Application Note

Restricted to customers under NDA

AUDIO CLOCKS FOR GAP9

Hardware Options and System Implications

Rel.1.0.a

April 4th, 2023

DISCLAIMER

This information is subject to change without notice.
Information on this document is provided “as is” without any warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, suitability for a particular purpose, or non-infringement. The information provided in this document is intended for informational purposes only. Information may be changed or updated without notice.

Table of contents

1. INTRODUCTION

This document describes possible options regarding clock sources to use in an Audio-oriented GAP9-based application, with some recommendations as well as indications of system-level impacts – esp. (asynchronous or synchronous) sample rate conversion needs.

2. PRECAUTIONS FOR GENERATION OF AUDIO-RELATED CLOCKS IN GAP9

2.1. USE FLL-BYPASS MODE FOR AUDIO CLOCKS

GAP9 generates all its internal clocks from a reference clock (1-50MHz) *'REF_FAST_clock'*, either obtained from its built-in crystal oscillator or provided by an external source through a dedicated pin.
There are 2 ways to generate each of those internal clocks (Fig. 1.x) :
- either using an internal FLL (Frequency Locked Loop) , which can generate any valid frequency from the reference clock – an FLL has the advantage of consuming less power than a fully-fledged PLL, but suffer from greater jitter.
- or, by a simple integer division of the reference clock, which avoids the jitter introduced by the FLL – but of course is less flexible in terms of reference clock / produced clock ratio.
Here, we are particularly interested in SAI clock options, as explained below.

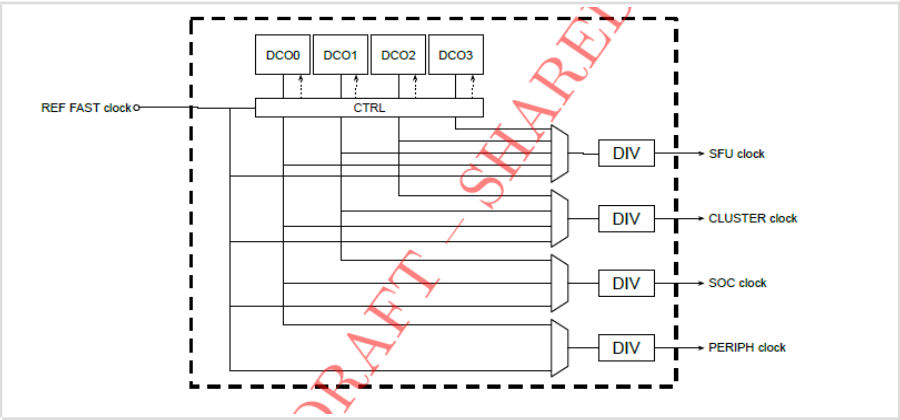
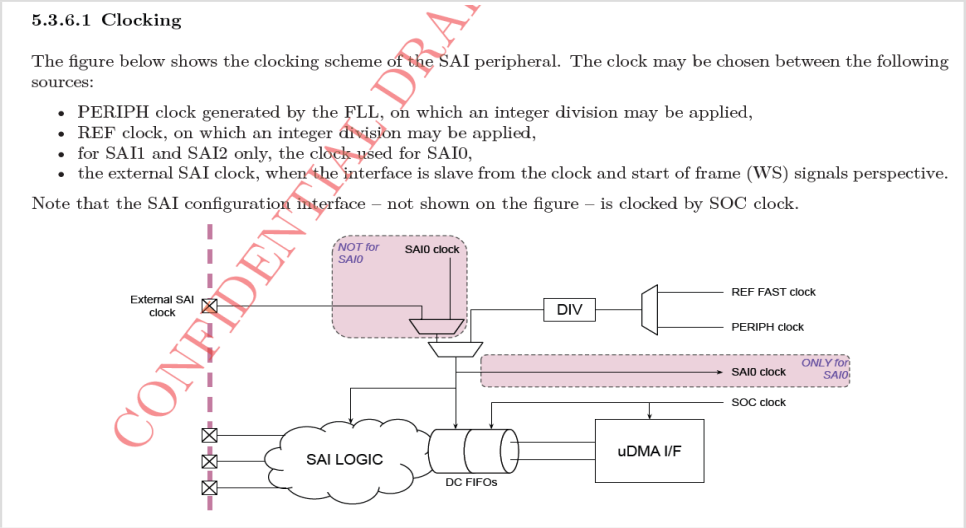


Figure 1.a – Generation of internal clocks in GAP9 (datasheet excerpt, 4.2.4.1)



5.3.6.1 Clocking

The figure below shows the clocking scheme of the SAI peripheral. The clock may be chosen between the following sources:

- PERIPH clock generated by the FLL, on which an integer division may be applied,
- REF clock, on which an integer division may be applied,
- for SAI1 and SAI2 only, the clock used for SAI0,
- the external SAI clock, when the interface is slave from the clock and start of frame (WS) signals perspective.

Note that the SAI configuration interface – not shown on the figure – is clocked by SOC clock.

Figure 1.b – Generation of SAI clocks in GAP9 (datasheet excerpt, 5.3.6.1)

In use cases where GAP9 connects to **microphones, A/D or D/A conversion chips (incl. Class-D amplifiers)**, to achieve good audio performance, **it is essential that these devices are fed with sufficiently clean (i.e., low-jitter) clocks.** The clock provided to these devices is the PCM bit clock (in I2S/TDM mode) or the PDM clock. This clock is provided by GAP9 on pin SAIx_SCK of its SAI interface (and the target devices use it either directly or through a PLL) . *[Note- in some (more rare) occurences, these devices may take in a separate 'master clock' – in that case, the below recommendations apply to the source of that clock]*

> Therefore : obtaining satisfactory performance from audio devices clocked by GAP9 requires that GAP9 is configured to **generate SAI clocks in FLL bypass mode.** This way, these clocks get generated by simple integer division of REF_FAST_clock and do not get polluted by FLL jitter. Of course, this requires REF_FAST_clock, the reference clock, to be clean itself – see below.

2.2. SOURCE OF REF_FAST_CLOCK

There are 2 options to provide REF_FAST_clock :

> using the on-chip clock oscillator (1-50MHz)
In that case a low-jitter crystal with suitable load capacitor must be connected between pins FAST_XTAL_A and FAST_XTAL_B of GAP9. Supported frequencies range from 1MHz to 50MHz. However, some frequencies are more 'audio-friendly', see below. 24.576 MHz or 12.288 MHz are often relevant choices. Do also pay attention to the relationship between reference clock frequency and maximum bit clock frequency (see below).

> using an externally provided reference clock

It is also possible to inject an external reference clock, up to 50MHz, through pin USART0_CLK/FAST_REF_CLK (ball K1 of GAP9). This signal must have 0-1.8V signaling levels. This clock must itself be very stable, since any jitter present at that point will directly impact generated audio clocks.

When an external reference clock is provided, the on-chip oscillator can be disabled (thus saving a little power) and no crystal is required on its pins.

An external clock source would typically be :

- either the radio (usually Bluetooth) SOC that is present in wireless (earbud/headphones) applications and connects to GAP9

- or, a specialized ultra-low power oscillator IC, such as SiTime's 8021 series, which can offer reduced footprint and lower power consumption compared to using a crsytal + on-chip oscillator (start-up time is longer than that of a crystal oscillator, which may or may not be relevant).

2.3. REFERENCE CLOCK FREQUENCY

> TYPICAL VALUES

Audio systems are typically designed to use specific audio sampling frequencies.
One common choice is 16 KHz or 48 KHz, and multiples (esp. power of 2) thereof: 96KHz, 192 Khz, 384 Khz, 768 Khz. Another choice is 44.1KHz and multiples thereof.

Therefore, to minimize the need for internal conversions and because the number of bits one wants to fit in an audio frame is typically a power of 2, it is convenient to **select a reference clock that is itself a power of 2 multiple of 48KHz** (or, possibly, 44.1 Khz if the system considered uses mor readily that base). As a result , **24.576 MHz or 12.288 MHz are interesting frequencies** to use for the reference clock. However, usage of other frequencies is also envisageable, with some precautions, see below.

> REFERENCE CLOCK FREQUENCY vs. MAX # OF CHANNELS, SAMPLING FREQUENCY, BITS PER CHANNEL

The frequency of the reference clock *REF_FAST_clock* bounds the maximum achievable # of channels, # of bits per channel and audio sampling frequency, as follows :

There must be enough bit clock cycles in on frame to accomodate all channels with their number of bits per channels. In addition, fractional slot configuration is not supported by the SAI interfaces (that is, no dummy bits are allowed within an audio frame).

This translates as the following requirement :

Num_channels x Num_bits_per_chann x FS = Bit_clk_fqcy = Ref_clk_fqcy/K , *with K an integer*

where :

Ref_clk_fqcy is the reference clock frequency (from on-chip quartz oscillator or external source)

Bit_clk_fqcy is the bit clock frequency *[an integer sub-multiple of the reference clock frequency – since FLL is not used here]*

Num_channels is the number of channels per audio frame (16 TDM channels maximum)

FS is the audio sampling frequency (i.e. audio frame rate)

For instance :

At 24.576 MHz reference clock frequency and maximum achievable bit clock , it is possible to fit up to 4 channels at 32 bits per channel and 192KHz audio sampling rate (4x32x192=24576).

Now, if the reference clock frequency and therefore maximum achievable bit clock frequency is , *e.g.*, down to 12.288 MHz, then the former 4 channels at 192KHz/32bits don't fit ; while, for instance, 12 channels at 64KHz/16 bits would fit (12x64x16=12288).

3. CHOICE OF AUDIO REFERENCE CLOCKS VS. HARDWARE OPTIMIZATIONS AND (A)SRC REQUIREMENTS

We consider here the common situation where the system contains GAP9 and a Bluetooth radio SOC, in an architecture along the lines of Fig.2: the BT SOC sends and receives compressed audio over a Bluetooth radio link and sends/receives uncompressed (PCM) audio to/from GAP9 using I2S or TDM protocol over one of GAP9's Serial Audio Interfaces (SAI). GAP9 also connects to a number of digital microphones and outputs analog audio to the speaker(s) of the headphone(s) or earbud(s) through an external D/A converter + amplifier chip.

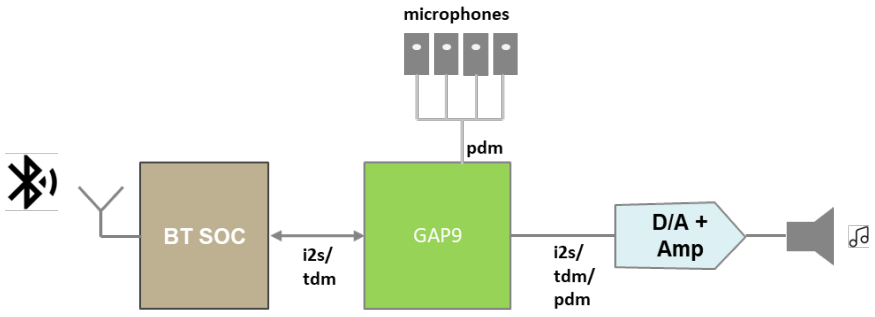


Figure 2 – Outline of Typical Earbud/Headphones Architecture with GAP9 and BT-SOC

- GAP9 may be Master on the SAI (I2S/TDM) interface, in which cases it supplies the bit clock and frame sync signal to the BT-SOC (and other slaves if any). This implies GAP9 imposes the audio sample rate (= frequency of frame sync signal) on the SAI interface.

- GAP9 may also be Slave on the I2S/TDM interface, in which it receives the bit clock and frame sync signal from the BT-SOC (along with other slaves if any). This implies the BT SOC imposes the audio sample rate (= frequency of frame sync signal) on the SAI interface.

3.1. OPTION 1 – INDEPENDENT CLOCKS FOR BT-SOC AND GAP9

Here we consider the case where the BT SOC and GAP9 run off their own independent reference clocks – typically, each with their own crystal.

Refer to Fig. 3.

Because GAP9 and BT-SOC reference clocks are asynchronous, an ASRC (asynchronous sample rate converter) is required on either side of the SAI interface, depending on which is side is master and which is slave.

- If GAP9 is master, it provides *fsync* and therefore defines the audio sample rate on the interface The BT SOC must adapt to this sample rate, taking into account uncertainties on GAP9's and its own reference clocks (tolerances w.r.t nominal frequency). To do so, the BT-SOC must implement an ASRC converter.

- If BT SOC is master, the situation is reversed. GAP9 now receives *fsync* from BT SOC and must adapt to this sample rate, again taking into uncertainties on reference clocks. To do so, GAP9 must implement an ASRC converter.

Example :

Assume BT SOC sinks audio from BT radio link at 44.1KHz nominal and pushes it as master on the SAI interface. Assume GAP9 resamples it to 48KHz nominal.

Assume BT SOC uses a crystal at 32MHz nominal (32MHz+$\varepsilon 1$ in reality) and GAP9 uses a crsytal at 24.576MHz nominal (24.576MHz+$\varepsilon 2$ in reality) .

Then :

- On BT SOC side, 44.1KHz = *Ratio1* x (32MHz+$\varepsilon 1$) with *Ratio1*=441/320

- On GAP9 side, 48KHz = *Ratio2x* (24.576MHz+$\varepsilon 2$) with *Ratio2*=48/24576=512

The conversion ratio is : *Ratio2*/*Ratio1* x (24.576MHz+ $\varepsilon 2$)/(32MHz+$\varepsilon 1$) , with $\varepsilon 1$ and $\varepsilon 2$ unknown, hence the need for ASRC.

If the 44.1 to 48KHz conversion were done on BT SOC side (master), GAP9 would still need to handle the fact that what the BT SOC considers to be 48KHz is not what GAP9 considers to be 48KHz (since their reference clocks have different uncertainties), i.e. handle an asynchronism. So better do a full ASRC on GAP9 (slave) side.

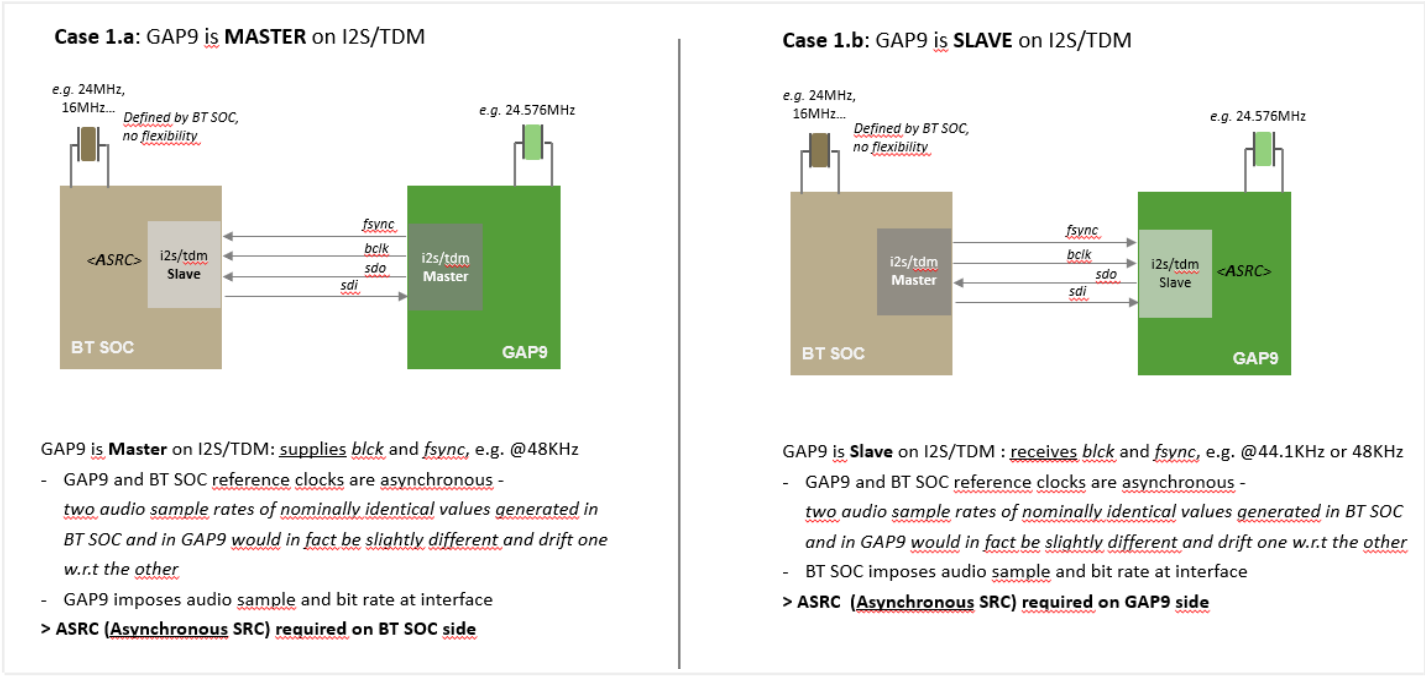


Figure 3 – Independent clocks on BT SOC side and GAP9 side

3.2. OPTION 2 – LOCKED REFERENCE CLOCKS FOR BT-SOC AND GAP9, WITH 'AUDIO-FRIENDLY' VALUE FOR GAP9

Here we consider the case where the BT SOC, running off a crystal at fixed frequency (typically, 12MHz/24MHz/48MHz or 13MHZ/26MHz or 16MHz/32MHz, depending on the BT-SOC model considered) integrates a low-jitter PLL and can output towards GAP9 a clean clock at an 'audio-friendly' frequency (power-of-2 multiple of typical sample rate – for instance, 24.576MHz = 512x48MHz).

Refer to Fig. 4.

Because GAP9 and BT-SOC reference clocks ultimately derive from the same source (the BT SOC crystal), they are locked and an SRC (synchronous sample rate converter) is sufficient, no need to handle any asynchronism.

The SRC can be done on any side of the SAI interface (or could even be split). The 'best' choice depends on application specific factors, like which side is able to handle SRC more efficiently or has little else to do, or, is it better from a system complexity point of view to settle for a single sample rate on the BT-SOC<>GAP9 interface across all use cases, etc.

Example :

Assume BT SOC may sink audio from BT radio link either at 44.1KHz or 48KHz. Assume GAP9 wants to work with 384KHz internal sample rate. Assume BT SOC is master on SAI interface.

One possible SRC scenario could then be : BT SOC does 44.1KHZ<>48KHz SRC if needed (when audio on BT radio link is 44.1KHz) to always communicate at 48KHz on interface. GAP9 always expects 48KHz on interface and does constant 48KHz<>384KHz conversion on its side.

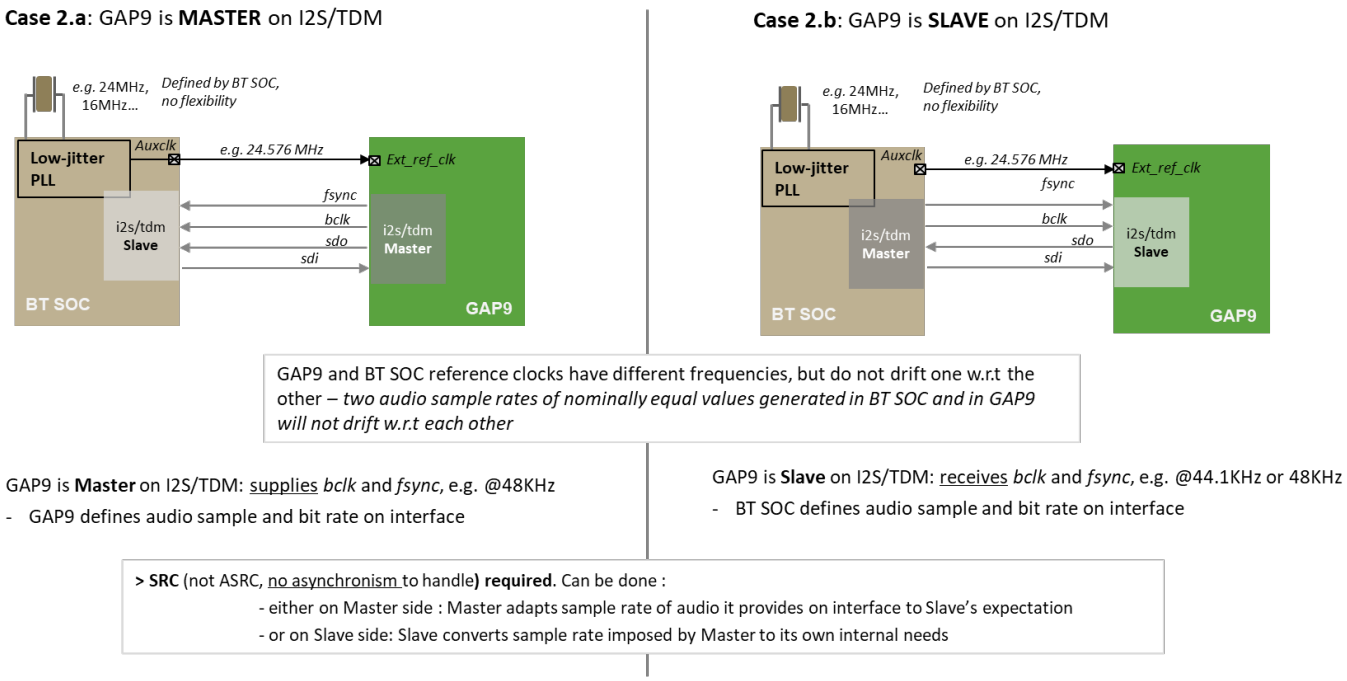


Figure 4 – Locked reference clocks, with 'audio-friendly' clock on GAP9 side

3.3. OPTION 3 – LOCKED REFERENCE CLOCKS FOR BT-SOC AND GAP9, WITH INTEGER RATIO

Here we consider the cases where GAP9 takes the same reference clock as the BT-SOC, or where the BT-SOC supplies GAP9 with an integer sub-multiple of its own reference clock. In some situations, this integer ratio case that may be worth considering too, depending on application requirements and constraints, as detailed below.

Refer to Fig. 5.

Like in previous case, because GAP9 and BT-SOC reference clocks ultimately derive from the same source (the BT SOC crystal), they are locked and an SRC (synchronous sample rate converter) is sufficient, no need to handle any asynchronism.

> Internally, with proper sample rate conversion, GAP9 is able to handle any audio sample rate to internal master clock ratio (provided the ratio is a rationale number). The cost of SRC, in terms of power consumption and – to some extent – in terms of resulting signal quality, will depend on what this ratio is. Integer ratios are preferable. For instance, with a 24MHz internal master cock, it is easy to handle 16KHz, 32KHz, 48KHz, 96 Khz or 192KHz.

However, one point to take into account if trying to use classic sample rates with non 'audio-friendly' master clock is the fact that **external devices connected through I2S/TDM to GAP9 may themselves impose some restrictions on the bit clock to frame rate ratio** – and therefore (since we're not using GAP9 FLL for bit clock generation) restrictions on the reference clock to audio sampling rate ratio. A typical case would be a DAC or ADC requiring this ratio to be one value in a set of valid values it can accept, most often a set of power-of-2 values (64, 128, 256…) and possibly power-of-2 multiples of 1.5 (96, 192, 768) , sometimes also a few other integer values.. The datasheet of such devices in the target application considered should be consulted.

> An alternative solution is to depart a little from 'classic' audio sample rates and work with the closest values that have a convenient relationship with the internal master clock.

- For example, with a 24MHz reference clock, instead of using : 48KHz (=24.576MHz/512), …., 768KHz (=24.576MHz/32) , use : 46.875KHz (=24MHZ/512), …, 750KHz (=24MHZ/32).

Similarly, typical PDM clocks would not be 1.536MHz/3.027MHz/6.144MHz but 1.489MHz/2.9782MHz/*5*.9564MHz.

The alternatives values are sufficiently close to the 'classic' ones to obtain almost identical audio quality.

- For 26MHz reference clock, one might decide to use 50.78125KHz (=26MHz/512) instead of 48KHz, …, 812.5KHz (=26MHz/32) instead of 768KHz. And as PDM clocks : 1.625MHz or 3.25MHz or 6.5MHz

- For 32MHz reference clock, it is more difficult to find 'classic' ratios that lead to audio sampling rates close to 'classic' values. For instance, instead of 48KHz, one could consider using 41.6667KHz (=32MHz/768) – but this is 13 % lower power-of-2 the classic 48KHz value, which starts to be significant.

Note : in any case, the ratio of audio sampling rates at input and output of SRC must be a rationale number.

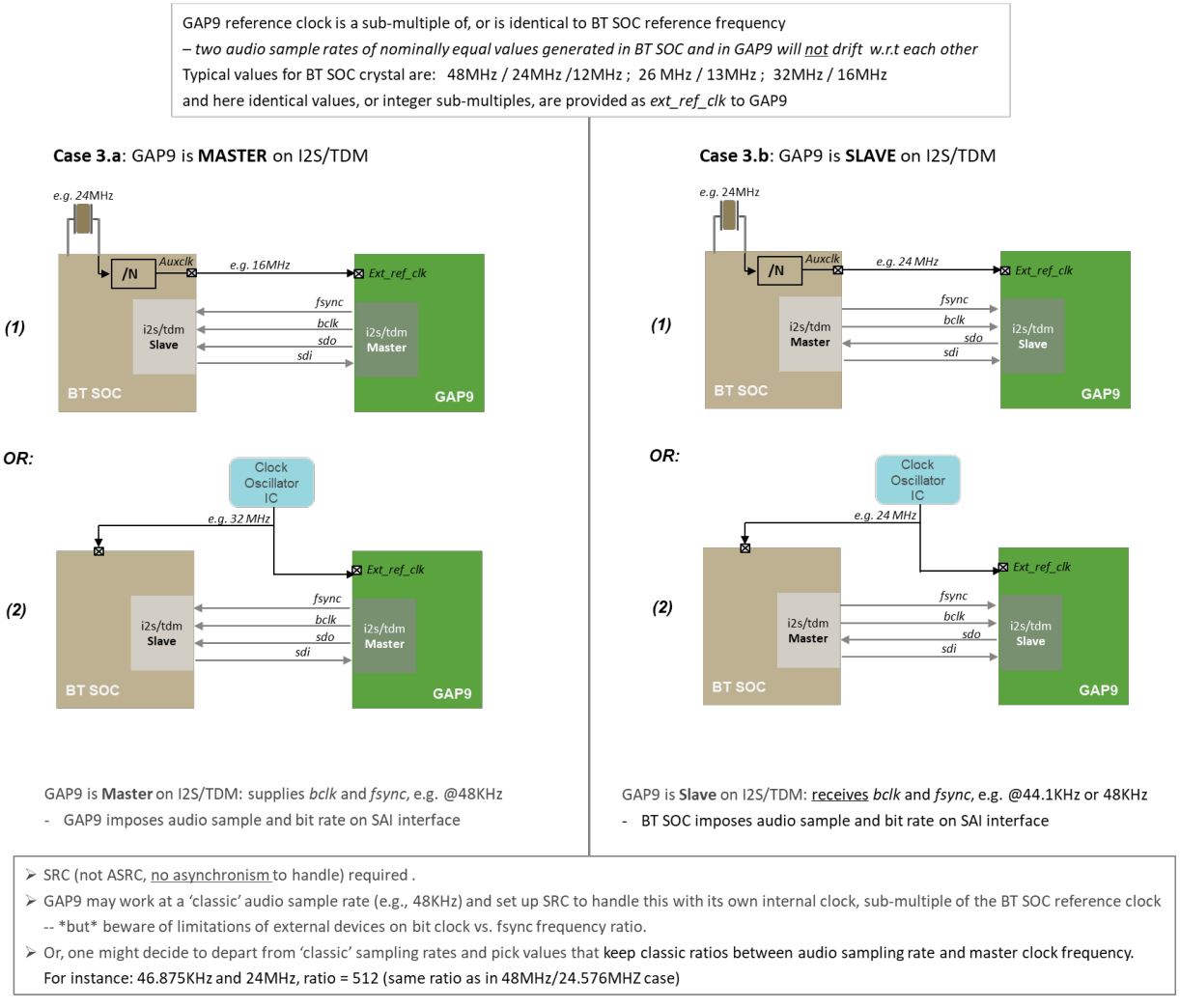


Figure 4 – Locked reference clocks, integer ratio for BT SOC and GAP9

4. GAP9 AS PURE COPROCESSOR

Now we are considering the type of architecture outlined in Fig. 5. GAP9 is used as a co-processor, where it receives audio data from a host (BT SOC or other), processes it and returns results to the host.

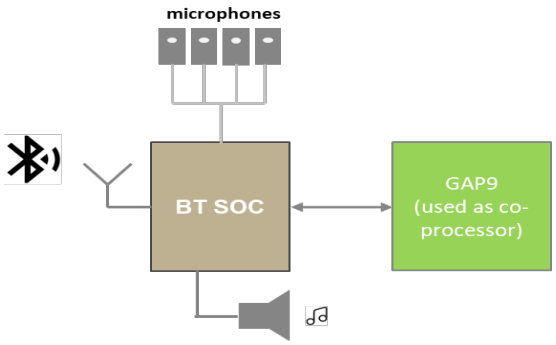


Figure 5 – GAP9 as coprocessor – Outline architecture

In this case there are no constraints on respective reference clocks. All that is required is that GAP9 processes the data quick enough to sustain the data rate required so the BT-SOC software meets its own real-time targets. The interface may be SAI (TDM/I2S), it may also be e.g. SPI or UART. The host may for example push buffers of data to process to GAP9 and pull buffer of processed data from GAP9.

DOCUMENT HISTORY

Drafts A-B *– March, 2023 (Xavier Cauchy)*
Initial drafts

Rel.1.0 *– 4.Apr, 2023 (XC)*
Initial release