernation.

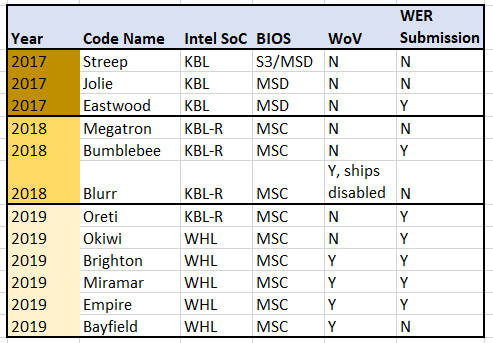
#### 3.2.1.4 Checklist for pre-DB

|  |  |
| --- | --- |
| **Item** | **Owner** |
| Target MSC battery life | Marketing |
| Component Selection to meet MSC battery life | EE – Architect |
| Power Budget by component | EE – Power Architect |
| MSC Power Map | EE - Architect |
| Device Drivers Meet Microsoft MSC Requirement | Component owner |
| System Bios meet Microsoft ACPI Design Guide for SOC platforms | Platform BIOS |

#### 3.2.1.5 LED functionality during MSC



#### 3.2.1.6 Modern Standby History



### 3.2.2 DB Phase – enablement and assessment

The focus during DB phase is to review MSC configuration, enable MSC, and provide baseline measurement of MSC power consumption. The ODM should confirm the EE – Power Architect’s Power Chart via physical power measurements (i.e. – normally using a DAQ). Finally, basic functionality (enter/resume MSC, play mp3 during MSC, etc.) should also verified.

#### 3.2.2.1 BIOS/Hardware Platform Configuration Review

BIOS/EC and EE need to go through the portion of Intel Low Power Features Debug Guide that discuss BIOS configuration (For Broadwell: 542852, Skylake is 557007, Coffee Lake 571210). *Note: Intel update these documents frequently so please ensure you have the latest version. In another word, Intel is developing this document at the same time HP is developing the platform.*

For Broadwell, BIOS/EC and EE needs to review the settings in Chapter 2 – Platform Interfaces and Configurations. There are some restrictions such as in Section 2.2.2 – Root Port Function Number Swap that may affect power management and prevent PCIe going to low power.

***Lessons learned from Broadwell***

These are some of the things we need to pay attention moving forward:

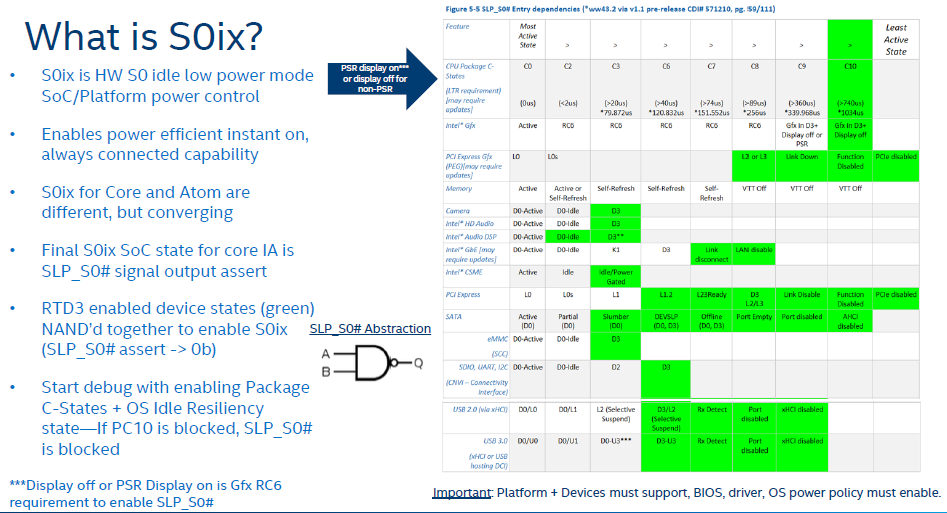
* Unused GPIO strapping (EC and HW) (Section 2.14 and manual check)
* ACPI tables – PEP checker (section 2.12)
* ME Power-gating during CS
* ICC = 0%

#### 3.2.2.2 Enable Connected Standby

BIOS needs to support enablement of MSC during DB phase. Component owners needs to provide drivers that support MSC. The system must enter, exit, and stay in MSC as well as transition into S4 Hibernation.

#### 3.2.2.3 Intel CPU/Chipset (PCH) Optimization

The starting point for optimization should be an examination of what’s required by the processor/chipset (PCH) to reach its lowest power state. See Intel’s explanation below.



Starting with Broadwell U and Y, PCH-LP platforms support a new feature - Windows\* InstantGo Power Save Mode. During S0 when the processor is in its lowest idle power state and SLP\_S0# is asserted, all of the 1.05V rails (including core, suspend and active sleep well) can be lowered to 0.95V. This feature also requires that the platform is configured for External Suspend VR Mode. For more details, refer to the Broadwell PCH-LP EDS (CDI / IBP #515621) and the Broadwell Platform Design Guide.

All components connected to the PCH needs to be in the correct state in order for the PCH to reach Low Power Mode 3. This means the hardware/software/firmware combination allows the component to enter its lowest power mode. If any of the devices does not do so and as a result prevent PCH to enter a low power state, then we cannot achieve PCH Low Power Mode 3.

Derived from the table *PCH low Power Mode 3: SLP\_S0# Low Prerequisites*, this is the complete list of hardware requirement:

1. CPU Package in C10. 9. SDIO in D3

2. Gfx in D3. 10. UART in D3

3. Screen off. 11. I2C buses in D3

4. PCIe in D3. 12. HD Audio in D3

5. SATA in D3. 13. Intel Audio DSP in D3

6. USB2.0 in D3. 14. Intel ME power gated

7. USB3.0 in U3. 15. Intel GbE disconnected or in D3

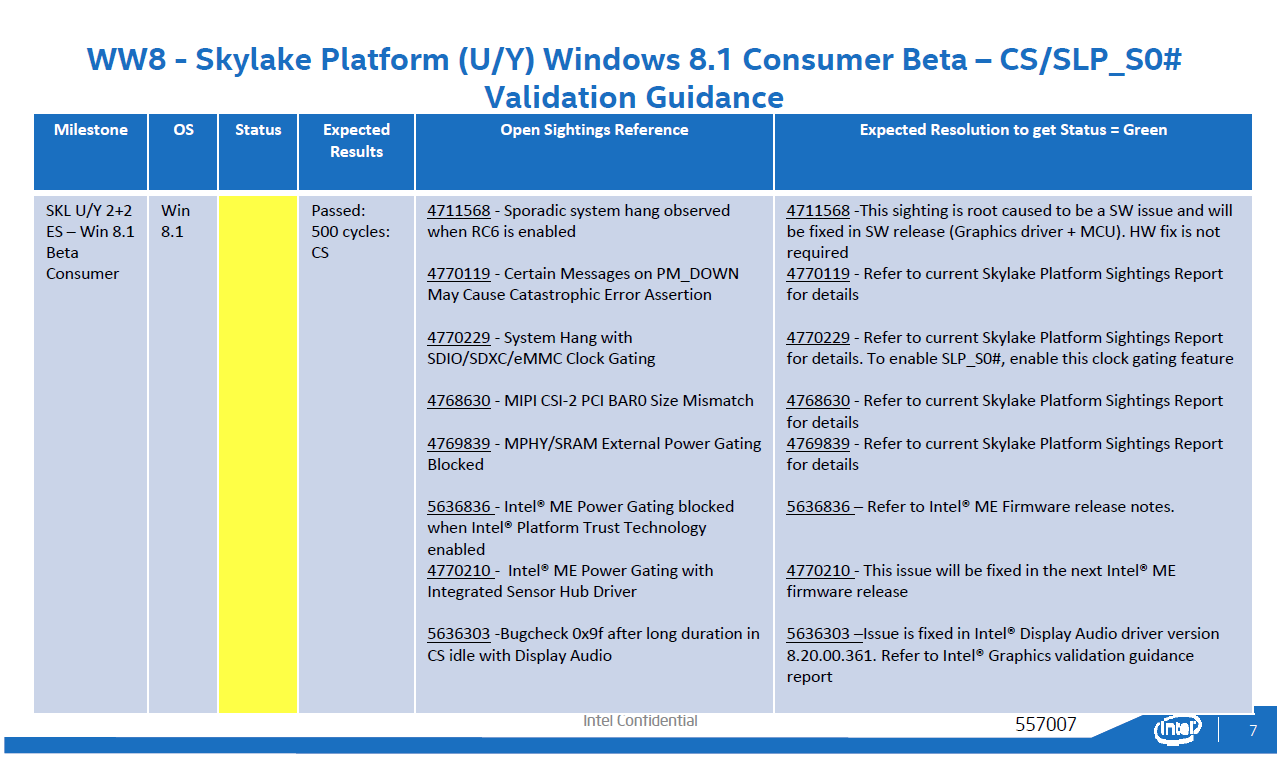
8. xHCI Controller in D3. 16. PCIe, SATA, & USB PHY’s power gated

Please refer to Intel Document 557007 - Skylake U/Y/S Platform – S0ix/Resets Validation Guidance for known sightings and limitations. Also, 571210 for the Coffee Lake S0ix Whitepaper.

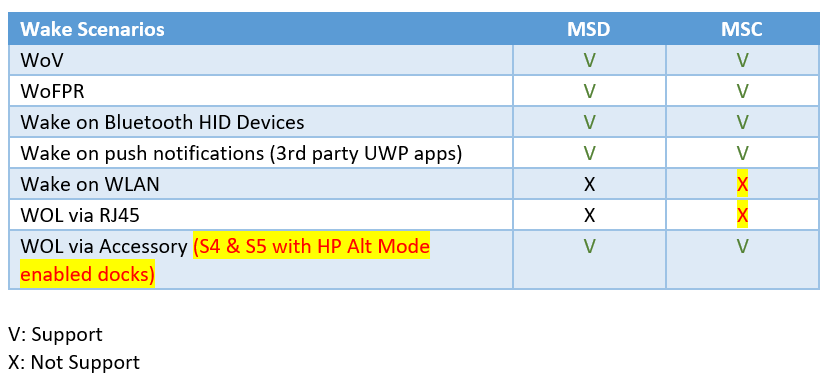
*A matrix showing the functionality for each controller is being created and will be shared on a Sharepoint. The link will be provided on the next version of the spec.*

Here is the list based on the latest revision 0.7 (3/30/2015):

Figure 2- Skylake SLP\_S0# Validation Guidance



#### 3.2.2.4 Communications Wake Support



***Lessons Learned***

On 2014 platforms (Olympia), HW design followed Intel CRB with some notable difference. This is not a complete list but highlights some major differences;

Some of the lessons learned from Olympia

1. Isolating FETs to gate the mentioned MPHYs for USB, PCIe, and SATA. Without these rails turned off SLP\_S0# would never assert
2. SLP\_S0#: this signal must be wired to the PMIC or main Core VR (per EC Design Guideline for BDW, Doc ID: 519280) The PMIC will use this signal to properly control PVCORE and bring the CPU into a LPM. If PVCORE does not go to 0V the CPU power will void any attempts to meet the power targets.
3. Interface with the EC: The system must have a bidirectional communication between the CPU and the EC so that they can properly enter MSC. The process is rather simple: the OS tells the EC that it will enter MSC so that the EC itself can go to LPM and place any devices or rails it controls in D3 or shut-down mode respectively. Reciprocally, the EC must have a way to wake the CPU/OS when a wake event occurs such as power button press, thermal shutdown, battery critically low, etc. In the case of BJR and Olympia we use internal commands to tell the EC that the CPU is going to MSC, we use the signal PCH\_SLP\_S0IX# to wake the EC (GPIO from the PCH to GPIO in the EC), and we use EXT\_SMI# so that the EC can wake the CPU.

#### 3.2.2.4 Measurement of MSC power consumption

An instrumented board is required to measure the average and floor power of the system during DB phase. We want to identify any major hardware changes required prior to SI phase.

#### 3.2.2.5 DB Phase Exit for Connected Standby

|  |  |
| --- | --- |
| **Item** | **Owner** |
| Hardware and BIOS team review platform implementation | EE/BIOS |
| Enable MSC | BIOS/Component Owners |
| Measurement of floor power and average power by component | EE/CS |
| Intel CPU/Chipset (PCH) Low Power Enablement (see table ??) | EE/CS/Component Owners |

## 3.2.3 SI Phase

The focus during SI phase is to optimize power consumption during MSC and validate full functionality.

#### 3.2.3.1 External Component Optimization

If we observe Slp\_S0# asserts during MSC but does not stay asserted, or if the system is drawing excessive power during MSC and Slp\_S0# stays asserted, then we need to exam the individual components.

The term external component refers to anything that’s not part of Intel processor and PCH, so this include hubs, peripherals and etc.

For each component, including those that’s attached to the buses listed above, we need to assess the following:

1. What is the lowest device state the component can achieve?
2. What should the lowest device state the component stays in during MSC (is it a wake source, is it a network access device, etc.)?
3. Does the component prevent PCH from entering Low Power Mode 3?
4. What is the expected power consumption during device low power idle?

#### 3.2.3.2 Floor Power Optimization

During CPU/PCH and external device optimization, the power estimate in pre-DB phase can be used as a reference for expected floor power for each component. By the end of SI phase, a reasonable floor power number should be achieved.

#### 3.2.3.3 Average Power Optimization

The environment, such as image, and internet connectivity affects the power consumption of the device under test. For consistency, the Modern Standby Core Team provides testing guideline. This method provides power consumption under best case scenario.

To understand average power consumption may require the system configured differently prior to MSC entry. It is in each component team’s best interest to add in their test plan to cover all usage case of that component. For example, audio team needs to test mp3 playback during MSC.

#### 3.2.3.4 SI Phase exit for Connected Standby

|  |  |
| --- | --- |
| **Item** | **Owner** |
| Modern Standby Connected 100% functional (wake source, component behavior, etc.) | MSC team/Component Owner/HW/BIOS |
| Floor Power meets Power estimate (compare to power budget) | MSC team/Component Owner/HW/BIOS |
| Average power meets marketing requirement | MSC team/Component Owner/HW/BIOS |

## 3.2.4 PV Phase

Execute plan to achieve desired average power consumption during MSC. Real world usage feedback from Dogfood program is analyzed and MSC issues resolved.

Test for average power consumption for every Image/BIOS release.

#### 3.2.4.1 PV Phase exit for MSC

|  |  |
| --- | --- |
| **Item** | **Owner** |
| Code Freeze Image meets power consumption requirement | MSC team/Component Owner/HW/BIOS |
| Address P1 issues from Dogfood program | MSC team/Component Owner/HW/BIOS |

## 3.2.5 MV Phase

Pay special attention to any hardware change between PV and MV that may affect MSC.

Interface with Microsoft’s WER team to confirm adherence to Exceptional requirements where required by HP Marketing. Here is a sample of the WER Exceptional requirements from Microsoft.

<https://hp.sharepoint.com/:x:/r/teams/CMITSE/Architecture/Modern_Standby/Microsoft_WER/WER%20Holiday%202018%20Exceptional%20Targets.xlsx?d=w221856695b8343fabd59633a6223b2b3&csf=1&e=pv3wmm>

## 3.2.6 Adaptive Hibernation

Adaptive Hibernation (and Battery Budget) is included in the Operating System by Microsoft. Please reference the following documents for its functionality.

<https://hp.sharepoint.com/:p:/r/teams/CMITSE/_layouts/15/Doc.aspx?sourcedoc=%7B94b01931-2564-4d85-a988-73cd0854e611%7D&action=edit&uid=%7B94B01931-2564-4D85-A988-73CD0854E611%7D&ListItemId=2018&ListId=%7BE24857D1-6801-41B0-AA4F-33950D39C31A%7D&odsp=1&env=prod>

<https://hp.sharepoint.com/:w:/r/teams/CMITSE/_layouts/15/Doc.aspx?sourcedoc=%7BDDE66959-4028-4E2B-A4D6-F7600090064C%7D&file=Adpative%20Hibernate.docx&action=default&mobileredirect=true>

### 3.2.6 “Hot Bag” Effect

Due to customer complaints, Windows no longer allows WiFi to wake/connect while the MSC machine is in DC mode (i.e. – battery power) unless the LID is open. Refer to the Microsoft email below:

<https://hp.sharepoint.com/teams/CMITSE/Architecture/Forms/AllItems.aspx?newTargetListUrl=%2Fteams%2FCMITSE%2FArchitecture&viewpath=%2Fteams%2FCMITSE%2FArchitecture%2FForms%2FAllItems%2Easpx&id=%2Fteams%2FCMITSE%2FArchitecture%2FModern%5FStandby%2FChange%20in%20Modern%20Standby%20Connected%20functionality%2Emsg&parent=%2Fteams%2FCMITSE%2FArchitecture%2FModern%5FStandby>

# How do we validate and debug MSC?

***Software and Hardware Tools***

A combination of software and hardware tool can be used to validate. The test procedure and tool guideline can be found in the MSC Test Guide. Please contact the MSC core team.

Here is a list of tools used and description.

Power House Mountain (formerly, Cactus Mountain) – Intel SW App. It uses Windows ETL and some Intel driver to capture system and device power states, and also processor C-state information. Note: Microsoft’s Windows Performance Analyzer can also be used to analyze the ETL files.

Battery Life Analyzer – Intel software. It is used to look at each device’s power states (??)

Sleep Study – Microsoft Windows built-in tool. Executed in an elevated command prompt and with the command “powercfg /sleepstudy”. Provides a high-level view of the previous MSC cycle.

DAQ – Data Acquisition – multiple vendors. Used to measure current and voltage for a component or power rail to access power consumption. The boards will need to be prepared for instrumentation.

***System/Functional Validation Test***

Please refer to MSC Test Guide document for test procedure.

***Debug Method***

Prerequisite: system board is prepared for DAQ or RPM2, see guideline in Appendix.Using the top down design, SLP\_S0# being at the top. First, we need to validate the following:

|  |  |
| --- | --- |
| **Item** | **Validation** |
| CPU Package in C10. | Measure PVCore, Power House Mountain |
| Gfx in D3. | Power House Mountain |
| Screen off. | visual |
| PCIe in D3. | Power House Mountain |
| SATA in D3. | Power House Mountain |
| USB2.0 in D3. | Power House Mountain |
| USB3.0 in U3. | Power House Mountain |
| XHCI Controller in D3. | Power House Mountain |
| SDIO in D3. | Power House Mountain |
| UART in D3. | Power House Mountain |
| I2C buses in D3. | Power House Mountain |
| HD Audio in D3. | Power House Mountain |
| Intel Audio DSP in D3. | Power House Mountain |
| Intel ME power gated. | Power House Mountain |
| Intel GbE disconnected or in D3. | Power House Mountain |
| PCIe, SATA, and USB PHYs power gated. | Power House Mountain |
| Power Consumption & DRIPS | Windows Sleep Study |

If one of the items is not met, there are two possibilities:

1. It is a specific component, such as Intel ME not gated, Intel Audio, then we need to investigate with component owner.
2. It is a bus or system behavior, such as CPU not enter C10 or xHCI controller not entering D3, then we need to involve PE.

By using DAQ, we can take power measurement over a period of time to isolate which component or rail is consuming excess power. We can then further isolate problematic component.

These are some of the possible root cause:

1. Component not behaving correctly during MSC (driver, component)
2. Component not power gated correctly (BIOS/EC, system board, component)

**Appendix A**

* **Checklist**

|  |  |
| --- | --- |
| **Item** | **Owner** |
| **Pre-DB** | |
| Target MSC battery life | Marketing |
| Component Selection to meet MSC battery life | EE – Architect |
| Power Budget by component | EE – Power Architect |
| MSC Power Map | EE - Architect |
| Device Drivers Meet Microsoft MSC Requirements | Component owner |
| System Bios meet Microsoft ACPI Design Guide for SOC platforms | Platform BIOS |
| **DB** | |
| Hardware and BIOS team review platform implementation | EE/BIOS |
| Enable MSC | BIOS/Component Owners |
| Measurement of floor power and average power by component | EE/MSC/PE |
| Intel CPU/Chipset (PCH) Low Power Enablement (see table 4-4 / Section 3.2.2.3) | EE/MSC/PE/Component Owners |
| **SI** | |
| MSC 100% functional (wake source, component behavior, etc) | MSC team/Component Owner/HW/BIOS |
| Floor Power meets Power estimate (compare to power budget) | MSC team/Component Owner/HW/BIOS |
| Average power meets marketing requirement | MSC team/Component Owner/HW/BIOS |
| **PV** | |
| Code Freeze Image meets power consumption requirement | MSC team/Component Owner/HW/BIOS |
| Address P1 issues from Dogfood program | MSC team/Component Owner/HW/BIOS |

* **Example Power Budget - Meritage**

[**https://hp.sharepoint.com/:x:/r/teams/CMITSE/Architecture/Modern\_Standby/Modern%20Standby%20and%20Low%20Power%20Enabling%20BIOS%20Checklist/Meritage\_CS\_Power.xlsx?d=w1838280a620b4225935dd93fd445a1e7&csf=1&e=5Doab1**](https://hp.sharepoint.com/:x:/r/teams/CMITSE/Architecture/Modern_Standby/Modern%20Standby%20and%20Low%20Power%20Enabling%20BIOS%20Checklist/Meritage_CS_Power.xlsx?d=w1838280a620b4225935dd93fd445a1e7&csf=1&e=5Doab1)

* **Example Power Map – Bazooka Jr.**

[**https://hp.sharepoint.com/:i:/r/teams/CMITSE/Architecture/Modern\_Standby/Modern%20Standby%20and%20Low%20Power%20Enabling%20BIOS%20Checklist/BJR\_SI2\_PowerMap\_PADs.png?csf=1&e=r2JPwH**](https://hp.sharepoint.com/:i:/r/teams/CMITSE/Architecture/Modern_Standby/Modern%20Standby%20and%20Low%20Power%20Enabling%20BIOS%20Checklist/BJR_SI2_PowerMap_PADs.png?csf=1&e=r2JPwH)

* **Example CS Component Selection - Broadwell**

[**https://hp.sharepoint.com/:x:/r/teams/CMITSE/Architecture/Modern\_Standby/Modern%20Standby%20and%20Low%20Power%20Enabling%20BIOS%20Checklist/Example\_CS\_Component\_Selection\_Broadwell.xlsx?d=w4c9a98c4f51e41f286d70da173c8fcee&csf=1&e=AVn01c**](https://hp.sharepoint.com/:x:/r/teams/CMITSE/Architecture/Modern_Standby/Modern%20Standby%20and%20Low%20Power%20Enabling%20BIOS%20Checklist/Example_CS_Component_Selection_Broadwell.xlsx?d=w4c9a98c4f51e41f286d70da173c8fcee&csf=1&e=AVn01c)

* **Intel – Runtime D3 recommendations**

<https://hp.sharepoint.com/:b:/r/teams/CMITSE/Architecture/Modern_Standby/Intel_Design_Whitepapers/567655_ICL_RTD3_SW_HW_Recomm_DG_Rev0p9.pdf?csf=1&e=BGTfRL>

* **Microsoft Design Papers**

[**https://hp.sharepoint.com/teams/CMITSE/Architecture/Forms/AllItems.aspx?newTargetListUrl=%2Fteams%2FCMITSE%2FArchitecture&viewpath=%2Fteams%2FCMITSE%2FArchitecture%2FForms%2FAllItems%2Easpx&id=%2Fteams%2FCMITSE%2FArchitecture%2FModern%5FStandby%2FMicrosoft%5FDesign%5FPapers**](https://hp.sharepoint.com/teams/CMITSE/Architecture/Forms/AllItems.aspx?newTargetListUrl=%2Fteams%2FCMITSE%2FArchitecture&viewpath=%2Fteams%2FCMITSE%2FArchitecture%2FForms%2FAllItems%2Easpx&id=%2Fteams%2FCMITSE%2FArchitecture%2FModern%5FStandby%2FMicrosoft%5FDesign%5FPapers)