Hi3796M V100 / Hi3798M V100 Low-Power Solution

User Guide

Issue	01
Date	2014-11-05

Trademarks and Permissions

Notice

The purchased products, services and features are stipulated by the contract made between HiSilicon and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

HiSilicon Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: http://www.hisilicon.com

Email: support@hisilicon.com

About This Document

Purpose

This document describes how to configure and debug software and hardware parameters of the dynamic voltage scaling solution.

Related Versions

The following table lists the product versions related to this document.

Product Name	Version
Hi3798M	V1XX
Hi3796M	V1XX

Intended Audience

This document is intended for:

- Technical support personnel
- Hardware development engineers
- Software development engineers

Change History

Changes between document issues are cumulative. Therefore, the latest document issue contains all changes made in previous issues.

Issue 01 (2014-11-05)

This issue is the first official release, which incorporates the following change:

Hi3796M V100 is supported.

The document name is changed.

Issue 00B01 (2014-10-11)

This issue is the first draft release.

Contents

Figures

Tables

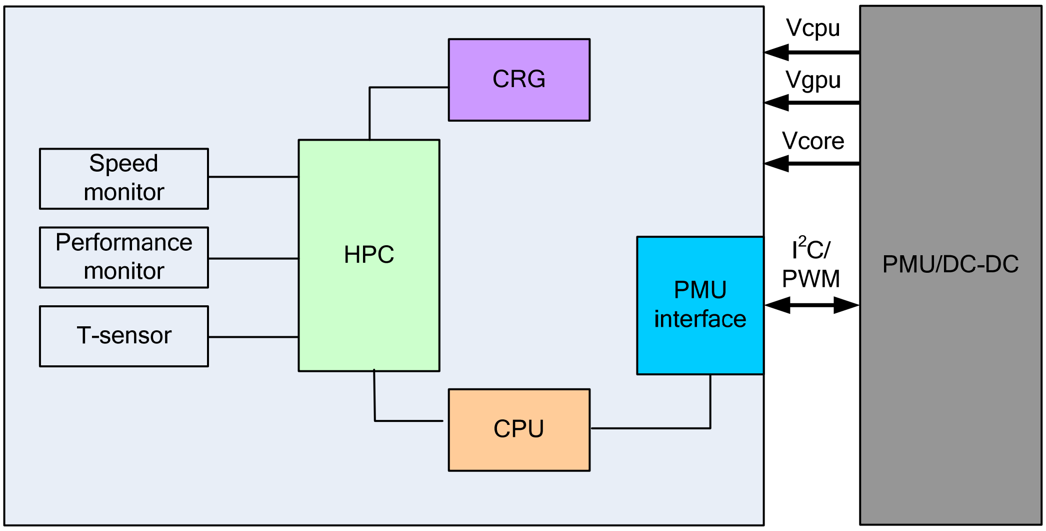
1 Overview

The dynamic voltage and frequency scaling (DVFS) technology indicates that the voltage and frequency are dynamically configured based on the application scenario to meet the requirements on the current circuit timings and performance. It minimizes power consumption by quickly switching the voltage and frequency based on the CPU and GPU usages for running services.

The adaptive voltage scaling (AVS) technology further reduces the chip power consumption on the basic of the DVFS by dynamically adjusting the voltage based on the chip manufacturing process, temperature, and circuit timings.

DVFS/AVS reduces the power consumption of a running chip while the chip performance remains unchanged. As shown in Figure 1-1, the hardware power controller (HPC) controls the external power management unit (PMU)/DC-DC over the PMU interface to dynamically adjust the CPU, GPU, and core voltages based on preset AVS and DVFS algorithms and the feedback data from the speed monitor, performance monitor, and T-sensor, reducing the average power consumption of the chip.

Figure 1-1 DVFS and AVS



For the DVFS, performance profiles must be created based on the system scenarios. For example, six frequency profiles (400 MHz, 600 MHz, 800 MHz, 1.0 GHz, 1.2 GHz, and 1.5 GHz) are created for the CPU, and each profile corresponds to a voltage value.

The DVFS and AVS technologies require that the CPU, GPU, and core voltages be dynamically adjustable. In the Hi379XM V100 solutions, dynamic adjustment of the CPU, GPU, and core voltages (the core and GPU power of Hi379XM V100 are combined) can be implemented by using a technology similar to the selective voltage binning (SVB) technology (combining DC-DC and PWM, applies to the CPU, GPU, and core).

2 Power Design

As the integration and frequency of the chip become increasingly higher, the number of transistors in the chip increases. The external power pin of the chip provides a common power supply node for the internal transistors. When the status of transistors changes, noises are caused and transmitted inside the chip.

To reduce noises, you are advised to use the power chip with excellent dynamic response performance for CPU and GPU power supplies.

- Recommendation 1: DC-DC chip with excellent dynamic response performance

  The working frequency of the DC-DC chip greatly affects its dynamic response performance. A higher frequency typically indicates a faster response speed. Therefore, the DC-DC chip with a high working frequency (higher than 640 kHz) is recommended.

- Recommendation 2: DC-DC chip with high Vref precision

  To improve the output voltage precision of the PWM dynamic voltage scaling, you are advised to select the DC-DC chip with high Vref precision (smaller than 2%).

3 Control Modes of Dynamic Voltage Scaling

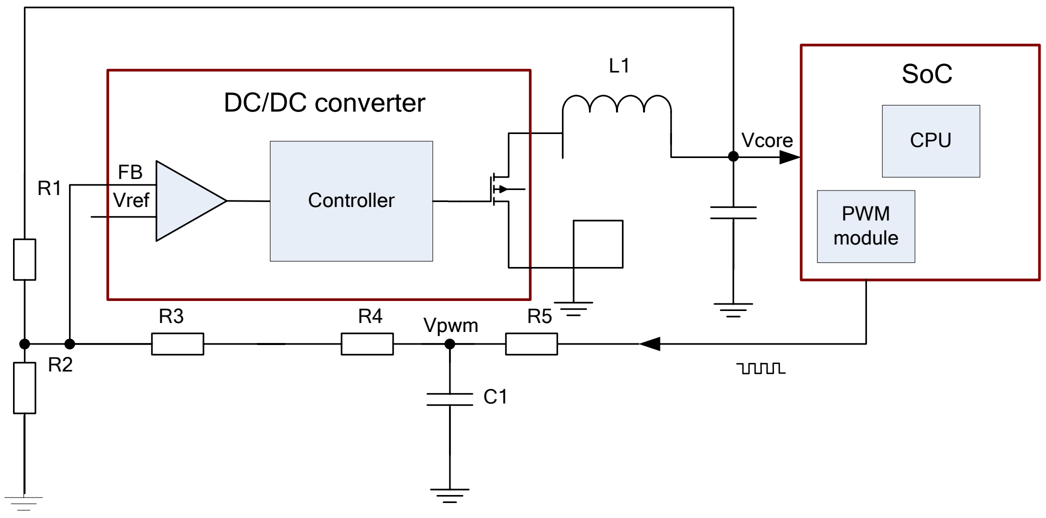
3.1 Dynamic Voltage Scaling by Using the PWM Technology

3.1.1 Function Implementation and Peripheral Resistor Configuration

The DC-DC circuit itself is a voltage feedback operational amplifier circuit. It implements output voltage scaling by adjusting the feedback voltage. As shown in Figure 3-1, the PWM signal acts as the compensation signal to the feedback signal. The PWM signals with different duty cycles output DC levels with voltages ranging from 0 V to 3.3 V after the signals pass through the low-pass filter (this process is equivalent to D/A conversion). The CPU dynamically compensates the DC-DC feedback by dynamically adjusting the PWM duty cycle. In this way, the DC-DC output voltage is dynamically adjusted.

Hi379XM V100 provides two PWM channels (PWM 0/PWM 1) for dynamically adjusting the CPU and core (GPU) input voltages respectively.

Figure 3-1 Implementing dynamic voltage scaling by using the PWM



Currently, the DC-DC reference voltage of each vendor is different from each other, and the DC-DC output voltage scaling amplitude also varies even when the PWM duty cycle is the same. To ensure that different DC-DC PWM adjustment solutions adopt the same set of software configurations, HiSilicon provides the power solution shown in Figure 3-1 in the reference design. The PWM compensation circuit is split into a simple circuit that contains R1 to R5 and C1. R5 and C1 values remain unchanged, and R1 to R4 are selected based on the Vref of the DC-DC.

As described in Table 3-1 and Table 3-2, the CPU voltage range is 0.92 V to 1.5 V, the GPU and core voltage range is 0.9 V to 1.32 V for Hi379XM V100. HiSilicon has verified the mainstream DC-DC and provided STB vendors with three sets of impedance configurations of R1 to R4 for adjusting the corresponding power supply based on the Vref value. If the Vref of the selected DC-DC is not in Table 3-1, you can submit an application to HiSilicon for verifying the DC-DC. Then HiSilicon provides a set of proper configurations and add it to the table.

For details about the configurations, see the *DC-DC Peripheral Impedance Selection for the CPU, GPU, and Core Power of Hi379XM V100* in the HiSilicon hardware package.

Table 3-1 Recommended peripheral impedance for PWM dynamic voltage scaling (V_{CPU}, 0.92 V to 1.5 V)

Recommended: Vmax = 1.5 V, Vmin = 0.92 V								
Vref (V)	**R1 (kΩ)**	**R2 (kΩ)**	**R3 (kΩ)**	**R4 (kΩ)**	**R5 (kΩ)**	**C1 (uF)**	**Vmax (V)**	**Umin (V)**
0.6	15	11.3	33	51	1	2.2	1.50	0.92
0.765	20	24.9	10	100	1	2.2	1.52	0.92
0.803	34.8	49.9	47	150	1	2.2	1.50	0.92
0.807	34.8	51	47	150	1	2.2	1.50	0.92

Table 3-2 Recommended peripheral impedance for PWM dynamic voltage scaling (V_{core}/V_{GPU}, 0.9 V to 1.32 V)

Recommended: Vmax = 1.32 V, Vmin = 0.9 V								
Vref (V)	**R1 (kΩ)**	**R2 (kΩ)**	**R3 (kΩ)**	**R4 (kΩ)**	**R5 (kΩ)**	**C (uF)**	**Vmax (V)**	**Umin (V)**
0.6	14.7	13.7	14.7	100	1	2.2	1.32	0.90
0.765	27	45.3	12	200	1	2.2	1.32	0.90
0.8	26.1	49.9	4.99	200	1	2.2	1.32	0.90
0.807	20	39.2	5.49	150	1	2.2	1.32	0.90
0.92	12	39	11	82	1	2.2	1.32	0.90
0.923	18.7	61.9	33	113	1	2.2	1.32	0.90

Recommended: Vmax = 1.32 V, Vmin = 0.9 V								
0.925	20	66.5	34.8	121	1	2.2	1.32	0.90

CAUTION

- The DC-DC with less than or equal to 2% Vref precision is recommended.
- The working frequency of the DC-DC must be greater than or equal to 640 kHz.

3.1.2 Voltage Configuration

According to the design in Figure 3-1 and configurations in Table 3-1 and Table 3-2, the real-time CPU and core (GPU) voltages of Hi379XM V100 each corresponds to a register value. As shown in Table 3-3, the corresponding configuration registers corresponding to the CPU, and GPU/core voltages of Hi379XM V100 are 0xF8A2301C and 0xF8A23018 respectively. The measured voltage value may have a ±20 mV error based on the actual load.

Take the Hi3798M V100 demo board as an example. The default CPU and core/GPU voltages during power-on are configured in the fastboot configuration table in the HiSilicon SDK:

- Vcpu = 1.38 V (The register whose address is 0xF8A2301C is set to **0x002F00E7**.)
- Vcore = 1.15 V (The register whose address is 0xF8A23018 is set to **0x004300A7**.)

You can query and modify the current voltage configuration by running commands in the fastboot command line. For details about how to query and modify the current configuration in kernel mode, see chapter 4 "SDK Debugging."

Table 3-3 Mapping between voltage and register values for Hi379XM V100 PWM dynamic voltage scaling

Mapping Between V_{CPU} and Register Values **Address for the Register Corresponding to the CPU: 0xf8a2301C** **(0.92 V to 1.5 V)**				**Mapping Between V_{core} and Register Values** **Address for the Register Corresponding to the core: 0xf8a23018** **(0.9 V to 1.32 V)**			
M	**Register Value**	**Voltage**	**Step**	**M**	**Register Value**	**Voltage**	**Step**
1	0x000100E7	1.495	0.005	1	0x000100A7	1.315	0.005
2	0x000300E7	1.49	0.005	3	0x000300A7	1.31	0.005
3	0x000500E7	1.485	0.005	5	0x000500A7	1.305	0.005
4	0x000700E7	1.48	0.005	7	0x000700A7	1.3	0.005
5	0x000900E7	1.475	0.005	9	0x000900A7	1.295	0.005
6	0x000B00E7	1.47	0.005	B	0x000B00A7	1.29	0.005
7	0x000D00E7	1.465	0.005	D	0x000D00A7	1.285	0.005

Mapping Between V_{CPU} and Register Values **Address for the Register Corresponding to the CPU: 0xf8a2301C** **(0.92 V to 1.5 V)**				**Mapping Between V_{core} and Register Values** **Address for the Register Corresponding to the core: 0xf8a23018** **(0.9 V to 1.32 V)**			
M	**Register Value**	**Voltage**	**Step**	**M**	**Register Value**	**Voltage**	**Step**
8	0x000F00E7	1.46	0.005	F	0x000F00A7	1.28	0.005
9	0x001100E7	1.455	0.005	11	0x001100A7	1.275	0.005
10	0x001300E7	1.45	0.005	13	0x001300A7	1.27	0.005
11	0x001500E7	1.445	0.005	15	0x001500A7	1.265	0.005
12	0x001700E7	1.44	0.005	17	0x001700A7	1.26	0.005
13	0x001900E7	1.435	0.005	19	0x001900A7	1.255	0.005
14	0x001B00E7	1.43	0.005	1B	0x001B00A7	1.25	0.005
15	0x001D00E7	1.425	0.005	1D	0x001D00A7	1.245	0.005
16	0x001F00E7	1.42	0.005	1F	0x001F00A7	1.24	0.005
17	0x002100E7	1.415	0.005	21	0x002100A7	1.235	0.005
18	0x002300E7	1.41	0.005	23	0x002300A7	1.23	0.005
19	0x002500E7	1.405	0.005	25	0x002500A7	1.225	0.005
20	0x002700E7	1.4	0.005	27	0x002700A7	1.22	0.005
21	0x002900E7	1.395	0.005	29	0x002900A7	1.215	0.005
22	0x002B00E7	1.39	0.005	2B	0x002B00A7	1.21	0.005
23	0x002D00E7	1.385	0.005	2D	0x002D00A7	1.205	0.005
24	0x002F00E7	1.38	0.005	2F	0x002F00A7	1.2	0.005
25	0x003100E7	1.375	0.005	31	0x003100A7	1.195	0.005
26	0x003300E7	1.37	0.005	33	0x003300A7	1.19	0.005
27	0x003500E7	1.365	0.005	35	0x003500A7	1.185	0.005
28	0x003700E7	1.36	0.005	37	0x003700A7	1.18	0.005
29	0x003900E7	1.355	0.005	39	0x003900A7	1.175	0.005
30	0x003B00E7	1.35	0.005	3B	0x003B00A7	1.17	0.005
31	0x003D00E7	1.345	0.005	3D	0x003D00A7	1.165	0.005
32	0x003F00E7	1.34	0.005	3F	0x003F00A7	1.16	0.005
33	0x004100E7	1.335	0.005	41	0x004100A7	1.155	0.005
34	0x004300E7	1.33	0.005	43	0x004300A7	1.15	0.005

Mapping Between V_{CPU} and Register Values **Address for the Register Corresponding to the CPU: 0xf8a2301C** **(0.92 V to 1.5 V)**				**Mapping Between V_{core} and Register Values** **Address for the Register Corresponding to the core: 0xf8a23018** **(0.9 V to 1.32 V)**			
M	**Register Value**	**Voltage**	**Step**	**M**	**Register Value**	**Voltage**	**Step**
35	0x004500E7	1.325	0.005	45	0x004500A7	1.145	0.005
36	0x004700E7	1.32	0.005	47	0x004700A7	1.14	0.005
37	0x004900E7	1.315	0.005	49	0x004900A7	1.135	0.005
38	0x004B00E7	1.31	0.005	4B	0x004B00A7	1.13	0.005
39	0x004D00E7	1.305	0.005	4D	0x004D00A7	1.125	0.005
40	0x004F00E7	1.3	0.005	4F	0x004F00A7	1.12	0.005
41	0x005100E7	1.295	0.005	51	0x005100A7	1.115	0.005
42	0x005300E7	1.29	0.005	53	0x005300A7	1.11	0.005
43	0x005500E7	1.285	0.005	55	0x005500A7	1.105	0.005
44	0x005700E7	1.28	0.005	57	0x005700A7	1.1	0.005
45	0x005900E7	1.275	0.005	59	0x005900A7	1.095	0.005
46	0x005B00E7	1.27	0.005	5B	0x005B00A7	1.09	0.005
47	0x005D00E7	1.265	0.005	5D	0x005D00A7	1.085	0.005
48	0x005F00E7	1.26	0.005	5F	0x005F00A7	1.08	0.005
49	0x006100E7	1.255	0.005	61	0x006100A7	1.075	0.005
50	0x006300E7	1.25	0.005	63	0x006300A7	1.07	0.005
51	0x006500E7	1.245	0.005	65	0x006500A7	1.065	0.005
52	0x006700E7	1.24	0.005	67	0x006700A7	1.06	0.005
53	0x006900E7	1.235	0.005	69	0x006900A7	1.055	0.005
54	0x006B00E7	1.23	0.005	6B	0x006B00A7	1.05	0.005
55	0x006D00E7	1.225	0.005	6D	0x006D00A7	1.045	0.005
56	0x006F00E7	1.22	0.005	6F	0x006F00A7	1.04	0.005
57	0x007100E7	1.215	0.005	71	0x007100A7	1.035	0.005
58	0x007300E7	1.21	0.005	73	0x007300A7	1.03	0.005
59	0x007500E7	1.205	0.005	75	0x007500A7	1.025	0.005
60	0x007700E7	1.2	0.005	77	0x007700A7	1.02	0.005
61	0x007900E7	1.195	0.005	79	0x007900A7	1.015	0.005

Mapping Between V_{CPU} and Register Values **Address for the Register Corresponding to the CPU: 0xf8a2301C** **(0.92 V to 1.5 V)**				**Mapping Between V_{core} and Register Values** **Address for the Register Corresponding to the core: 0xf8a23018** **(0.9 V to 1.32 V)**			
M	**Register Value**	**Voltage**	**Step**	**M**	**Register Value**	**Voltage**	**Step**
62	0x007B00E7	1.19	0.005	7B	0x007B00A7	1.01	0.005
63	0x007D00E7	1.185	0.005	7D	0x007D00A7	1.005	0.005
64	0x007F00E7	1.18	0.005	7F	0x007F00A7	1	0.005
65	0x008100E7	1.175	0.005	81	0x008100A7	0.995	0.005
66	0x008300E7	1.17	0.005	83	0x008300A7	0.99	0.005
67	0x008500E7	1.165	0.005	85	0x008500A7	0.985	0.005
68	0x008700E7	1.16	0.005	87	0x008700A7	0.98	0.005
69	0x008900E7	1.155	0.005	89	0x008900A7	0.975	0.005
70	0x008B00E7	1.15	0.005	8B	0x008B00A7	0.97	0.005
71	0x008D00E7	1.145	0.005	8D	0x008D00A7	0.965	0.005
72	0x008F00E7	1.14	0.005	8F	0x008F00A7	0.96	0.005
73	0x009100E7	1.135	0.005	91	0x009100A7	0.955	0.005
74	0x009300E7	1.13	0.005	93	0x009300A7	0.95	0.005
75	0x009500E7	1.125	0.005	95	0x009500A7	0.945	0.005
76	0x009700E7	1.12	0.005	97	0x009700A7	0.94	0.005
77	0x009900E7	1.115	0.005	99	0x009900A7	0.935	0.005
78	0x009B00E7	1.11	0.005	9B	0x009B00A7	0.93	0.005
79	0x009D00E7	1.105	0.005	9D	0x009D00A7	0.925	0.005
80	0x009F00E7	1.1	0.005	9F	0x009F00A7	0.92	0.005
81	0x00A100E7	1.095	0.005	A1	0x00A100A7	0.915	0.005
82	0x00A300E7	1.09	0.005	A3	0x00A300A7	0.91	0.005
83	0x00A500E7	1.085	0.005	A5	0x00A500A7	0.905	0.005
84	0x00A700E7	1.08	0.005	A7	0x00A700A7	0.9	0.005
85	0x00A900E7	1.075	0.005	-	-	-	-
86	0x00AB00E7	1.07	0.005	-	-	-	-
87	0x00AD00E7	1.065	0.005	-	-	-	-
88	0x00AF00E7	1.06	0.005	-	-	-	-

Mapping Between V_{CPU} and Register Values **Address for the Register Corresponding to the CPU: 0xf8a2301C** **(0.92 V to 1.5 V)**				**Mapping Between V_{core} and Register Values** **Address for the Register Corresponding to the core: 0xf8a23018** **(0.9 V to 1.32 V)**			
M	**Register Value**	**Voltage**	**Step**	**M**	**Register Value**	**Voltage**	**Step**
89	0x00B100E7	1.055	0.005	-	-	-	-
90	0x00B300E7	1.05	0.005	-	-	-	-
91	0x00B500E7	1.045	0.005	-	-	-	-
92	0x00B700E7	1.04	0.005	-	-	-	-
93	0x00B900E7	1.035	0.005	-	-	-	-
94	0x00BB00E7	1.03	0.005	-	-	-	-
95	0x00BD00E7	1.025	0.005	-	-	-	-
96	0x00BF00E7	1.02	0.005	-	-	-	-
97	0x00C100E7	1.015	0.005	-	-	-	-
98	0x00C300E7	1.01	0.005	-	-	-	-
99	0x00C500E7	1.005	0.005	-	-	-	-
100	0x00C700E7	1	0.005	-	-	-	-
101	0x00C900E7	0.995	0.005	-	-	-	-
102	0x00CB00E7	0.99	0.005	-	-	-	-
103	0x00CD00E7	0.985	0.005	-	-	-	-
104	0x00CF00E7	0.98	0.005	-	-	-	-
105	0x00D100E7	0.975	0.005	-	-	-	-
106	0x00D300E7	0.97	0.005	-	-	-	-
107	0x00D500E7	0.965	0.005	-	-	-	-
108	0x00D700E7	0.96	0.005	-	-	-	-
109	0x00D900E7	0.955	0.005	-	-	-	-
110	0x00DB00E7	0.95	0.005	-	-	-	-
111	0x00DD00E7	0.945	0.005	-	-	-	-
112	0x00DF00E7	0.94	0.005	-	-	-	-
113	0x00E100E7	0.935	0.005	-	-	-	-
114	0x00E300E7	0.93	0.005	-	-	-	-
115	0x00E500E7	0.925	0.005	-	-	-	-

Mapping Between V_{CPU} and Register Values **Address for the Register Corresponding to the CPU: 0xf8a2301C** **(0.92 V to 1.5 V)**				**Mapping Between V_{core} and Register Values** **Address for the Register Corresponding to the core: 0xf8a23018** **(0.9 V to 1.32 V)**			
M	**Register Value**	**Voltage**	**Step**	**M**	**Register Value**	**Voltage**	**Step**
116	0x00E700E7	0.92	0.005	-	-	-	-

4 SDK Debugging

4.1 Checking Low-Power Information

4.1.1 Checking the CPU Frequency and Voltage

Run the following command over the serial port:

```
cat /proc/msp/pm_cpu
```

If the following information is displayed, the current CPU frequency is 400 MHz and the CPU voltage is 1000 mV.

```
CPU: freq = 400000 (kHz) volt = 1000 (mv)
```

4.1.2 Checking the Core Voltage

Run the following command over the serial port:

```
cat /proc/msp/pm_core
```

If the following information is displayed, the core voltage is 1150 mV.

```
core: volt = 1150 (mv)
```

4.1.3 Checking the GPU Frequency

Run the following command over the serial port:

```
cat /proc/msp/pm_gpu
```

If the following information is displayed, the current GPU frequency is 200 MHz.

```
---------Hisilicon GPU Info---------
Frequency                    : 200000 (kHz)
...
```

4.1.4 Checking the Low-Power Policies Supported by the CPU

Run the following command over the serial port:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_governors
```

The following information may be displayed over the serial port:

```
conservative ondemand userspace powersave interactive performance
```

The preceding policies are described as follows:

- conservative: The frequency and voltage are adjusted by step.
- ondemand: The frequency and voltage are dynamically adjusted based on the CPU load. It is slower than the interactive policy.
- userspace: The voltage and frequency are set by the user. The SDK does not automatically adjust them.
- powersave: The frequency is always set to the lowest to ensure low power consumption.
- interactive: The frequency and voltage are dynamically adjusted based on the CPU load. This policy is used by default.
- performance: The frequency is always set to the highest to ensure high performance.

4.1.5 Checking the Low-Power Policy Used by the CPU

Run the following command over the serial port:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

If the following information is displayed, the interactive policy is used.

```
interactive
```

4.1.6 Checking the CPU Multi-Core Information

Run the following command over the serial port:

```
cat /proc/cpuinfo
```

If the following information is displayed (processor 0 and processor 1 exist), there are two cores. If the information contains only processor 0, there is only a single core. If the information contains processors 0–3, there are four cores.

```
Processor      : ARMv7 Processor rev 1 (v7l)
processor       : 0
BogoMIPS     : 2371.58
processor      : 1
BogoMIPS     : 2396.16
```

4.2 Setting Low-Power Parameters

4.2.1 Setting the CPU Low-Power Policy

Run the following command over the serial port (*xxx* is the policy name):

```
echo xxx > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

- To disable dynamic CPU frequency and voltage scaling, set the low-power policy to userspace.

```
echo userspace >
/sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

- To enable dynamic CPU frequency and voltage scaling, set the low-power policy to interactive.

```
echo interactive >
/sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

4.2.2 Setting the CPU Frequency

The current SDK has six default CPU profiles (400 MHz, 600 MHz, 800 MHz, 1000 MHz, 1200 MHz, and 1500 MHz).

To manually set the CPU frequency, disable dynamic CPU frequency and voltage scaling by running the following command:

```
echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
```

Then set the CPU frequency by running the following command:

```
echo 400000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_setspeed
```

Where, 400000 is in kHz. That is, the CPU frequency is set to 400 MHz. Note that the CPU voltage is also synchronized to the preset stable voltage that supports this frequency.

4.2.3 Setting the CPU Frequency and Voltage

To adjust the CPU frequency and voltage separately, run the following commands:

```
echo volt=xxxx > /proc/msp/pm_cpu
echo freq=xxxx > /proc/msp/pm_cpu
echo freq=xxxx volt=xxxx > /proc/msp/pm_cpu
```

freq is the frequency value (in kHz) and **volt** is the voltage value (in mV).

For example, if you want to set the CPU voltage to 1100 mV, run the following command:

```
echo volt=1100 > /proc/msp/pm_cpu
```

If you want to set the CPU frequency to 600 MHz, run the following command:

```
echo freq=600000 > /proc/msp/pm_cpu
```

If you want to set the CPU frequency to 600 MHz and voltage to 1100 mV at the same time, run the following command:

```
echo freq=600000 volt=1100 > /proc/msp/pm_cpu
```

⚠ **CAUTION**

The frequency value must be one of the six default frequency values because the CPU cannot work at any frequency. If the input value is invalid, the configuration fails. The voltage can be set to any value within a certain range, but it must be sufficient to support the CPU at the current frequency. The voltage range varies with the board and chip. If the configured voltage is too low, the board may be unstable or suspended immediately.

4.2.4 Setting the Core Voltage

The core frequency is fixed and cannot be dynamically adjusted. To adjust the core voltage, run the following command:

```
echo volt=xxxx > /proc/msp/pm_core
```

The unit of the parameter is mV.

For example, to adjust the core voltage to 1150 mV, run the following command over the serial port:

```
echo volt=1150 > /proc/msp/pm_core
```

⚠ **CAUTION**

The core voltage significantly affects system stability. Therefore, use the default value and do not adjust it typically.

4.2.5 Setting the GPU Frequency

The GPU frequency adjustment is enabled by default. To manually adjust the GPU frequency, you need to disable the automatic adjustment function by running the following command:

```
echo 0 > /sys/module/mali/parameters/mali_dvfs_enable
```

Then run the following commands over the serial port:

```
echo freq=xxxx > /proc/msp/pm_gpu
```

freq is the frequency value (in kHz).

For example, if you want to set the GPU frequency to 432 MHz, run the following command:

```
echo freq=432000 > /proc/msp/pm_gpu
```

CAUTION

The configured frequency must be a frequency supported by the GPU; otherwise, the GPU may run abnormally.

4.2.6 Enabling/Disabling the CPU AVS Function

The CPU AVS function is enabled by default. To manually disable the CPU AVS function, run the following command:

```
echo avs=off > /proc/msp/pm_cpu
```

To manually enable the CPU AVS function, run the following command:

```
echo avs=on > /proc/msp/pm_cpu
```

5 Configuration Instance

The following takes the PWM configurations in the HiSilicon hardware package for Hi3798M V100 as an example.

- Selecting PWM signals

  Hi3798M V100 reserves two PWM channels to implement dynamic CPU and GPU/core voltage scaling. The relationship between PWM pin multiplexing and core/GPU and CPU control is as follows:

  - PWM channel 0, multiplexing pin name PMC_PWM0, controlling the core/GPU power
  - PWM channel 1, multiplexing pin name PMC_PWM1, controlling the CPU power

- Setting the initial voltage

  The default voltages are configured by the fastboot configuration table when the system boots. To view the initial voltage, you can check the corresponding register value in the fastboot command line and obtain the current voltages based on Table 3-3. The voltages configured in the HiSilicon SDK are as follows:

  - Vcpu = 1.38 V (The register whose address is 0xF8A2301C is set to **0x000700E7**.)
  - Vcore = 1.15 V (The register whose address is 0xF8A23018 is set to **0x004300A7**.)

  In fastboot mode, the DVFS function is disabled. You can run commands in the fastboot command line to query the values of voltage configuration registers, and then find the corresponding voltage value in Table 3-3.

  - CPU: md 0xf8a2301C
  - Core: md 0xF8A23018
  - GPU: md 0xF8A23018

- Querying the current voltage and frequency in kernel mode

  In kernel mode, the AVS and DVFS functions are enabled. The CPU voltage changes in real time based on the current load, and only the current voltages can be queried.

  You can run the following command:

  - `cat /proc/msp/pm_cpu`
  - `cat /proc/msp/pm_core`