Application Note

# GAP9
# HARDWARE INTEGRATION GUIDE

## for WLCSP package variant

Rel.1.7
7. June, 2024

---

**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 Application Note is intended to help hardware designers integrate GreenWaves GAP9 ultra-low power Application Processor chip for edge AI into their board design.
It is relevant for the initial WLCSP-100 (Wafer Level Chip Scale Package) version of GAP9. GAP9 may also come in the form of bare die (KGD, Known Good Die) or in a different package such as BGA with more I/Os, appropriate instructions for those variants will be provided in due time.

*Note:* Marks noted *« [Ref.X] »* in this text point to references provided at the end of the document.

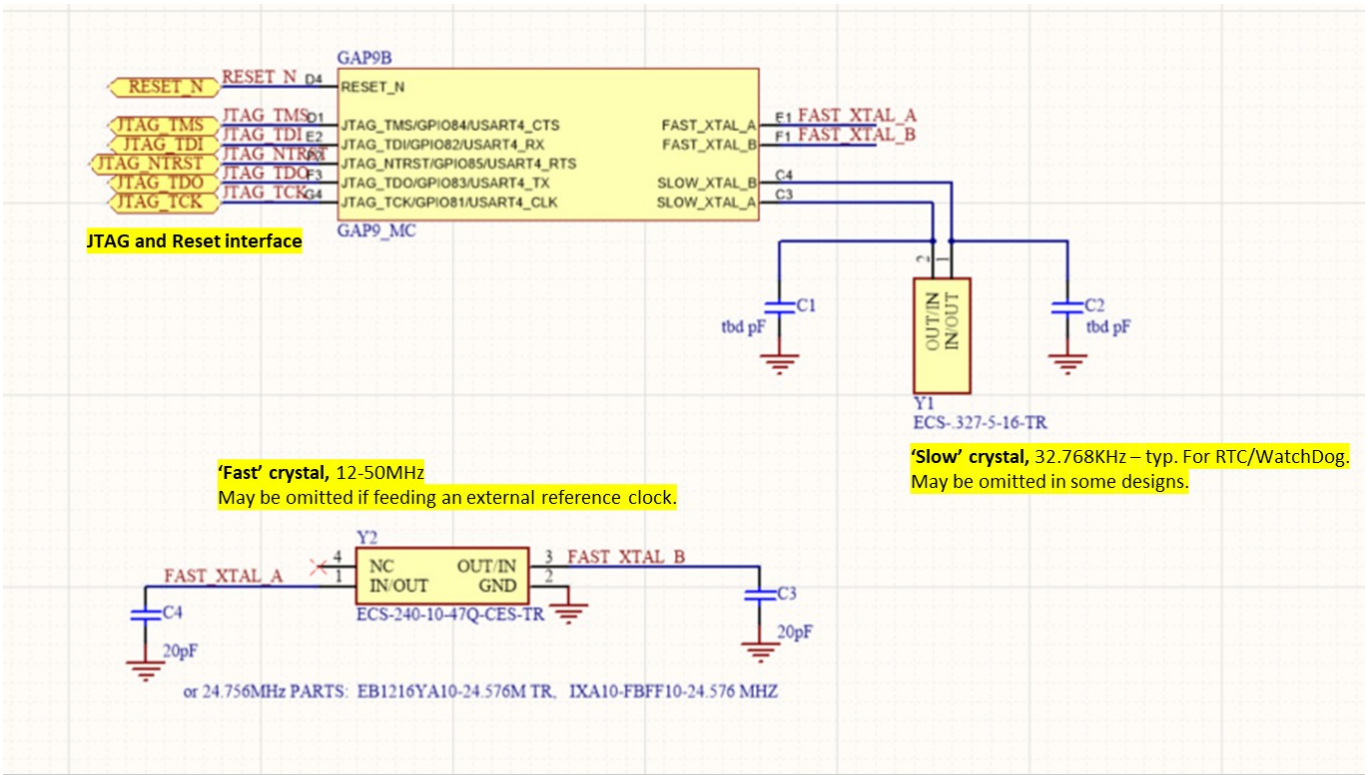
## 2. REFERENCE CORE DESIGN

Figures 1.a to 1.f provide an example PCB design around GAP9, with recommended connections to external memory, to crystal, to passives for internal DC-DC converter and to power supply sources.

This is a generic example that needs to be tuned to your specific system. In particular, a number of optimizations may be possible, depending on your own application requirements.

The next sections will offer advice and recommendations on specific aspects of GAP9 hardware integration.

*Fig. 1.a - JTAG , Crystals, Input Power Supplies*



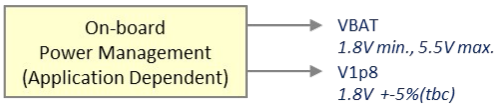
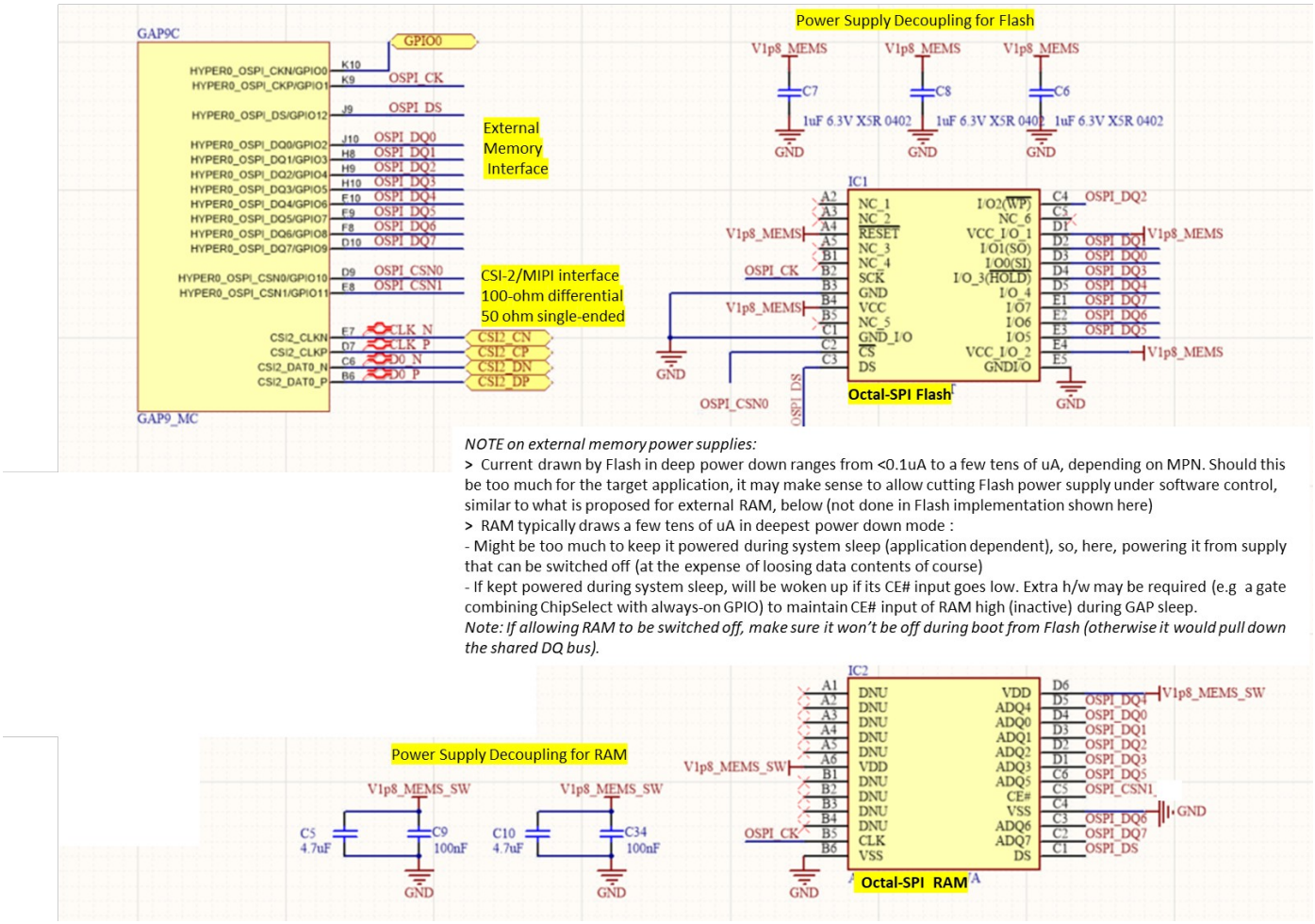



*Fig. 1.b - Memories & CSI-2*




*NOTE on external memory power supplies:*

> Current drawn by Flash in deep power down ranges from <0.1uA to a few tens of uA, depending on MPN. Should this be too much for the target application, it may make sense to allow cutting Flash power supply under software control, similar to what is proposed for external RAM, below (not done in Flash implementation shown here)

> RAM typically draws a few tens of uA in deepest power down mode :

- Might be too much to keep it powered during system sleep (application dependent), so, here, powering it from supply that can be switched off (at the expense of loosing data contents of course)
- If kept powered during system sleep, will be woken up if its CE# input goes low. Extra h/w may be required (e.g a gate combining ChipSelect with always-on GPIO) to maintain CE# input of RAM high (inactive) during GAP sleep.

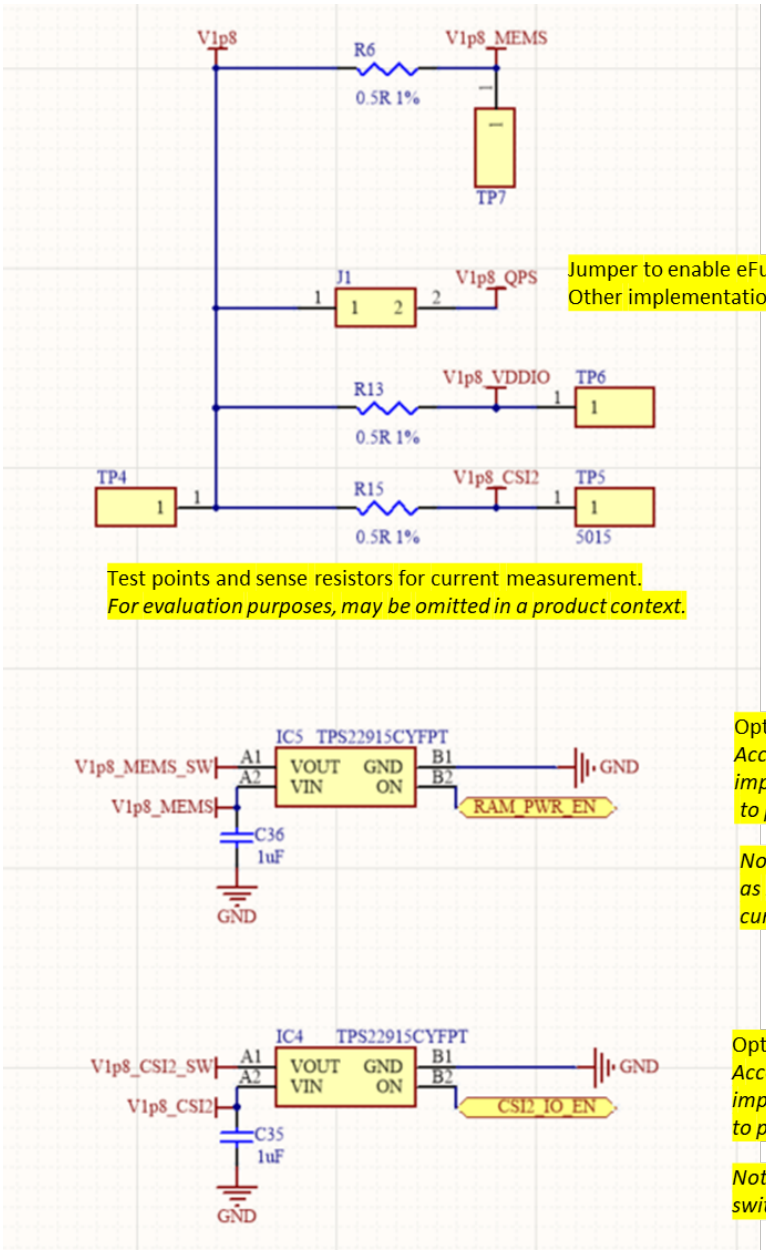
*Note: If allowing RAM to be switched off, make sure it won't be off during boot from Flash (otherwise it would pull down the shared DQ bus).*

***Please note :***

***(1) usage of external memories is optional, interest depends on complexity of the target algorithms and application. Refer to Section 7 for details.***

***(2)*** *even though the /RESET pin of the Flash is tied to the power supply in the above, an alternative solution is to tie it to the system hardware reset (same signal as that going to NRESET pin of GAP9). This ensures the Flash goes back to its default configuration (typ. single SPI) in case of system reset and can simplify things in some corner cases.*

*Fig. 1.c - Memories & CSI-2 Power Supplies*




Jumper to enable eFuse programming power supply.
Other implementations possible.

Test points and sense resistors for current measurement.
*For evaluation purposes, may be omitted in a product context.*

Optional switch on RAM power supply.
*According to applications needs,*
*implement or not and connect RAM_PWR_EN*
*to pull-up, pull-down, GPIO or external agent*

*Note – Flash power supply not made switchable here as Flash typically offer deep power down mode with current <100nA. Decision is design dependent.*

Optional switch on CSI-2 interface power supply.
*According to applications needs,*
*implement or not and connect CSI2_IO_EN*
*to pull-up, pull-down, GPIO or external agent*

*Note - CSI2 core power supply (V0p65_CSI2) not made switchable here as power consumption of CSI-2 core is ultra-low.*

*Fig. 1.d - GAP9 Power Supplies*

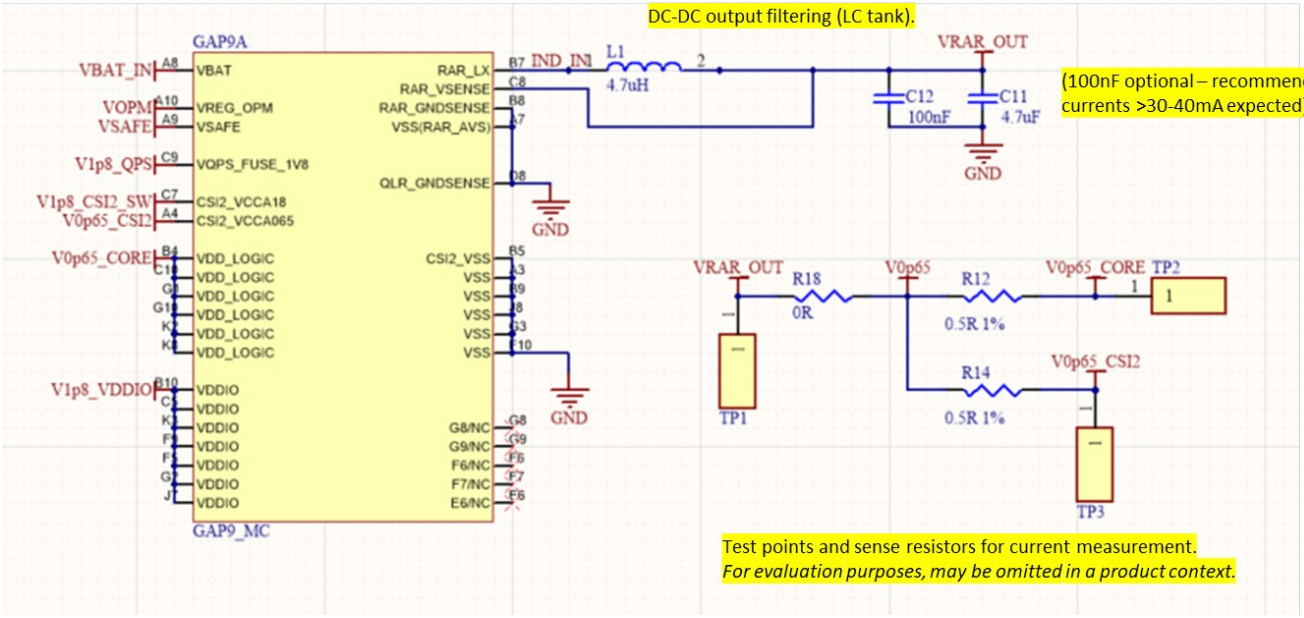



*Fig. 1.e - GAP9 Power Supply Decoupling*

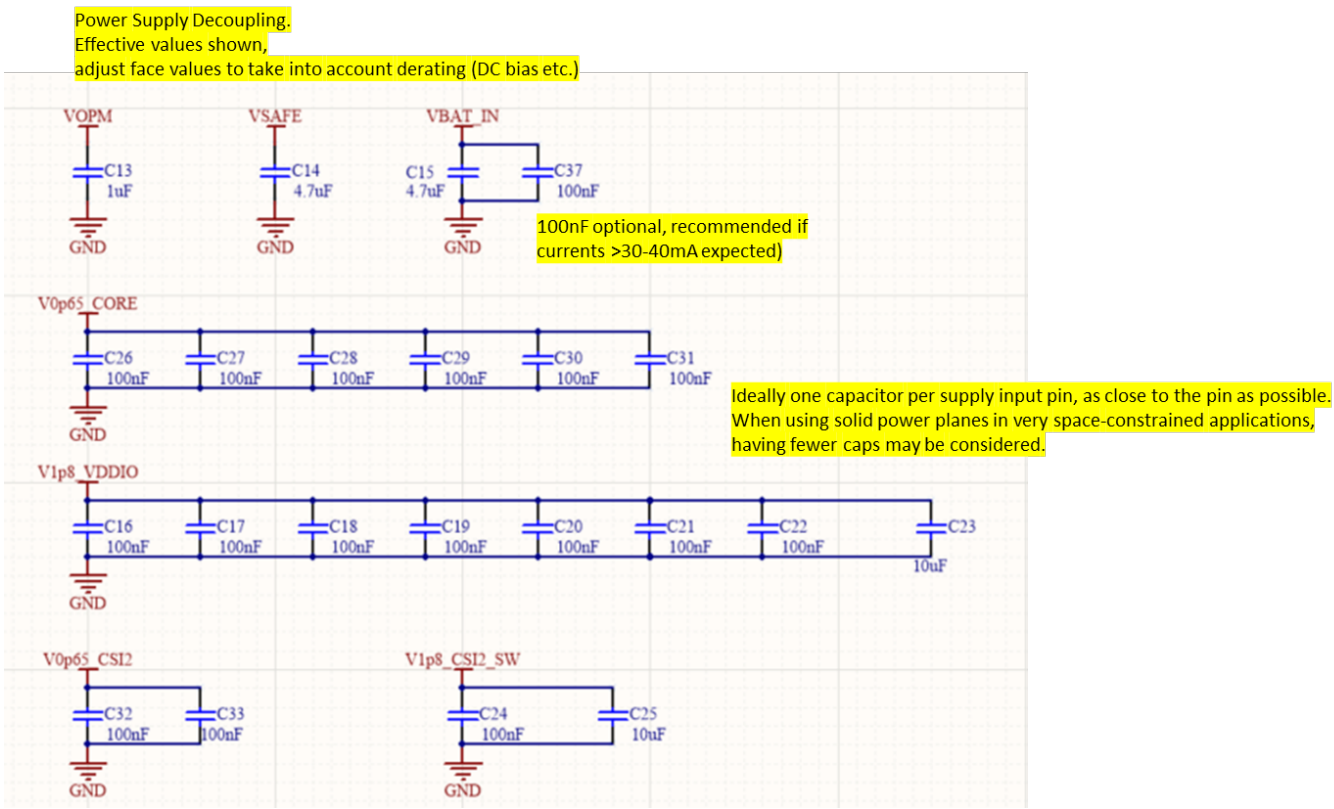
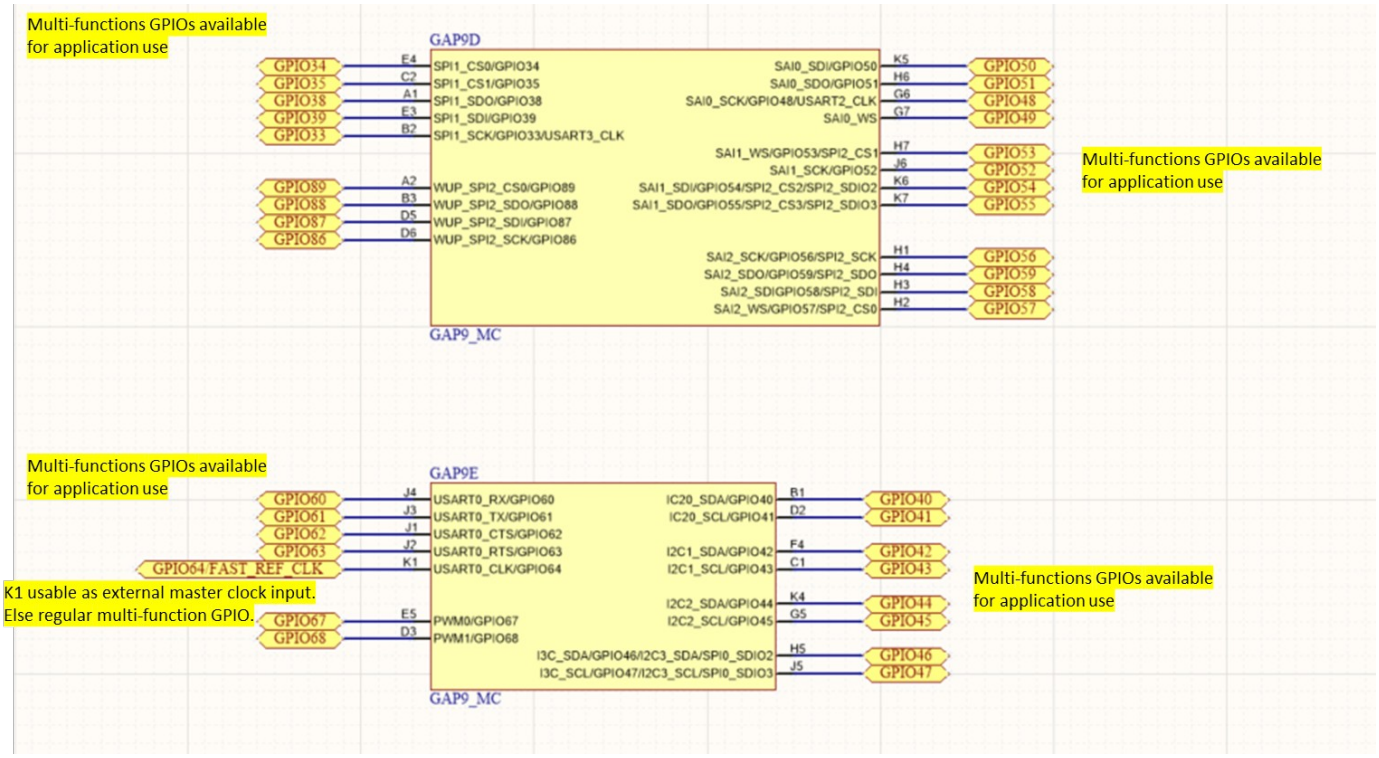


*Fig. 1.f - GAP9 Multi-function GPIOs*




## 3. POWER SUPPLIES

### > Core power supply

GAP9 embeds a **DC-DC converter** that takes an **input voltage between 1.8V and 5.5V**, to be provided on pin **VBAT (ball A8)**.

GAP9's embedded DC-DC converts this voltage to a voltage between 0.65V and 0.8V (software controlled value) to be used by all internal logic and embedded memories of GAP9.

The VBAT input voltage would typically come from the system's battery, either directly or through some voltage conversion stage.
The internal DC-DC converter generates a switching voltage on pin **RAR_LX (B7)**. As illustrated in Figure 1 and GAP9's datasheet *[Ref.1]*, this signal must be passed through an LC tank to obtain a stable voltage (*VDD_LOGIC*) for GAP9's internal logic. This output voltage can be set between 0.65V and 0.8V under software control. This output voltage must be fed back to the DC-DC through pin **RAR_VSENSE (C8)** and also used as GAP9 core power supply, injected though pins **VDD_LOGIC (B4, C10, G1, G10, K2, K8)**.

The switching signal on RAR_LX is a potential noise source (agressor) to surrounding signals ; it is therefore recommended to take adequate precautions. Please refer to the specific section on DC-DC requirements further down this document.

### > I/O ring -

GAP9's I/Os are all powered from 1.8V to be supplied through pins **VDDIO (B10, C5, F5, F9, G2, J7, K3).**
There are 2 type of I/Os : 'always-on' and 'swicthed', see next section.

### > eFuse Power Supply -

An additional power supply, **VQPS_FUSE_1V8 (C9)**, is dedicated to programming the embedded eFuse cells, which are one-time programmable bits (default value = Logic-0, can be set to Logic-1 by 'blowing' the eFuse cell). eFuse bits control different settings and behaviors of GAP9 (see datasheet for details). Except in some particular cases, it is mandatory to be able to program the eFuses on the production line and therefore have some means on the board to provide VQPS=**1.8V+-5%**. Current draw during programming is **12 mA** typ. and **22mA** worst case.
Power-up and -down of VQPS must meet certain timing constraints, as illustrated below (where VDD_EFUSE is an internal supply picked from the core logic supply provided on pins VDD_LOGIC).

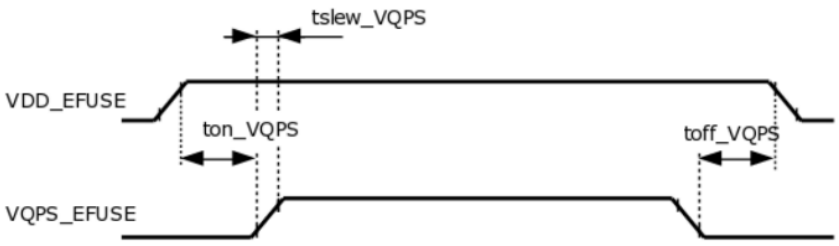



Figure 4: Power sequencing

| Item | Description | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| ton_VQPS | On-time of VDD_EFUSE before VQPS_EFUSE | 10 | | - | us |
| toff_VQPS | Off-time of VQPS_EFUSE before VDD_EFUSE | 10 | | - | us |
| tslew_VQPS | Rising time of VQPS | 50 | | - | us |
| NOTES: 1. time values are given between 90% and 10% levels of rising slopes, or 10% to 90% levels of falling slopes, for all powers and signals | | | | | |

Because VDD_EFUSE, derived from VDD_LOGIC, is gated internally according to the state of the SOC domain, fulfilling the above requirement also means VQPS_EFUSE must not be active before or during boot or reset, nor while GAP9 is in sleep mode, nor while VBAT is switched off. Refer to GAP9 datasheet section 5.2.14 for details.

Practically, **it is recommended to leave pin VQPS_FUSE_1V8 either open (disconnected) or tied to GND at all times except when fuses need to be programmed.** To program eFuses, apply 1.8V to VQPS_FUSE_1V8 once GAP9 has booted only, run the code to program eFuses, and once done remove 1.8V from VQPS_FUSE_1V8.
In terms of implementation, one solution is to provision pads or test points on the board that can connect to a test/programming jig (typically through pogo pins) on the production line. On a development or prototyping board, it may be convenient to provision a jumper between VQPS_FUSE_1V8 and some suitable on-board 1.8V supply. Alternative solutions are possible, provided the timing constraints are met.

*Practical note :* one way to control the slew rate of VQPS_EFUSE_1V8 is to derive it from a suitable 1.8V supply through a low-resistance RC filter. For instance : R=1ohm, C=22uF would ensure a sufficiently slow transition while limiting the worst case voltage drop across the resistor to only 20mV or so.

Lastly, if VQPS_EFUSE is derived from the same root supply as VBAT input of GAP9, then the overall design should ensure the inrush current into VQPS_EFUSE may not cause a droop on pin VBAT below 1.8V-5%, to keep within the specified operating conditions of GAP9's internal power management unit.

### > Bypass capacitors -

All power supply pins should be properly bypassed to GND through low ESR (Equivalent Series Resistance) caps, placed as close as possible to the pins they decouple.
Ceramic capacitors with X5R or X7R dielectric type are recommended. Keep in mind this type of capacitor exhibits dependency to DC bias, meaning the actual capacitance matches nominal capacitance at 0V bias but (especially in small form factors such as 0402) can drop rather sharply as the DC bias applied to the cap increases.

### > Power Supply Sequencing -

Chapter '*Boot Process*' in the GAP9 Datasheet describes the differents steps involved in this process, including start up of internal regulators and power-on reset.
On the board, VDDIO is allowed to rise before *VBAT*, simultaneously with *VBAT* or after *VBAT*; however in all cases **VDDIO must have stabilized when internal reset is relased**. When VDDIO rises before VBAT, the internal POR (Power-On Reset) ensures this, but if on the board VDDIO rises later than VBAT, then it is recommended to maintain GAP9 hardware reset input low until VDDIO has fully stabilized. One way to implement this on the PCB is to pull the hardware reset line to VDDIO (1.8V nom.) through, *e.g.*, a 10K resistor, and to **provision a 100nF capacitor between the hardware reset signal and GND**.

## 4. GAP9 INPUT & OUTPUT PINS (I/O) – INCL. BOOT PINS

*Also refer to GAP9 I/O Datasheet.*

GAP9 uses 2 types of digital I/Os :

### > « always-on » I/Os :
These have very low leakage and **stay on when GAP9 is put into (retentive or deep) sleep mode**, but can only offer moderate switching speed. Because these I/Os stay powered, they can drive a user-configured value during sleep and some can be used as wake-up source (see GAP9 I/O datasheet for a complete list).

### > « switched » I/Os :
These can reach higher switching speed but have higher leakage ; **in general** those I/Os are **switched off when GAP9 is in sleep mode**. There is however a 'light sleep' mode that can be entered and exited with GAP9 'switched' I/Os retaining their configuration.

GAP9's I/O datasheet specifies which I/Os are 'always-on' and which are 'switched' as well as their behavior in different power modes.

**Specificity of switched I/Os :**
Because a switched I/O is not powered during sleep mode, **any attempt to externally pull it up or drive it to Logic-1 during sleep will trigger protection diodes and parasitic structures**, creating a flow of current from the pin to the switched off internal VDDIO line and resulting in unwanted current paths in the switched off I/O and unability to reliably maintain the Logic-1 level at pin level.
**In addition**, because of their specific behavior at power-up, switched I/Os **must not be externally pulled or driven to Logic-1 during power-up.**

This has some implications if GAP9 needs to be put in sleep mode with external devices – typically memories – also in power-down or sleep mode. Consider a case where an external memory is put into some power-down or sleep mode by an appropriate OSPI command. As soon as GAP9 then enters sleep mode, the (active low) ChipSelect from GAP9 is no longer maintained to Logic-1, which may wake up the memory (depending on wake-up mechanism implemented by the external memory). One solution for such cases is the insertion of a logic gate controlled by an always-on GPIO.

### > GPIO default and alternate functions :

All digital I/Os offer a number of alternative functionalities, labeled Alternate0, Alternate1, etc. in the I/O datasheet.
In addition :
- drive strength is selectable (from 1 or 2 to 12mA),
- some of those I/Os can be used as wake-up source to exit GAP9 from sleep mode,
- most feature optional integrated weak pull-up and pull-down resistors

**At power-up and hardware reset, all those I/Os start up in Alternate 1 mode (=GPIO, General Purpose Input/Output), configured as input, without internal pull.**
The only exceptions are NRESET and JTAG pins.

### > GPIO remapping :

On a number of I/Os, an additional level of mutiplexing inside the chip allows to replace the regular Alt.0 functionality by yet another signal. This additional mux has access to a selection of 36 peripheral interface signals. Some of them can only be accessed that way. A number of precautions must be taken to successfully configure remappable pins. Refer to GAP9 datasheet, section *« Pin Description »* for details.

### > Using GPIO as wake-up source :

A selected number of 'always-on' GPIOs (not all of them – 12 exactly) can be used as wake-up source : an event on those pins can cause GAP9 to exit retentive or deep sleep. Those specific pins are marked as such in GAP9 datasheet, section *« Pin Description »,* last column ot the Table presented there.

### > BOOT pins :

GAP9 can boot from the following sources : JTAG, embedded MRAM, HyperFlash on external memory interface, SPI Flash on external memory interface (single SPI to Octal SPI Flash are supported).
**Selection of boot source is done :**
- either through the setting of **specific eFuse bits**
- or through the setting of **2 specific pins that are examined at reset**

Those pins are **BOOT1 = GPIO87** and **BOOT0 = GPIO86**. They are both sampled at exit from reset; if both are Logic0 then boot source is decided by eFuse bits, else those pins determine boot source. It is nevertheless possible to disable consideration of BOOT pins by burning an appropriate eFuse.
Decoding of BOOT pins is as follows :

BOOT[1:0] = 00 (binary) → eFuse-defined boot source
01 → boot from JTAG
10 → boot from HyperFlash
11 → boot from eMRAM

*Note : booting from (Octal-)SPI Flash is performed either through secondary boot after a primary boot from eMRAM, or through eFuse defined direct boot.*

## 5. CRYSTAL OSCILLATORS

GAP9 features two crystal oscillators :
- a 'slow' oscillator intended for a 32.768KHZ crystal.
- a 'fast' oscillator intended for a 12 to 50MHz crystal

### > 'Slow' 32.768KHz oscillator

Usage of this oscillator is **optional**. It would typically be implemented when on-chip RTC or WatchDog functionality is required.
*Note:* use of the watchdog is also possible without that oscillator (*i.e.*, using the 'fast' oscillator described below instead), with the limitation of a shorter maximum refresh period (up to 2**24 x T*refclk*, refclk being a sub-multiple of the 'fast' clock -- *refclk can be* as slow as *fast_clk/128*, but beware that *refclk* speed impacts different peripherals – refer to the clocking scheme described in GAP9 datasheet)

The slow oscillator is tuned to operate with a 32.768KHz crystal connected across pins **SLOW_XTAL_XA (C3)** and **SLOW_XTAL_XB (C4)**. This oscillator cannot be bypassed, that is, no external 32.768KHz clock source can be passed through the XTAL pins.

It is advised to select a crystal with low ESR and low load capacitance, as detailed below. The crystal and its capacitors should be placed close to GAP9, to limit parasitics as much as possible. Ceramic capacitors of C0G dielectric type are a suitable choice. Start-up time of the 32.768KHz oscillator depends on exact crsytal characteristics and selected load capacitors but can be expected to be of the order of several hundreds of ms, up to over 1 second in extreme cases.

The selected crystal should have the following characteristics :

- crystal load capacitance **$C_L$ = 4pF to 7pF** ; ideally no greater than 4pF for best results

- **maximum ESR** (Equivalent Series Resistance) **< 60kOhm, ideally** – moderately higher values possible depending on load capacitance (at the expense of slightly degraded power and frequency accuracy), as follows :

| Maximum equivalent resistance $ESR_{MAX}$ (Ohms) | Load capacitance $C_L$(pF) | | | |
|---|---|---|---|---|
| | 4 | 5 | 6 | 7 |
| 35 k | √ | √ | √ | √ |
| 40 k | √ | √ | √ | √ |
| 50 k | √ | √ | √ | √ |
| 55 k | √ | √ | √ | √ |
| 60 k | √ | √ | √ | √ |
| 65 k | √ | √ | √ | |
| 70 k | √ | √ | √ | |
| 75 k | √ | √ | | |
| 80 k | √ | √ | | |

*NOTE : the effective load capacitance CL is not the capacitance placed on each pin of the crystal ; rather, its definition combines these 2 external capacitors with pin capacitances, internal capacitances and stray capacitance of package and PCB. Refer to Appendix for details on load capacitance calculation and relationship with value of the external capacitor connected to each pin of the crystal.*

GAP9 pre-integrates a 1pF capacitor on each crystal pin. **This internal 1pF on each terminal must be taken into account when calculating $C_L$.** In some cases, together with parasitics, this may be sufficient to properly operate the oscillator at low CL .

Below are some examples of 32.768KHz crystal offering very low ESR and CL, in small footprint SMD packages (suggestions only, to be properly checked by board designer).

**> in 3.2x1.6mm package :**

- Manufacturer: Abracon - MPN: ABS07W-32.768KHz-D/J/K-1/2-T

- Manufacturer: CTS - MPN: TFE32xT32K7680R ('T' ordering code for lowest CL)

**> In 2.0x1.2mm package :**

- Manufacturer Micro Crystal - MPN : CM8V-T1A 32.768 kHz 4pF XX XX XX

- Manufacturer: Abracon - MPN: ABS06W-32.768KHZ-D/J/K-1/2-T

- Manufacturer: Abracon - MPN: ABS06-32.768KHZ-x-(H/W)-(1/4)-(T)

- Manufacturer: CTS - MPN: TFE20xW32K7680R

**> PCB Placement and Layout Recommendation for crystal oscillator :**
It is recommended to implement a guard ring around the crystal + load capacitors and dedicate a ground « island » to this area.

### > 'Fast' 12-50MHz Oscillator and External Clock Options

It is mandatory to supply a 'fast' clock input to GAP9. This can be done :
- either using the **on-chip oscillator**, with a crystal (12MHz to 50MHz) connected across pins **FAST_XTAL_A (E1)** and **FAST_XTAL_B (F1)**,
- or by injecting an **externally generated, full swing (0-1.8V),** 50% duty cycle clock into pin **FAST_REF_CK/USART0_CLK (K1)**

*[ The fast oscillator can also be operated in a bypass mode where a reduced swing 0-0.8V clock would be provided to FAST_XTAL_A; however the pass-through significantly degrades duty cycle and therefore this solution is not recommended ]*

Selection of on-chip oscillator vs. externally generated clock is done though eFuse or through second stage bootloader.

Pay attention to the fact that FAST_REF_CK/USART0_CLK (K1) is a 'switched' I/O, which will be off when GAP9 is in sleep mode (except possibly in 'light sleep' mode). Therefore, the external clock source should not toggle nor drive a Logic1 on that pin while GAP9 is sleeping, else current will flow through protection and parasitic structures of the I/O.

## 6. INTERNAL DC-DC CONVERTER – POWER SUPPLY IMPLEMENTATION

Figure 2 describes the implementation of power supplies in GAP9 and specifies what external components are required. Pay atttention to the fact that indicated component values are effective values, that is, derating should be applied to take into account operating conditions (such as the reduction of effective capacitance of X5R/X7R dielectric capacitors under DC bias, especially when small component sizes are selected). Below are a few additional recommendations.

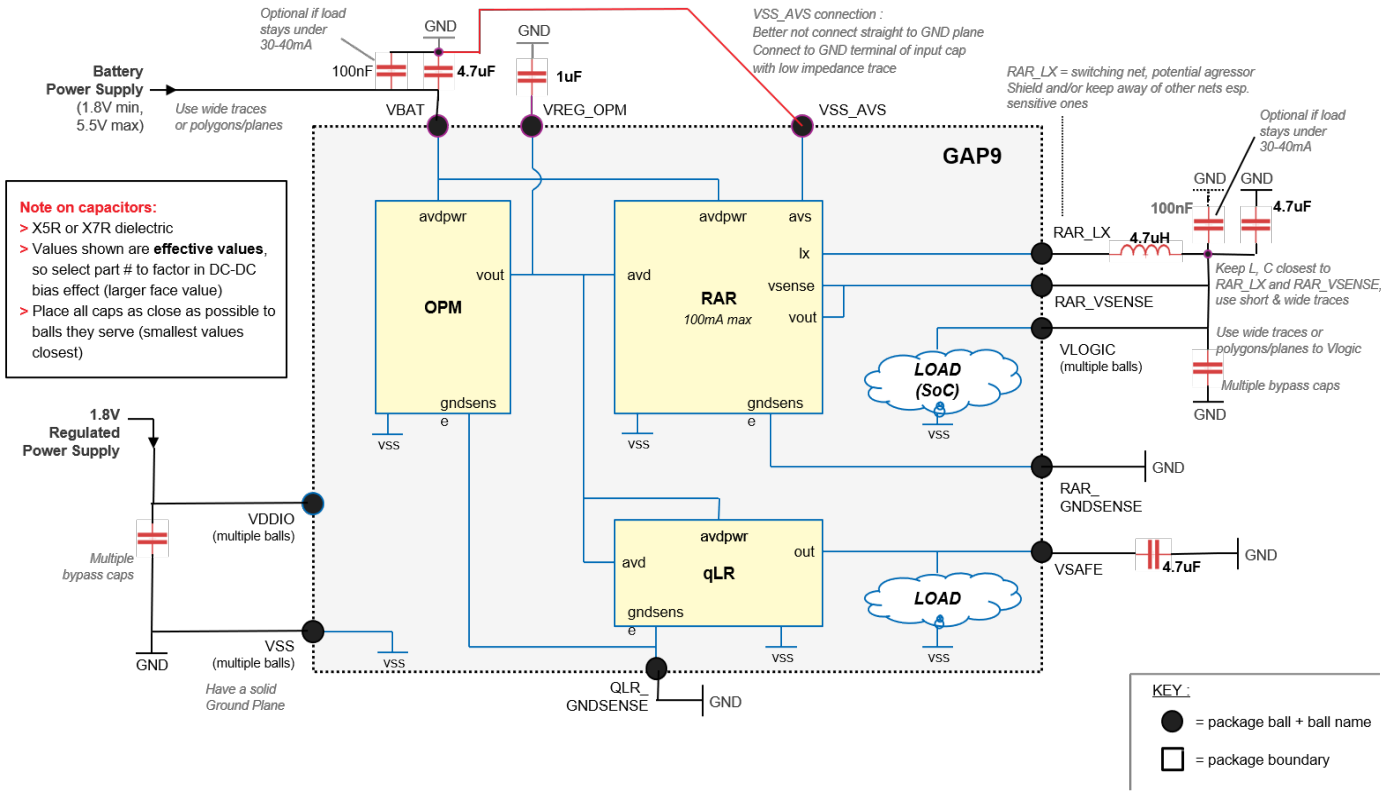



*Fig. 2 – Structure of internal DC-DC converter and implementation of related power supplies*

### > Inductor Selection

A **4.7uH** inductor value is recommended. It is advised to select a shielded, **low DCR** inductor (ideally better than 0.5ohm, but up to 1.5ohm or so remains reasonable) – lower equivalent DC resistance (DCR) typically comes at the expense of larger overall dimensions so this should be weighted against application needs and constraints. Low DCR helps achieve good conversion efficiency ; shielding limits the amount of noise radiated by the inductor and therefore risks to pollute neighboring signals.

The selected inductor should exhibit a saturation current **$I_{sat}$ large enough to sustain the peak (not average) current from the DC-DC** (as measured before LC filtering, so including large ripple). If current

through the inductor is beyond Isat, then the inductor does not behave as an inductor anymore and the result is unpredictable.
The peak current through the DC-DC can be obtained from the following formula :

***Ipeak = Idc+ 0.5xDelta_IL***

where :

- I*dc* is the maximum average load current drawn by the application – for GAP9 this can conservatively be taken as 60mA
- *Delta_IL* is the ripple current, obtained from the following formula :

***Delta_IL = [ 1 / (F.L) ] x [ Vlogic x (1 – (Vlogic/Vbat)) ]***

where :

. *F* is the switching frequency of the interal DC-DC oscilalltor
– for GAP9 : 0.7MHz min, 1MHz typ., 1.3MHz max
. *Vlogic* is the output regulated voltage : 0.65V to 0.8V, user programmable (default: 0.8V)
. *Vbat* is the input power supply voltage : defined by system design, within the 1.8V to 5.5V range

To calculate *Delta_Il* and therefore *Ipeak* and in turn, minimum required *Isat* that will be suitable for the application under any operating condition, we must use the worst case values in the above formula, taking into account component tolerances etc.

*Note* : Inductor manufacturers define *Isat* as the current for which effective inductance drops by a certain ratio w.r.t nominal value – however they don't all use the same threshold, for instance some will define this ratio as -50 % and others as -30 %.
If the manufacturer provides a curve showing effective inductance vs. current, then make sure I*peak* is well within the flat region of the curve, where inductance does not depart from nominal value by more than a few %.
Else, to be absolutely sure I*sat* is sufficient, we can consider the following (conservative) criterion :
*Isat @30 %* must be at least 3x higher than calculated *Ipeak*

**EXAMPLE :**
The table below calculates I*peak* values for a number of example operating conditions. All parameters are taken at their worst case and it is assumed GAP9 will be maximally loaded, a case for which we consider core consumption may top 60mA *[this is conservative and be tuned after full silicon characterization]* and up to 0.8V logic core voltage will be required (to reach maximum operating frequency).

**Ipeak = Idc + 0.5 x Iripple**

with : **Idc** = max. DC current drawn from converter

**Iripple** = ripple current = **[1/(F.L)] x [ Vlogic(1-Vlogic/Vbat) ]**

*where :* F = switching frequency of GAP9's internal DC-DC oscillator : 0.7MHz min., 1MHz typ. 1.3MHz max

V*logic* = regulated voltage fed to GAP9's logic – 0.65V to 0.8V, user programmable (default : 0.8V)

V*bat* = input power supply range (typ. from battery but could also come from on-board intermediate voltage regulator), Defined by sytsem design, must be within 1.8V-5 .5V range

| | | |
|---|---|---|
| Min. F (worst case for Isat calculation) | 0,7 | MHz |
| Min. L (worst case) = 4.7uH -20 % | 3,76 | uH |

| *Example use case* | V*logic* (V) | V*bat* (V) | **Iripple (mA)** | **Idc (mA)** | **Ipeak (mA)** | Min. Isat (mA) With 3x margin |
|---|---|---|---|---|---|---|
| *USB power, max. processor activity* | 0,8 | 5 | **255** | 60 | **188** | 563 |
| *Fully loaded Li-Ion/LiPo battery* | 0,8 | 4,2 | **246** | 60 | **183** | 549 |
| *Battery 3.3V max voltage* | 0,8 | 3,3 | **230** | 60 | **175** | 525 |
| *On-board DC-DC converter, 2.5V* | 0,8 | 2,5 | **207** | 60 | **163** | 490 |
| *On-board DC-DC converter, 1.8V* | 0,8 | 1,8 | **169** | 60 | **144** | 433 |

*Table 1 : Ipeak and recommended Isat for selected use case, worst case conditions*

For the larger input voltages (*Vbat*), larger *Ipeak* and therefore *Isat* requirements (which translates as more bulky inductor) are required.
Example of inductors suitable for use with **up to 5V input power supply and maximum GAP9 current draw** are :
- in **1080** (imp.) package size : **Murata DFE252007F-4R7M**
- in **0806** package size : **Murata  DFE201610E-4R7M**

For severely space constrained applications, it is possible to go down to smaller package size with some trade-offs. For instance :
- in **0603** package size, a relevant part is : **Taiyo Yuden MBKK1608H4R7M**
DC resistance is only slightly above 0.5 ohm (0.73 ohm max) ; I*sat*@30 % is 410mA which is not quite 3x I*peak* if considering the above use cases, however all I*peak* values envisaged in the table are well within the flat region of the effective inductance vs. current curve (see part datasheet).

### > Capacitor Selection

Recommended value for both input capacitor Cin (on Vbat) and output capacitor on RAR_LX is 4.7uF.
**X5R or X7R** dielectric capacitors should be used for their low ESR characteristics. However, such capacitors display a significant dependence to DC bias (the indicated nominal value is typically for 0V DC bias and effective capacitance drops when voltage increases) – especially when they come in small form factor e.g. 0402 or 0201.

**The 4.7uF target for Cin and Cout is the value that should me matched *under the expected DC Bias*.**
At DC-DC input, worst case DC bias (max input voltage) may be 1.8V to 5.5V, depending on system implementation choices; at output, worst case bias is 0.8V (max volttage delivered by DC-DC converter). Please check the behavior of the selected capacitor under these conditions and select a nominal (0V bias) capacitance value greater than 4.7uF if needed.

### > PCB Placement and Layout Recommendation for internal DC-DC converter :

Besides the usual good PCB design practices :

- Pin  **RAR_LX (B7)** is the unfiltered switching output of the internal DC-DC converter and the associated net will therefore be a **noise source** – so it is important to keep it away from sensitive signals (and perhaps shield it using some grounded guard ring). Make the trace between pin RAR_LX and the inductor wide and as short as possible.

- All ground pins should connect to the PCB ground plane with traces as wide as possible.
Pin **VSS(RAR_AVS) (A7)** is a **noisy ground** that should not connect straight to the ground plane. Instead, connect it to the Ground pin of the input capacitor (Cin on VBAT) through a (wide) low-impedance trace – and then to the global ground plane.

-  All passives related to the DC-DC converter should be placed as close as possible to the pins they serve.

## 7. [ OPTIONAL ] EXTERNAL MEMORIES

GAP9 embeds a 1.5 MByte L2 volatile memory (SRAM) and 2 Mbytes of non-volatile memory (eMRAM, embedded Magneto-Resistive RAM – which, although based on a different technology, may be viewed as an equivalent of embedded Flash). These internal memories can be complemented with external Flash and/or external RAM.

For small applications, usage of the embeded MRAM to store code and constants may be sufficient and no external Flash is required. For richer applications, external Flash is required. The required capacity is application dependent (typically expected in the 16Mbit to 128Mbit range).

Usage of an external RAM is optional and directly depends on the requirements of the applicative software, as does sizing of the RAM. Some algorithms might be fairly lightweight in that respect and live with the 1.5MB internal RAM, others will be more demanding and will require extra RAM.

The external memory interface of GAP9 can be configured as :
- **HyperBus** interface, to support HyperFlash and HyperRAM devices,
- **Quad-SPI** or **Octal-SPI** interfaces (as well as single- and dual-SPI if useful), to support SPI Flash (typ. Serial NOR) and SPI RAM (typ. PSRAM) including those compliant with the **Xccela** standard
- **SDIO** interface is also supported. However, given there is a single memory interface on GAP9 in its WLCSP-100 package variant, usage of the SDIO interface may be problematic if the same interface is also used by external Flash or RAM at high speed.

Maximum clock speed on this interface is 185MHz. DDR (dual data rate) data transfer is supported. This leads to a peak data rate of 370MB/s.

To obtain best throughput performance, it is strongly advised to select RAM models that support linear burst with page boundary crossing capability (option ‘RBX’ at APM).

### - MEMORY VENDORS -

*Note : the information below is indicative ; please do your own research and check with memory manufacturers.*

**> HyperBus** uses 8-bit wide data, with DDR data transfer.
Manufacturers of HyperBus devices include :
- Cypress (RAM, Flash as well as RAM+Flash Multi-Chip module)
- ISSI (RAM, Flash)
- Winbond (HyperRAM only).

Those devices tend to be automotive-qualified and come into BGA-24 6x8 package (not suitable for all applications).

**> Octal-SPI** (Xccela standard, *www.xcella.org*) also uses 8-bit wide data with DDR data transfer capability, but the protocol is slightly different from HyperBus.
Manufacturers of Octal-SPI/Xccela devices include :
- Octal-SPI Flash (32Mbit to 512+Mb): Macronix, ISSI, GigaDevice, Renesas (formerly Adesto then Dialog)
- Octal-SPI RAM (32Mb to 512Mb) : APMemory, ISSI

Those devices come in standardized BGA-24 6x8 package as well as in much smaller packages, down to WLCSP for some of them.
*** BEWARE *** Some Octal-SPI RAM (from ISSI at least) handle Command/Address bits in a peculiar way, with Reserved bits between Lower Column Address bits and Upper Column Address bits. This is not supported by GAP9.

**> Quad-SPI** use 4-bit wide data, DDR or SDR (both supported by GAP9) – can be an interesting choice when bandwidth requirements are not too high, to save on cost
There is a large choice of Quad-SPI Flash vendors ; including the above and others leaders in NOR Flash. Quad-SPI RAM can be obtained from e.g. APMemory.

***Note* : Specific Precautions when Interfacing with Quad-SPI Memories :**
Some QPI memories (typically, those housed in an 8-pin package) multiplex a HOLD#/RESET# signal on pin SDIO3. If there is a need to boot directly from this type of memory, it is necessary to connect SDIO3 to a resistor (e.g. 50K-100K) weakly pulling up to VDDIO – unless weak pull-up is already provided internally to the Flash (vendor dependent, check datasheet). This is because, at power-up, the memory will start in single SPI mode and regard SDIO3 as HOLD#/RESET#, but SDIO3 being not driven by GAP9 at power-up, it may be taken as Logic0 and therefore asserted, blocking acces to the Flash.

### - PCB Placement and Layout Recommendations for external Memories -

Clock on the memory interface can be up to 185MHz with double data rate (DDR) -- so bit rate up to 370Mbps . Although not extremely high, these are respectable speeds and careful layout of the clock and data lines is required to preserve signal integrity. This includes keeping trace lengths relatively short and balanced, avoiding as far as possible – or at least limiting – the number of vias on those traces, etc.
Cypress, the main provider of HyperBus memories, has produced an Application Note AN 211622 *[Ref.2]* *« HyperFlash and HyperRAM Layout Guide »* that provides multiple guidelines. Although they are somewhat on the conservative side, especially with regard to signal length balancing, they are a good starting point. It is recommended to follow them as much as possible, weighting them against practical constraints. This can also be a source of inspiration when routing an Octal-SPI bus.
Current surges from Flash and RAM memories may be significant, so proper decoupling of power supply pins (with sufficient bypass capacitors very close to the power supply pins) is essential. Please refer to memory supplier's recommendations.

## 8. CSI-2 / MIPI INTERFACE

GAP9 provides a CSI-2 MIPI (D-PHY) interface, able to support reception of CSI-2 data at up to 1.5Gb/s. Only receive functionality is supported (GAP9 can not send CSI-2 data to a receiver). The GAP9 design supports dual-lane MIPI, however the WLCSP-100 package variant of GAP only supports **single lane MIPI** (as the second data lane is not exposed, to limit ball count and be able to fit within this small package).

GAP9's CSI-2 PHY handles all aspects of the CSI-2 protocol.

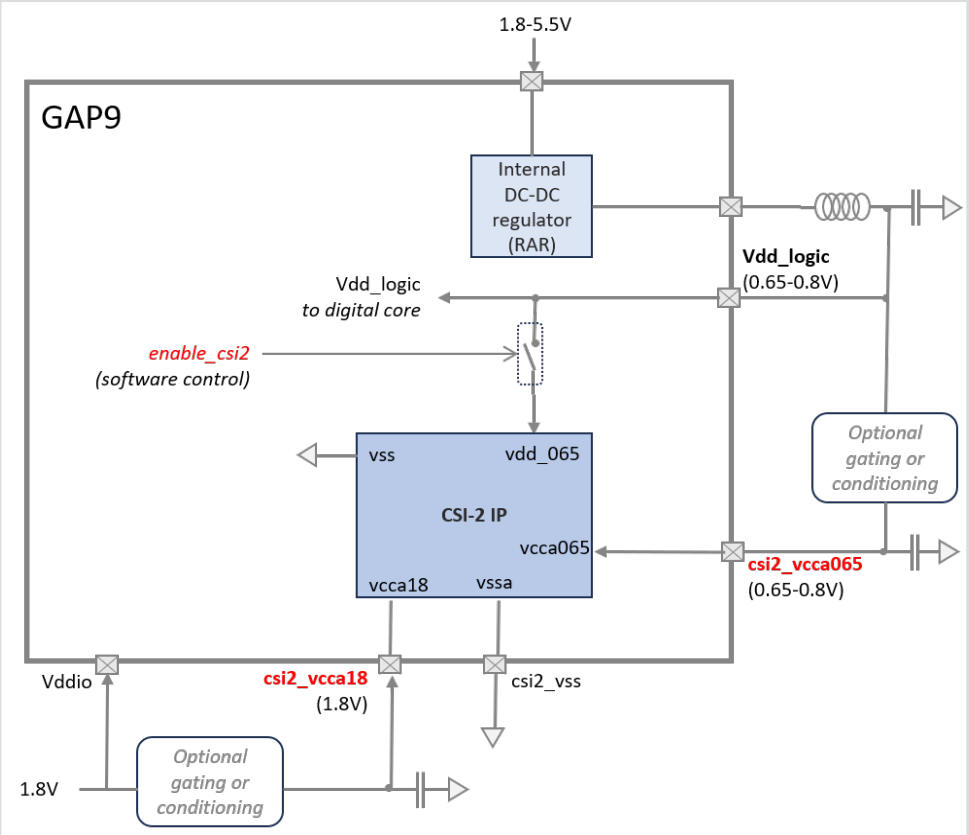
CSI-2 signal pins are the following :
- CSI2_DAT0_P and CSI2_DAT0_N (balls B6 and C6) – differential MIPI (single) data lane
- CSI2_CLKN and CSI2_CLKP (balls E7 and D7) – differential clock lane

### > CSI-2 Power Supplies

Power supply to the CSI-2 PHY of GAP9 is provided through dedicated balls (refer to GAP9 I/O datasheet) :
- **CSI2_VCCA065** on ball A4 powers the digital core of the CSI-2 PHY and may range from 0.65V to 0.8V. It must be same voltage level as VDD_LOGIC which powers the digital core of GAP9. Typically, the former can simply be derived from the latter – as is the case in the reference core design provided in section 2.

- **CSI2_VCCA1.8** on pin C7 powers the I/O part of the CSI-2 PHY and should be 1.8V+-10 %
- **CSI2_VSS** on pin B5 is a dedicated Ground pin for the CSI-2 PHY
The IP also uses a 'digital' core supply, enabled by software and internall derived from **VDD_LOGIC**.

The following figure shows how CSI-2 power supplies are typically controlled and the table describes which combinations are valid and which are not.
In particular ;
- an application that never uses CSI-2 can leave supply pins CSI2_VCCA18 and CSI2_VCCA065 open or tie them to Ground.
- an application that makes permanent use of CSI-2 (outside sleep) and is very power-conscious may want to externally switch off the CSI-2 I/O supply (CSI2_VCCA18) during GAP9 sleep,
- an application that wishes to completely cut CSI-2 power consumption anytime CSI-2 is not used, including outside GAP9 sleep, should power off not only the CSI-2 I/O supply (CSI2_VCCA18) but also the CSI-2 core supply (CSI2_VCCA065).




| *CSI-2 External & Internal Power Supplies* | | | *State of CSI-2 Interface* |
|---|---|---|---|
| **CSI2_VCCA18** | **CSI2_VCCA065** | **Enable_CSI2** | |
| 1.8V | 0.65V-0.8V | OFF | Valid - CSI-2 IP off, ready to be enabled.<br>Standby current ~ *TBD* uA *<...give a value for each power supply ? >* |
| 1.8V | 0.65V-0.8V | ON | Valid - CSI-2 IP enabled and ready for image capture. Idle current in this mode is *TBD* uA. *<IP DS says 200uW typ. 290uW max but does not specifiy how it's split over the different supplies >* |
| 1.8V | OFF | OFF | Valid for stand-by, although not optimal. This situation typically occurs if GAP9 enters sleep mode (→ Vdd_logic=0 → csi2_vcca065=0) and 1.8V is still present. Stand-by current in that configuration may reach *TBD* uA, and up to *TBD* worst case at high temperature. Acceptability depends on application. |
| OFF | OFF | OFF | Valid - CSI-2 interface off – either temporarily (duty cycling of camera) or permanently (no camera used in application) |
| Any other combination | | | Out of spec. In particular, software must not enable CSI-2 IP unless both CSI2_VCCA065 and CSI2_VCCA18 are present. |

### > CSI-2 Trace Routing

The clock and data lanes are **high-speed differential pairs** and should be carefully routed as such, with **controlled characteristic impedance**.
Good practices for the routing of high-speed signals shoud be adhered to. They're not repeated here as many resources on that topic exist (including some good Application Notes from reputable silicon vendors which can *e.g.* be found by searching for « *high speed PCB trace routing recommandations »* on the Internet.

Required electrical characteristics and timings of the CSI-2 interface are listed in the MIPI Alliance *« Specification for D-PHY »* document. Key take-aways and recommandations include :

- the clock and data lanes should be routed as very **loosely coupled differential pairs, with 50-ohm single ended trace impedance (100-ohm differential)**. Total trace length should be minimized and the number of vias on those traces should absolutely be minimized. If vias are unavoidable, they should appear as much as possible at the same location on both signals of the differential pair. Make sure a trace on the bottom layer also runs over a reference plane (preferably GND) to maintain the characteristic impedance, and place GND vias close to the layer change (or 10nF cap between GND and PWR if the latter is the reference plane), to properly handle return current.

- keep CSI-2 traces away from other high-speed signals. To keep loose coupling, traces within a differential pair should be separated by at least 3x the trace width. Never route them over a reference plane split.

- **propagation time differences (skew)** between signals of a same pair (intra-pair skew) and between a data and a clock lane (inter-pair skew) **should be minimal.** This typically requires the use of 'serpentine' PCB traces to balance their length. The balancing (length matching) should be done as close as possible to the source of unbalance to have tightly in-phase signals along the longest possible portion of the differential pair. Maximum skew should be considered from an end-to-end point of view, targeting values in line with the D-PHY official spec, – nevertheless, the following values are a good starting point for PCB design: maximum intra-pair skew <1ps (~150um), maximum clock-data skew <10ps (~1.5mm),

- the maximum allowed 'fligth time' from transmitter to receiver is 2ns according to the D-PHY specification. For most dielectrics this translates as 25-30cm. Pay attention to the fact this is for the complete path, inlcuding any FCC cable etc.

- addition of test points should be considered very carefully as these cause impedance discontinuity and therefore affect signal integrity

## 9. SERIAL AUDIO INTERFACES (SAI)

GAP9 provides 3 bidirectional serial audio interfaces (SAI). Each of them can be configured to support the following :

- **I2S** (Inter-IC Sound) protocol – for transport of up to 2 **PCM** streams (i.e. stereo)

- **TDM** (Time Division Multiplex) – for transport of multiple **PCM** streams

- **PDM** (Pulse Density Modulation) – typically for connection of up to 4 digital microphones or some PDM/DSD compatible D/A converter / codec / digital amplifier

Each SAI interface involves 4 signals : SAIx_SCK (serial clock, up to 50MHz), SAIx_WS (Word Select *a.k.a* Frame Sync), SAIx_SDO (serial data out – from GAP9 point of view), SAIx_SDI (serial data in).

While the names are largely self-explanatory when mapping SAI to an I2S or TDM interface, the PDM configuration is special.

### > PCM over I2S or TDM interface -

Figure 3 outlines the connection of an I2S or TDM equipment to GAP9 using an SAI.

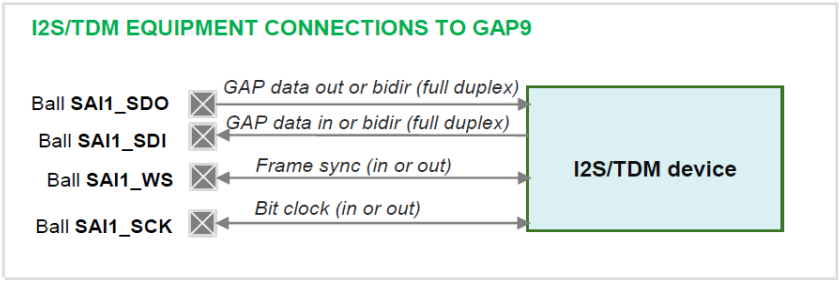



*Fig. 3 - Connection of an I2S device to, e.g., SAI1*

### > PDM interface -

Fig. 4 illustrates how the internal PDM engine connects to the SAI pins.

This scheme can be used in different ways :

**> PDM inputs :**

- **'standard' clocked PDM mode** – with DDR (double data rate) support for L/R multiplexing, as typically used by digital microphones – so 2 data streams per PDM channel

**> PDM outputs :**

- **clocked PDM mode** – without DDR support i.e. 1 data stream per PDM channel
- **clockless differential mode.**

These modes are explicited in the next paragraphs.

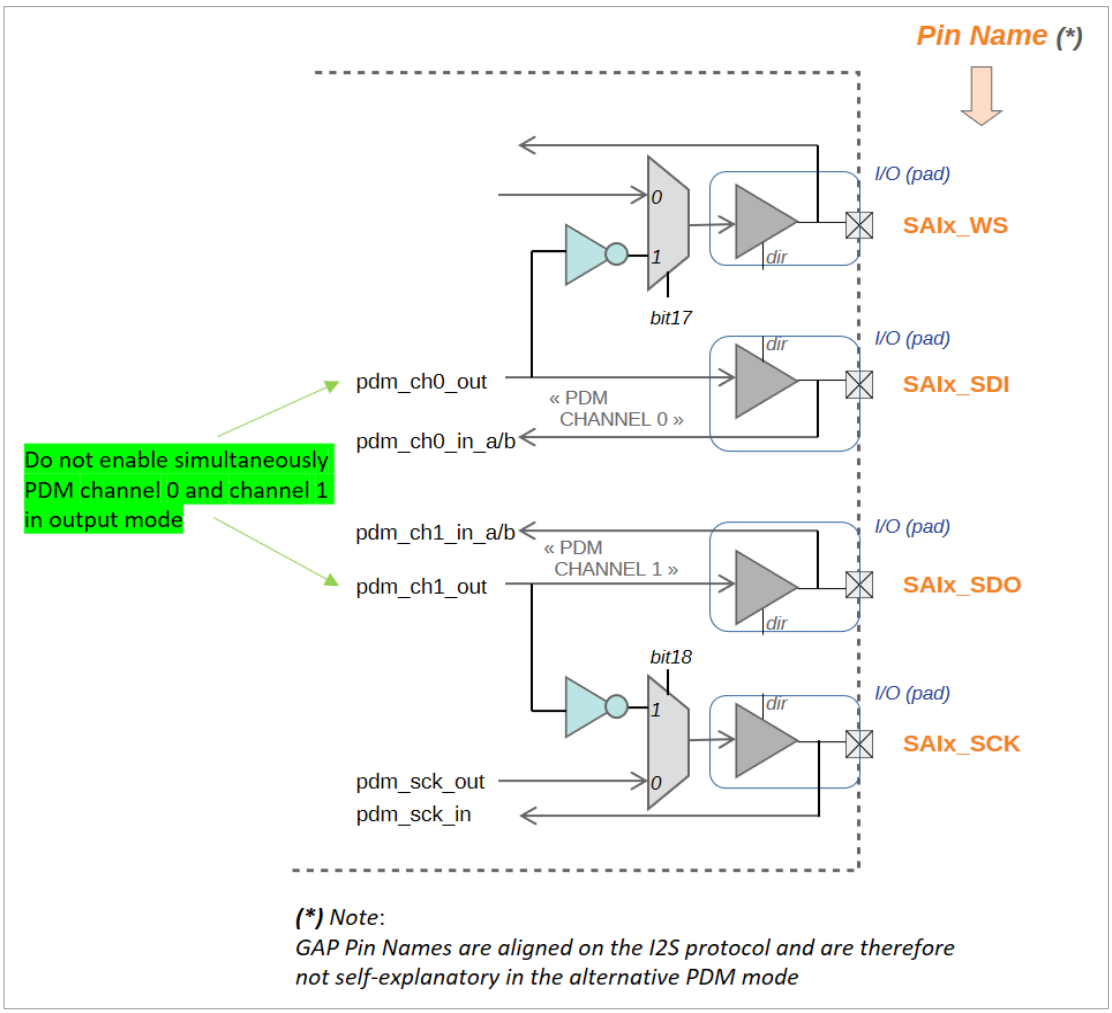



*Fig. 4 - Mapping of PDM signals to SAI interface*

**(1) PDM inputs** – typ. for digital microphones

A PDM microphone typically provides its output data on one selectable clock edge of the PDM clock and tristates its data I/O on the other clock edge. This enables connecting 2 microphones, (a) and (b) (for instance, Left and Right channels) to the same data line with temporal multiplexing of the 2 channels. Fig.5 illustrates this.

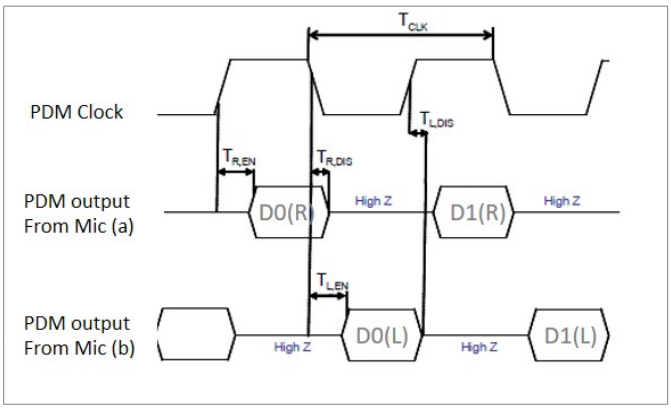



*Fig. 5 – Typical PDM clock and PDM data waveforms for a digital microphone*

Two PDM input channels (each bearing 2 multiplexed data streams, so 4 data streams overall) can be born by one SAI interface of GAP9. Refer to the tables at the end of this section for pin assignment.

Fig. 6 illustrates how up to 4 digital microphones can connect to GAP9 through one SAI interface (each microphone normally has a L/R input tied to either Logic0 or Logic1, not shown here, to decide on which phase of the PDM clock it drives data and on which phase it tri-states).

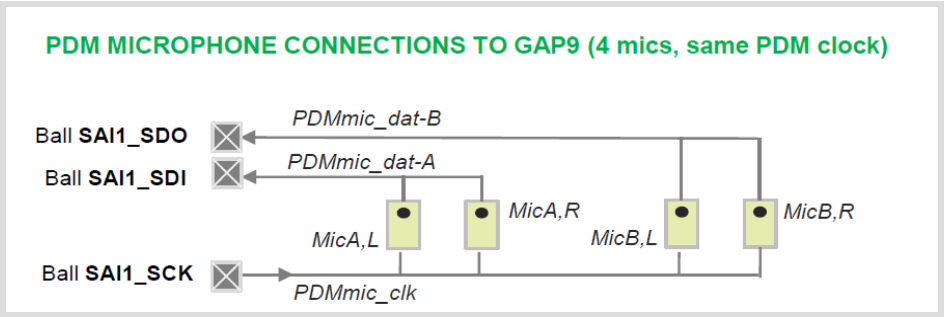



*Fig. 6 - Connection of 4 digital PDM Microphones to, e.g., SAI1*

### (2) PDM outputs

Two PDM output modes are supported : 'standard' clocked PDM mode (without DDR support though) and differential PDM mode.

**IMPORTANT NOTE :** although 2 PDM channels can technically be enabled on a given SAI channel, **only one PDM output channel should be enabled at a time**. See details in dedicated section below.

#### - 2.a - PDM outputs in single-ended mode

In this mode, GAP9 provides PDM ouput data on the falling edge of the PDM clock (to be sampled by the receiving device, typically an ADC, on the rising edge of the PDM clock). It does not go to high impedance state on the other clock edge nor does it perform any multiplexing. Refer to Fig.7. Refer to the tables at the end of this section for possible pin assignments.

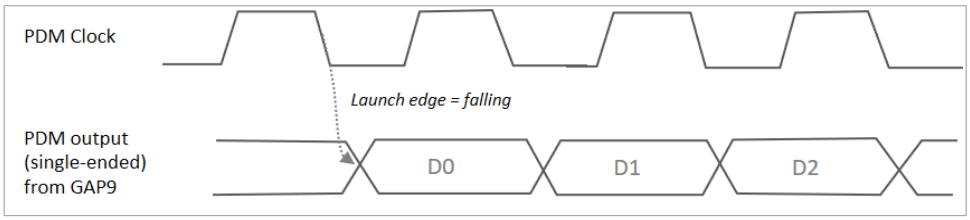



*Fig. 7 – GAP9 'standard' PDM waveforms (clock + data, single-ended)*

#### - 2.b - PDM outputs in differential mode

'Differential PDM' can be output by GAP9 through two I/Os, one carrying a pulse-density modulated stream (sigma-delta) representative of an audio signal and the other carrying its complement. The swing is 0-1.8V, meaning common mode is 0.9V. Fig. 8 provides an illustrative waveform.

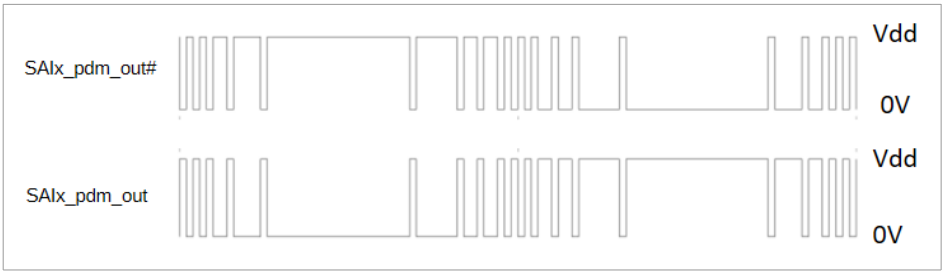



*Fig. 8 – Differential PDMoutput format from GAP9*

This a format specifically intended to drive a 'differential PDM Amplifier' essentially performing active analog low-pass filtering to reconstruct the analog signal encoded by the PDM pulses. One may consider the input PDM is treated as an analog differential input. No clock is required.

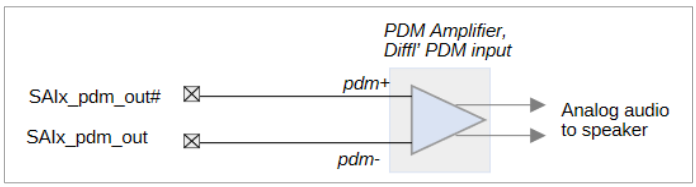



*Fig. 9 – Usage of differential PDM output*

## > Number of PDM channels per SAI

**> In input mode,** it is possible to clock in up to 2 multiplexed (e.g. Left/Right) streams, typ. from PDM microphones, on pdm_ch0_in and pdm_ch1_in – so 2 channels providing 4 PDM data streams overall after internal demultiplexing.

**> In output mode**, GAP9 can **output** a **single PDM channel per SAI** (be it differential or standard). Therefore, when two PDM channels need to be output (typically for stereo playback from a single GAP9), the Left channels must be output on SAIx while the Right channel must be output on SAIy, with 'x' different from 'y'.

As one PDM output channel consumes 2 lines of an SAI interface (pdm_chX_out and pdm_chX_out# in clockless differential mode, pdm_sck_out/in and pdm_chX_out in standard mode), the two other lines can still be used as PDM *inputs* (to connect e.g. 2 PDM microphones). Therefore, it is possible to map 2 PDM output channels and up to 4 PDM input channels on 2 serial audio interfaces as illustrated in Fig.10.a and Fig.10.b

One constraint though is that the PDM output and input channels sharing the same SAI must internally be clocked out/in by the same PDM clock.

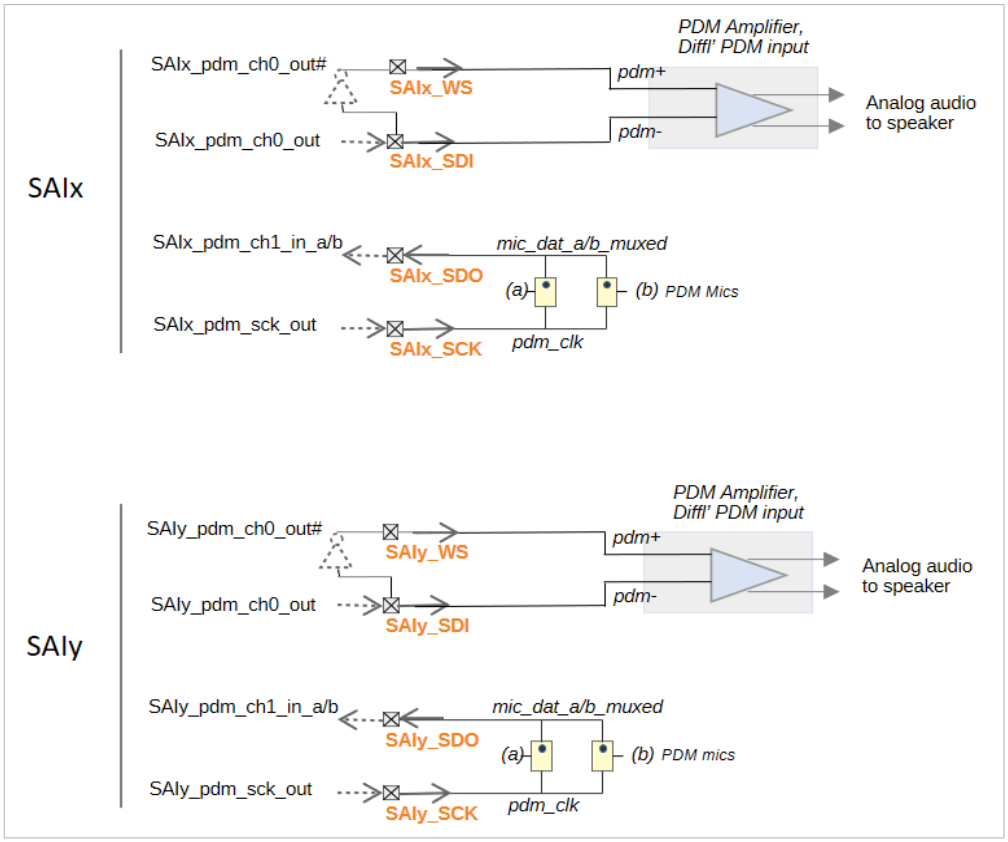



*Fig. 10.a – Mapping 2 differential PDM output channels and 4 PDM input channels onto 2 SAI of GAP9*

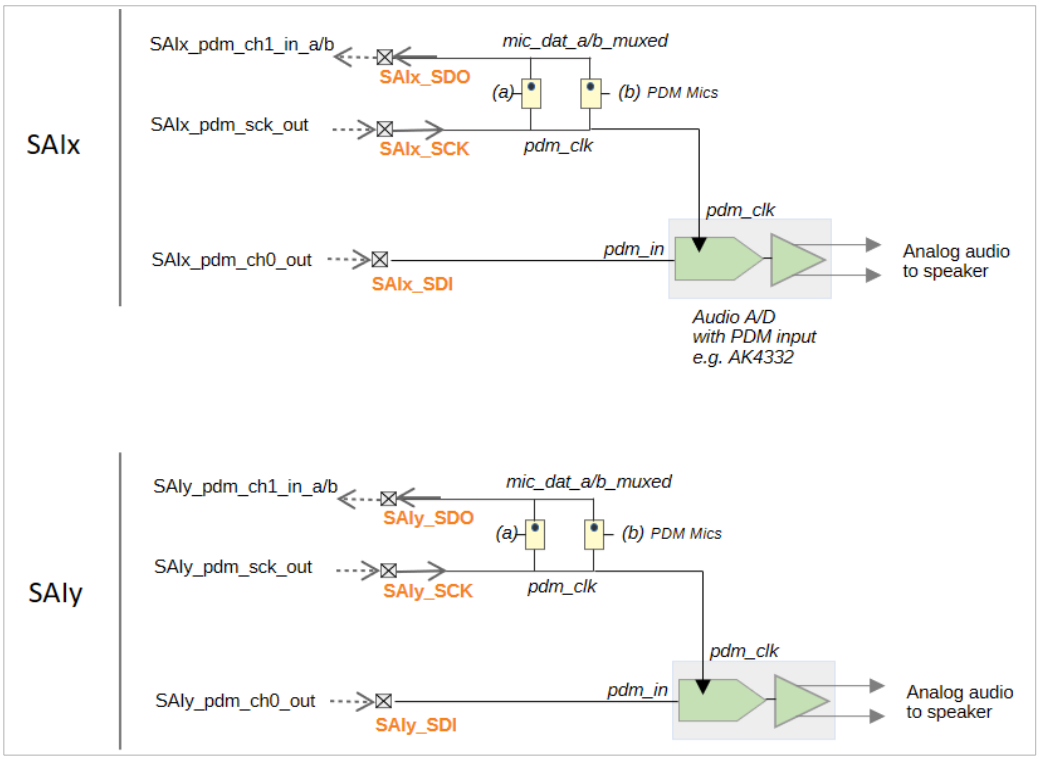



*Fig.10.b – Mapping 2 standard PDM output channels and 4 PDM input channels onto 2 SAI of GAP9*

## > Summary -

The following tables recap all mapping options that **one** SAI can support :

### PCM over I2S or TDM – up to 16 input or output channels (*)

| GAP9 Pin Name | Pin function | Direction | Purpose |
|---|---|---|---|
| SAIx_WS | i2s-tdm_ws_in or i2s-tdm_ws_out | In (slave) or Out (master) | Audio Frame Sync |
| SAIx_SDI | i2s-tdm_sdi | **In** | **PCM serial data input** |
| SAIx_SDO | i2s-tdm_sdo | **Out** | **PCM serial data output** |
| SAIx_SCK | i2s-tdm_sck_in or i2s-tdm_sck_out | In (slave) or Out (master) | PCM clock |

*(*) up 16 TDM (or multi-channel I2S) slots supported*
*– with the constraint that bit clock frequency must be equal to or greater than: 'number of slots' x 'bits per slot' x 'audio sample rate'*

### PDM – 1, 2, 3 or 4 inputs (typ. from digital mics)

| GAP9 Pin Name | Pin function | Direction | Purpose |
|---|---|---|---|
| SAIx_WS | n/a | | |
| SAIx_SDI | pdm_ch0_in_a/b | **In** | **PDM data input(s)** channel 0, DDR support *Data_a(e.g. left)/b(e.g. right) muxed, on opposite clock phases) - typ. from 1 or 2 digital microphone(s)* |
| SAIx_SDO | pdm_ch1_in_a/b | **In** | **PDM data input(s)** channel 1, DDR support *Data_a(e.g. left)/b(e.g. right) muxed, on opposite clock phases) - typ. from 1 or 2 digital microphone(s)* |
| SAIx_SCK | pdm_sck_in *or* pdm_sck_out | In *(slave)* or Out *(master)* | PDM serial clock |

### PDM – 1 (single-ended) output

| GAP9 Pin Name | Pin function | Direction | Purpose |
|---|---|---|---|
| SAIx_WS | n/a | | |
| SAIx_SDI | unused | | |
| SAIx_SDO | pdm_ch1_out | **Out** | **PDM (single-ended) output**<br>*typ. to A/D with PDM/DSD input support* |
| SAIx_SCK | pdm_sck_in *or*<br>pdm_sck_out | In *(slave)* or<br>Out *(master)* | PDM serial clock |

Alternative equivalent: swap roles of SAIx_SDI and SAIx_SDO, i.e. use **SAIx_SDI as pdm_ch0_out** and SAIx_SDO becomes unused

### PDM – 1 differential output

| GAP9 Pin Name | Pin function | Direction | Purpose |
|---|---|---|---|
| SAIx_WS | unused | | |
| SAIx_SDI | unused | | |
| SAIx_SDO | pdm_ch1_out | **Out+** | **PDM differential output**<br>*typ. to specific 'PDM Amplifier'* |
| SAIx_SCK | pdm_ch1_out# | **Out-** | **PDM differential output** (complement)<br>*typ. to specific 'PDM Amplifier'* |

Alternative equivalent: swap roles of [SAIx_SCK+SAIx_SDO] and [SAIx_WS+SAIx_SDI] ,
i.e. use **SAIx_SDI as pdm_ch0_out + SAIx_WS as pdm_ch0_out#** and other pair becomes unused

### PDM – 1 (single-ended) output + 1 or 2 inputs

| GAP9 Pin Name | Pin function | Direction | Purpose |
|---|---|---|---|
| SAIx_WS | n/a | | |
| SAIx_SDI | pdm_ch0_out | **Out** | **PDM (single-ended) output**<br>*typ. to A/D with PDM/DSD input support* |
| SAIx_SDO | pdm_ch1_in_a/b | **In** | **PDM data inputs**, channel 1 (DDR a/b)<br>*typ. from 1 or 2 digital microphone(s)* |
| SAIx_SCK | pdm_sck_in *or*<br>pdm_sck_out | In *(slave)* or<br>Out *(master)* | PDM serial clock<br>*Common to input and output PDM* |

**Alternative equivalent**: swap roles of SAIx_SDI and SAIx_SDO, i.e. use **SAIx_SDO as pdm_ch1_out** and **SAIx_SDI as pdm_ch0_in_a/b**

### PDM – 1 differential output + 1 or 2 inputs

| GAP9 Pin Name | Pin function | Direction | Purpose |
|---|---|---|---|
| SAIx_WS | pdm_ch0_out | **Out+** | **PDM differential output**<br>*typ. to specific 'PDM Amplifier'* |
| SAIx_SDI | pdm_ch0_out# | **Out-** | **PDM differential output** (complement)<br>*typ. to specific 'PDM Amplifier'* |
| SAIx_SDO | pdm_ch1_in_a/b | **In** | **PDM data inputs**, channel 1 (DDR a/b)<br>*typ. from 1 or 2 digital microphone(s)* |
| SAIx_SCK | pdm_sck_in *or*<br>pdm_sck_out | In *(slave)* or<br>Out *(master)* | PDM serial clock<br>*Common to input and output PDM* |

## 10. JTAG, H/W RESET & FOTA

### > JTAG Access

A GAP9-fitted board normally needs to provide access to GAP9's JTAG interface, as GAP9 programming and debug is performed through JTAG (altough, in specific cases and in a production context, other solutions might be envisaged). The JTAG interface on GAP9 consists of pins JTAG_TDI, TDO, TMS, TCK and NTRST. The JTAG programming probe will also require access to the harwdare reset pin NRESET (system reset). Most JTAG probes will also make use of a target power supply pin (used to detect presence and/or voltage level of connected I/Os) to be connected to GAP9's VDDIO power rail.
Please note the following :
> **NRESET** : **the JTAG probe needs to be able to assert the chip reset line** – i.e., drive it to low level. When not asserting reset, most probes will not drive NRESET, which should therefore be pulled up to VDDIO on the board through a resistor of e.g. 10-50 Kohm.
> **JTAG_NTRST** : the JTAG Test Access Port interface (TAP) of GAP9 can be reset through JTAG_NTRST (different from system reset above). However, some JTAG probes do not provide this signal and some JTAG probe adapters simply pull JTAG_NTRST low (which is the 'asserted' level of JTAG_NTRST). In the sake of flexiblity, it is recommended to **route JTAG_NTRST to the JTAG connector through a series resistor which would be non-fitted by default** and to **pull JTAG_NTRST to GAP9's VDDIO** (1.8V) on GAP9's side (through a few 10Kohm).

Space permitting, it is recommended to insert a series resistor of a few 100s ohm on each JTAG line to protect it from unintended over-voltage when manipulating/connecting the JTAG probe and to limit ringing (as external wires may be of significant length).

### > Hardware Reset and FOTA

Firmware Over-The-Air update, FOTA for short, is the process by which new resident firmware can be obtained by a product through a radio link and flashed into a processor to replace a previous version (or as initial firmware).

The GAP9 SDK comes with mechanisms that make FOTA possible. Code would typically be downloaded through the radio (*e.g.,* Bluetooth) SOC and, at some point after download is complete, this SOC will need to reset GAP9. This would typically be done by having a GPIO of the radio SOC driving pin NRESET of GAP9 to Logic-0 (asserted) – with the PCB design providing means to do so. Pay attention to the fact the NRESET pin may also need to be drivable by other agents, including in particular the JTAG programming probe (see former section). This means it is preferable to drive NRESET in open-drain mode and, in any case, it is essential to make sure no electrical conflict will occur when the JTAG probe or any other agent asserts NRESET.

## 11. OTHER RECOMMENDATIONS

### > Test Points for real-time algorithm tuning

Certain audio applications require some specific tuning of algorithms, which needs to be performed late in the design process, for instance on a prototype/product with (almost) final form factor – *e.g.,* to take into account its exact acoustic characteristics. This tuning typically requires real-time access to internal data. Interfaces of choice to export this data out of GAP9 would typically be SPI (possibly quad-SPI if bandwidth

needs are very high) or UART (which can achieve very decent throughput esp. with flow control). For applications falling in that category, it is therefore recommended to select suitable GAP9 interface signals and route them to adequate test points.

### > eMRAM Magnetic Immunity <TBD – Keep of not?>

Embedded MRAM technology is inherently sensitive to strong magnetic fields. If your application makes use of GAP9's eMRAM and your product design is such that GAP9 may come in very close vicinity (a few mm) of a strong magnet, either internal or external to the product, then you may need to take some precautions. Please refer to GreenWaves Technologies Application Note, « GAP9 Magnetic Immunity – with a focus on Hearables Devices ».

# APPENDIX

## – Crystal Load Capacitance Definition –

The effective load capacitance $C_L$ is the combination of internal capacitance on each pin of the oscillator, external capacitance on each pin of the oscillator and stray capacitance of PCB and chip package (see figure below).

It is calculated as :

$$C_L = \frac{C_{XA}.C_{XB}}{C_{XA} + C_{XB}} + C_{stray}$$

where :

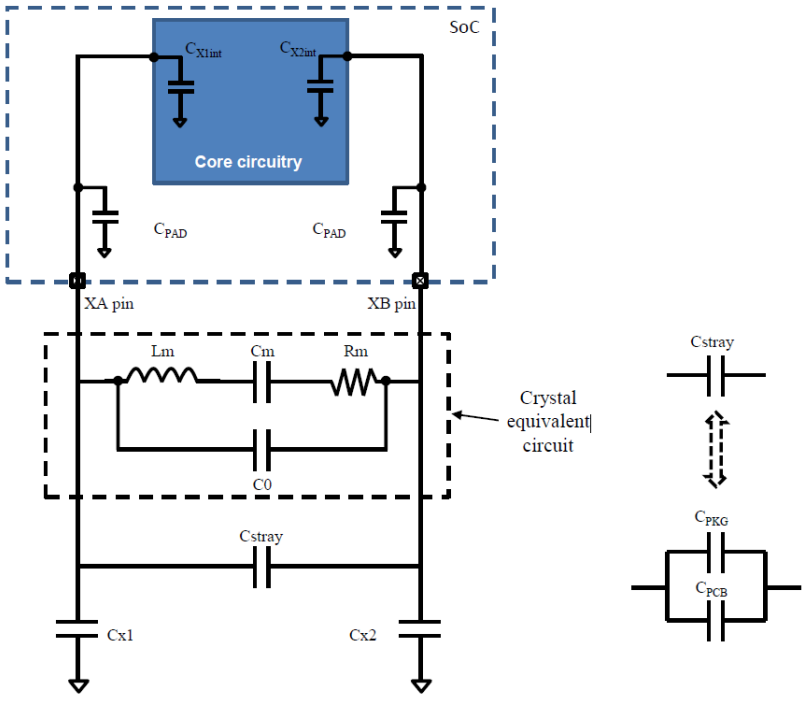
- $C_{XA}$ is the capacitance on pin XA of the crystal and consists of pad capacitance $C_{pad}$ (~0.5pF), external capacitance added on that pin ($C_{X1}$) plus, in the case of GAP8, an internal capacitance $C_{X1,INT}$ already provisioned on each pin (1pF) – and same for $C_{XB,}$ , i.e. :

$$C_{XA} = C_{X1} + C_{pad} + C_{X1_{int}} \text{ and } C_{XB} = C_{X2} + C_{pad} + C_{X2_{int}}.$$

- $C_{stray}$ consists of stray capacitance of the package and the PCB ; it is therefore board-dependent and may be 2-3pF with crystal kept very close to the chip.

*Example* :
With $C_{X1,INT}$ = $C_{X2,INT}$ = 1pF, $C_{pad}$ = 0.5pF and assuming external caps $C_{X1}$ = $C_{X1}$ = 1pF,
then $C_{XA}$ = $C_{XB}$ = 2.5pF ;
and assuming $C_{stray}$ = 2.5pF (PCB design dependent),
then $C_L$ = (2.5*2.5)/(2*2.5) + 2.5 = 3.75pF

This calculated $C_L$ value must be lower than or equal to the total load capacitance supported by the selected crystal. Lower $C_L$ value will favor fast start-up at the expense of somewhat degraded frequency accuracy.



## DOCUMENT HISTORY

***Draft A to F –*** *August-October, 2021 (Xavier Cauchy)*
Initial drafts

***Rel.1.0 –*** *10.Jan, 2022 (XC)*
First official release
Remove prior licensing info, mark as restricted to NDA Customers

***Rel.1.1 –*** *19.Jan, 2022 (XC)*
Rework Section 9 Serial Audio Interfaces
- impossiblity to output 2 PDM channels onto a single SAI interface – need to dispatch 2 PDM outputs over 2 SAI
- PDM output format of GAP9 not DDR capable

***Rel.1.2 –*** *22.Feb, 2022 (XC)*
GAP9 cannot invert Chip Select to RAM or Flash after all – Update ref design and explainations accordingly.
Add instruction to always provide JTAG_NTRST as part of JTAG interface.

***Rel.1.3 –*** *16.May, 2022 (XC)*
Add information on eFuse current consumption.
Add information about CSI-2 core power consumption
Update information about JTAG_NTRST.

***Rel.1.4 –*** *8.Dec, 2022 (XC)*
Change recommended output cap on Vsafe from 1uF to 4.7uF.
Add explicit recommendation (in Power Supply Sequencing) to slightly delay reset signal wrt VDDIO
Add information in GPIO section, regarding remapping and usage as wake-up source
Add short note on test points for algorithm tuning

***Rel.1.5 –*** *29.March, 2023 (XC)*
Add information about Hardware Reset and FOTA

***Rel.1.6 –*** *16.May, 2024 (XC)*
Specify that some Octal-SPI RAM have a peculiar handling of address bits and won't be compatible.
Further highlight that usage of external memories is optional.
Advise usage of RBX option for RAM.
Suggest to tie Flash reset to system hardware reset.

***Rel.1.7 –*** *18.July, 2024 (XC)*
Update recommendations regarding eFuse power supply (to avoid risks of accidental corruption)
Complete and clarify CSI-2 power supply recommendations