# Dynamic scaling of CPU core frequency and p-states:

The processor we're working on is the i7-4770k, from the series of processors called Haswell

**Objective:**

To set each CPU core to a different frequency (as per the requirements of the user).

**Approach:**

The various power governor modes allow for dynamic frequency setting, and the *userspace* mode was first used. Using the *userspace* power governor settings, it was possible to make each core request for a frequency independent of the other cores. A benchmark application with about a million multiplications and additions was run under a variety of scenarios.

**Observations:**

Case 1: All cores were set to a frequency of 0.8 GHz. The RAPL energy counter values were noted. It was noted that the value was not unique or the same during every execution, due to a host of other noises and factors. Hence the range of these energy counter values was noted. Let this range be indicated by X.

Case 2: All cores were set to a frequency of 3.4 GHz. The range of the RAPL energy counter values was noted. Let this range be indicated by Y, and the starting value in the range Y was greater than the largest value in the range X.

Case 3: Core 0 was set to 0.8 GHz and all the other cores were set to 3.4 GHz. The execution was confined to core 0 by setting the affinity of the thread of execution to core 0 within the scheduler. It was expected that the energy counter values would fall in the range of X, as the core which was running the application was understood to be running at 0.8 GHz. Instead, the energy counter values belonged to the range of Y, which indicated that core 0 was running at 3.4 GHz and not 0.8 GHz.

Many other scenarios were tried and it was noted that, irrespective of the frequency requested by a CPU core, all non-idle cores were running at the maximum of all frequencies requested by the different CPU cores. For example, we requested for 0.8 GHz on core 0, 1.1 GHz on core 1, and so on. The maximum of all the requested frequencies was 2.8 GHz, which was requested only by core 5, and all the other cores asked for a lower frequency. However, it seems like all the cores were running at 2.8 GHz, as it was the request with the maximum frequency.

**Conclusion based on the above:** It might not be possible to make different CPU cores run at different (user-specified) frequencies. They (provided they are non-idle) would all run at the same frequency, which would be the maximum frequency requested by any of the CPU cores at an instant of time.

**In support of the above conclusion:**

1. An article explaining the concepts of p-states and dynamic voltage-frequency scaling in Haswell processors by an Intel researcher conveyed that all cores were running on the same voltage plane, and when one core had to be run at a high frequency, running the others at a lower frequency would be inefficient, and hence, all cores would either be idle or run at this maximum requested frequency.
2. Another power governor mode called the *performance* mode was tried. This allows the user to set a maximum limiting frequency for the cores, and when the *performance* mode is enabled, the cores run at this maximum frequency. However, it was not possible to set a different limiting frequency for each core, separately. But, this option does exist and may be possible on other processors.

Also, it was noted that controlling voltage directly might not be possible, and there are no concrete methods to read the voltage values through any MSRs. One way to map each frequency to a voltage would be a compute the voltage by measuring all the other required parameters, such as dynamic power, frequency, etc.

It makes sense to assume that every discrete available core frequency corresponds to a P-state. For this Haswell processor we are working on, the discrete available frequencies are : 3.4 GHz, 3.2 GHz, 3.1 GHz, 2.9 GHz, 2.7 GHz, 2.5 GHz, 2.4 GHz, 2.2 GHz, 2.0 GHz, 1.8 GHz, 1.7 GHz, 1.5 GHz, 1.3 GHz, 1.1 GHz, 1.0 GHz and 0.8 GHz. Thus, there might be **16 available P-states**, each corresponding to one of the above discrete frequencies and a corresponding voltage too.