White Paper

# Power Efficiency – Analysis and SW Development Recommendations for Intel® Atom™ based MID platforms

**Intel® Software & Services Group**

**Petter Larsson**

Contributions by:
Ajaya Durg, Adarsh Jain, Srinivas Sripathi, Bob Steigerwald, James Rose, Padmashree Apparao

Explore power characteristics of the Intel® Atom™ platform geared for the MID (Mobile Internet Device) category of devices. Providing recommendations for SW developers on how to best optimize applications for power efficiency for this new line of Intel processors/platforms. Some topics also apply to Intel® Atom™ processor based NetBooks.

***March 2009***

**Intel Corporation**

# Contents

# 1. Background

The objective of this paper is to investigate Intel® Atom™ based MID (Mobile Internet Device) platform power (including chipset components) characteristics for typical workloads, such as media playback, browsing, idle, etc. The results are aimed at providing recommendations for SW developers on how to best optimize applications for power efficiency.

The platform power characteristics were captured using two methods:

- Instrumented Crown Beach/Menlow Software Development Platform (SDP) (including Intel® Atom™ processor and chipset) enabling discrete component power measurements using Fluke NetDAQ* equipment
- Linux* tool, PowerTOP* (http://www.lesswatts.org/projects/powertop/), to extract information on processor C-state (sleep) residency, P-state (execution) residency and wakeup characteristics.

The Intel® Atom™ processor provides low power features such as:

- New core x86 micro-architecture
  - In-order execution
  - Multithreading support. Hyper Threading Technology (HT) aka. Simultaneous Multi-Threading (SMT). *Not available for all Atom SKUs.*
  - Enhanced macro-op execution
- 45nm technology
- Enhanced Intel® SpeedStep® Technology
- Up to SSSE3 instruction set support
- Deep Power-Down Technology - C6 processor sleep state
- Dynamic cache sizing
- Enhanced dynamic clock gating
- Enhanced L2 data pre-fetcher

The chipset provides features such as:

- Graphics HW acceleration (including video acceleration used by media player)
- Intel® Display Power Saving Technology
- Intel® Rapid Memory Power Management

The OS/SW stack used for this investigation is Moblin* Linux (http://moblin.org/) with the Helix* media framework (https://helixcommunity.org/) using HW accelerated video codecs. Other OS or SW stacks such as RedFlag*, MIDinux*, or Microsoft Windows* Vista are out of scope for this investigation.

The “Recommendations” chapter summarizes the findings from the analysis and provides guidelines on how to develop power efficient SW on the MID platform. The following chapters are divided into “Technical Background”, which describes some key technical concepts, “Measurement procedure”, which outlines the methods used to capture the data presented in this paper and “Workloads”, which breaks down results for each of the workloads analyzed.

# 2. Recommendations

From the Intel® Atom™ platform power data analyzed in the following chapters a number of recommendations were assembled on how to design SW for power efficiency on the MID platform.

In essence the goal is to maintain the performance while improving application power efficiency leading to extended device use time between charges.

## 2.1. Reduce application demands for processor wakeups

By minimizing the frequency (or coalescing) of wakeups, the processor is able to move to, and reside longer in, deeper sleep states such as C6, allowing lower average processor power. Using C6 deep sleep state the processor is able to operate at an average power of 80mW when idle.

Optimally, a power efficient application in idle shall showcase the same (or minimal difference) processor wakeups per second as when system is in idle (application not running). A basic approach to measure the above is measuring “power at the wall” or verifying application wakeup behavior using PowerTOP. PowerTOP provides a very good method for measuring the C/P state residency of the system and for finding main sources of wakeups.

Refer to chapter 5.1.1: Processor sleep behaviour” and “5.2: Video playback” for details.

## 2.2. Utilize an energy-efficient UI and framework technology

It is important to be aware of the UI power implications regardless if your application is native or if it runs on-top of a runtime environment or framework. If possible, when developing the application try to select a UI, runtime and framework that are energy efficient. For instance, an application based on Flash[1] will be less power friendly due to the high number of wake ups in idle and the runtime overhead.

Although somewhat OEM specific, if possible, utilize a home screen solution (application launcher) with low average idle power footprint such as an HTML or OpenGL based solution.

Refer to chapter “5.1.2: Processor & Chipset power usage in Idle modes” for details.

[1] Flash* version 9 was used in this study. Recent versions of Flash 9 and 10 have improved power characteristics.

## 2.3. Thread applications well, even for single core processors

When HT is enabled it will allow an additional thread to execute per processor core in case the first thread is stalled. This will in many cases improve performance for well threaded workloads. Additionally, threaded video workloads often experience increased frame rate, due to the increased performance.

The use of HT for threaded workloads increases average power but due to improved performance workloads will complete faster, resulting in lower net energy compared to executing the workload with HT disabled. In essence, if an application exploits multi threading well, energy can be decreased significantly, improving battery life.

Observe that some of the Intel® Atom™ processor SKUs do not feature HT.

Refer to chapter "5.3: Benefits of HT on threaded workloads": Benefits of HT on threaded workloads" for details.

## 2.4. Media Workloads: Take full advantage of hardware accelerated codecs

Media frameworks such as the Helix framework, provide HW accelerated codecs for video standards such as H.264. HW acceleration greatly improves average power footprint and device up-time. Another benefit with HW acceleration is that processor is offloaded allowing it to perform other tasks.

Higher definition video scales well and does not significantly increase average power footprint of chipset and processor. Although, it is important to note that the average power required by memory increases rapidly for higher definition content due to greatly increased memory access.

It is not recommended to use SW codecs for Intel® Atom™ based platforms as the processor gets maxed out even at low resolutions (480p and below) resulting in very high average power.

Refer to chapter ”5.2: Video playback” for details.

HW accelerated codecs for the Linux MID platform are available through the Helix community, https://rp4mid.helixcommunity.org/.

## 2.5. General power efficiency recommendations

a. Be aware that one power inefficient application in Idle is enough to cripple battery life for the whole system.

b. Develop context aware applications that adapt to system state changes
Refer to http://software.intel.com/en-us/articles/energy-efficient-software-developing-power-aware-apps for articles on developing for context awareness.

    1. Power state change (system sleep/wake transitions): The application shall handle system power state change gracefully and not incur unnecessary delays. The application shall not prevent system from changing power state unless absolutely necessary.

    2. Battery state change: Adapt behavior due to switch from AC to/from battery power. Some features, such as automatic updates, could potentially be deactivated to save power when powered by battery.

    3. Network state change: Adapt application behavior to network connect/disconnect events. Some features could potentially be deactivated, saving power when not connected to network (flight mode).

c. Develop “Data Efficient” applications that minimize data movement to/from external storage or RAM. Also investigate if data movement can be grouped or batched to decrease frequency of drive spin up/down.

d. Utilize extended SIMD instruction sets such as SSE3, improving performance and indirectly reducing application energy usage.

e. Utilize tools to achieve power optimizations or identify power optimization opportunities

    1. Utilize the Intel® Compiler for MID to compile the code into a binary specifically adapted to the Atom™ micro-architecture. This has the potential to greatly improve performance and in some cases also energy efficiency

    2. Utilize PowerTOP to find components waking up the system excessively

    3. Utilize Application Energy Toolkit to graph/analyze energy consumption

## 2.6. Platform configuration recommendations

Note that the following recommendations are somewhat OEM SW/OS specific and sometimes not applicable to application development.

a. Make sure the system is configured to power the screen off if there is no user input for a suitable time interval. This reduces the average power in the chipset and allows additional power savings from LCD power down.

Refer to chapter “5.1.2: Processor & Chipset power usage in Idle mode” for details.

b. Using on-demand governor to regulate P-state, depending on workload, improves system power efficiency.

## 2.7. Suggested power optimization process

a. Optimize to meet performance goals

b. Establish baseline before power optimizations

    1. Gather data using PowerTOP in single threaded mode (Powertop currently not as accurate in HT mode)
        a. Wakeups/s
        b. Cx state residency
        c. Px state residency

    2. If possible utilize external power meter to measure average power during application idle compared to system idle and during key workload execution

c. Investigate behavior during application idle compared to system idle. Explore difference in wakeups/s, Cx/Px states. Optimally an application in idle should have none or minimal impact on total system idle power

d. Use Intel® VTune™ for MID to identify functions that show high C0 residency running key application workloads. Utilize processor “unhalted ref clock” counter to assess C0 residency. Note that “Unhalted core clock” is unreliable due to frequency change (if Intel® SpeedStep® Technology is enabled) and that “Time stamp counter” is unreliable at and below C4 due to disabled PLL. “unhalted ref clock” / wall clock time can be used to compute true C0 residency.

e. Re-architect/re-code
    1. Look for activities that force processor into C0 state (interrupts)
    2. Coalesce or remove unnecessary activities to reduce wakeups
    3. Look into data efficiency bottlenecks (disk, memory, network)
    4. Note that optimizing for speed may increase C0% which might be OK since overall energy reduces due to performance speed up

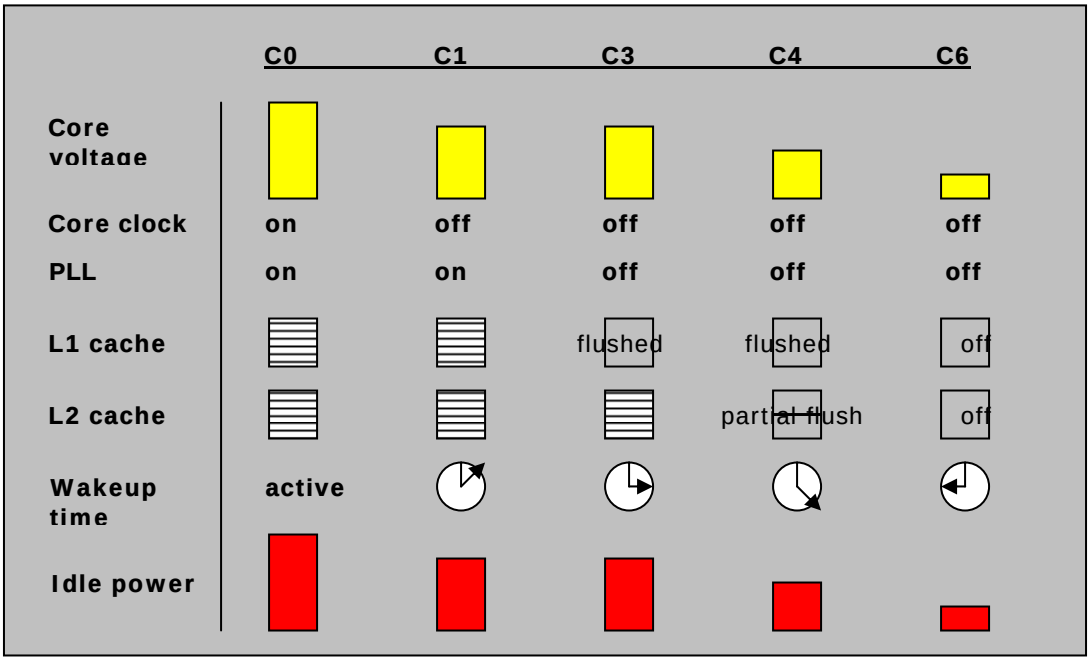
For further details on the above and related topics please find detailed articles at Intel Software Network (ISN): Energy Efficient Software http://software.intel.com/en-us/articles/energy-efficient-software/

# 3. Technical Background

This chapter explains some of the technologies in focus for this paper.

## 3.1. C6 - Deep Power Down State

The Intel® Atom™ processor features a new deep sleep state named C6. The new sleep state enables the processor to move into deeper sleep during inactive intervals between executions. The benefit of the new state is a much lower average power in idle mode in comparison to the C4 sleep state.




The above illustration shows the impact on the processor when C6 is active. For instance in deep power down the core voltage is substantially decreased (~0.3V) and the caches are turned off (flushed) leading to significant power savings. Observe that the bars do not depict the exact values but illustrate the value relative to other states.

The deeper sleep state has a side effect in that the processor has a much longer wake-up time compared to wake-up from C4. Depending on the frequency of processor wakeups the benefit of C6 will vary. On average, if the processor has ~200 wakeups per second the benefits of C6 are lost due to the latency/power cost sleep state transitions. For clear benefits of C6 we recommend <100 wakeups per second.

The applications running on the platform are very much affecting the frequency of the processor wakeups. For instance, by carefully selecting periodic timers with longest possible interval, the application allows the processor to move to and stay longer in deeper sleep states.

## 3.2. Hyper-Threading Technology (HT)

Hyper-Threading Technology (also Simultaneous Multi-threading (SMT)) improves processor performance under certain workloads by providing useful work for execution units that would otherwise be idle. The benefit is especially useful for a single core processor such as Intel® Atom™, as

it enables execution of an additional thread while execution is stalled for first thread, for instance due to a cache miss.

Observe that the OS scheduler perceives the processor as having two cores when HT is enabled.

HT may improve the performance for multithreaded code running on the processor. The exact performance benefit depends on the specific workload and on how well threaded the code is.

# 4. Measurement Procedure

Note that this paper does not analyze all component parts of a typical MID form factor device. MID devices will feature a range of wireless communication components such as WiFi, WiMax, WWAN or BT all increasing the average platform power footprint. Extended analysis of form factor devices might include networked workloads such as audio/video streaming, Chat or Email, VOIP, P2P solutions, connected browsing and gaming.

## 4.1. System setup

The system BIOS menu provides access to settings for several key processor features such as turning HT on/off, selecting “deepest” sleep state (C4 and C6 used for this study).

After system start-up, the “on-demand” power governor was activated, enabling the processor to throttle P-state depending on the performance needs of the workload.

Refer to chapter 7: “Appendix B” for details on system configuration and setup.

## 4.2. NetDAQ measurements

Using Fluke NetDAQ equipment we were able to measure voltage and current for discrete components on the instrumented Crown Beach/Menlow platform board. The board was instrumented by connecting wires from 4 NetDAQ measurement modules to sense resistors on the board. This enabled us to capture voltage and current for components such as processor, chipset, RAM, PATA and PCI-bus etc. and calculate the power usage per component or group of components.

Several measurements for all workloads were collected, abnormal captures omitted and most common representative collection of measured data selected.

Refer to chapter 7: “Appendix B” for details on NetDAQ setup.

## 4.3. PowerTOP measurements

PowerTOP (http://www.lesswatts.org/projects/powertop/) is a Linux tool that measures how well the system uses various hardware power-saving features. It also highlights culprit software components that prevent optimal usage of hardware power savings and provides tuning suggestions. Furthermore the tool provides measured C (sleep) and P (execution) state residency, and the number of wakeups/s for the processor as a whole and for top software component culprits.

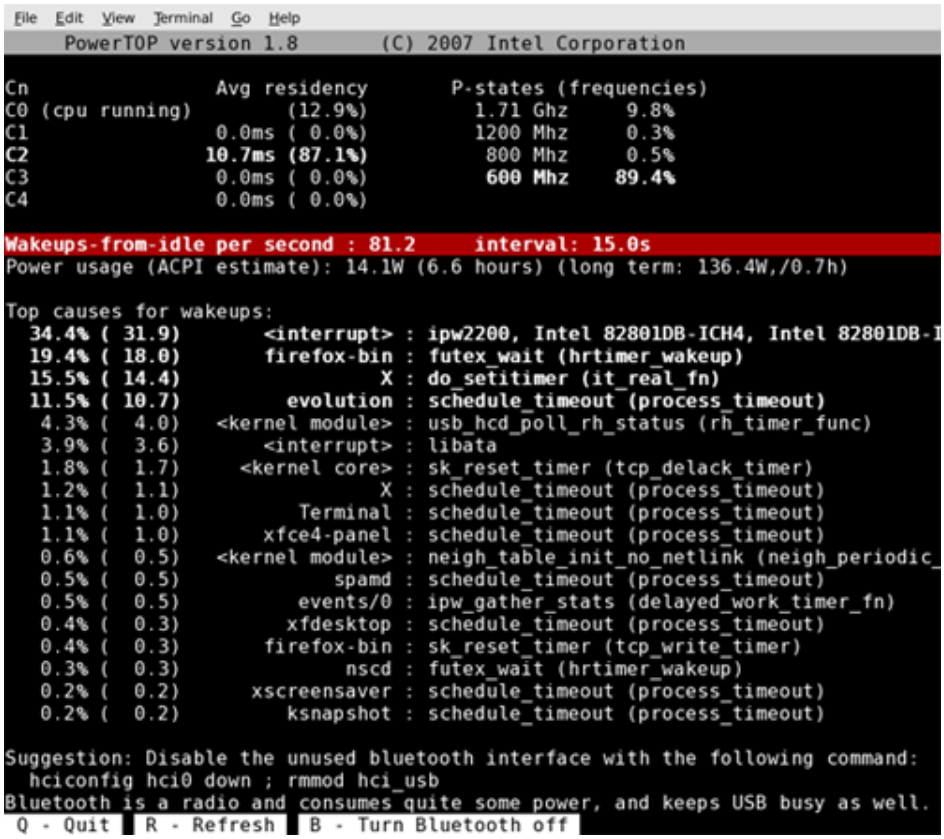
For this investigation some automated scripts were created to launch the workload and then capture PowerTOP data over an interval of time. Several measurements for all workloads were collected, abnormal captures omitted and most common representative collection of measured data was selected.

The PowerTOP measurements were not synchronized with measurements gathered using the NetDAQ, as a direct correlation technique was not available.

To examine the actual processor C-state residency it is important to understand the mapping between the ACPI C-state (presented by PowerTOP) and the actual processor C-state. Unfortunately ACPI in its current form does not provide greater granularity. Refer to the below table for mapping ACPI C-state to processor C-state (depending on lowest sleep mode selected in BIOS).

| **ACPI: C-state** | **Processor: C-state (with C6 activated in BIOS)** | **Processor: C-state (with C6 deactivated in BIOS)** |
|---|---|---|
| **C1** | C1 | C1 |
| **C2** | C2 | C2 |
| **C3** | C4 | C4 |
| **C4** | C6 | - |

Below is an example of a typical PowerTOP dialog, where C0 is computed by PowerTOP from the captured C1-C4 (ACPI) data.




*In the above example, PowerTOP indicates that, over the measurement interval, the system was in C2 87.1% of the time and at the lowest P-state (600MHz) 89.4% of the time. Wakeups per second during the interval averaged 81.2. PowerTOP also suggests that turning off Bluetooth will save power.*

# 5. Workloads

The following MID workload categories were analyzed:

1. Idle behaviour in typical system idle states
2. Video playback using Moblin Media Player & Helix framework
3. Multi-threaded Video decode, audio transcode and Flash workloads used for HT impact analysis
4. Browsing using Moblin Browser

All workload measurements are performed in steady state unless otherwise noted. Linux* kernel version 2.6.22 was used.

## 5.1. Idle modes

Power data was captured for different processor configurations (modified via BIOS):

- HT on/off
- C6 on/off

The following idle modes were analyzed:

1. Home Screen – HTML
2. Home Screen – HTML – Screen off
3. Home Screen – Clutter (OpenGL)
4. Home Screen – Flash (UI created in Flash, embedded in HTML)
5. XTerm
6. Moblin Browser with default home page loaded

*Note: As the Flash Home Screen feature was broken, no Flash application icons were visible just a Flash shell. This was considered an acceptable approximation. After launch, a regular terminal shell was invoked (ctrl-alt-F1), the "on-demand" governor activated and measurements were made.*

*Observe that this is by no means an exhaustive list. Additional measurements are needed to provide a more complete overview of possible application/launcher user interfaces.*

### 5.1.1. Processor sleep state behaviour

The following NetDAQ data was captured with HT turned on. C6 or C4 (C6 off) was configured as the lowest possible processor sleep state.

Below data was captured for when in idle on HTML Home Screen.

| C-State Configuration | Average CPU Idle Power |
|---|---|
| **C6 off** | ~160mW |
| **C6 on** | ~80mW |

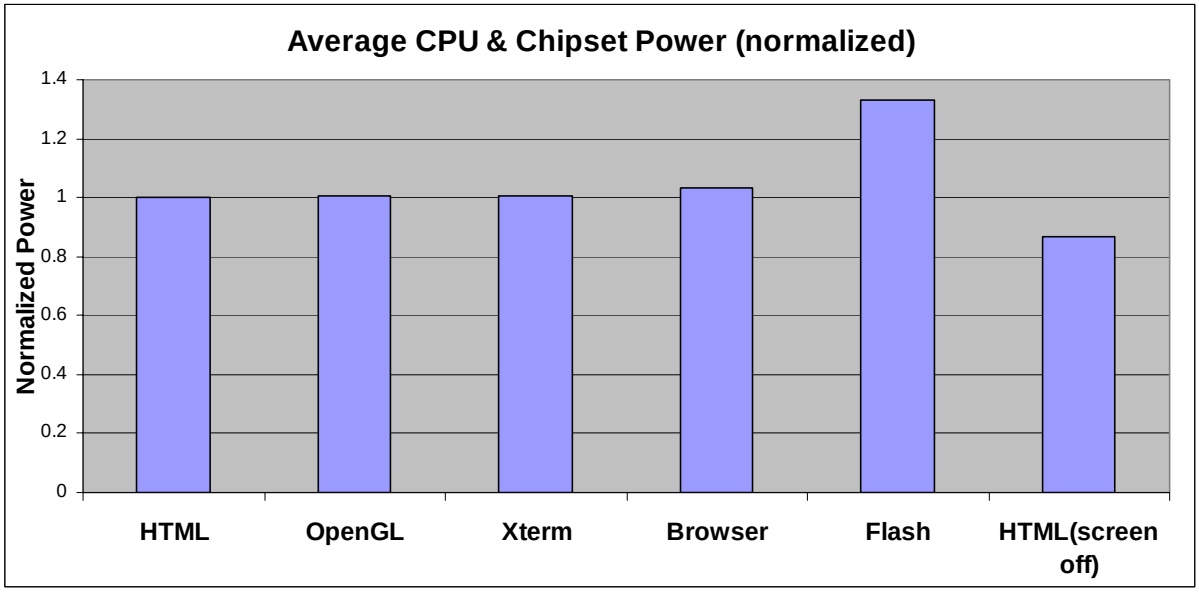
Note the ~80mW difference in average power for C6 on vs. off. It is clear that the C6 state has an important positive impact on average power when processor is experiencing low load or is mostly in idle. In essence the new C6 deep sleep state significantly reduces average Intel® Atom™ processor power in idle.

Refer to chapter 6: "Appendix A" for details on Intel® Atom™ sleep states and their characteristics.

### 5.1.2. Processor & Chipset power usage in Idle modes

The following NetDAQ data was captured with C6 and HT turned on. Idle power behaviour for the different Home Screens (application launchers) was compared to the Browser and Xterm in idle.

The Home Screens tested represent various UI technologies used for displaying application icons and launching applications. The UI technologies tested was HTML, OpenGL (Clutter) and Flash[1].




The above graph illustrates the normalized power behaviour for the 6 targeted idle workloads.

Contrary to what might be expected the Clutter home screen (OpenGL) does not lead to increased chipset average idle power. The HTML and OpenGL UI home screen solutions used are quite power friendly.

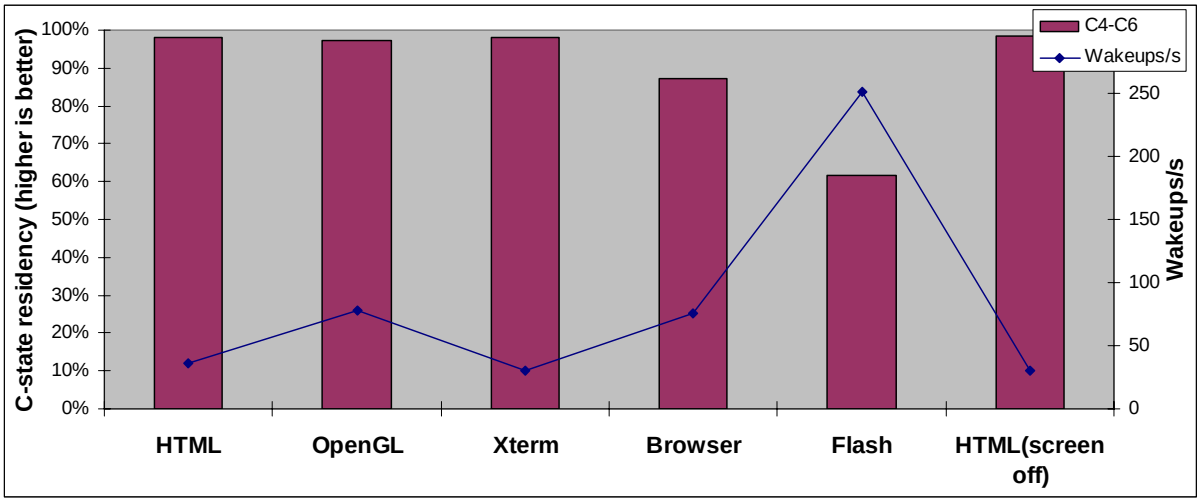
Automatically turning the screen off after an interval of no user input lowers chipset average power significantly (~13.5% below HTML idle). More importantly, powering off the screen also saves LCD power.

Moblin browser in idle mode on the default home page consumes a slightly higher average power for processor/chipset (~3.5 % above HTML idle). Observe that the default page did not have any advanced content such as Flash* or Ajax*, etc.

When idle on Flash home screen a completely different pattern is revealed. Due to the high number of interrupts, 250 wakeups/s (captured by PowerTOP, see below), the benefit of C6 sleep state is not fully utilized (even though the processor sometimes moves into C6). This is apparent from the much higher average power usage for processor and chipset (33% above HTML idle).

An alternative view of the power behaviour is available from the data captured by PowerTOP. Wakeup/s and deeper C state residency is captured in the graphs below.

[1] Flash version 9 was used in this study. Recent versions of Flash 9 and 10 have improved power characteristics.




From the above data it is easy to see the impact of a high number of wakeup/s on processor deep sleep state residency. For instance, the Flash home screen wakes up the processor ~250 times/s resulting in ~60% of the time spent in C4-C6, while the HTML home screen wakes up the processor 35 times/s resulting in ~98% of the time spent in C4-C6 resulting in significant power improvements.

Recent measurement using latest versions of Flash 9 and 10 reveals an improved Flash wakeup pattern resulting in improved power characteristics. Still, even during playback of the simplest Flash content with a frame rate of 1 fps, the wakeups per second in idle does not move below ~100 wakeups/s.

For all workloads, the processor spent > 95% in the lowest frequency mode (LFM, 800MHz) execution state (P-state).

## 5.2. Video playback

Power measurements were captured while Moblin Media player (utilizing the Helix framework) played back video of the following formats:

| Video standard | Fps | Rate (kbps) | Resolution(WxH) |
|---|---|---|---|
| **CIF** | 30 | 510 | 352x288 |
| **480p** | 30 | 1480 | 720x480 |
| **720p** | 30 | 4400 | 1280x720 |
| **1080p** | 30 | 9800 | 1920x1080 |

The media was encoded with H.264 for the video stream and AAC for the audio stream. Furthermore media playback with and without Helix HW acceleration was measured.

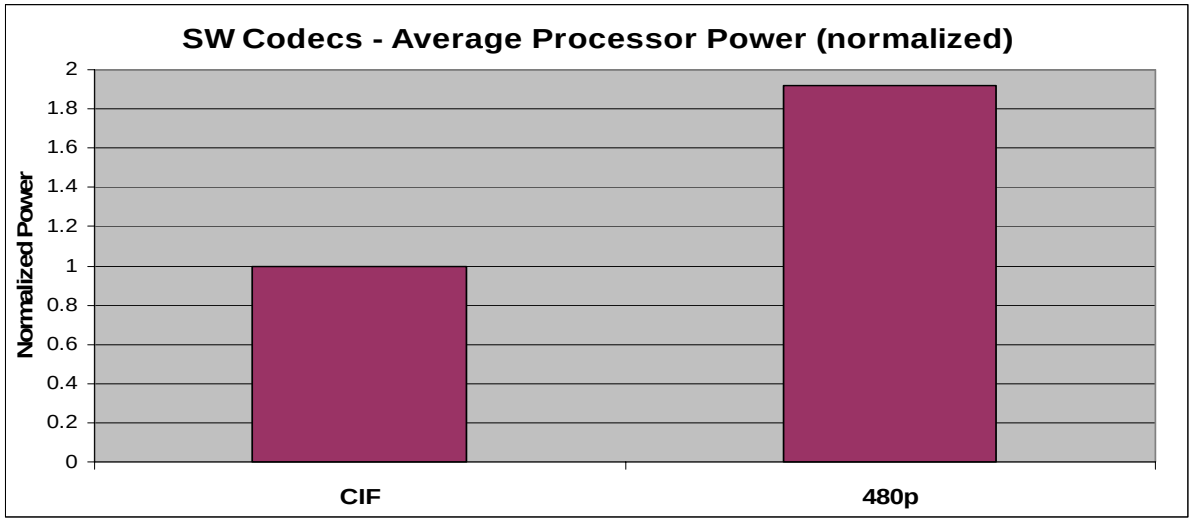
Observe that 1080p is only supported for some of the Intel® Atom™ SKUs.

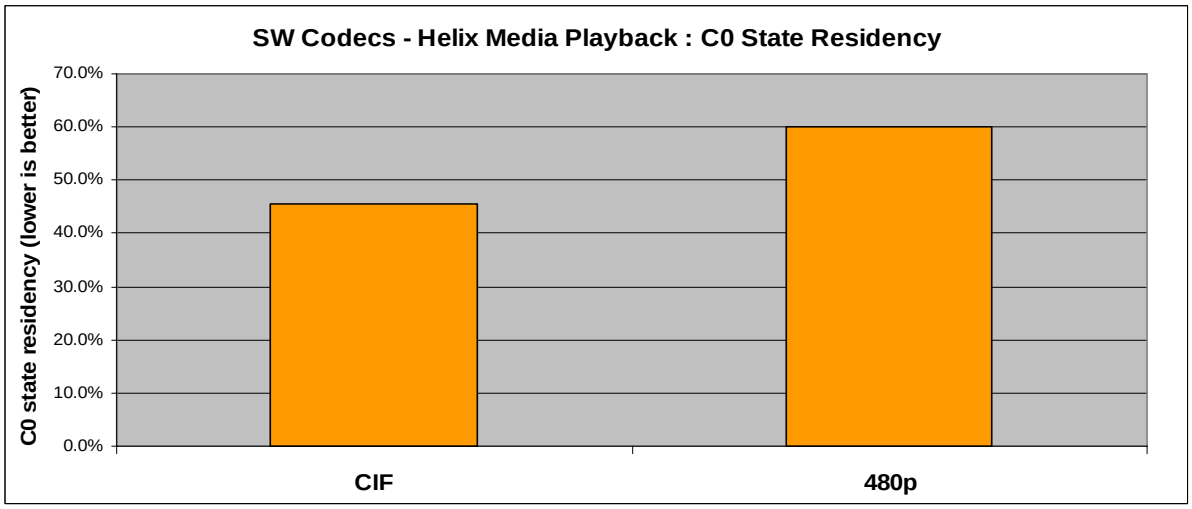
Note that other media frameworks such as GStreamer* also feature HW acceleration of video for the Intel® Atom™ platform.

NetDAQ data was captured with C6 turned on and HT was toggled on/off depending on the target measurement.

### 5.2.1. SW codecs vs. HW accelerated codecs

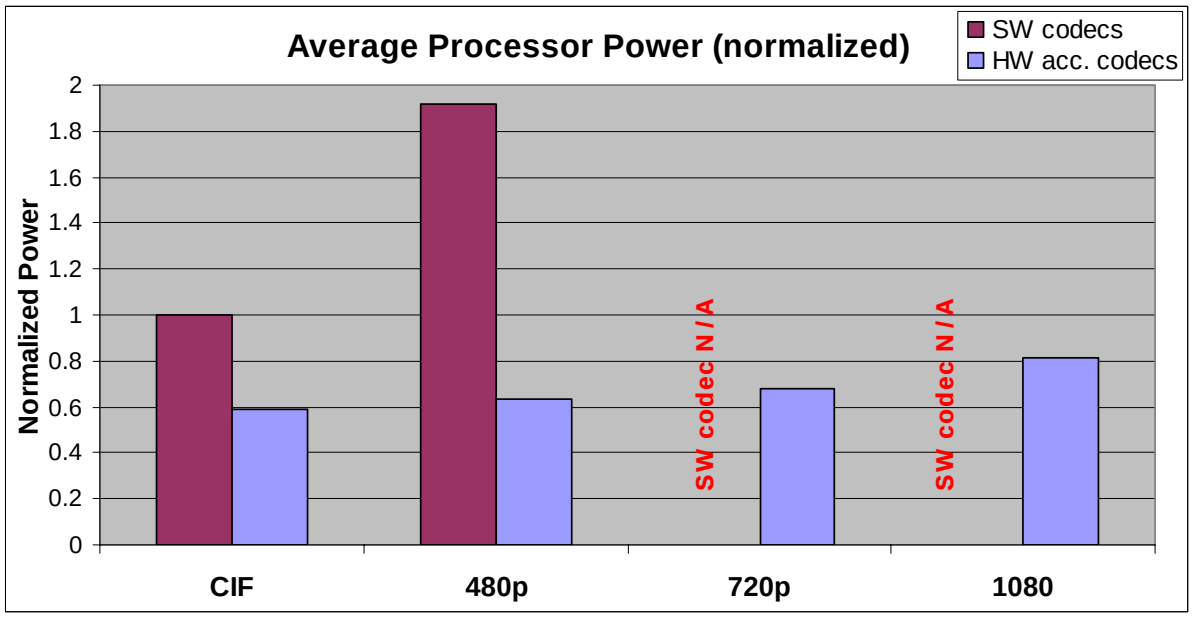
The normalized graphs below compare average normalized power and C0 state residency during SW codec media playback.

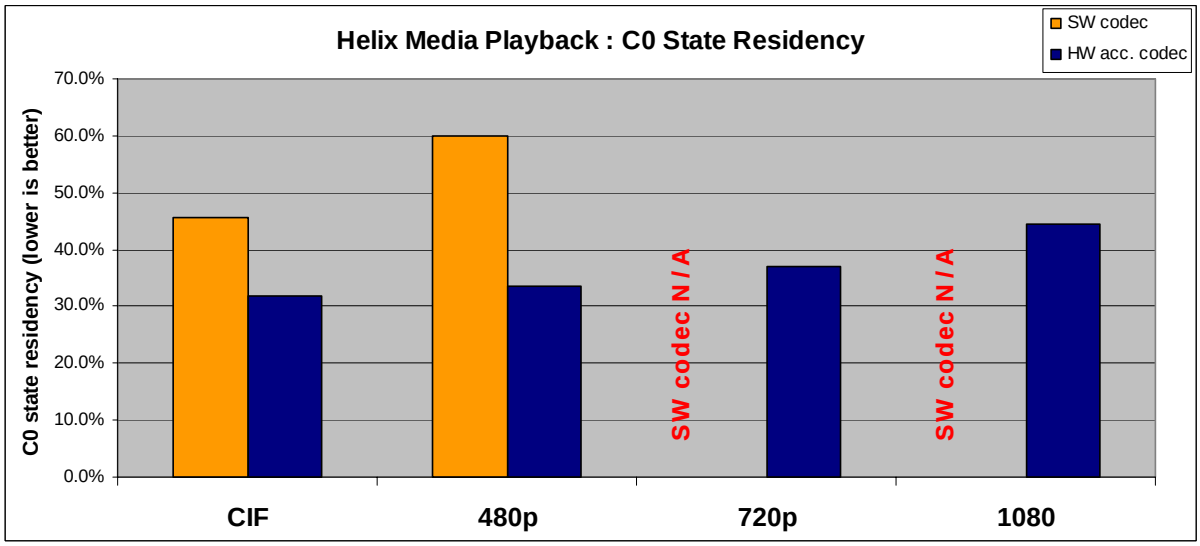



Decoding 480p (~3.5x more data than CIF handled) results in almost 2x in average processor power compared to CIF. Note that in using SW codecs, the processor is heavily utilized even for low resolution playback, as can be seen from the C0 residency graph.

There is clearly a need for HW acceleration to allow playback of higher resolution video workloads. The following graph compares average normalized power and C0 state residency during playback with SW codecs vs. playback using HW accelerated codecs.




From the above graphs it is clear that HW accelerated codecs have great benefits with regards to average power during high definition video playback. The required processor power scales gracefully for increased video resolutions. For instance, using HW acceleration, the platform is able to process 20x more data playing back 1080p vs. CIF with only a minor increase (~25%) in processor power. The average chipset power using HW acceleration also scales well, as will be illustrated below.

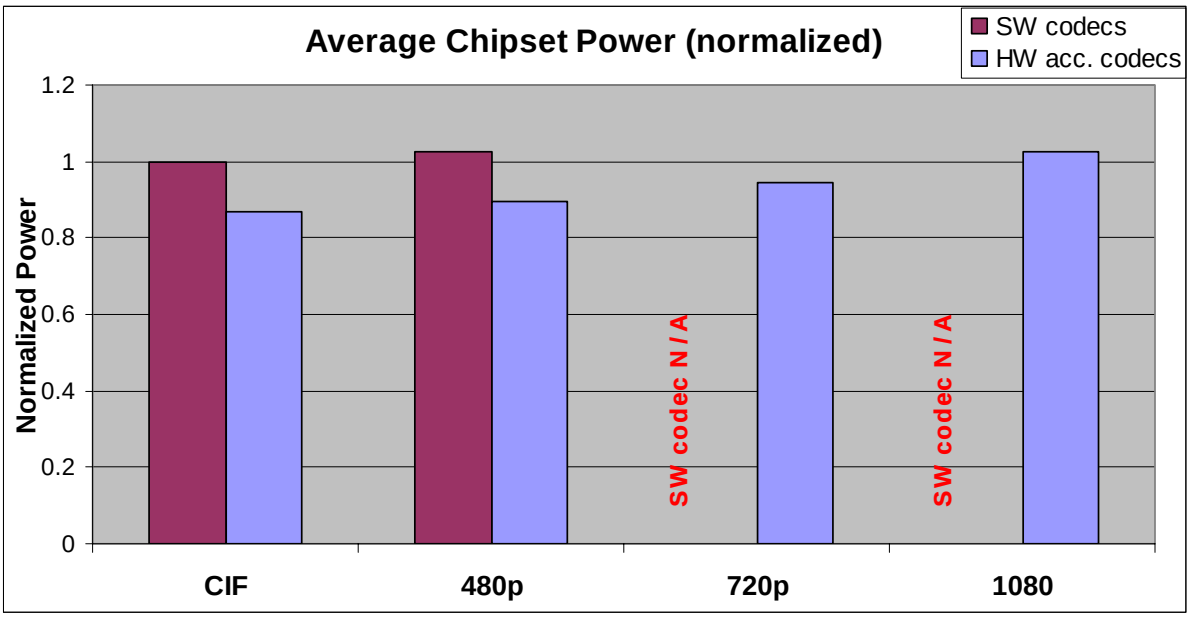
With regards to C0 state residency the processor for high resolution playback using HW acceleration, the processor shows moderate utilization.

***Note: The release of the Helix framework used was not optimized for the MID platform and some kernel bottlenecks were identified. These issues have been addressed in more recent Moblin releases leading to much fewer wakeups (reaping the benefits of C6 sleep state) and much improved C0 state residency all leading to lower average power footprint for HW accelerated video playback.***

Note that using SW codecs, media with resolution greater than 480p cannot be played back due to performance limitations.

For all HW accelerated workloads the processor spent > 90% in the lowest frequency mode (LFM, 800MHz) P-state. Using SW codecs, the P-state residency indicates very high processor load. For instance, during 480p playback the processor spent just 3% in LFM.

The normalized graph below compares average chipset for HW accelerated codec media playback vs. SW codec media playback.

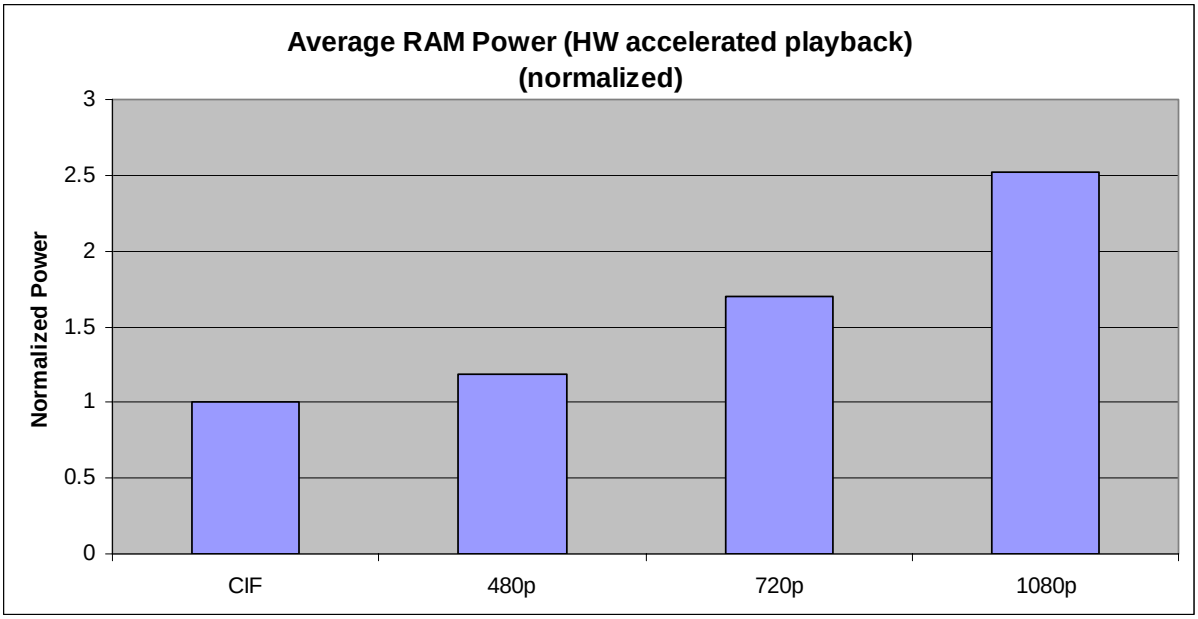



From the graph is clear that not only does HW acceleration enable playback of 1080p, it also improves the average chipset power. As can be seen in the graph the chipset average power used scales gracefully for higher resolution content. Also note that playing back 1080p using HW acceleration requires about the same average chipset power as playing back CIF content using SW codecs.

Data collected with PowerTOP indicates that the processor wakes up 300-600 times/s during video content playback, depending on the media format. The processor therefore has very limited benefits of the C6 sleep state.

### 5.2.2. Memory load impact on power

Another important aspect of media playback is how frequent data is read/written to memory. Large volumes of data transferred to/from memory results in increased average system power. The normalized graph below illustrates the increased playback power use for various definitions of video content.




Platform memory subsystem uses on average 2.5x more power for 1080p playback vs. CIF playback.

The RAM is exercised approximately to the same degree for both HW accelerated playback and playback using SW codecs.

Note that several memory access improvements have been introduced to the framework and recent Linux kernels such as 2.6.28 have lead to improved average RAM power.

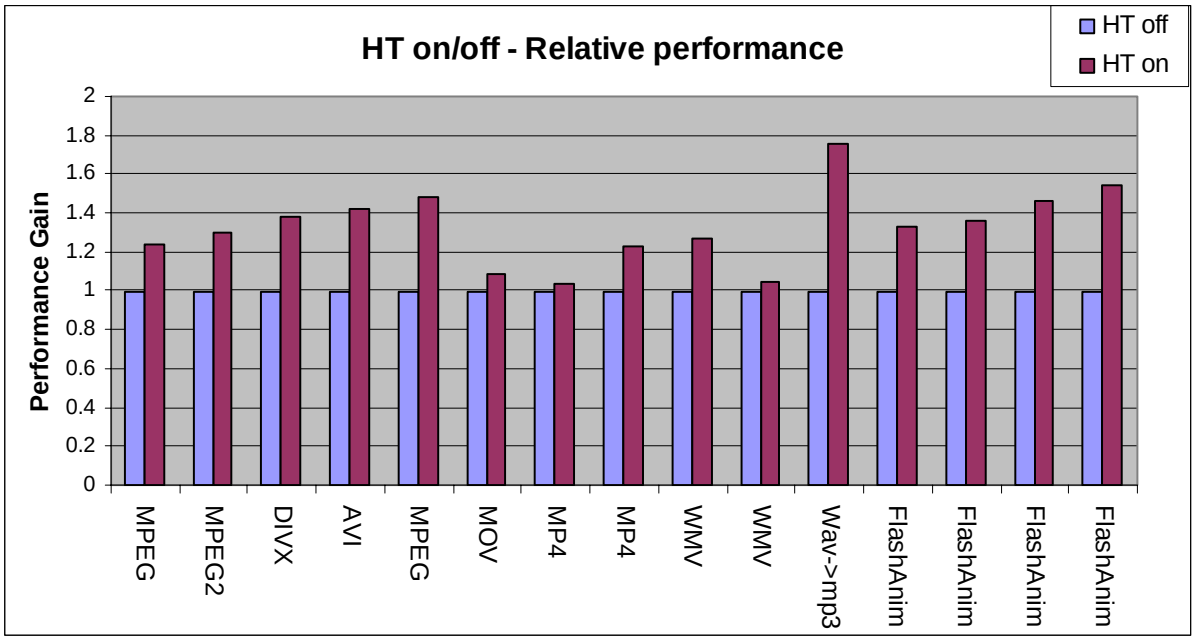
## 5.3. Benefits of HT on threaded workloads

Below processor data was captured with HT turned on/off while running various multi-threaded SW video decode and audio transcode workloads. Observe that HW acceleration was not used for the following workloads.

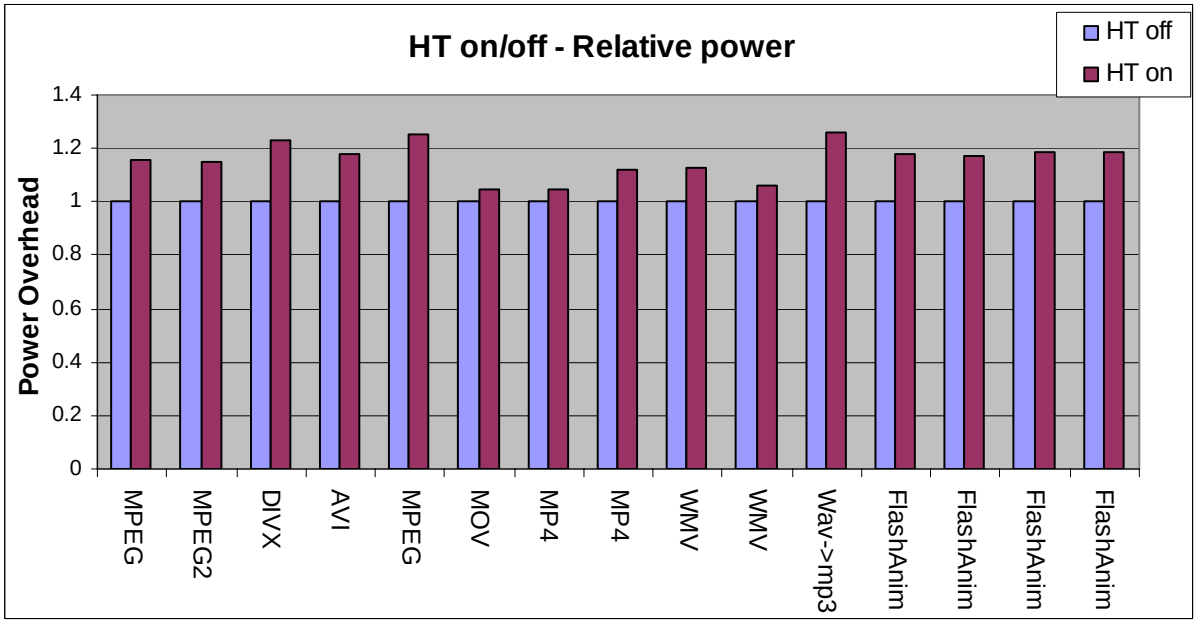
The following three graphs show typical relative processor performance, power and energy for the workloads with HT turned on/off. Analysis included a range of video workloads with various resolutions, one audio workload (transcoding “wav” to “mp3”) and four Flash[1] animation (no Flash video) workloads. The workloads tested were all multithreaded to take advantage of multithreaded processor architectures.

Decoding was performed at highest possible rate (disregarding specified media rate) completing workload as fast as possible. When HT was turned on this generally resulted in higher power during workload execution, faster completion and thereby overall energy savings.

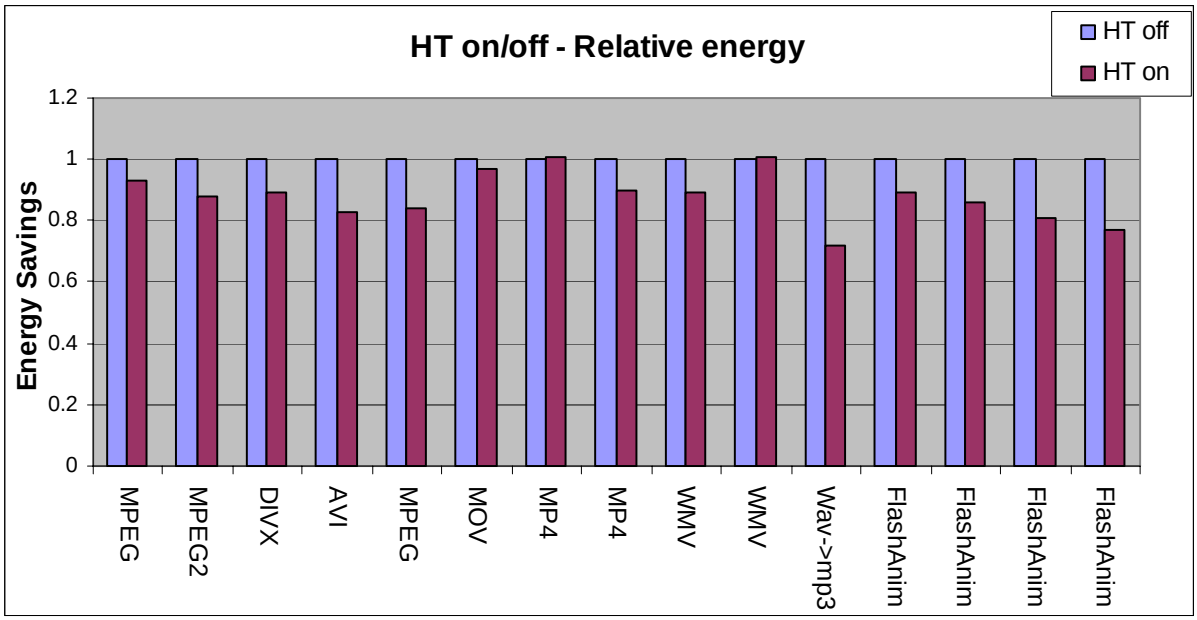
[1] Flash version 9 was used in this study. Recent versions of Flash 9 and 10 have improved power characteristics.




The graph clearly shows the performance benefit of the HT feature. Over the measured workloads we see a 32% geomean performance gain. The gain in performance naturally comes at an expense of average power. The graph below details the relative power overhead when HT is turned on/off.



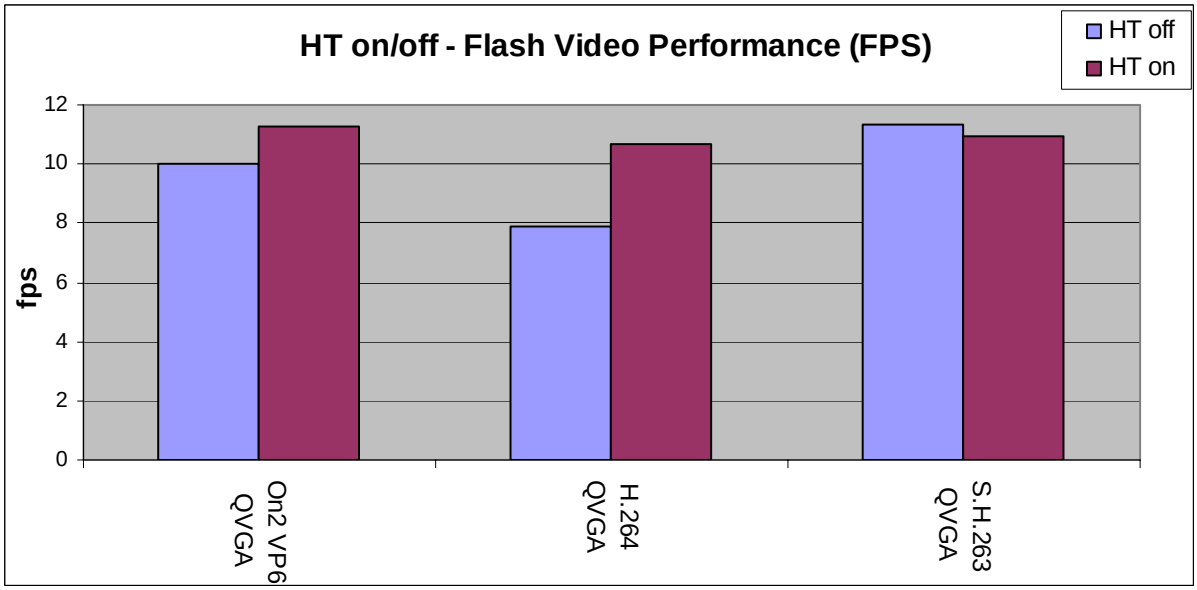

From the measured data we see a geomean power overhead of 15%. Due to the increased performance the workloads generally completed faster which has an impact on the energy used by the processor. Below graph details the calculated processor energy benefits of the HT feature.




From the above graph we see 14% energy savings for the measured workloads.

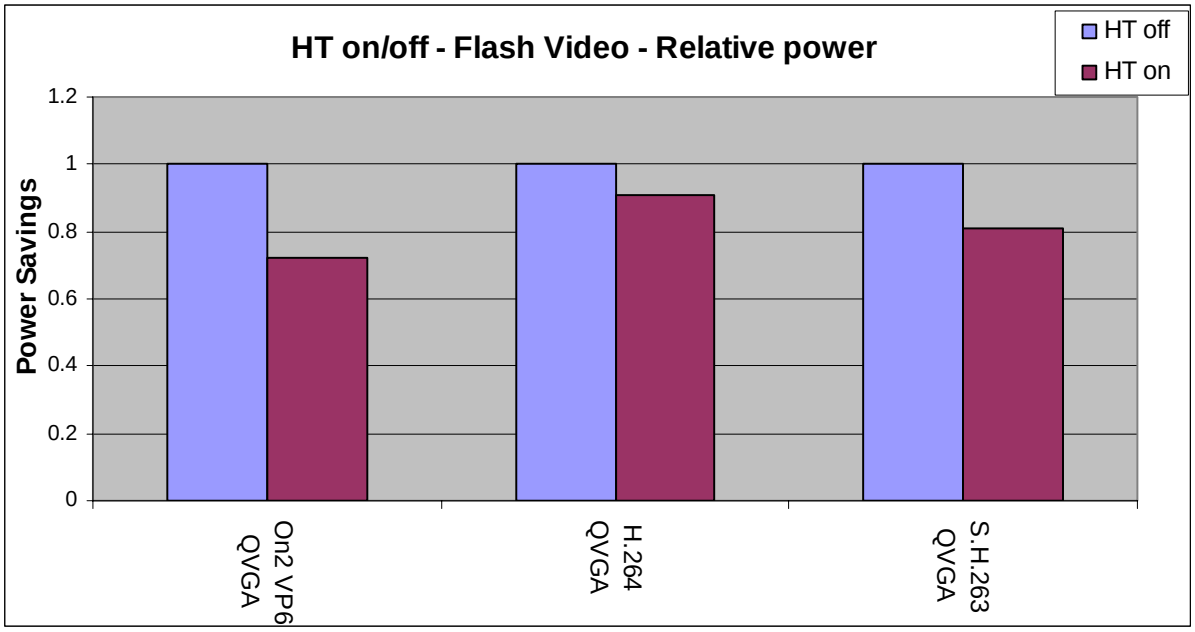
The benefits of increased performance can also be seen for workloads such as Flash[1] video playback. Contrary to former workloads the following workloads do not complete faster with HT turned on. Instead the increased performance gained by HT enabled translates into an increased frame rate (Note that Flash will play back the media at highest possible frame rate, up to the specified media frame rate).

The below graph shows measured frame rate for three different Flash video workloads with HT turned on/off.




[1] Flash version 9 was used in this study. Recent versions of Flash 9 and 10 have improved power characteristics.

From the above graph we measured a geomean frame rate gain of 14%. Besides the frame rate gains the use of HT also improves average power as the below graph illustrates.




From the measured data we found a geomean power saving of 19%, mainly due to extended time spent in lower P-state.

The reason for the decreased power, for Flash video, with HT enabled is due to the increased ability to move to lower P-states. Note that this behaviour is dependent on power management policy and workload.

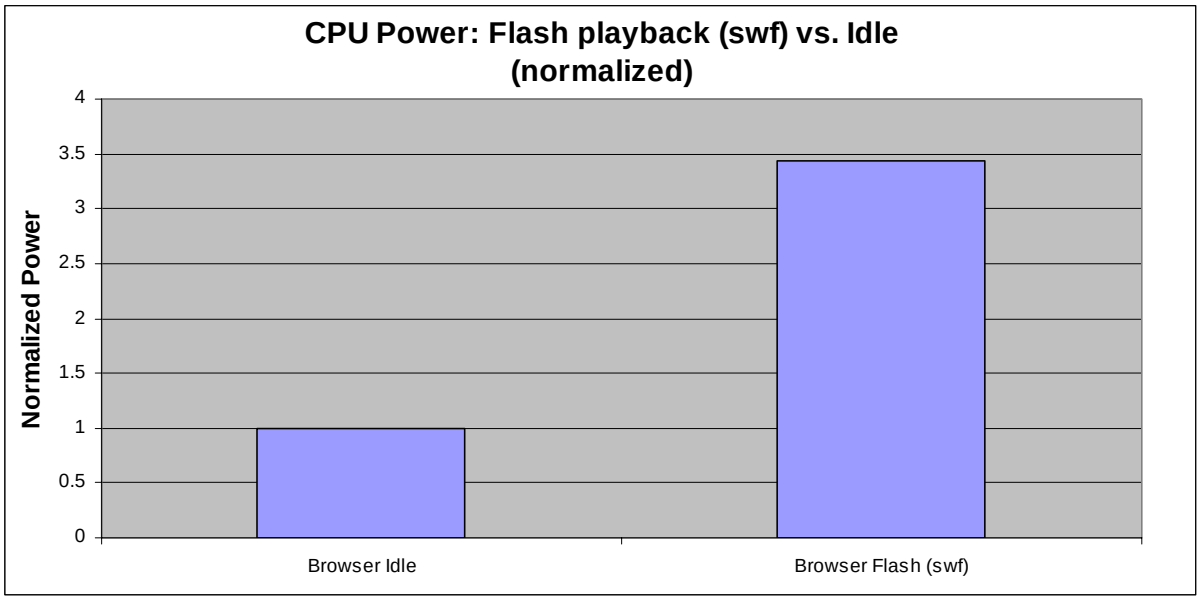
In summary, for the workloads tested, the overall performance gain with HT activated compared to HT disabled was ~32% (ranging from 4-76%) while the overall power overhead with HT activated compared to HT disabled was ~15% (ranging from 5-26%). Due to the increased performance, with HT activated, the workload completion time was shortened leading to a net energy savings of ~14% (ranging from 0-28%). Flash video workloads showcase a frame rate gain of 14% and power savings of 19%.

## 5.4. Browsing

The following NetDAQ data was captured with both C6 and HT enabled.

Moblin browser, based on Firefox 3, was used. A very simple Flash[1] content (swf) media file was opened up in browser. Flash content, displaying small flashing text was measured in “idle”. The processor and chipset power was measured and compared to Browser idle power behaviour.

[1] Flash version 9 was used in this study. Recent versions of Flash 9 and 10 have improved power characteristics.




From the above normalized graph it is clear that even for the simplest Flash[1] content the processor is very active. Additional data captured with PowerTOP reveals that the processor wakes up ~340 times/s compared to ~75 times/s when the Browser is in idle on default home page.

If Flash content is heavily used during browsing, frequent processor activity will cause a significant decrease in the amount of time available on one battery charge.

*As future editions of the Flash engine evolve make sure to utilize the latest Flash release as future releases does feature improved power efficiency.*

[1] Flash version 9 was used in this study. Recent versions of Flash 9 and 10 have improved power characteristics.

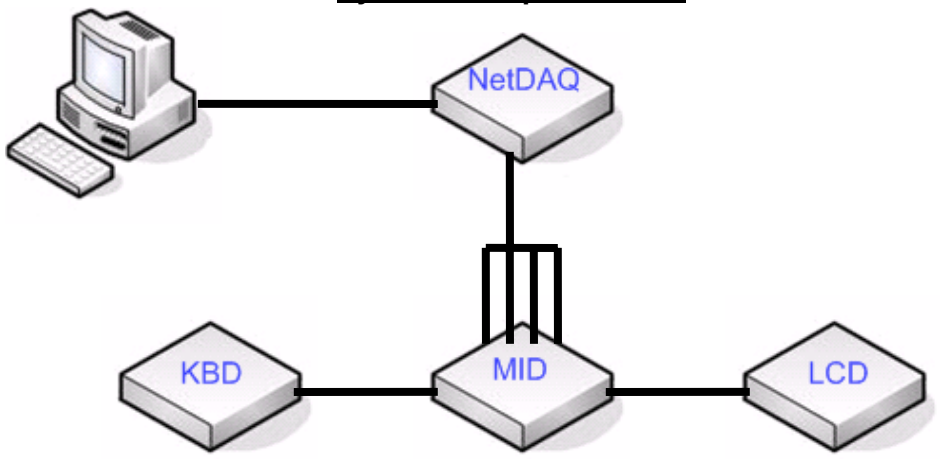
# 6. Appendix A – Atom™ processor specifications

Details on available Intel® Atom™ processor SKUs including data on clock speed, TDP, idle power, FSB, sleep state details and more can be found on the Intel® Atom™ Processor Technology resource site.

http://www.intel.com/products/atom/index.htm

# 7. Appendix B - HW setup and NetDAQ power measurement setup

**System setup overview**




**Platform specification and configuration**

| | |
|---|---|
| Intel® Atom™ processor | B1 |
| Intel® Poulsbo chipset (SCH) | C0 |
| FSB implementation | CMOS type |
| RAM | 1 Gb |
| LFM | 800 |
| HFM | 1600 |
| BIOS revision | 70 |
| PSB drivers | |
| Drivers | 0.9 |

| | |
|---|---|
| Video | 0.15 |
| Xpsb | 0.7 |
| Helix codecs | 0.262 beta 3 |
| Resolution | 1024x600 |
| Screen brightness | Default |
| processor frequency governor | On-demand |

The SDP processor used is considered equivalent to the **Intel® Atom™ Z530** SKU.

**NetDAQ configuration**

| | |
|---|---|
| Sample interval | 0.05s (20 times/s) |
| Measurement modules | 4 |
| Measurement points | 36 |
| Key power measurement objects | |
| Total processor | VCC, VCC_CORE, VTT_processor |
| Total SCH (main) | VTT_SCH, SCH_VCORE, SCH(2), SCH_SUS, SM_SCH |
| RAM | DIMM |
| Total (main) | Total processor + Total SCH (main) + RAM |
| Other SCH | PCIE, DVLDS, SDVO, DPLLA, DPLLB, PCIEPLL, HPLL, AUSBPLL |
| Other board | MINIPCIE, DDR2, PWH, CH, KBC, PATA, USB, IMVP, processor_PHASE |
| Total | Total (main) + Other SCG + Other board |

Instrumented SDP was connected to NetDAQ for measurements via the 4 modules connected to the sense resistors on board. The NetDAQ was in its turn connected via Ethernet loopback cable to host PC where the measurements were collected. The SDP was additionally connected to external LCD kit via USB and to keyboard via PS/2.

Fluke NetDAQ collects measured current and voltage from the board sense resistors and transfers the data to the NetDAQ SW tool on the host machine which calculates, adjusts for board specific offsets and accumulates power data according to the power measurement objects listed above.

# 8. Acronyms

| | |
|---|---|
| **CIF** | Common Intermediate Format. Video media format of resolution 352x288 |
| **C-state** | Processor Sleep state. C1 - C6 sleep states are available on Intel® Atom™. (Observe that C0 should be considered running state) |
| **FSB** | Front Side Bus. Interface between processor and SCH |
| **HDD** | Hard Disk Drive |
| **Helix** | Multimedia framework used as part of Moblin stack. https://helixcommunity.org/ https://rp4mid.helixcommunity.org/ |
| **HFM** | Highest Frequency Mode |
| **HTT** | Hyper Threading Technology. Also Simultaneous Multi-Threading (SMT) |
| **LFM** | Lowest Frequency Mode |
| **MID** | Mobile Internet Device. Category of mobile devices based on the Intel® Atom™ processor. |
| **Moblin** | Open Source community for sharing and creating Linux reference stack for MID. http://moblin.org/ |
| **NetDAQ** | Networked Data Acquisition Unit. HW from Fluke used to measure discrete component data such as Voltage and Current. http://us.fluke.com/usen/products/NetDAQ.htm?catalog_name=FlukeUnitedStates |
| **OpenGL** | Open Graphics Library. Used by the Clutter reference UI. |
| **PATA** | Parallel ATA (Advanced Technology Attachment). Interface for connecting storage devices such as HD or CD/DVD |
| **PowerTOP** | Measures system/application use of various hardware power-saving features. http://www.lesswatts.org/projects/powertop/ |
| **PSB** | Intel® Atom™ MID chipset (also SCH) |
| **P-state** | Processor Execution state. LFM -> HFM |
| **SCH** | System Controller Hub, aka. Poulsbo. |
| **SDP** | Software Development Platform. For the targeted platform also named Crown Beach. |
| **SIMD** | Single Instruction Multiple Data allowing data level parallelism |
| **SSE** | Streaming SIMD Extensions. Extended instruction set |
| **SMT** | Simultaneous Multi-Threading. Also. Hyper Threading Technology (HTT) |
| **SSD** | Solid State Disk |
| **TDP** | Thermal Design Power. Represents the maximum amount of power the thermal solution is required to dissipate |
| **Workload** | Isolated execution object with well defined behavior |