Application note

DA1468x Power measurements

AN-B-037

Abstract

This document provides all the information needed to perform correctly power measurements and the typical values expected for the DA1468x family of chips.
For the measurements the following is used: the DA14681 ProDK the Agilent N6705 power analyzer and the SW tools provided by Dialog.

Contents

Abstract **1**

Contents **2**

Figures **2**

Tables **2**

1 Terms and definitions **4**

2 References **4**

3 Introduction **5**

4 Preparation of the SW, HW, instruments and equipment **5**

4.1 ProDK HW preparation 5

4.2 Power analyser use 8

4.3 BLE advertisement and connection power consumption 8

4.3.1 HW Preparation 8

4.3.2 SW preparation 8

4.3.3 Measurement procedure 9

4.3.4 Results 12

4.4 CPU power measurement 13

4.4.1 HW Preparation 13

4.4.2 SW preparation 13

4.4.3 Measurement procedure 15

4.4.4 Results 18

5 Useful tips **26**

Revision history **28**

Contacting Dialog Semiconductor **29**

Figures

Figure 1: DA1468x ProDK with the default (OOB) jumper placement 6
Figure 2: Shielded box and N6705B 7
Figure 3: Advertising, footprint on datalogger 10
Figure 4: Advertising average over longer period of time. 11
Figure 5: Sleep period between two advertising events 11
Figure 7: The unzipped project under SDK folder structure 14
Figure 8: running the uA_per_MIPS_bare project 15
Figure 8: M0 MIPS per operation Frequency 17
Figure 9: MIPS, Clock & Power with HCLK/1 19
Figure 10: 23

Tables

Table 1: BLE operation power consumption, VBat=3.8V 12
Table 2: CPU clock select 15
Table 3: HCLK prescaler selection 16
Table 4: PCLK prescaler selection 16
Table 5: M0 MIPS per operation Frequency 16
Table 6: MIPS, Clock & Power with HCLK/1 18

Table 7: MIPS, Clock & Power with HCLK/2 20
Table 8: MIPS, Clock & Power with HCLK/4 22
Table 9: MIPS, Clock & Power with HCLK/8 24
Table 10: Retained RAM power budget per cell 27

1 Terms and definitions

LP clock	Low Power clock
HCLK	AMBA High Speed Bus clock
PCLK	AMBA Peripheral Bus clock
RCX	RC 10.5MHz internal LP clock
RC32K	RC 32KHz internal LP clock
RC16M	RC 16MHz internal clock
XTAL32K	32.768KHz external crystal LP clock
XTAL16M	16MHz external crystal clock
PLL48	PLL 48MHz internal clock
PLL96	PLL 96MHz internal clock
MB	Mother Board
DB	Daughter Board
ProDK	Pro Development Kit
FSM	Finite State Machine

2 References

[1] DA1468x, Datasheet, Dialog Semiconductor

[2] Agilent Technologies, DC Power Analyser Model N6705 User's Guide, © Agilent Technologies, Inc. 2007 - 2012

[3] UM-B-060, User Manual DA1468x/DA1510x PRO-Development kit

[4] AN-B-061, DA1468x Application hardware design guidelines

[5] UM-B-047, User manual, DA1468x Getting Started with the Development Kit

[6] UM-B-056, User manual, DA1468x Software Developer's Guide

3 Introduction

This document describes the procedure for performing power consumption measurements on the DA1468x platforms using the DA1468x ProDK or just the DA1468x daughterboard. Please refer to the User Manual for the DA1468x PRO-Development kit and related documentation.

The DA1468x includes a complete power management IC, making it not just a SoC but SoC+PMU in one package. To compare the power consumption of the DA1468x, the reader must think of it like the complete PCB of a solution with all the power supply and charger, packed in a tiny package without any additional parts requirements other than the decoupling capacitors and one inductor.

The measurements described in this document are performed by default on the DA1468x ProDK daughterboard only, unless otherwise clearly stated.

For measurements accuracy, especially for the very low power cases like sleep states and LP clocks, a power analyser should be used. For the measurements in this case the Agilent N6705B power analyser is used. Please refer to the instrument documentation for details about it. It is suggested to use the sampling at the N6705B instead of SW as the sampling is faster this way and the measurements more accurate.

The document also covers the preparation of the HW and equipment needed for the measurements, explains the parameters to set in the SW for accurate measurements.

Dialog provides also an embedded tool in the ProDK Motherboard which provides power measurements capabilities. This tool is called Power Profiler and it is very accurate for measuring the power consumption while the device is active, but is not accurate enough for the extremely low currents when the DA14680 is in sleep modes.

CPU power consumption using all possible clocks and clock dividers is also covered. This allows the calculation of power consumption per MHz and the power consumption per MIPS. Normally this is not needed but for very special cases might provide useful information to the customer.

IMPORTANT NOTE: Make sure that the 16MHz XTAL in the DA1468x is properly trimmed. For measurements the mass production chips must be used rather than engineering samples. The DA1468x hardware guidelines application note explains how to recognize the version on the package. [4]

4 Preparation of the SW, HW, instruments and equipment

4.1 ProDK HW preparation

The standard DA1468x ProDK does not need any modifications to be used for getting the power measurements. What need to remember is that the mother board must be on power to avoid having any leakages and

a. If the shielded box is used then have the device for the connection measurements also placed in the box.

b. If the daughterboard (DB) is supplied with power at the coin cell contacts at the bottom of the DB then remove ALL the jumpers of the motherboard (MB). This is the suggested method.

c. If the DB is supplied from the motherboard remove all the jumpers of the MB EXCEPT of the J9 and J13. This is going to add some offset to the measurements.

Figure 1: DA1468x ProDK with the default (OOB) jumper placement

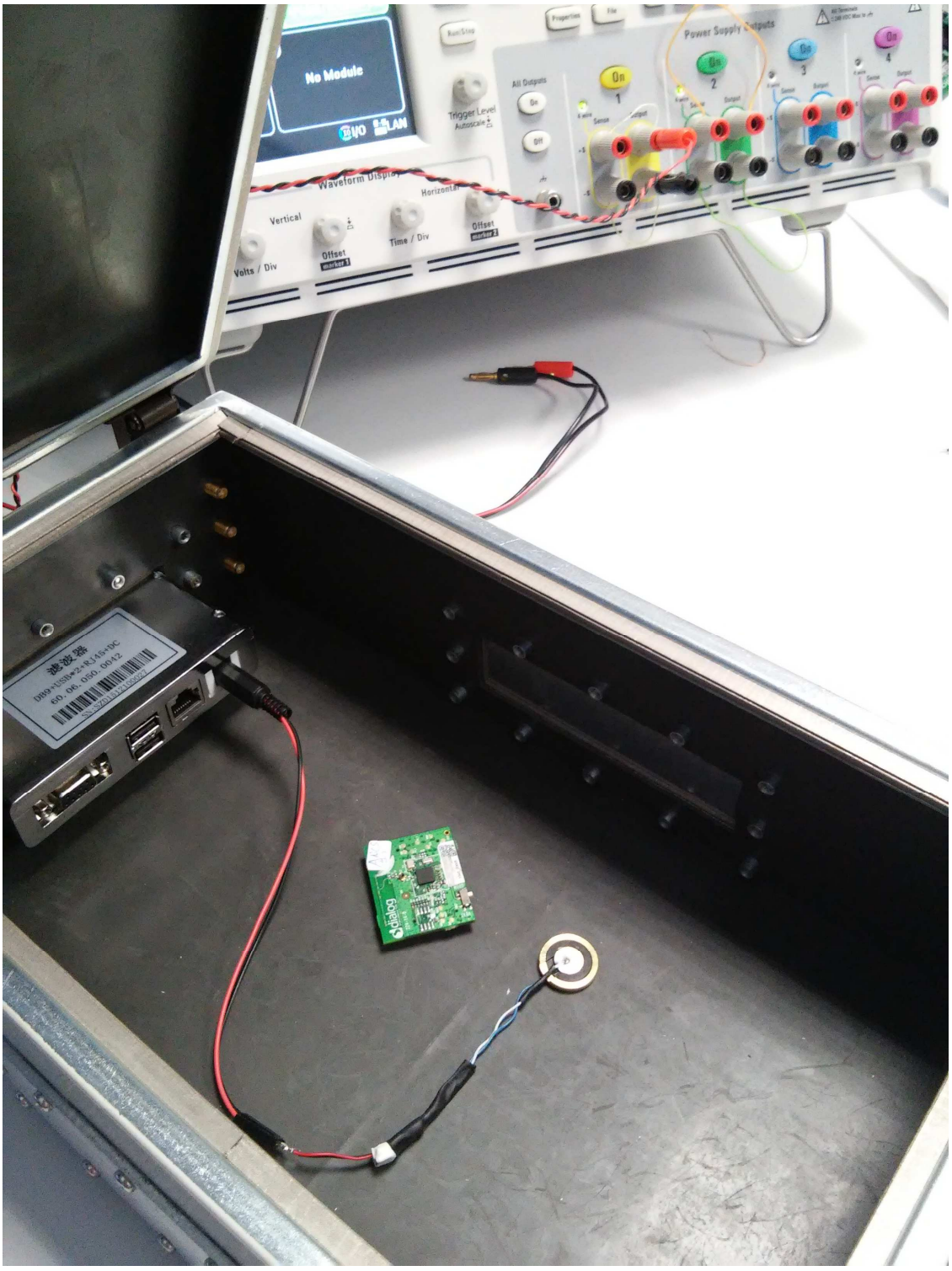


Figure 2: Shielded box and N6705B

4.2 Power analyser use

There are many configurations in which the power analyser can be used. In this document we use it as power supply source for measuring the current and log its changes.

For the correct connection and use of the power analyser please refer to the instruments documentation.

The power consumption can be measured with the DB either unplugged from the MB or with the DB plugged on the MB.

Prefer to have the DB unplugged from the MB for measurements not requiring the breakout connectors of the MB. Have the DB plugged on the MB with all the jumpers of the MB removed when the test requires connecting either loads for testing or GPIOs for control/testing to DA1468x.

IMPORTANT NOTE:

Avoid using low sampling rates in the instrument settings. Low sampling rates may lead to aliasing. Sampling rate of at least 10 ksamples/s is suggested to avoid aliasing and probably wrong measurements especially for the sleep modes.

4.3 BLE advertisement and connection power consumption

4.3.1 HW Preparation

Voltage: 3.8V

Power supply of the DB: through the coin cell contacts

SW project to use: *ble_adv_demo*, included in the generally available SDK.

To prepare the DB for the measurement use the Smartsnippets Studio to compile and write the code to the FLASH of the DB. Keep the DB switch to VBAT position and the J13, J9, J14 and J8 in place, so the DB will have power supply and the J-Link will be attached in order to FLASH the device through JTAG interface.

Once the DB is programmed, unplug it from the MB and connect power supply supply at the coin cell contacts, using soldered leads or a coin cell adapter. Flip the switch to COIN on the DB. If the measurement will be performed on the MB, keep only the J9 and J13 jumpers in place.

If the environment is too noisy with large number of Wi-Fi and BLE/BT devices it is suggested to perform the measurements using a shielded box to avoid increased power consumption due scan responses and re-transmissions.

4.3.2 SW preparation

For this measurement the *ble_adv* from the standard SDK is used. There are a couple of parameters that needs attention before building and using the application.

In the **custom_config_qspi.h** file check the definitions to match the following ones:

The default power configuration profile is OK.

```
#define dg_configPOWER_CONFIG                    (POWER_CONFIGURATION_2)
```

Choose the LP clock to use during sleep.

```
#define dg_configUSE_LP_CLK                      LP_CLK_32768
```

The PRODUCTION_MODE will remove all the ASSERTIONS. This is how the final product code will be built.

For the measurements the BLE parameters can be easily set in the “ble_config.h” file under “sdk\ble\config\” path of the SmartSnippets Studio project.

Alternatively, the developer can set the BLE parameters using the provided API calls. The function *ble_error_t* ***ble_gap_adv_intv_set***(*uint16_t adv_intv_min, uint16_t adv_intv_max);* should be used prior to start advertising to set the advertising interval.

The connection parameters are set by the master (central) device. In case the DA14680 is also used as master, the connection parameters are passed as arguments to the function *ble_error_t* ***ble_gap_connect***(***const*** *bd_address_t *peer_addr,* ***const*** *gap_conn_params_t *conn_params);*

```
#define dg_configIMAGE_SETUP                    PRODUCTION_MODE
```

Finally select the FLASH to use POWER DOWN mode.

```
#define dg_configPOWER_FLASH                    (1)
#define dg_configFLASH_POWER_DOWN               (1)
#define dg_configFLASH_POWER_OFF                (0)
```

Leave the rest of the definitions in the custom_config_qspi.h to their default values.

Having all the changes made, compile the application using the **DA146881-01-Release_QSPI (DA14680/1 Release build configuration for cached QSPI mode)** configuration and FLASH the device.

4.3.3 Measurement procedure

For the measurement, have the power analyser set at 3.8V and the current supply capability at 0.1A. Have the power analyses supply/measurement channel OFF.

For average current measurements, data logger mode should be used on N6705B. Select the current only trace to be captured or current and power if required and set an appropriate capture time. Suggested is to set 3-5 min capture time.

In order to measure individual parts of the event interval, such as current during sleep or during active state, use the SCOPE mode which provides higher sampling rate, thus increased accuracy.

By adjusting time per division and number of points, sampling period is altered. After capture is stopped, markers may be used to limit the area of interest and provide the required measurements within it. Measurement settings allow for various useful values to be selected (average, max/min, charge etc). It is suggested that measurements are repeated for a couple of events, so that rough averaging may be performed. When using the SCOPE function, especially when sleep current measurements are intended, it is important that ‘AUTO’ is selected in the ‘Ranges…’ setting window, found or the lower part of the screen to the right. This way, the ‘seamless range’ option of the Power Analyser is employed which effectively increases the dynamic range, allowing mA and uA measurements at the same time, without having to adjust the range.

To measure, detach the programmed DB from MB and connect the power analyser leads to the battery coin connectors at the bottom of the DB. Then power ON the supply/measurement channel of the power analyser.

On the power analyser use the Data Logger function to capture the current and power if required for 3-5 minutes to have realistic averaging.

Configure the golden unit using with the connection interval parameters for the specific test.

The Advertising interval can be set in the project to match the required scenario settings. Please refer to programming guide.

The actual footprints for advertising and average over longer period of time when advertising is shown on Figure 3, Figure 4 and Figure 5.

The device will get in/out of sleep as needed based on the parameters set for the BLE. Developer does not have to change anything.

In case there is need to measure various power save modes of the DA1468x, just change accordingly the parameter in the `pm_set_sleep_mode(`*`pm_mode_extended_sleep`*`)`; in the main() function of the project.

Figure 3: Advertising, footprint on datalogger

In Figure 3 the first large spike during the wakeup period is due to capacitors charging. While the system is in sleep the charged capacitors will slowly discharge. When the system wakes up, the capacitors will recharge. The spike gets bigger for longer sleep periods as the capacitors will discharge more with time.

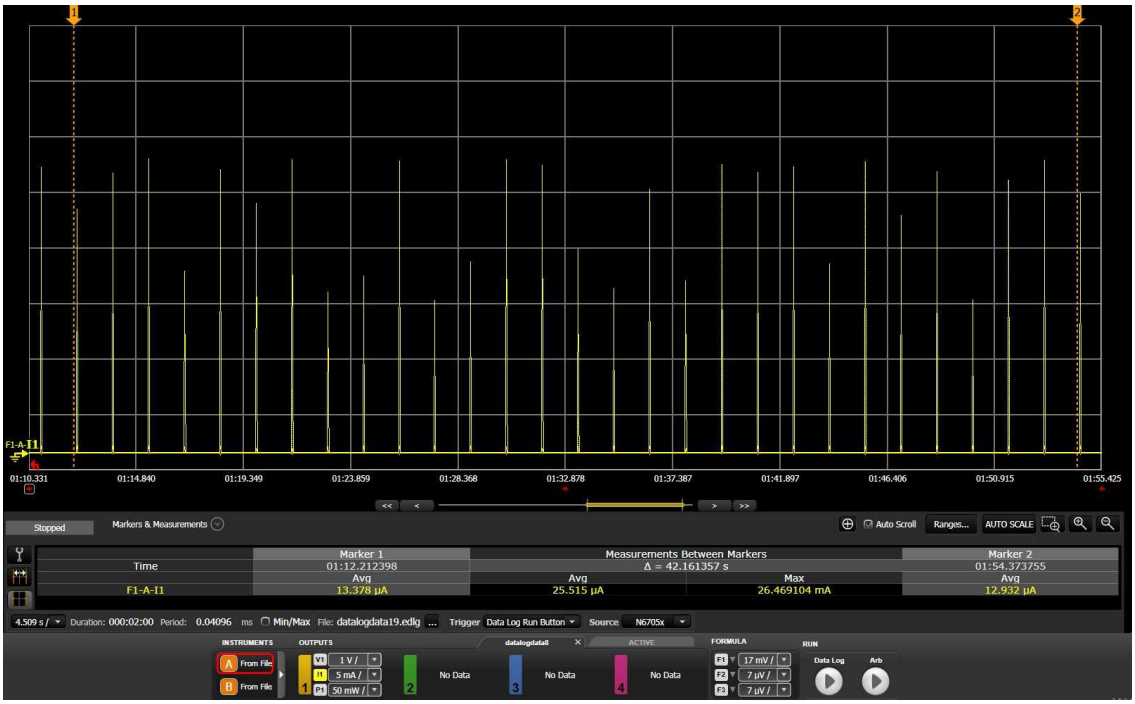


Figure 4: Advertising average over longer period of time.

Every vertical line is an advertising active state as shown on Figure 3

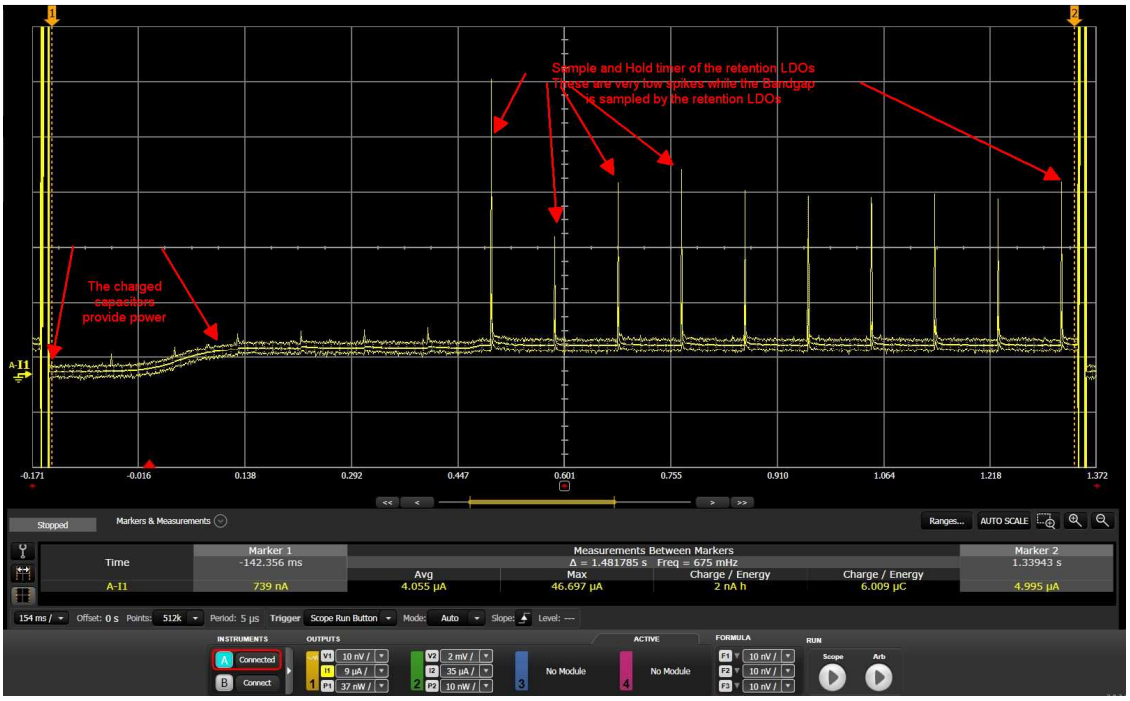


Figure 5: Sleep period between two advertising events

IMPORTANT NOTE: The device will not power save for the first 8 second since power up (or boot). Start getting the measurement after the 8 seconds to get the correct results. 8 Seconds is the maximum possible time in worst case to stabilize XTAL32K which is necessary for sleep. Usually much less time is needed, but the safe value is the 8 sec to allow sleep.

On Figure 5, initially the current is very low even negative in some cases. This is because the charged capacitors are discharging providing energy back to the system, thus for a short time the current on VBat drops.

The periodic low spikes are the Sample & Hold timer function for the retention LDOs. A part of the system is waking periodically by the sleep timer, samples the Bandgap and corrects the Sample & Hold Retention LDOs voltage.

4.3.4 Results

Measuring as described in paragraphs 4.3.1 to 4.3.3, the expected power consumption is as presented in Table 1.

Test Case	average **charge** per **active** state per interval (uC)	average **current** per **active** state per interval (mA)	average **energy** per **active** state per interval (uJ)	average **sleep current** per interval (uA)	average **sleep energy** per interval (uJ)	average **energy** per interval (uJ)	average **current** for capture (uA)	peak **current** (mA)	peak **power** (uW)
Adv 100 ms	16.5091	1.4278	62.7337	0.3221	0.1286	62.8623	157.5473	10.7792	0.041
Adv 1000 ms	17.3739	1.5032	66.0179	3.0823	11.7913	77.8118	20.3713	25.7034	0.0976
Adv 1500 ms	17.5355	1.517	66.6317	3.6971	21,2142	87,8479	15,3515	27,46	0,1043
Connect 50 ms	9,925	1,1464	37,7145	0,2866	0,0545	37,7695	198,7764	10,591	0,0402
Connect 100 ms	10,5273	1,2037	40,0033	0,3227	0,1227	40,1271	105,5916	11,0286	0,0419
Connect 400 ms	12,8064	1,4125	48,6619	1,7381	2,6456	51,3079	33,7523	23,8571	0,0906
Connect 1000 ms	16,1443	1,6602	61,3457	3,1402	11,9699	73,3164	19,2837	25,4778	0,0967
deep sleep							3.52		
Hibernation							0.85		

Table 1: BLE operation power consumption, VBat=3.8V

Note 1 Measured on DA14681-01 silicon with DB unplugged from MB.

Note 2 The charge per active state per interval (uC) differs because the decoupling capacitors discharging level differs. The longer the sleep period the more the capacitors will discharge. This results in higher current peaks on wake-up. This is also why the peak current differs for each case.

Note 3 **Average charge per active state per interval:** This is how much charge is used only for the active part of the interval. The active state is from wake to sleep where the system is awake and there is BLE RX/TX activity like advertising event.
The fact that the 100ms advertising case uses less uC is due to less discharging of the capacitors comparing to 1000ms and 1500ms cases. This is even more obvious in the connection cases.

The number is an average over many packets.

Note 4 **Average current per active state per interval:** Is the average current over many packets during the active state of the interval.

Note 5 **Average energy per active state per interval:** Accordingly this is the energy used during the active state

Note 6 **Average sleep current per interval:** It is the average current over many packets for the sleep state of an interval. Sleep state is from sleep to wake during which the device is at low power and is not having any activity.
At the beginning of the sleep state the current is very low, even negative in some cases. This is because the capacitors discharge and provide energy to the system.
For longer sleep periods the current is getting higher, until it's max value for the sleep mode. This is because as the time is passing the capacitors are drained of energy and there is need for energy from the power supply.

Note 7 **Average sleep energy per interval:** Likewise as Note 6 above the energy used for the sleep state of an interval.

Note 8 **Average energy per interval:** It is the total energy for an interval including sleep and active state.

Note 9 **Average current for capture:** Is the average current of a longer time capture for more intervals (sleep+active). This is the average current expected over time for the selected operation mode.

4.4 CPU power measurement

This is a special measurement, useful for the characterization of the CPU in the chip. The table 6 provides all the related information including the uA/MIPs.

The goal is to measure the power consumption of the CPU alone, isolating the power consumption of all the surrounding logic and circuitry in the chip and outside of it.

This will be the offset in our measurements. This offset includes the power consumption on bandgap on RC16M which is always running, on DCDC and the leakages.

To avoid compiling the code for all the possible combinations we have designed a minimal bare metal application using a while(1) statement to cause the load on the CPU for the measurement and enabling the user to change dynamically the clocks using combinations of GPIOs.

The process is carefully designed not to cause additional power consumption, but requires the GPIOs to be disconnected after the desired clock is selected. The steps are described in details in paragraph 4.4.3 Measurement procedure further below in this chapter.

4.4.1 HW Preparation

Voltage: 3.0V

Power supply of the DB: through the coin cell contacts

SW project to use: *uA_per_MIPS_bare.*, The project is provided on demand.

To prepare the DB for the measurement use the Smartsnippet Studio to compile and run the code in RAM. In this test we do not want to activate or use the QSPI and the related subsystem. Keep the DB switch to COIN position. All the jumpers except the J5 (including the JTAG jumpers J14 and J8) must be removed once the code is downloaded and running in the DB.

Check that the dg_configBATTERY_LOW_LEVEL should be set at value higher than 3.0V.

4.4.2 SW preparation

The project is not provided by default with the SDK. It is provided by the Customer Support group of Dialog Semiconductor on demand to selected customers.

The project name is *uA_per_MIPS_bare*. Once the ZIP file with the project is available, unzip it under the "*projects\dk_apps\templates\uA_per_MIPS_bare*" directory of the SDK release (preferably the latest one) so that the structure of the SDK will look like in Figure 6.

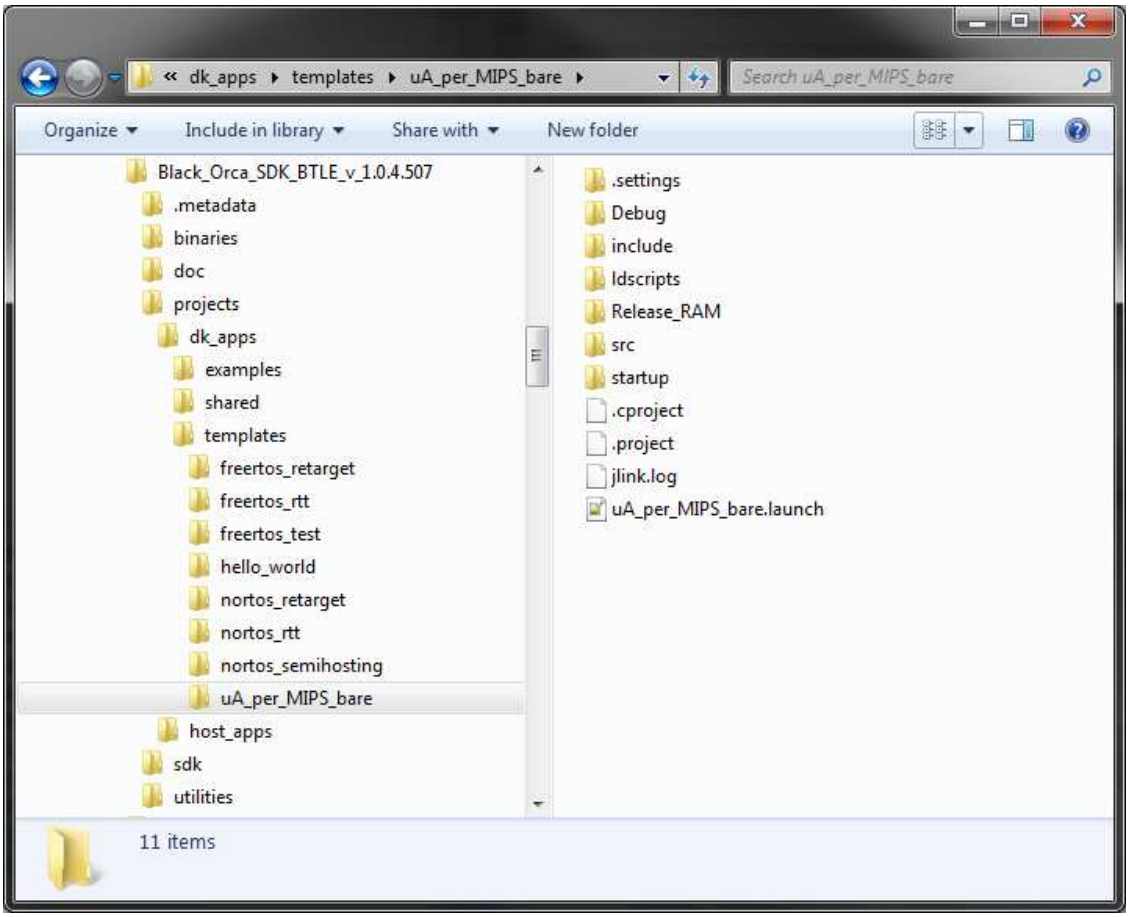


Figure 6: The unzipped project under SDK folder structure

Once the files are unzipped in place as described above, start the SmartSnippets Studio and import the project just like any other project (check the related documentation).

Compile the project. There is only one build configuration for Release RAM. Once the project is built, run it using the debug button and selecting the uA_per_MIPS_base configuration as shown in Figure 7.

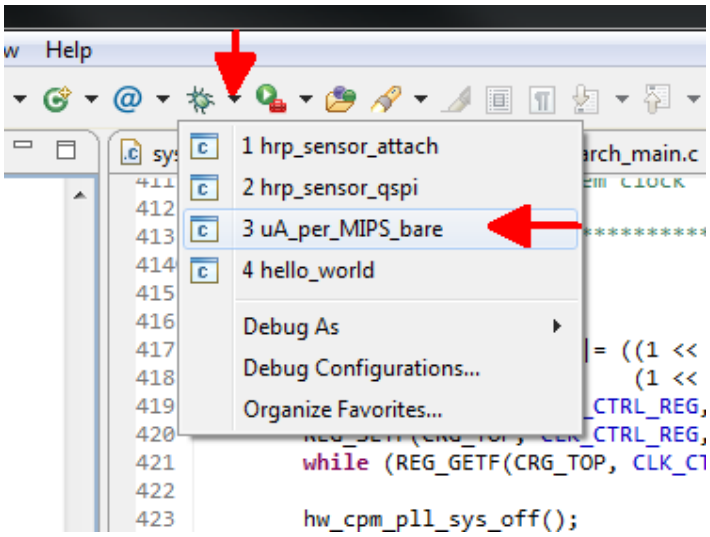


Figure 7: running the uA_per_MIPS_bare project

Finally detach the JTAG by physically removing the J8 and J14. The device is ready to be used for the measurements. The code is running from RAM and in case of power loss the development station must be re-connected, the JTAG link must be re-established and the process to run the code should be repeated up to this point.

4.4.3 Measurement procedure

To perform one or more measurements, once the code is downloaded and running in RAM, user can choose dynamically the CPU clock to measure with. There is only one mode which will not allow changing the mode after that. That is the calculation of the offset because it will stop the CPU clock thus the device will not respond any more.

The initial running clock is the XTAL16M.

There are three sets of GPIOs used for selecting the CPU clock, the CPU clock prescaler and the peripherals domain prescaler as shown in Table 2, Table 3 and Table 4.

Table 2: CPU clock select

Clock to select	P3_2	P3_1	P3_0	Decimal value
XTAL16M	0	0	0	0
PLL48	0	0	1	1
PLL96	0	1	0	2
RC16M	0	1	1	3
RC32K	1	0	0	4
XTAL32K	1	0	1	5
RCX	1	1	0	6
offset	1	1	1	7

Table 3: HCLK prescaler selection

HCLK prescaler	P3_4	P3_3	Decimal Value
DIV /1	0	0	0
DIV /2	0	1	1
DIV /4	1	0	2
DIV /8	1	1	3

Table 4: PCLK prescaler selection

PCLK prescaler	P3_6	P3_5	Decimal Value
DIV /1	0	0	0
DIV /2	0	1	1
DIV /4	1	0	2
DIV /8	1	1	3

The daughter-board has to be plugged on the motherboard, the daughter-board must be power supplied by the coin cell contacts from the external source (preferably directly from the N6705) and leave only the J5 jumper in place on the motherboard. All the rest jumpers including J8 and J14 for the JTAG must be removed.

Keep the motherboard connected thought the debug USB port to the PC to avoid leakages.

The trigger to activate a GPIO selection is the P1_3 pin which momentarily must go to logical HIGH.

When the change is accepted the after P1_3 is at logical HIGH, the D2 LED will blink 3 times indicating the acceptance.

After blinking disconnect all the GPIOs used for the selection and the P1_3.

Now the daughter-board is running at selected clock and prescalers. It is ready to measure on N6705 the power consumption.

The MIPS for the M0 CPU core used in the DA1468x are available in Table 5.

Table 5: M0 MIPS per operation Frequency

Frequency (MHz)	ARM M0 MIPS
8	6.72
12	10.08
16	13.44
24	20.16
48	40.32
96	80.64

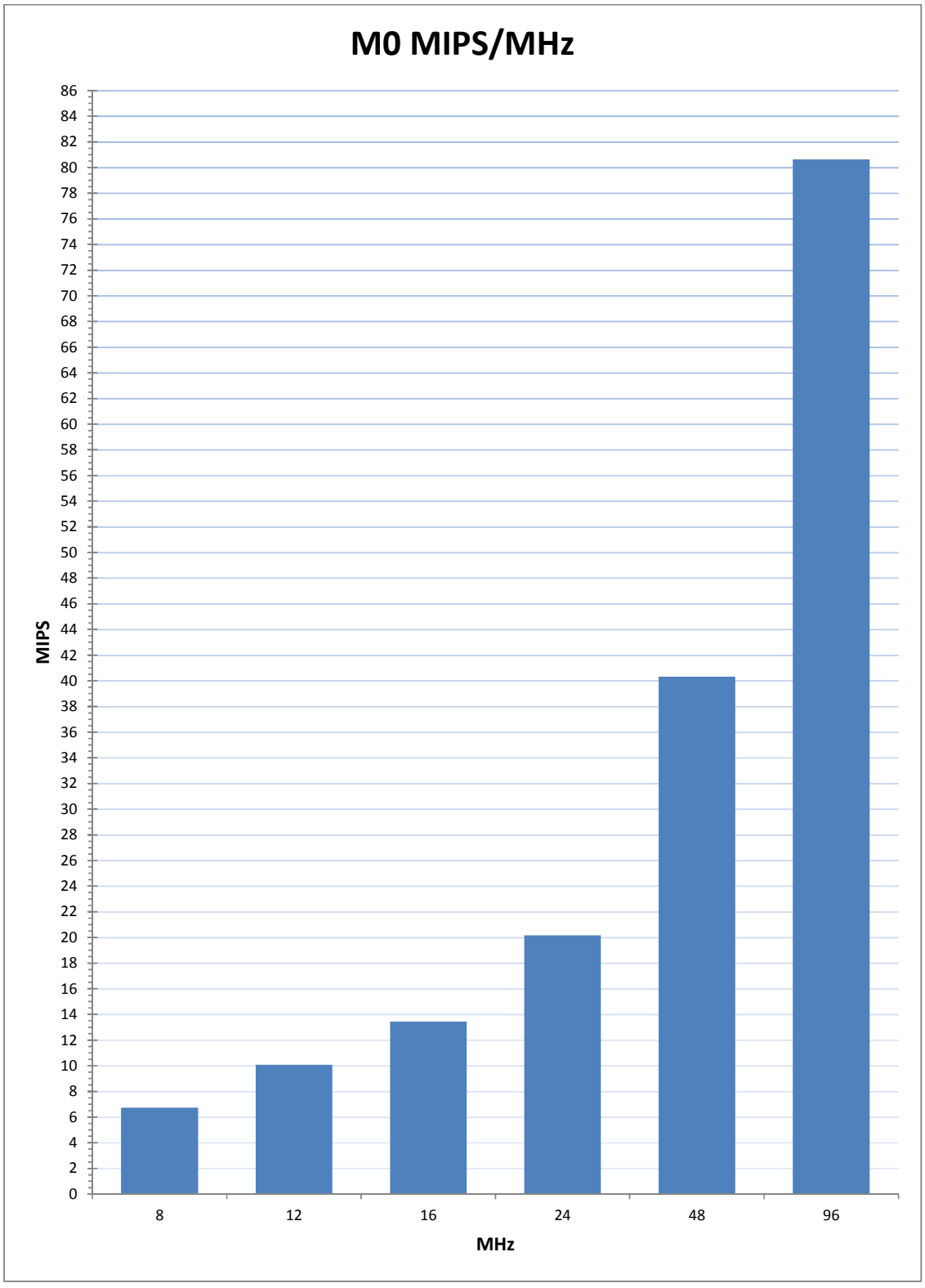


Figure 8: M0 MIPS per operation Frequency

4.4.4 Results

Table 6: MIPS, Clock & Power with HCLK/1

		HCLK/1			
		PCLK/1	**PCLK/2**	**PCLK/4**	**PCLK/8**
RC32K	*mA*	0,275103	N/A	N/A	N/A
	KHz	32			
	uA/MHz	0,041	N/A	N/A	N/A
	MIPS	0,0287			
	uA/MIPS	46,092	N/A	N/A	N/A
XTAL32K	*mA*	0,275341	N/A	N/A	N/A
	KHz	32			
	uA/MHz	0,049	N/A	N/A	N/A
	MIPS	0,0287			
	uA/MIPS	54,396			
RC16	*mA*	0,562	0,553	0,531	0,525
	MHz	16			
	uA/MHz	18,004	17,470	16,046	15,713
	MIPS	13,44			
	uA/MIPS	21,433	20,797	19,103	18,706
XTAL16M	*mA*	1,208	1,193	1,184	1,181
	MHz	16			
	uA/MHz	58,383	57,424	56,909	56,702
	MIPS	13,44			
	uA/MIPS	69,504	68,362	67,748	67,502
PLL48M	*mA*	3,413	3,345	3,301	3,278
	MHz	48			
	uA/MHz	65,400	63,993	63,059	62,595
	MIPS	40,32			
	uA/MIPS	77,857	76,182	75,070	74,518
PLL96M	*mA*	5,331	5,193	5,105	5,061
	MHz	96			
	uA/MHz	52,678	51,238	50,324	49,872
	MIPS	80,64			
	uA/MIPS	62,712	60,998	59,910	59,371

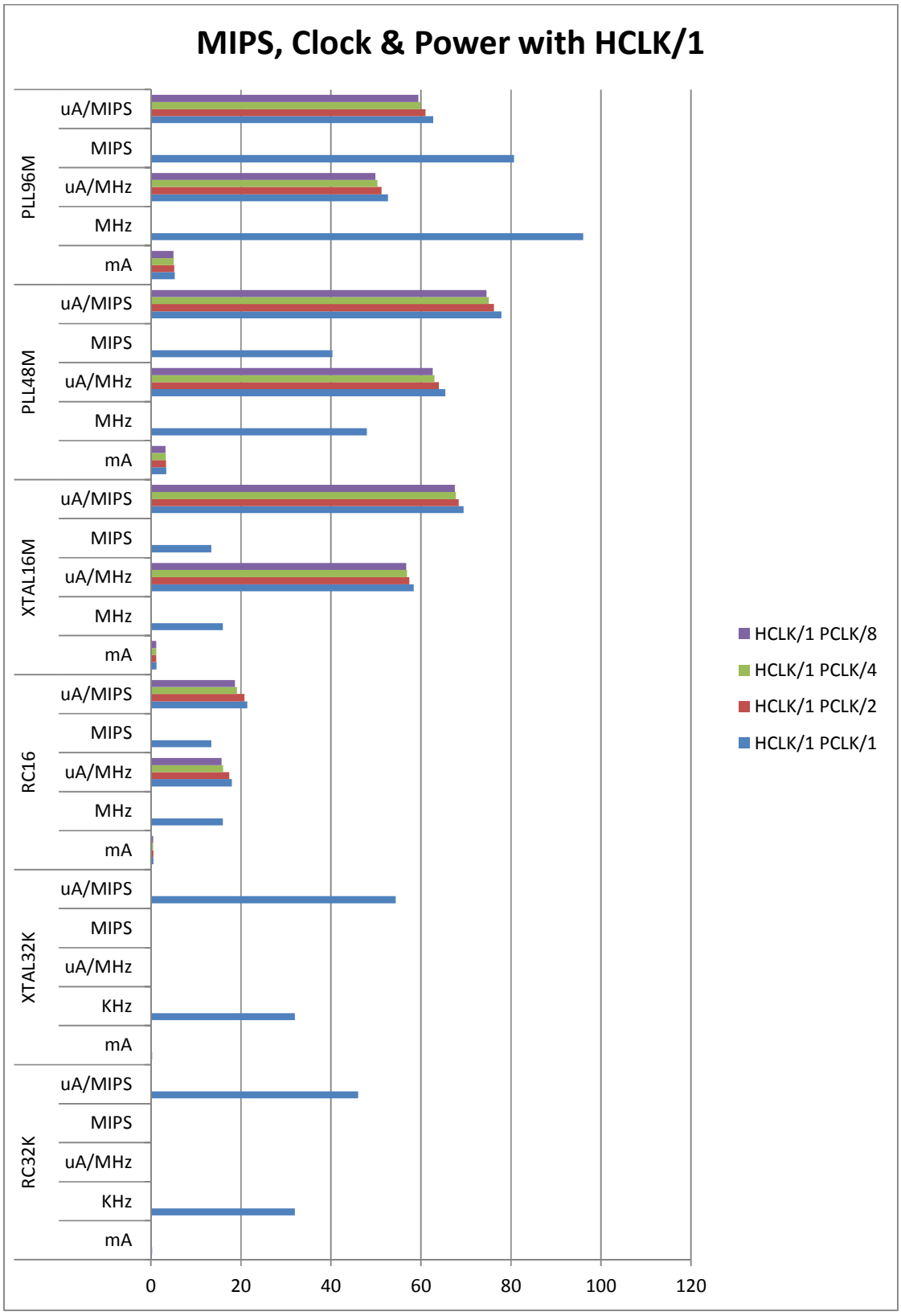


Figure 9: MIPS, Clock & Power with HCLK/1

Table 7: MIPS, Clock & Power with HCLK/2

		HCLK/2			
		PCLK/1	**PCLK/2**	**PCLK/4**	**PCLK/8**
RC16	*mA*	0,411	0,406	0,403	0,402
	MHz	8			
	uA/MHz	17,138	16,538	16,188	16,047
	MIPS	6,72			
	uA/MIPS	20,403	19,688	19,271	19,104
XTAL16M	*mA*	0,845	0,831	0,818	0,814
	MHz	8			
	uA/MHz	71,394	69,696	68,040	67,541
	MIPS	6,72			
	uA/MIPS	84,992	82,972	81,000	80,406
PLL48M	*mA*	2,523	2,479	2,457	2,447
	MHz	24			
	uA/MHz	93,736	91,900	90,960	90,532
	MIPS	20,16			
	uA/MIPS	111,591	109,404	108,285	107,776
PLL96M	*mA*	3,742	3,653	3,608	3,585
	MHz	48			
	uA/MHz	72,250	70,405	69,453	68,983
	MIPS	40,32			
	uA/MIPS	86,012	83,815	82,682	82,123

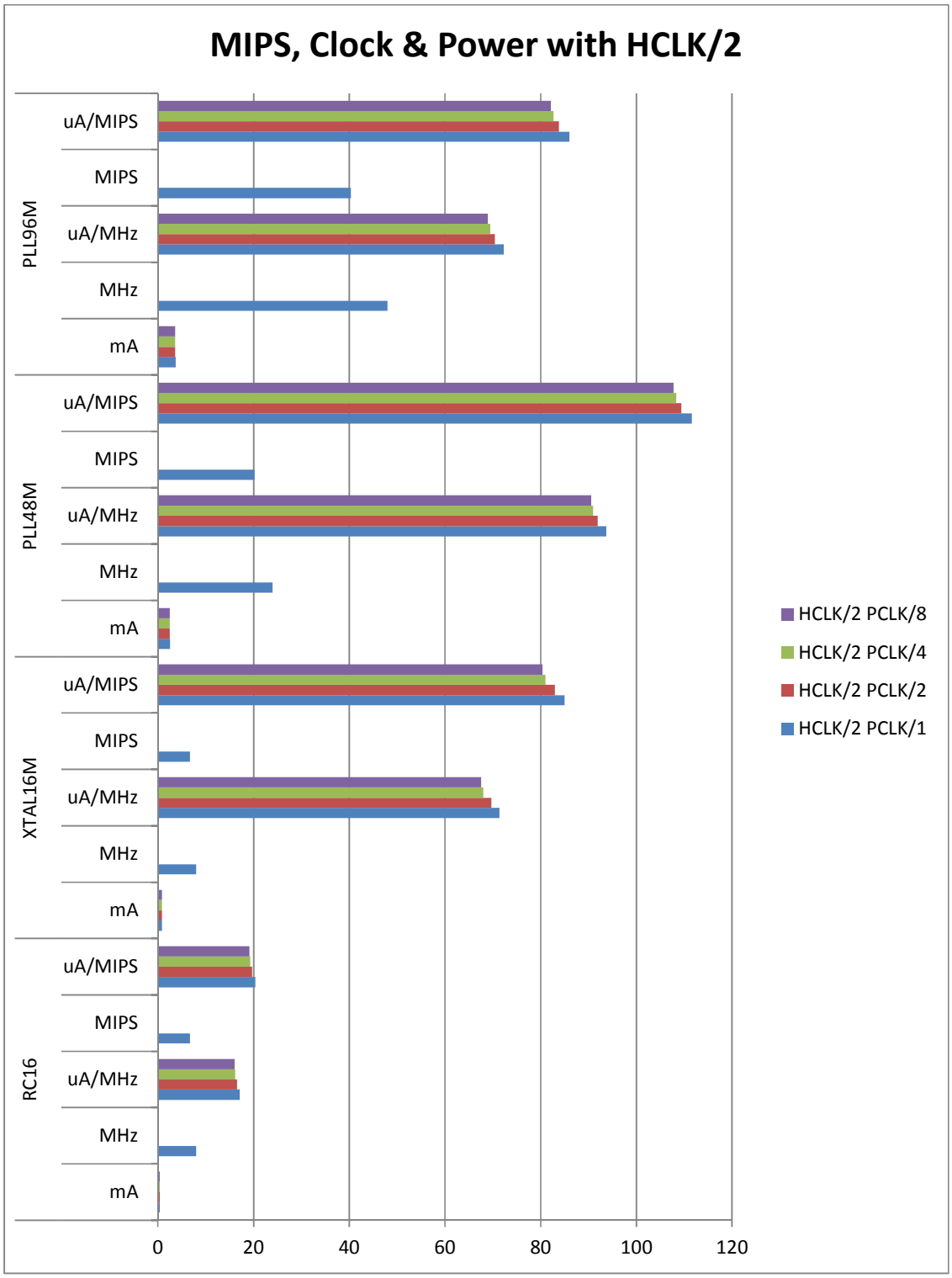


Table 8: MIPS, Clock & Power with HCLK/4

		HCLK/4			
		PCLK/1	**PCLK/2**	**PCLK/4**	**PCLK/8**
RC16	*mA*	0,355	0,353	0,352	0,351
	MHz	4			
	uA/MHz	20,357	19,799	19,499	19,299
	MIPS	3,36			
	uA/MIPS	24,234	23,570	23,213	22,975
XTAL16M	*mA*	0,696	0,688	0,683	0,680
	MHz	4			
	uA/MHz	105,623	103,572	102,249	101,549
	MIPS	3,36			
	uA/MIPS	125,742	123,300	121,725	120,892
PLL48M	*mA*	1,989	1,967	1,956	1,951
	MHz	12			
	uA/MHz	142,974	141,074	140,189	139,796
	MIPS	10,08			
	uA/MIPS	170,207	167,945	166,892	166,424
PLL96M	*mA*	2,671	2,627	2,604	2,593
	MHz	24			
	uA/MHz	99,881	98,036	97,077	96,639
	MIPS	20,16			
	uA/MIPS	118,906	116,710	115,567	115,046

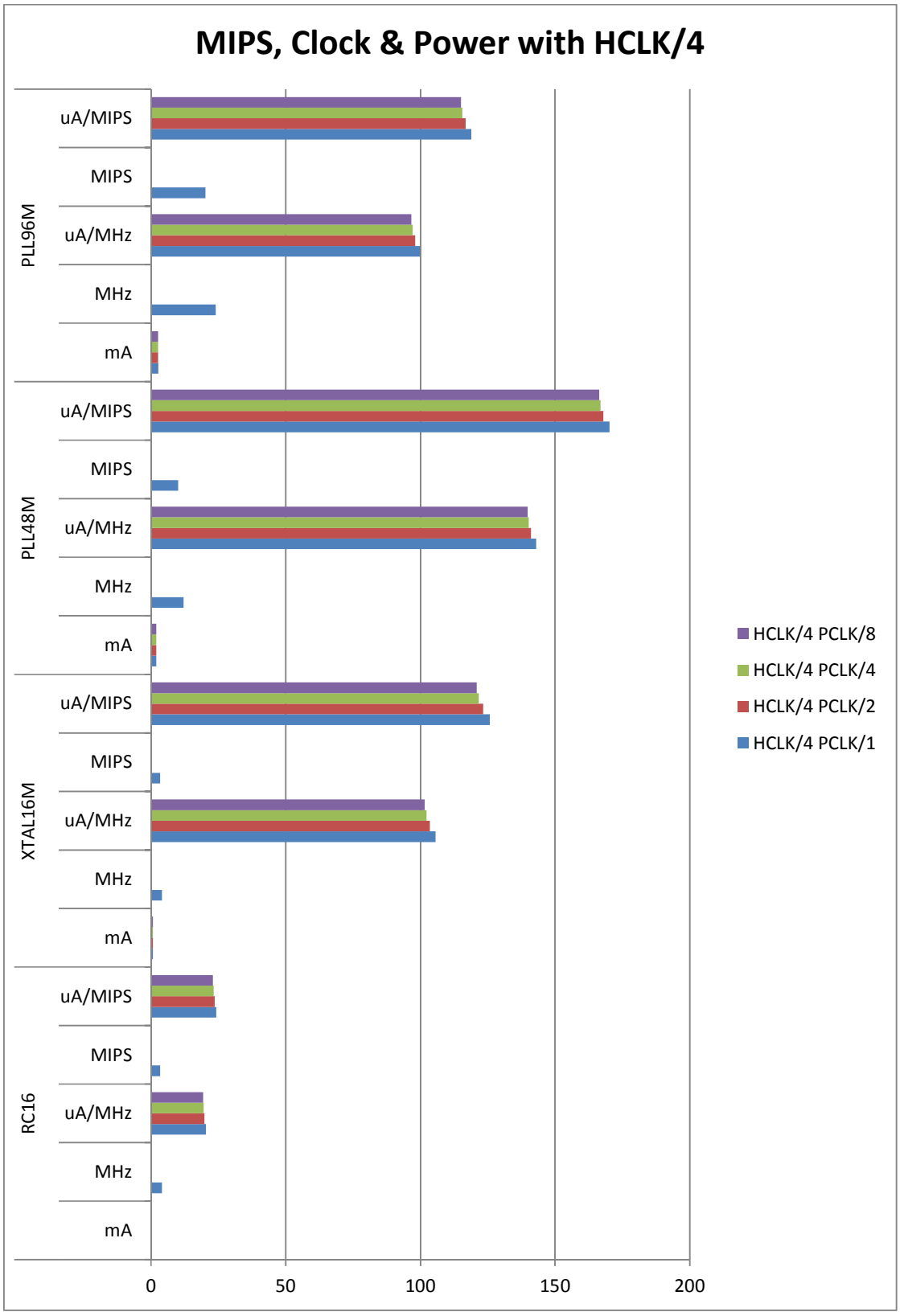
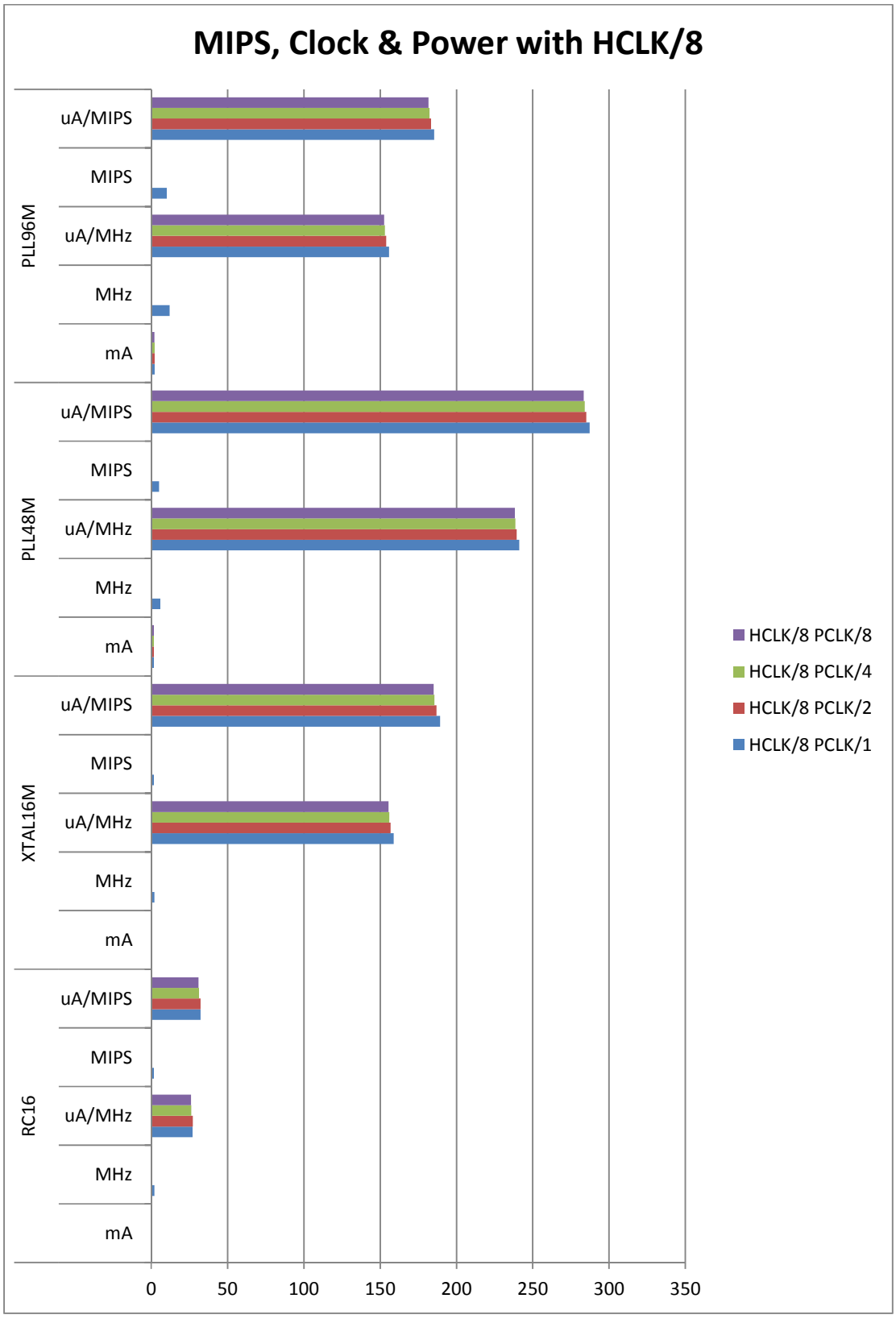


Figure 10:

Table 9: MIPS, Clock & Power with HCLK/8

		HCLK/8			
		PCLK/1	**PCLK/2**	**PCLK/4**	**PCLK/8**
RC16	*mA*	0,328	0,328	0,326	0,326
	MHz	2			
	uA/MHz	27,129	27,223	26,203	25,986
	MIPS	1,68			
	uA/MIPS	32,296	32,408	31,194	30,935
XTAL16M	*mA*	0,592	0,588	0,586	0,585
	MHz	2			
	uA/MHz	158,972	156,945	155,905	155,445
	MIPS	1,68			
	uA/MIPS	189,252	186,839	185,601	185,053
PLL48M	*mA*	1,721	1,711	1,705	1,703
	MHz	6			
	uA/MHz	241,246	239,484	238,614	238,155
	MIPS	5,04			
	uA/MIPS	287,198	285,100	284,065	283,518
PLL96M	*mA*	2,142	2,122	2,112	2,105
	MHz	12			
	uA/MHz	155,689	154,019	153,146	152,635
	MIPS	10,08			
	uA/MIPS	185,344	183,356	182,317	181,709



5 Useful tips

Each product and each application code is unique. Having said that the reader should realize that there is no single rule to achieve the optimal power consumption for all products, all applications and all configurations.

However there are some key factors to help choose the most appropriate resource to use for the product design.

Retention RAM, CACHE, QSPI FLASH

Retention RAM, CACHE and QSPI FLASH use are very tightly interconnected between them. Some common questions and their answers about this are:

1) How much retention RAM to use?

   There is not a golden rule here. The use case can even change this number. In general keep in the RAM the data, literals and code which is used very often and the access to QSPI FLASH will result in higher power consumption and slight degradation of the overall performance.

2) When is it better to increase the use of retention RAM?

   When there are pieces of the code critical for the execution (e.g. ISR sensitive to delays), or used all the time (e.g. a timer with very short period) then these pieces of the code might me better to put them in retention RAM.

   Also if there are constants and lookup tables which the code has to use often, then these might be a good idea also to keep them in the retention RAM.

3) Should I bypass the CACHE?

   No, never. Although it is possible to put the CACHE in bypass mode and have the code running directly from FLASH, the chip is designed to work with CACHE. Running code without cache results to 15-20 times slower code execution due to delays inserted by the QSPI FLASH speed.

4) When to use Power Down and when Power OFF modes for the QSPI FLASH?

   Power OFF mode for the QSPI FLASH is suitable to be used and result in lower power consumption only if the device mainly Deep sleeps for long time periods (e.g. minutes). For all the rest of the use cases where the device is waking even every few seconds, the Power Down mode is more efficient.

How much retention RAM will be used is defined in the custome_confir_qspi.h by setting the following definitions:

```
#define dg_configMEM_RETENTION_MODE                 (0x1F)
#define dg_configMEM_RETENTION_MODE_PRESERVE_IMAGE  (0x1F)
```

These two must have the same value and it is the retention RAM value to put in the register `PMU_CTRL_REG` in `RETAIN_RAM` field. Refer to DA1468x datasheet for details on the register entry and the possible values.

The power budget for each block of retained RAM is shown in Table 10.

Table 10: Retained RAM power budget per cell

RAM cell	Deep Sleep Mode (No clock)	Extended Sleep Mode (RCX)
RAM1 (8 kB)	190	260
RAM2 (24 kB)	431	543
RAM3 (32 kB)	570	698
RAM4 (32 kB)	623	714
RAM5 (32 kB)	790	980

Note 1 Measured on DA14680-AD silicon. The values are in nA

```
#define dg_configSHUFFLING_MODE                (0x3)
```

This defines the RAM shuffling to be used on the SYS_CTRL_REG field REMAP_RAMS. Refer to DA1468x datasheet for details on the values valid for this field of the register.

Deep Sleep, Extended Sleep or Hibernation modes

The power-save modes we are discussing here refer to the system blocks of DA1468x, not the BLE block. The BLE block does power save on its own by default.

For normal operation the modes of interest are the Extended Sleep and the Deep Sleep modes.

In Extended sleep the BLE is available and the rest of the system is sleeping to lower the power consumption. The system wakes periodically and runs scheduled tasks or wakes on interrupt (BLE or external HW source) and runs the ISR.

In Deep Sleep there is no clock, the BLE is turned off and only an external interrupt signal on the configured pins can wake the device.

Hibernation mode is a special mode to be used for shipping the final product to market without draining the battery.

Low Power (LP) Clocks: RCX, XTAL32K, RC32K

There are three LP clocks available. The RC32K is not an accurate clock and mainly is used when waking from clockless sleep modes like Deep Sleep to run the HW FSM. RC32K is the only clock which will be enabled upon event while in Deep Sleep and the only running clock if waking from VBUS interrupt (#define dg_configLOW_VBAT_HANDLING 1).

RCX and XTAL32K are accurate clocks to be used with extended sleep modes where we need to have accuracy on wake-sleep cycles in order to serve on time the BLE operations upon wakeup. The provided SDK is taking care of the clock use and the developer should just select the one to use in the custom_config_qspi.h or custom_config_ram.h files.

RCX Accuracy is less than 200ppm.

Fast clocks: RC16M, XTAL16M, PLL48MHz and PLL96MHz

RC16M is a fast internal clock, but not very accurate. Is used primarily by the DCDC, and the ROM-Booter to start the device. If it is disabled the DCDC cannot be used.

The XTAL16M is an accurate clock dependent on the external 16MHz crystal. It is used for normal operation with the BLE protocol enabled.

PLL48 and PLL96 are PLL clocks internally generated. They require the XTAL16M clock and they offer higher performance but higher power consumption too.

Avoid current leakage on peripheral GPIOs

Make sure that all the unused peripheral GPIOs are configured as input_pulldown which is the default power-up mode for the GPIOs.

Revision history

Revision	Date	Description
1.0	18-May-2016	Initial version.
1.1	25-May-2016	Added note for suggested sampling rate in chapter 4.2
1.2	20-Jul-2016	Updated measurements for DA14681-01 silicon Updated dataloger screenshots Added explanations for the measurements in Table 1

Status definitions

Status	Definition
DRAFT	The content of this document is under review and subject to formal approval, which may result in modifications or additions.
APPROVED or unmarked	The content of this document has been approved for publication.

Disclaimer

RoHS Compliance

Dialog Semiconductor complies to European Directive 2001/95/EC and from 2 January 2013 onwards to European Directive 2011/65/EU concerning Restriction of Hazardous Substances (RoHS/RoHS2).
Dialog Semiconductor's statement on RoHS can be found on the customer portal https://support.diasemi.com/. RoHS certificates from our suppliers are available on request.

Contacting Dialog Semiconductor

United Kingdom (Headquarters)
Dialog Semiconductor (UK) LTD
Phone: +44 1793 757700

Germany
Dialog Semiconductor GmbH
Phone: +49 7021 805-0

The Netherlands
Dialog Semiconductor B.V.
Phone: +31 73 640 8822

Email:
enquiry@diasemi.com

North America
Dialog Semiconductor Inc.
Phone: +1 408 845 8500

Japan
Dialog Semiconductor K. K.
Phone: +81 3 5425 4567

Taiwan
Dialog Semiconductor Taiwan
Phone: +886 281 786 222

Web site:
www.dialog-semiconductor.com

Singapore
Dialog Semiconductor Singapore
Phone: +65 64 8499 29

Hong Kong
Dialog Semiconductor Hong Kong
Phone: +852 3769 5200

Korea
Dialog Semiconductor Korea
Phone: +82 2 3469 8200

China (Shenzhen)
Dialog Semiconductor China
Phone: +86 755 2981 3669

China (Shanghai)
Dialog Semiconductor China
Phone: +86 21 5424 9058