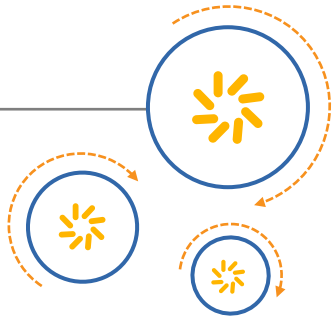
Qualcomm Technologies, Inc.



DragonBoard™ 410c based on Qualcomm® Snapdragon™ 410E Processor

APQ8016E Clock Plan

January 11, 2017

Questions or comments: http://www.96boards.org/forums/forum/products/dragonboard/

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

LM80-P0436-55 Rev. A

Revision history

Revision	Date	Description
A	January 2017	Initial release

Contents

Figures

Tables

1 Introduction

1.1 Purpose

This document describes the clock plan for the APQ8016E ASIC. It describes the crystals, PLL plans, partition of clock ownership to different subsystems, and performance levels of NOCs, the processors, multimedia modules, and DDR.

This document is written for APQ8016E customers to help them understand the implementation of clock plans in the released software.

1.2 Conventions

Function declarations, function names, type declarations, attributes, and code samples appear in a different font, for example, `#include`.

1.3 Technical assistance

Forum support for the DragonBoard 410c is available at 96boards.org.

2 Clock plan description

This chapter provides details of the clock plan. Figure 2-1 shows an overview of cores and buses in the chipset. The Wireless Connectivity Subsystem (WCNSS), Applications Processor Subsystem (APSS), Multimedia Subsystem (MMSS), and Resource Power Manager (RPM) are the cores in the chipset. Network-on-a-Chip (NoC) represents the interconnects that connect subsystems and blocks.

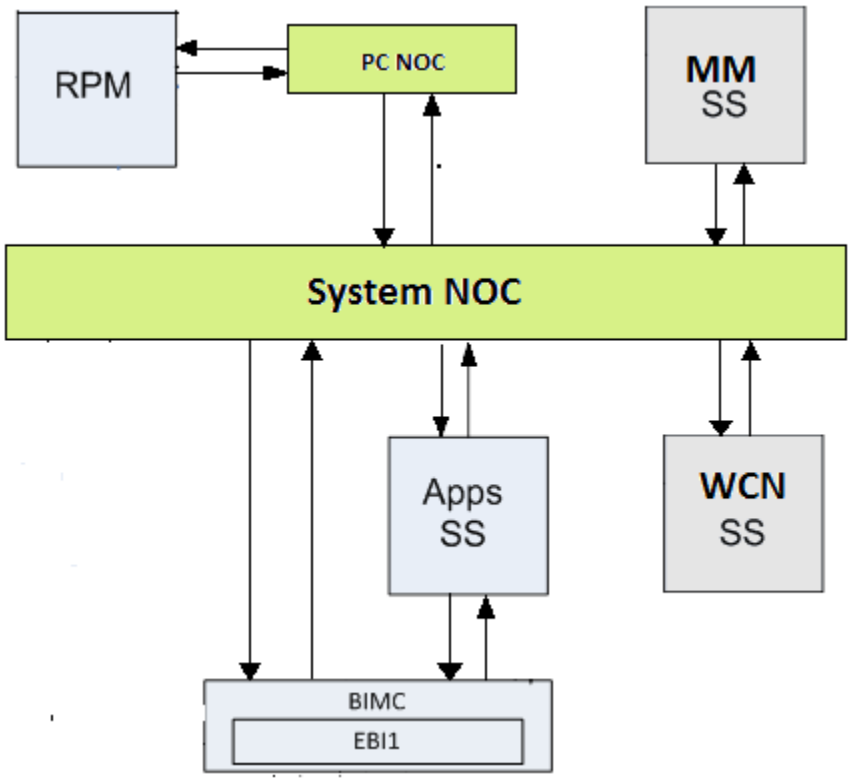


Figure 2-1 APQ8016E bus and core topology

2.1 Source crystals

The APQ8016E chip has one crystal, XO as listed in Table 2-1. These crystals makes up the source for all PLLs and can be the source for other clocks. The main source of clocks is the crystal oscillator which operates at 19.2MHz. All other clocks required by the different subsystems are derived from this clock.

Table 2-1 Oscillators

Name	Frequency (MHz)
XO	19.200

The chip does not have a separate sleep crystal. The sleep clock, whose oscillation frequency is 32.768 kHz, is used as the source for time tick and is generated from XO using a divisor of 586.

2.2 PLL configuration

The APQ8016E chip has six PLLs, as listed in Table 2-2. PLLs are partitioned to subsystems. Only the core that owns each PLL can configure it. Ownership of the PLLs is noted in the last column of Table 2-2.

The clocks to the different subsystems are derived from these PLL's, depending upon the frequency requested by the subsystem. The PLLs are configured during boot time and the clock frequencies cannot be changed.

Table 2-2 PLLs

PLL	Source	Output frequency (MHz)	Voteable	Notes	Subsystem
GPLL0	XO	800.0000	Yes	General purpose	RPM
GPLL1	XO	903.1680	Yes	LPAAUDIO_PLL	RPM
GPLL2	XO	930.0000	Yes	VFE (Turbo)	APPS
BMIC_PLL	XO	1066.6667	Yes	BIMC (DDR)	RPM
WCNSS_PLL	XO	800.0000	Yes	WCNSS PLL	WCNSS
APSS_PLL	XO	Varies	No	ACPU PLL 1.2 GHz max	APPS

Figure 2-2 shows the clock block diagram. This diagram is a high-level overview of the clock.

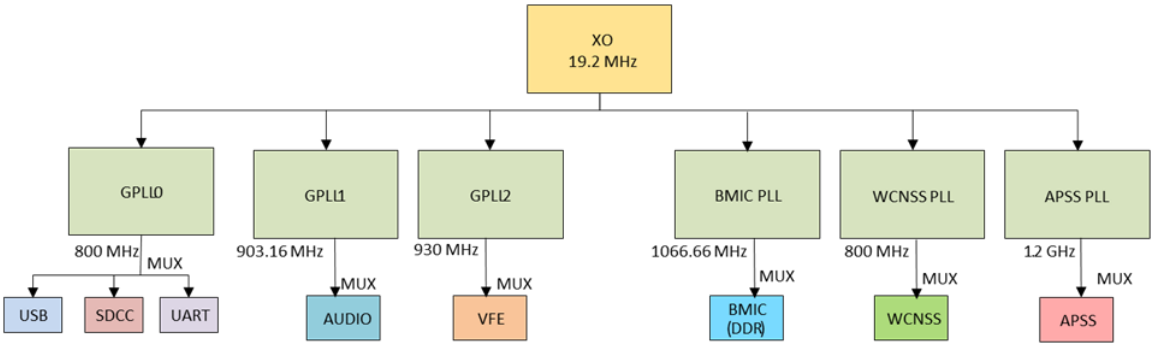


Figure 2-2 Clock block diagram

2.3 NoC (bus) and EBI1 configuration

The RPM own control for configuration of the bus and DDR clocks. The RPM chooses a performance level that satisfies all outstanding requests for a bus. Requests for clock speed can come from any client/subsystem to RPM (e.g., the MSS, APSS, or WCNSS). The RPM aggregates all the clock requests and applies the specific settings to the subsystem or client.

The tables in this section reflect the data available at the time this document was published.

2.3.1 System NoC

Table 2-3 lists the system NoC performance levels.

Any client or subsystem (e.g., USB, SDCC, UART, etc.) can request or vote for a particular frequency using the APIs mentioned in Section 3. The RPM clock driver will aggregate and configure the system NoC clock to the max voted level.

Table 2-3 System NoC performance levels

Performance level	Frequency (MHz)	Source
0	19.200	XO
1	50.000	GPLL0
2	100.000	GPLL0
3	133.333	GPLL0
4	160.000	GPLL0
5	200.000	GPLL0
6	266.667	GPLL0

2.3.2 Peripheral and configuration NoC

Table 2-4 shows the peripheral and configuration NoC performance levels.

Any client or subsystem (e.g., USB, SDCC, UART, etc.) can request or vote for a particular frequency using APIs mentioned in Section 3. The RPM clock driver will aggregate and configure the PC NOC clock to the max voted level.

Table 2-4 Peripheral NoC performance levels

Performance level	Frequency (MHz)	Source
0	19.200	XO
1	50.000	GPLL0
2	100.000	GPLL0
3	133.333	GPLL0

2.3.3 Bus Interface Memory Controller (BIMC)

Table 2-5 shows the BIMC performance levels.

Any client or subsystem (e.g., ACPU, graphics, display, USB, SDCC, UART, etc.) can request or vote for a particular frequency using APIs mentioned in Section 3. The RPM clock driver will aggregate and configure the BIMC clock to the max voted level.

Table 2-5 BIMC performance levels for APQ8016E

Performance level	Frequency (MHz)	Source
0	9.600	XO
1	50.000	GPLL0
2	100.000	GPLL0
3	200.000	GPLL0
4	400.000	GPLL0
5	533.000	BIMC_PLL

2.4 Processor performance levels

The various processors in the system can be configured to operate at various speeds. This section outlines the processors that can have their operating speeds changed.

2.4.1 RPM processor

This processor is part of the RPM subsystem and is a Cortex M3-based processor. The clock rates cannot be modified by the user.

The performance levels for the RPM processor are listed in Table 2-6.

Table 2-6 RPM processor core performance levels

Performance level	Frequency (MHz)	Source
0	19.200	XO
1	100.000	GPLL0
2	177.800	GPLL0

2.4.2 Application processors

The application processor core performance levels are shown in Table 2-7. This table provides clock plan data for the Cortex A53 processors (four cores). This table reflects the data available at the time this document was published.

All the APPS CPU cores powered from VDD_APC rail support three different corner voltages.

- Called SVS (low voltage – 1.05v)
- Nominal (medium voltage – 1.15v)
- Turbo (high voltage – 1.35v).

These voltage levels are applied based on the load of the APPS (APPS CPU cores) subsystem.

Table 2-7 Application processor core performance levels for 1.2 GHz speed chips

Performance level	Frequency (MHz)	Source	VDD_APC
0	200.000	GPLL0	SVS
1	400.000	GPLL0	SVS
2	533.330	GPLL0	Nominal
3	800.000	GPLL0	Nominal
4	998.400	APSSPLL	Turbo
5	1094.400	APSSPLL	Turbo
6	1152.000	APSSPLL	Turbo
7	1209.000	APSSPLL	Turbo

2.4.3 WCNSS

The APQ8016E has a WCNSS that runs on an ARM9™ processor. The performance levels for the processor in the WCNSS are listed in Table 2-8. These clocks cannot be modified by the user.

Table 2-8 WCNSS processor core performance levels

Performance level	Frequency (MHz)	Source
0	19.200	XO
1	32.000	WCNPLL
2	120.000	WCNPLL
3	240.000	WCNPLL

2.5 Multimedia module performance levels

This section describes performance levels for various modules in the MMSS. These include graphics, video, JPEG, multimedia display subsystem (MDSS), and video frontend (VFE) modules. The performance levels are controlled by the APSS. For actual data that is used for the release, see the source code:

- Android builds – `kernel/drivers/clk/qcom/clock-gcc-8916.c`
- Debian builds – `kernel/drivers/clk/qcom/…`

Table 2-9 Graphics (3D graphics core) performance levels (struct ftbl_gcc_oxili_gfx3d_clk[])

Performance level	Frequency (MHz)	Source
0	19.200	XO
1	50.000	GPLL0
2	80.000	GPLL0
3	100.000	GPLL0
4	160.000	GPLL0
5	177.780	GPLL0
6	200.000	GPLL0
7	266.670	GPLL0
8	294.912	GPLL1
9	310.000	GPLL2
10	400.000	GPLL0

Table 2-10 Video core performance levels (struct ftbl_gcc_venus0_vcodec0_clk[])

Performance level	Frequency (MHz)	Source
0	100.000	GPLL0
1	160.000	GPLL0
2	228.57	GPLL0

Table 2-11 MDSS performance levels (struct ftbl_gcc_mdss_mdp_clk[])

Performance level	Frequency (MHz)	Source
0	50.000	GPLL0
1	80.000	GPLL0
2	100.000	GPLL0
3	160.000	GPLL0
4	177.780	GPLL0
5	200.000	GPLL0
6	266.670	GPLL0
7	320.000	GPLL0

Table 2-12 VFE performance levels (struct ftbl_gcc_camss_vfe0_clk[])

Performance level	Frequency (MHz)	Source
0	50.000	GPLL0
1	80.000	GPLL0
2	100.000	GPLL0
3	160.000	GPLL0
4	177.780	GPLL0
5	200.000	GPLL0
6	266.670	GPLL0
7	320.000	GPLL0
8	400.000	GPLL0
9	465.000	GPLL2

NOTE: VFE clock is not enabled by default. The clock will be enabled when the camera is connected.

`gcc_pdm_ahb_clk` is not enabled by default. Once it is enabled, only 100MHz is seen, as this is the maximum frequency in Table 2-13.

The power manager aggregates the requirement votes from audio/voice clients and then sends those aggregated votes to the clock driver to achieve the desired frequency. The clock driver chooses the next available frequency corner based on the votes and sets the clock.

Table 2-13 ULT AUDIO AHB performance levels

Performance level	Frequency (MHz)	Source
0	3.200	XO
1	6.400	XO
2	9.600	XO
3	19.200	XO
4	40.000	GPLL0
5	66.670	GPLL0
6	80.000	GPLL0
7	100.000	GPLL0

AHB (`gcc_blsp1_ahb_clk`) is 100MHz. Less than 100MHz cannot be set.

Table 2-14 ULT AUDIO LPAIF performance levels (I2S clock)

Performance level	Frequency (MHz)	Source
0	0.256	XO
1	0.512	XO
2	0.7056	GPLL1
3	0.768	XO
4	0.800	XO
5	1.024	GPLL1
6	1.4112	GPLL1
7	1.536	XO
8	1.6	XO
9	2.048	GPLL1
10	2.4	XO
11	2.8224	GPLL1
12	3.072	GPLL1
13	4.096	GPLL1
14	4.8	XO
15	5.6448	GPLL1
16	6.144	GPLL1
17	8.192	GPLL1
18	9.6	XO
19	11.2896	GPLL1
20	12.288	GPLL1

2.6 Clock ownership

This section lists the clocks by ownership. Only the owner of the clock can control the clock.

2.6.1 RPM-owned clocks

The RPM owns and manages the globally shared resource clocks such as NoCs and DDR. Other cores use messages to send requests to enable, disable, or set a minimum rate for these clocks.

- SYSTEM_NOC_AXI_CLK
- PERIPH _CONFIG_NOC_AHB_CLK
- BIMC_CLK
- RPM_PROC_CLK
- SPMI_AHB_CLK
- SPMI_SER_CLK
- RBCPR_CLK
- QDSS clocks

2.6.2 APSS-owned clocks

The APSS owns and manages the clocks discussed in this section. All the clocks mentioned here are defined in clock-gcc-8916.c.

2.6.2.1 System-related clocks (bus, peripheral, and others)

- CE1_CLK
- SDC[n]_APPS_CLK
- USB and HSIC clocks
- BLSP_QUP_I2C clocks
- BLSP_QUP_SPI clocks
- BLSP_UART clocks
- GP clocks

2.6.2.2 Multimedia-related clocks

- VCODEC0_CLK
- JPEG clocks
- CAMERA, VFE, and CSI clocks
- DISPLAY and MDP clocks

2.6.3 WCNSS-owned clocks

The clocks owned and managed by WCNSS are:

- WCN_CPU and AHB clocks
- BT_32MHz_CLK
- BT_FM_19.2MHz_CLK
- WLAN_PHY_FAST clocks
- WLAN_PHY clocks
- WLAN ADC/DAC clocks
- WLAN_RFIF_CLK

3 Clock APIs

3.1 struct clk *clk_get

Returns a structure clk corresponding to the clock producer.

```
struct clk *clk_get(struct device *dev, const char *id);
```

Return value: valid structure; or valid IS_ERR() condition containing the error.

3.2 clk_prepare

`clk_prepare` may sleep, which differentiates it from `clk_enable`. In a simple case, `clk_prepare` can be used instead of `clk_enable` to ungate a clk if the operation may sleep. In fact, `clk_prepare` must be called before `clk_enable`.

```
int clk_prepare(struct clk *clk);
```

Return value: 0 on success; or ERROR otherwise.

3.3 clk_unprepare

`clk_unprepare` may sleep, which differentiates it from `clk_disable`. In a simple case, `clk_unprepare` can be used instead of `clk_disable` to gate a clk if the operation may sleep. In fact, `clk_disable` must be called before `clk_unprepare`.

```
void clk_unprepare(struct clk *clk);
```

3.4 clk_enable

Enables the clock. `clk_enable` must not sleep, which differentiates it from `clk_prepare`. In a simple case, `clk_enable` can be used instead of `clk_prepare` to ungate a clk if the operation will never sleep. One example is a SoC-internal clk that is controlled via simple register writes. In the complex case, a clk ungate operation may require a fast and a slow part. It is this reason that `clk_enable` and `clk_prepare` are not mutually exclusive. In fact, `clk_prepare` must be called before `clk_enable`.

```
int clk_enable(struct clk *clk);
```

Return value: 0 on success; or ERROR otherwise.

3.5 clk_disable

`clk_disable` must not sleep, which differentiates it from `clk_unprepare`. In a simple case, `clk_disable` can be used instead of `clk_unprepare` to gate a clk if the operation is fast and will never sleep. One example is a SoC-internal clk which is controlled via simple register writes. In the complex case, a clk gate operation may require a fast and a slow part. It is this reason that `clk_unprepare` and `clk_disable` are not mutually exclusive. In fact, `clk_disable` must be called before `clk_unprepare`.

```
void clk_disable(struct clk *clk);
```

3.6 clk_prepare_enable

Using this API, we can prepare and enable the specified clock at once in a non-atomic context.

```
static inline int clk_prepare_enable(struct clk *clk);
```

3.7 clk_disable_unprepare

Using this API, we can disable and unprepare the specified clock at once in the non-atomic context.

```
static inline void clk_disable_unprepare(struct clk *clk);
```

3.8 clk_set_rate

Using this we can specify a new rate for clk. The clk whose rate is being changed and the rate new rate for that particular clock.

```
int clk_set_rate(struct clk *clk, unsigned long rate);
```

Return value: 0 on success or –ERROR otherwise

3.9 clk_get_rate

Using this we can verify/get the clock rate.

```
unsigned long clk_get_rate(struct clk *clk);
```

Return value: returns the rate of the clock.

Example code:

```
Struct clk *pwm_gp1_clk;
pwm_gp1_clk = clk_get(&pwm_dev, "gcc_camss_gp1_clk");
if (IS_ERR(pwm_gp1_clk))
{
printk("%s: Get cam_gp1_clk error!!!\n", __func__);
pwm_gp1_clk = NULL;
return;
}
pr_err("%s: clk_set_rate \n", __func__);
clk_set_rate(pwm_gp1_clk, 100000000);

pr_err("%s: clk_prepare_enable \n", __func__);
clk_prepare_enable(pwm_gp1_clk);
```

Code Reference: /kernel/drivers/clk/clk.c

4 Clock debugging using ADB commands

The following ADB commands help to enable, disable, and measure the different subsystem clocks.

- adb devices
- adb root
- adb remount
- adb shell
- root@msm8916_64:/sys/kernel/debug/clk/gcc_gp1_clk # cat list_rates

  19200000

root@msm8916_64:/sys/kernel/debug/clk/gcc_gp1_clk # cat print

xo_clk_src

xo_clk_src

gp1_clk_src

CMD_RCGR: 0x80000000

CFG_RCGR: 0x00000000

M_VAL: 0x00000000

N_VAL: 0x00000000

D_VAL: 0x00000000

gcc_gp1_clk

CBCR: 0x80000000

root@msm8916_64:/sys/kernel/debug/clk/gcc_gp1_clk # cat measure

0

root@msm8916_64:/sys/kernel/debug/clk/gcc_gp1_clk # cat enable

0

root@msm8916_64:/sys/kernel/debug/clk/gcc_gp1_clk # echo 1 > enable

root@msm8916_64:/sys/kernel/debug/clk/gcc_gp1_clk # cat enable

1

root@msm8916_64:/sys/kernel/debug/clk/gcc_gp1_clk # cat measure

19200036

root@msm8916_64:/sys/kernel/debug/clk/gcc_gp1_clk # cat print

xo_clk_src

```
xo_clk_src
gp1_clk_src
        CMD_RCGR: 0x00000000
        CFG_RCGR: 0x00000000
          M_VAL: 0x00000000
          N_VAL: 0x00000000
          D_VAL: 0x00000000
gcc_gp1_clk
           CBCR: 0x00000001
root@msm8916_64:/sys/kernel/debug/clk/gcc_gp1_clk # echo 0 > enable
root@msm8916_64:/sys/kernel/debug/clk/gcc_gp1_clk # cat enable
0
root@msm8916_64:/sys/kernel/debug/clk/gcc_gp1_clk # cat measure
0
root@msm8916_64:/sys/kernel/debug/clk/gcc_gp1_clk # cat print
xo_clk_src
xo_clk_src
gp1_clk_src
        CMD_RCGR: 0x80000000
        CFG_RCGR: 0x00000000
          M_VAL: 0x00000000
          N_VAL: 0x00000000
          D_VAL: 0x00000000
gcc_gp1_clk
           CBCR: 0x80000000
root@msm8916_64:/sys/kernel/debug/clk/gcc_gp1_clk #
```

Debugging of 3D graphics clock (gcc_oxili_gfx3d_clk)

==========

```
root@msm8916_64:/sys/kernel/debug/clk/gcc_oxili_gfx3d_clk # cat print
xo_clk_src
xo_clk_src
gfx3d_clk_src
```

```
        CMD_RCGR: 0x80000000
        CFG_RCGR: 0x00000001
gcc_oxili_gfx3d_clk
          CBCR: 0x80004ff0
root@msm8916_64:/sys/kernel/debug/clk/gcc_oxili_gfx3d_clk # cat enable
0
root@msm8916_64:/sys/kernel/debug/clk/gcc_oxili_gfx3d_clk # echo 1 > enable
root@msm8916_64:/sys/kernel/debug/clk/gcc_oxili_gfx3d_clk # cat enable
1
root@msm8916_64:/sys/kernel/debug/clk/gcc_oxili_gfx3d_clk # cat print
xo_clk_src
xo_clk_src
gfx3d_clk_src
        CMD_RCGR: 0x80000000
        CFG_RCGR: 0x00000001
gcc_oxili_gfx3d_clk
          CBCR: 0x80004ff1
root@msm8916_64:/sys/kernel/debug/clk/gcc_oxili_gfx3d_clk # cat measure
0
root@msm8916_64:/sys/kernel/debug/clk/gcc_oxili_gfx3d_clk # cat rate
19200000
root@msm8916_64:/sys/kernel/debug/clk/gcc_oxili_gfx3d_clk # echo 50000000 > rate
root@msm8916_64:/sys/kernel/debug/clk/gcc_oxili_gfx3d_clk # cat rate
50000000
root@msm8916_64:/sys/kernel/debug/clk/gcc_oxili_gfx3d_clk # cat print
xo_clk_src
xo_clk_src
gpll0_aux_clk_src
       APPS_VOTE: 0x00000001
gfx3d_clk_src
        CMD_RCGR: 0x80000000
        CFG_RCGR: 0x0000031f
gcc_oxili_gfx3d_clk
          CBCR: 0x80004ff1
root@msm8916_64:/sys/kernel/debug/clk/gcc_oxili_gfx3d_clk #
```

Debugging of system NOC clock (snoc_clk)

==========

root@msm8916_64:/sys/kernel/debug/clk # cd snoc_clk

root@msm8916_64:/sys/kernel/debug/clk/snoc_clk # ls

root@msm8916_64:/sys/kernel/debug/clk/snoc_clk # cat measure

199998901

root@msm8916_64:/sys/kernel/debug/clk/snoc_clk # cat enable

1

root@msm8916_64:/sys/kernel/debug/clk/snoc_clk # echo 266000000 > rate

root@msm8916_64:/sys/kernel/debug/clk/snoc_clk # cat measure

266667704

root@msm8916_64:/sys/kernel/debug/clk/snoc_clk # cat print

snoc_clk

root@msm8916_64:/sys/kernel/debug/clk/snoc_clk #

NOTE: `snoc_clk` is 200Mhz (199998901 Hz) because the peripherals (USB debugging cable connected) have requested the same clock so that the RPM aggregates all the clocks requested by peripherals, and sets the maximum of that. The `snoc_clk` rate cannot be set below that frequency.

See Qualcomm Snapdragon 410E Processor APQ8016E Hardware Register Description (LM80-P0436-13) for all the hardware register configuration information.

- https://developer.qualcomm.com/download/sd410/hardware-register-description-qualcomm-snapdragon-410.pdf

Check clock availability in: `/kernel/drivers/clk/qcom/clock-gcc-8916.c`

NOTE: If the APQ8016E DragonBoard schematics are downloaded, you can also debug the clocks using CRO or Logic Analyzer.

5 Source code references

Android source code:

Android source code is published on http://codeaurora.org (CAF).

The following document shows how to setup a Linux computer to pull the source code from CAF (and elsewhere):

- https://developer.qualcomm.com/download/db410c/linux-android-software-build-and-installation-guide.pdf

The following zip files have a shell script that actually runs the pull:

- https://developer.qualcomm.com/download/db410c/android_board_support_package_vla.br_.1.2.7-01010-8x16.0-4.zip
- https://developer.qualcomm.com/hardware/dragonboard-410c/tools
- https://developer.qualcomm.com/download/db410c/android-bootloader-v2.zip

After running the script, the user will have all of the open source and a number of proprietary binary files that can be used to build a complete Android image for the DragonBoard 410c.

Clock related source code:

`/kernel/drivers/clk/clk.c`

`/kernel/drivers/clk/qcom/clock-gcc-8916.c`

A References

A.1 Acronyms and terms

Acronym or term	Definition
ACPU	Application CPU
AHB	Advanced High-performance Bus
API	Application Program Interface
APSS	Applications Processor Subsystem
ASIC	Application Specific Integrated Circuit
BMIC	Bus Interface Memory Controller
DDR	Double Data Rate RAM
DRAM	Dynamic Random Access Memory
EBI1	External Bus Interface 1 (DRAM)
LPAAUDIO	Low Power Audio
LPAIF	Low Power Audio Interface
MDSS	Multimedia Display Subsystem
MMSS	Multimedia Subsystem
NoC	Network On a Chip
OCMEM	On Chip Memory
PBL	Primary Boot Loader
PC NOC	Peripheral Configuration - Network On a Chip
PLL	Phase Locked Loop
RAM	Random Access Memory
RPM	Resource Power Manager
SBL1	Secondary Boot Loader
SoC	System On a Chip
SR	Supply Regulated
SRPLL	Supply Regulated Phase Locked Loop
ULT AUDIO	Ultra Low Tier Audio
VFE	Video Front End
WCNSS	Wireless Communication Subsystem
XO	Crystal oscillator

EXHIBIT 1

PLEASE READ THIS LICENSE AGREEMENT ("AGREEMENT") CAREFULLY. THIS AGREEMENT IS A BINDING LEGAL AGREEMENT ENTERED INTO BY AND BETWEEN YOU (OR IF YOU ARE ENTERING INTO THIS AGREEMENT ON BEHALF OF AN ENTITY, THEN THE ENTITY THAT YOU REPRESENT) AND QUALCOMM TECHNOLOGIES, INC. ("QTI" "WE" "OUR" OR "US"). THIS IS THE AGREEMENT THAT APPLIES TO YOUR USE OF THE DESIGNATED AND/OR ATTACHED DOCUMENTATION AND ANY UPDATES OR IMPROVEMENTS THEREOF (COLLECTIVELY, "MATERIALS"). BY USING OR COMPLETING THE INSTALLATION OF THE MATERIALS, YOU ARE ACCEPTING THIS AGREEMENT AND YOU AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO THESE TERMS, QTI IS UNWILLING TO AND DOES NOT LICENSE THE MATERIALS TO YOU. IF YOU DO NOT AGREE TO THESE TERMS YOU MUST DISCONTINUE AND YOU MAY NOT USE THE MATERIALS OR RETAIN ANY COPIES OF THE MATERIALS. ANY USE OR POSSESSION OF THE MATERIALS BY YOU IS SUBJECT TO THE TERMS AND CONDITIONS SET FORTH IN THIS AGREEMENT.

1.1 **License.** Subject to the terms and conditions of this Agreement, including, without limitation, the restrictions, conditions, limitations and exclusions set forth in this Agreement, Qualcomm Technologies, Inc. ("QTI") hereby grants to you a nonexclusive, limited license under QTI's copyrights to use the attached Materials; and to reproduce and redistribute a reasonable number of copies of the Materials. You may not use Qualcomm Technologies or its affiliates or subsidiaries name, logo or trademarks; and copyright, trademark, patent and any other notices that appear on the Materials may not be removed or obscured. QTI shall be free to use suggestions, feedback or other information received from You, without obligation of any kind to You. QTI may immediately terminate this Agreement upon your breach. Upon termination of this Agreement, Sections 1.2-4 shall survive.

1.2 **Indemnification.** You agree to indemnify and hold harmless QTI and its officers, directors, employees and successors and assigns against any and all third party claims, demands, causes of action, losses, liabilities, damages, costs and expenses, incurred by QTI (including but not limited to costs of defense, investigation and reasonable attorney's fees) arising out of, resulting from or related to: (i) any breach of this Agreement by You; and (ii) your acts, omissions, products and services. If requested by QTI, You agree to defend QTI in connection with any third party claims, demands, or causes of action resulting from, arising out of or in connection with any of the foregoing.

1.3 **Ownership.** QTI (or its licensors) shall retain title and all ownership rights in and to the Materials and all copies thereof, and nothing herein shall be deemed to grant any right to You under any of QTI's or its affiliates' patents. You shall not subject the Materials to any third party license terms (e.g., open source license terms). You shall not use the Materials for the purpose of identifying or providing evidence to support any potential patent infringement claim against QTI, its affiliates, or any of QTI's or QTI's affiliates' suppliers and/or direct or indirect customers. QTI hereby reserves all rights not expressly granted herein.

1.4 **WARRANTY DISCLAIMER.** YOU EXPRESSLY ACKNOWLEDGE AND AGREE THAT THE USE OF THE MATERIALS IS AT YOUR SOLE RISK. THE MATERIALS AND TECHNICAL SUPPORT, IF ANY, ARE PROVIDED "AS IS" AND WITHOUT WARRANTY OF ANY KIND, WHETHER EXPRESS OR IMPLIED. QTI ITS LICENSORS AND AFFILIATES MAKE NO WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THE MATERIALS OR ANY OTHER INFORMATION OR DOCUMENTATION PROVIDED UNDER THIS AGREEMENT, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR AGAINST INFRINGEMENT, OR ANY EXPRESS OR IMPLIED WARRANTY ARISING OUT OF TRADE USAGE OR OUT OF A COURSE OF DEALING OR COURSE OF PERFORMANCE. NOTHING CONTAINED IN THIS AGREEMENT SHALL BE CONSTRUED AS (I) A WARRANTY OR REPRESENTATION BY QTI, ITS LICENSORS OR AFFILIATES AS TO THE VALIDITY OR SCOPE OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT OR (II) A WARRANTY OR REPRESENTATION BY QTI THAT ANY MANUFACTURE OR USE WILL BE FREE FROM INFRINGEMENT OF PATENTS, COPYRIGHTS OR OTHER INTELLECTUAL PROPERTY RIGHTS OF OTHERS, AND IT SHALL BE THE SOLE RESPONSIBILITY OF YOU TO MAKE SUCH DETERMINATION AS IS NECESSARY WITH RESPECT TO THE ACQUISITION OF LICENSES UNDER PATENTS AND OTHER INTELLECTUAL PROPERTY OF THIRD PARTIES.

1.5 **LIMITATION OF LIABILITY.** IN NO EVENT SHALL QTI, QTI'S AFFILIATES OR ITS LICENSORS BE LIABLE TO YOU FOR ANY INCIDENTAL, CONSEQUENTIAL OR SPECIAL DAMAGES, INCLUDING BUT NOT LIMITED TO ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL DAMAGES, ARISING OUT OF THE USE OR INABILITY TO USE, OR THE DELIVERY OR FAILURE TO DELIVER, ANY OF THE MATERIALS, OR ANY BREACH OF ANY OBLIGATION UNDER THIS AGREEMENT, EVEN IF QTI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THE FOREGOING LIMITATION OF LIABILITY SHALL REMAIN IN FULL FORCE AND EFFECT REGARDLESS OF WHETHER YOUR REMEDIES HEREUNDER ARE DETERMINED TO HAVE FAILED OF THEIR ESSENTIAL PURPOSE. THE ENTIRE LIABILITY OF QTI, QTI's AFFILIATES AND ITS LICENSORS, AND THE SOLE AND EXCLUSIVE REMEDY OF YOU, FOR ANY CLAIM OR CAUSE OF ACTION ARISING HEREUNDER (WHETHER IN CONTRACT, TORT, OR OTHERWISE) SHALL NOT EXCEED US$10.

2. **COMPLIANCE WITH LAWS; APPLICABLE LAW.** You agree to comply with all applicable local, international and national laws and regulations and with U.S. Export Administration Regulations, as they apply to the subject matter of this Agreement. This Agreement is governed by the laws of the State of California, excluding California's choice of law rules.

3. **CONTRACTING PARTIES.** If the Materials are downloaded on any computer owned by a corporation or other legal entity, then this Agreement is formed by and between QTI and such entity. The individual accepting the terms of this Agreement represents and warrants to QTI that they have the authority to bind such entity to the terms and conditions of this Agreement.

4. **MISCELLANEOUS PROVISIONS.** This Agreement, together with all exhibits attached hereto, which are incorporated herein by this reference, constitutes the entire agreement between QTI and You and supersedes all prior negotiations, representations and agreements between the parties with respect to the subject matter hereof. No addition or modification of this Agreement shall be effective unless made in writing and signed by the respective representatives of QTI and You. The restrictions, limitations, exclusions and conditions set forth in this Agreement shall apply even if QTI or any of its affiliates becomes aware of or fails to act in a manner to address any violation or failure to comply therewith. You hereby acknowledge and agree that the restrictions, limitations, conditions and exclusions imposed in this Agreement on the rights granted in this Agreement are not a derogation of the benefits of such rights. You further acknowledges that, in the absence of such restrictions, limitations, conditions and exclusions, QTI would not have entered into this Agreement with You. Each party shall be responsible for and shall bear its own expenses in connection with this Agreement. If any of the provisions of this Agreement are determined to be invalid, illegal, or otherwise unenforceable, the remaining provisions shall remain in full force and effect. This Agreement is entered into solely in the English language, and if for any reason any other language version is prepared by any party, it shall be solely for convenience and the English version shall govern and control all aspects. If You are located in the province of Quebec, Canada, the following applies: The Parties hereby confirm they have requested this Agreement and all related documents be prepared in English.