**IN THE UNITED STATES DISTRICT COURT**
**FOR THE EASTERN DISTRICT OF TEXAS**
**MARSHALL DIVISION**

| | |
|---|---|
| LIBERTY PATENTS LLC,<br>Plaintiff,<br>v.<br>TEXAS INSTRUMENTS, INC.,<br>Defendant. | CIVIL ACTION NO. 2:21-cv-50<br>ORIGINAL COMPLAINT FOR PATENT INFRINGEMENT<br>**JURY TRIAL DEMANDED** |

## ORIGINAL COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Liberty Patents LLC ("Liberty Patents" or "Plaintiff") files this original complaint against Defendant Texas Instruments, Inc. ("TI" or "Defendant"), alleging, based on its own knowledge as to itself and its own actions and based on information and belief as to all other matters, as follows:

## PARTIES

1. Liberty Patents is a limited liability company formed under the laws of the State of Texas, with its principal place of business at 2325 Oak Alley, Tyler, Texas, 75703.

2. Defendant Texas Instruments, Inc. is a company organized and existing under the laws of the state of Delaware. Texas Instruments, Inc. may be served with process through its registered agent, CT Corporation System at 1999 Bryan St., Suite 900, Dallas, Texas, 75201.

3. TI is "a global semiconductor company that designs, manufactures, tests and sells analog and embedded processing chips."[1] It "design[s], manufacture[s], test[s] and sell[s] analog and embedded semiconductors" for "nearly 100,000 customers around the globe."[2] It has a

[1] *See* https://www.ti.com/about-ti/company/overview.html.

[2] *See* https://www.ti.com/about-ti/company/what-we-do.html.

portfolio of about 80,000 products with 14 manufacturing sites worldwide, operating "10 wafer fabs, seven assembly and test factories, and multiple bump and probe facilities."[3]

## JURISDICTION AND VENUE

4. This is an action for infringement of a United States patent arising under 35 U.S.C. §§ 271, 281, and 284–85, among others. This Court has subject matter jurisdiction of the action under 28 U.S.C. § 1331 and § 1338(a).

5. This Court has personal jurisdiction over TI pursuant to due process and/or the Texas Long Arm Statute because, *inter alia*, (i) TI has done and continues to do business in Texas; (ii) TI has committed and continues to commit acts of patent infringement in the State of Texas, including making, using, offering to sell, and/or selling accused products in Texas, and/or importing accused products into Texas, including by Internet sales and/or sales via retail and wholesale stores, inducing others to commit acts of patent infringement in Texas, and/or committing at least a portion of any other infringements alleged herein in Texas, and (iii) TI is registered to do business in Texas.

6. Venue is proper in this district under 28 U.S.C. § 1400(b). Venue is further proper because TI has committed and continues to commit acts of patent infringement in this district, including making, using, offering to sell, and/or selling accused products in this district, and/or importing accused products into this district, including by Internet sales and/or sales via retail and wholesale stores, inducing others to commit acts of patent infringement in this district, and/or committing at least a portion of any other infringements alleged herein in this district.

[3] *Id.*

7. TI also has a regular and established place of business in this district, including at least at 6412 US-75, Sherman, Texas, 75090:

6412 US-75,
Sherman, TX 75090
Texas Instruments Inc

**Source:** https://goo.gl/maps/QsyZgk86RKkHp7qb8; https://goo.gl/maps/rBda8qr16vW3LPHa8.

## BACKGROUND

8. The patents-in-suit describe key improvements in electronic circuitry relating to the enhancement of processors through reduced power consumption, increased processing, and greater performance.

9. U.S. Patent Nos. 7,509,504 ("the '504 Patent"), 8,127,156 ("the '156 Patent"), and 8,458,496 ("the '496 Patent") are part of the same family of patents, which generally relates to body biasing voltages used in integrated circuits (ICs). The patents teach application of body biasing techniques that improve circuit performance and reduce power consumption in one or more power modes. Specifically, they disclose systems and methods for generating body biasing voltages so that a processor can operate with increased power savings. For example, the family of patents describes techniques for using body biasing voltages to decrease power consumption during high performance applications. Other examples disclose use of body biasing voltages during low power modes.

10. The use of body biasing voltages in ICs has become an increasingly necessary design feature in many applications today. Coupled with the explosive demand for ICs over the last few years, the more stringent requirement that ICs consume less and less power has focused the industry towards using body biasing voltages. Body biasing voltage techniques are now being used in applications ranging from automotive technologies to industrial IoT devices. Whether an application requires high performance circuitry or ultra-low power modes (or both), body biasing techniques have become essential.

11. The technology described by the '504 Patent, the '156 Patent, and the '496 Patent was developed by engineers at Transmeta Corp. Transmeta was a technology company formed in 1995 and best known for designing high performance processors, such as the Crusoe and the Efficeon in the early 2000s. In particular, Transmeta's major focus was on developing low

power, high performance ICs. To achieve such high power savings, one of the major techniques used by Transmeta engineers was to apply body biasing voltages to its ICs.

12. Industry experts have recognized the technological innovation of Transmeta's processors, and some have noted that Transmeta's energy-saving processors were ahead of their time.[4] The inventions disclosed in these patents are extremely important to multiple industries, and have been cited by major technology companies and processor developers like Canon, Freescale Semiconductor (now part of NXP), Nvidia, Packet Digital, and Smart Technologies (now part of Foxconn).

13. U.S. Patent No. 6,535,959 ("the '959 Patent") covers technology used in a wide array of electronic devices and applications, including computer processors for mobile and automotive industries, SoCs, graphics display controllers, LCD systems, etc. More particularly, it describes techniques for more efficient handling of computer instructions in processors for faster processing.

14. In particular, the '959 Patent discloses a processor that includes a set-associative instruction cache comprising multiple blocks. Claim 1 of the '959 patent, as an example, is directed to a processor that generates a power reduction signal, which indicates whether the subsequent instruction to be executed resides in the same block of the instruction cache as the current instruction that is being executed. This advantageously allows, for example, the processor to read consecutive instructions (or instructions that are in the same block) quickly,

[4] *See, e.g.*, *Chip Hall of Fame: Transmeta Corp. Crusoe Processor*, IEEE Spectrum (June 30, 2017) ("Ahead of its time, this chip heralded the mobile era when energy use, not processing power, would become the most important spec."), https://spectrum.ieee.org/tech-history/silicon-revolution/chip-hall-of-fame-transmeta-corp-crusoe-processor.

without multiple additional steps. The novel system results in a processor with increased operating speed and decreased power consumption.

15. The invention described in the ’959 Patent was the result of research conducted by two inventors at Conexant Systems, Inc., which was—at the time—the world’s largest, standalone communications-IC company. Conexant, itself, was a spin-off from the semiconductor division of the well-known and well-regarded Rockwell International Corp. Conexant was known as a leading supplier of innovative semiconductor solutions for imaging, audio, embedded modem, and video surveillance applications.[5] Recently, Conexant was acquired by Synaptics, the leading developer of human interface solutions for over $300 million. Since its formation, Conexant has been an innovator in the semiconductor field (and others) with more than a thousand patents assigned to it.

16. The ’959 Patent has been cited by multiple technology companies—as recently as 2017—including, Apple, ARM, Fujitsu, Hewlett-Packard, Honeywell, IBM, Intel, Panasonic (Matsushita), Oracle, Samsung, STMicroelectronics, Toshiba, and Transmeta.

## COUNT I

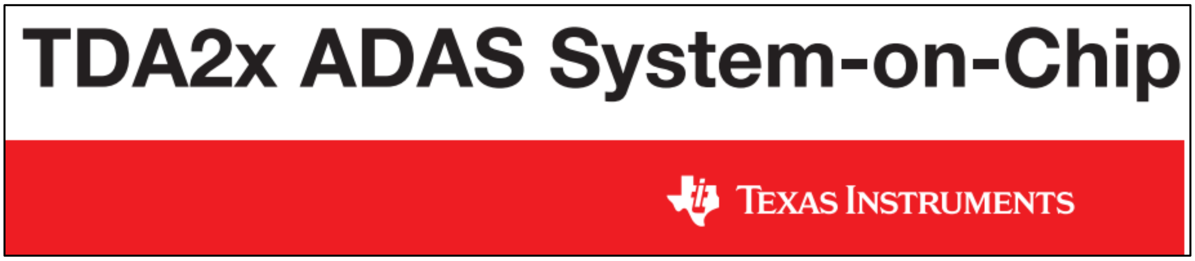
## DIRECT INFRINGEMENT OF U.S. PATENT NO. 7,509,504

17. On March 24, 2009, U.S. Patent No. 7,509,504 (“the ’504 Patent”) was duly and legally issued by the United States Patent and Trademark Office for an invention entitled “Systems and Methods for Control of Integrated Circuits Comprising Body Biasing Systems.”

[5] *See* Conexant’s Audio Solution Named CES Innovations 2011 Awards Honoree, BUSINESS WIRE (Nov. 9, 2010), http://finance.sunnyvale.com/camedia.sunnyvale/news/read/15498866/conexant%E2%80%99s_audio_solution_named_ces_innovations_2011_awards_honoree.

18. Liberty Patents is the owner of the ’504 Patent, with all substantive rights in and to that patent, including the sole and exclusive right to prosecute this action and enforce the ’504 Patent against infringers, and to collect damages for all relevant times.

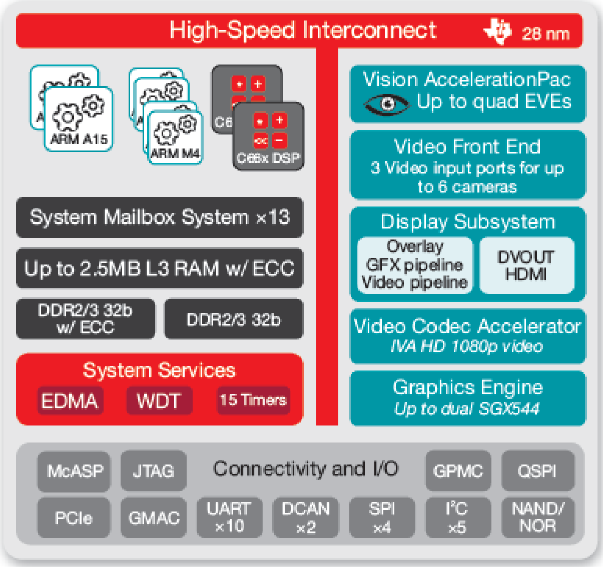
19. TI made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, TI’s TDA2x System-on-Chip devices and other products[6] that are capable of applying body biasing voltages during a power mode of a processor or other integrated circuit (“accused products”):



TDA2x ADAS System-on-Chip

Texas Instruments

[6] *See, e.g.*, TI AM5706, AM5708, AM5716, AM5718, AM5718-HIREL, AM5726, AM5728, AM5729, AM5746, AM5748, AM5749, DRA710, DRA712, DRA714, DRA716, DRA718, DRA722, DRA724, DRA725, DRA726, DRA744, DRA745, DRA746, DRA74P, DRA750, DRA756, DRA75P, DRA76P, DRA77P, DRA790, DRA791, DRA793, DRA797, OMAP4430, OMAP4460, OMAP4470, OMAP5430, OMAP5432, TDA2E, TDA2EG-17, TDA2HF, TDA2HG, TDA2HV, TDA2LF, TDA2P-ABZ, TDA2P-ACD, TDA2SA, TDA2SG, TDA2SX, TMDXIDK5718, TMDSEVM572X, TMDSIDK572, TMDSIDK574, D3-3P-ADAS-DK, D3-3P-TDAX-DK, DRA71XEVM, DRA72XEVM, DRA79XEVM, 703664-1001, 705851-1001, 705852-1001, DRA76XP-DRA77XP-TDA2PX-ACD-CPUBOARD, J6EVM5777, OMAP5432-EVM, TDA2EXEVM, TDA2PXEVM, EVM5777BG-03-00-00, EVM5777G-03-40-00, EVM5777VISION-V2-0, etc.



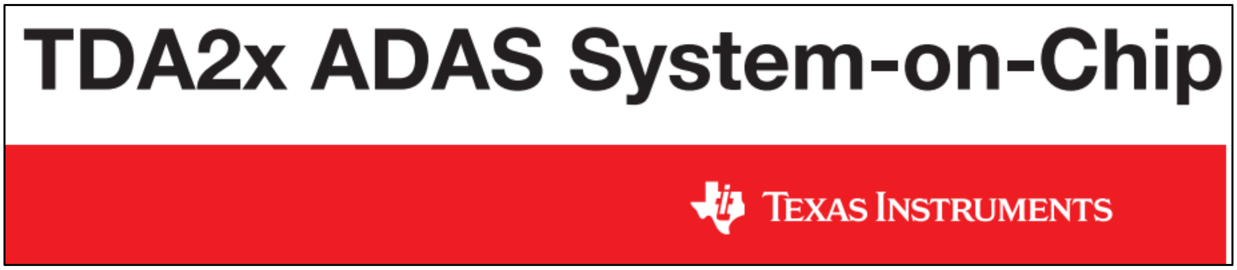
▲ *Figure 1. Block diagram for TDA2x SoC*

**Source:**
https://www.ti.com/lit/ml/sprt681/sprt681.pdf?ts=1597386693056&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 1)

20. By doing so, TI has directly infringed (literally and/or under the doctrine of equivalents) at least Claims 1 and 9 of the ’504 Patent. TI’s infringement in this regard is ongoing.

21. TI’s TDA2x System-on-Chip devices are exemplary accused products. The devices include a computer system.

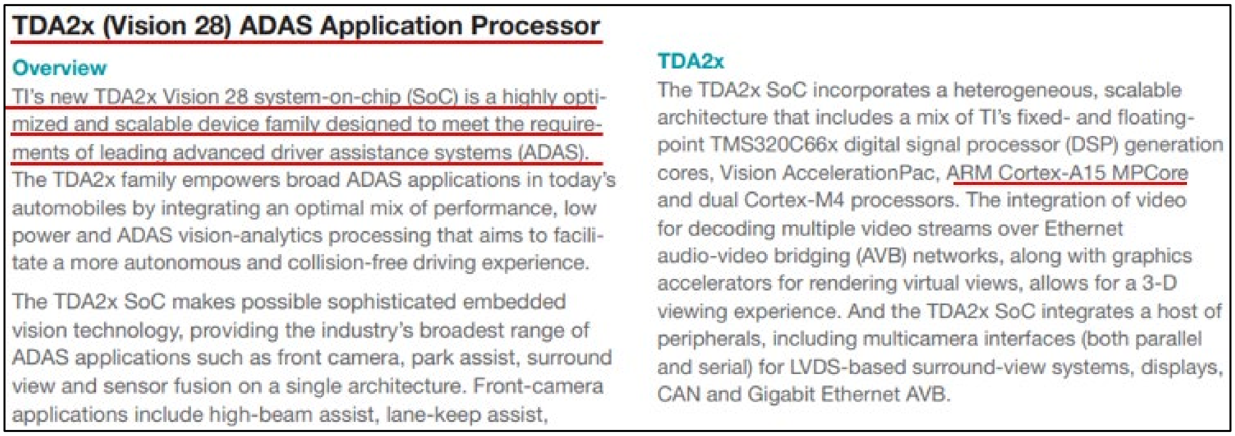
22. For example, TI’s TDA2x System-on-Chip devices include a dual core processor module along with different memory elements and peripherals (“computer system”). The devices support Adaptive Body Bias (ABB), which can apply a forward or reverse body bias voltage to the transistors of the device using ABB.



TDA2x ADAS System-on-Chip

TEXAS INSTRUMENTS

**Source:** https://www.ti.com/lit/ml/sprt681/sprt681.pdf?ts=1597386693056&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 1)



TDA2x (Vision 28) ADAS Application Processor

Overview

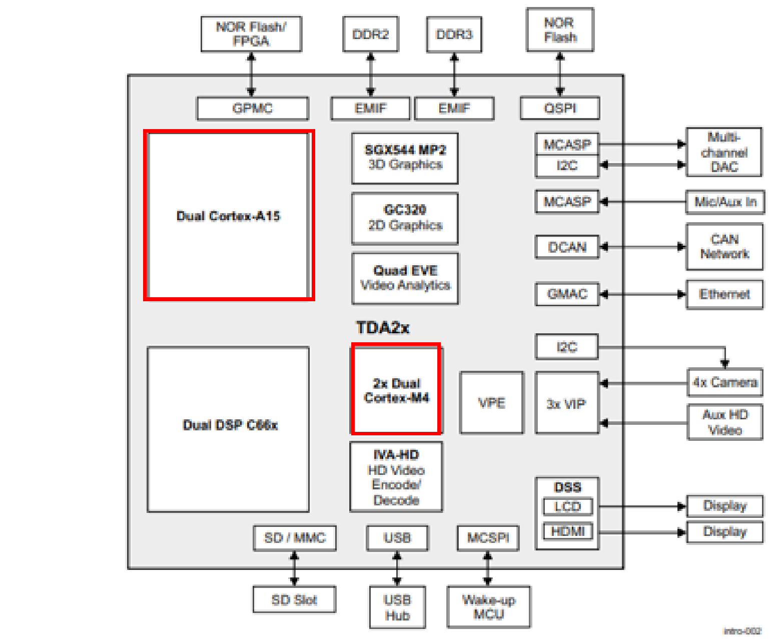
TI's new TDA2x Vision 28 system-on-chip (SoC) is a highly optimized and scalable device family designed to meet the requirements of leading advanced driver assistance systems (ADAS). The TDA2x family empowers broad ADAS applications in today's automobiles by integrating an optimal mix of performance, low power and ADAS vision-analytics processing that aims to facilitate a more autonomous and collision-free driving experience.

The TDA2x SoC makes possible sophisticated embedded vision technology, providing the industry's broadest range of ADAS applications such as front camera, park assist, surround view and sensor fusion on a single architecture. Front-camera applications include high-beam assist, lane-keep assist,

TDA2x

The TDA2x SoC incorporates a heterogeneous, scalable architecture that includes a mix of TI's fixed- and floating-point TMS320C66x digital signal processor (DSP) generation cores, Vision AccelerationPac, ARM Cortex-A15 MPCore and dual Cortex-M4 processors. The integration of video for decoding multiple video streams over Ethernet audio-video bridging (AVB) networks, along with graphics accelerators for rendering virtual views, allows for a 3-D viewing experience. And the TDA2x SoC integrates a host of peripherals, including multicamera interfaces (both parallel and serial) for LVDS-based surround-view systems, displays, CAN and Gigabit Ethernet AVB.

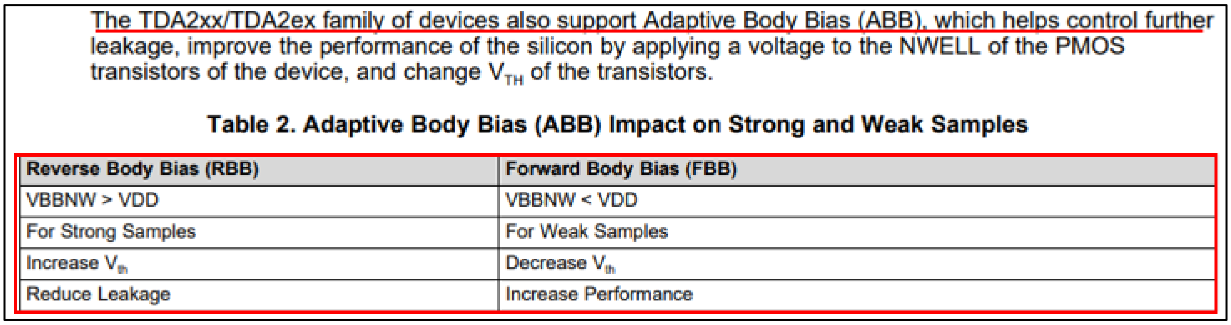
**Source:** https://m.eet.com/media/1241324/slyy044.pdf (Page 5)



A The speeds shown are the highest design targeted, non-binned OPP.

Figure 1. TDA2xx Device Block Diagram

**Source:** https://www.ti.com/lit/an/sprac21a/sprac21a.pdf?ts=1597386703137&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples

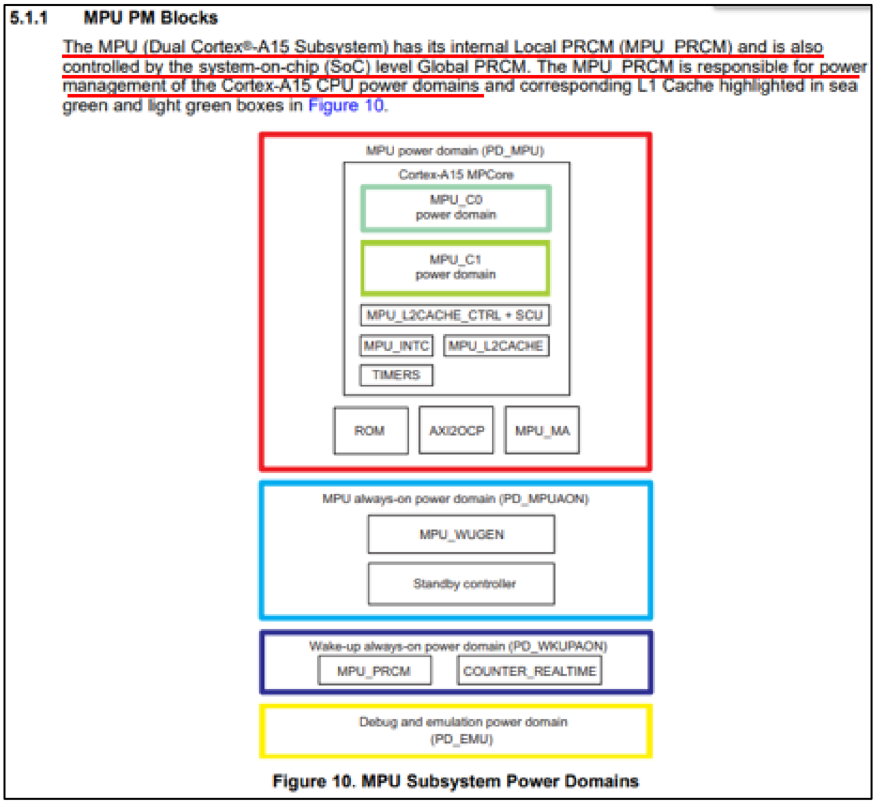
| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:** https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)

23. TI's TDA2x System-on-Chip devices include a plurality of data storage locations comprising body biasing voltage information corresponding to a plurality of power modes of the computer system. At least one of the plurality of power modes enables operation of the computer system at at least two different clock frequencies.

24. For example, TI's TDA2x System-on-Chip devices include a dual core processor module. Further, the System-on-Chip includes an MPU_PRCM module for power management of the processor. The TDA2x family of devices supports multiple power states (e.g., a power state in which both cores are on, a power state in which one core is forced off, a power state in which one core is forced off and the other core is in retention, etc.).

25. The devices support Adaptive Body Bias (ABB). ABB voltage values are available for each OPP (Operating Performance Point), and therefore, for each power state. The OPP is the operating condition for the device and is defined by the voltage and frequency applied to the processor. The ABB voltage values ("body biasing voltage information") corresponding to the different power states ("plurality of power modes") are stored in Efuse registers ("data storage locations"). A power state changes based on the processor's performance, and each power state is associated with a different clock frequency and/or a different voltage.



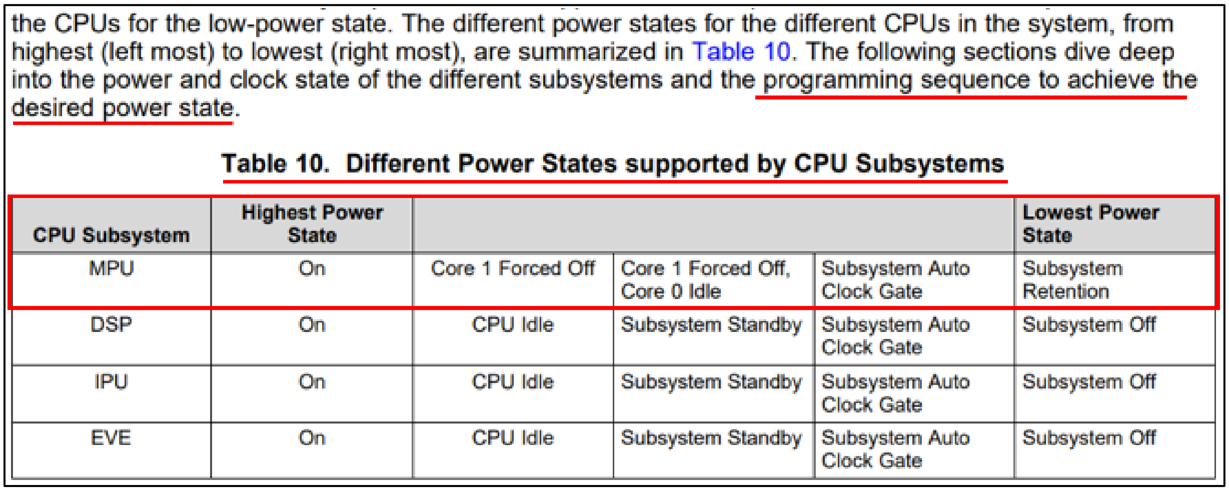
**5.1.1 MPU PM Blocks**

The MPU (Dual Cortex®-A15 Subsystem) has its internal Local PRCM (MPU_PRCM) and is also controlled by the system-on-chip (SoC) level Global PRCM. The MPU_PRCM is responsible for power management of the Cortex-A15 CPU power domains and corresponding L1 Cache highlighted in sea green and light green boxes in Figure 10.

**Figure 10. MPU Subsystem Power Domains**

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 19)



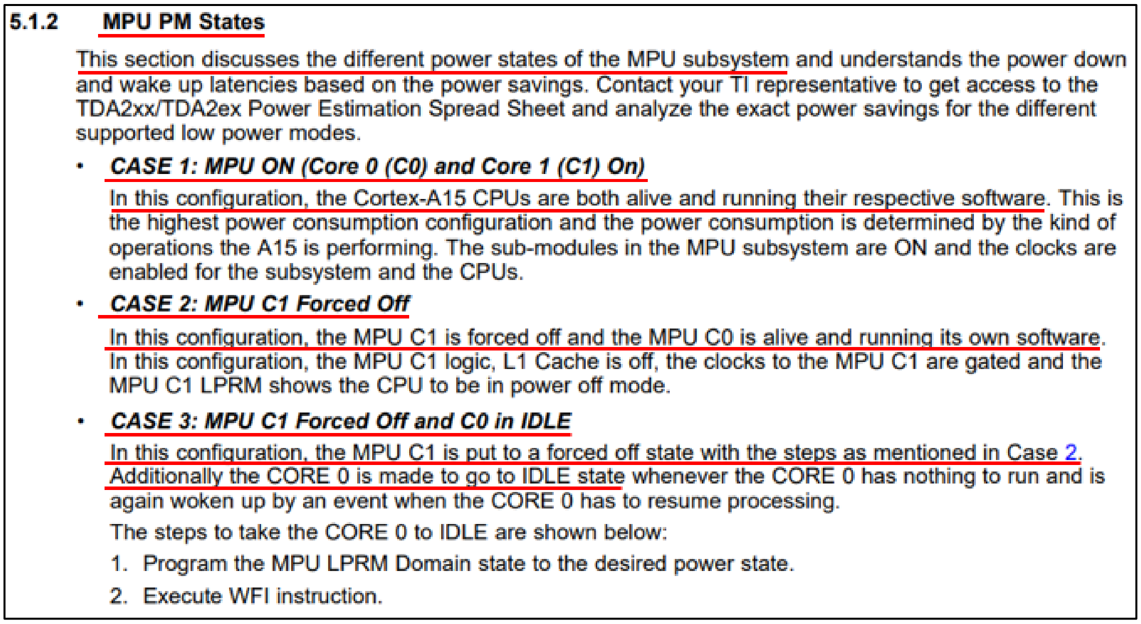
the CPUs for the low-power state. The different power states for the different CPUs in the system, from highest (left most) to lowest (right most), are summarized in Table 10. The following sections dive deep into the power and clock state of the different subsystems and the programming sequence to achieve the desired power state.

**Table 10. Different Power States supported by CPU Subsystems**

| CPU Subsystem | Highest Power State | | | | Lowest Power State |
|---|---|---|---|---|---|
| MPU | On | Core 1 Forced Off | Core 1 Forced Off, Core 0 Idle | Subsystem Auto Clock Gate | Subsystem Retention |
| DSP | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| IPU | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| EVE | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 18)



5.1.2 **MPU PM States**

This section discusses the different power states of the MPU subsystem and understands the power down and wake up latencies based on the power savings. Contact your TI representative to get access to the TDA2xx/TDA2ex Power Estimation Spread Sheet and analyze the exact power savings for the different supported low power modes.

- ***CASE 1: MPU ON (Core 0 (C0) and Core 1 (C1) On)***

  In this configuration, the Cortex-A15 CPUs are both alive and running their respective software. This is the highest power consumption configuration and the power consumption is determined by the kind of operations the A15 is performing. The sub-modules in the MPU subsystem are ON and the clocks are enabled for the subsystem and the CPUs.

- ***CASE 2: MPU C1 Forced Off***

  In this configuration, the MPU C1 is forced off and the MPU C0 is alive and running its own software. In this configuration, the MPU C1 logic, L1 Cache is off, the clocks to the MPU C1 are gated and the MPU C1 LPRM shows the CPU to be in power off mode.

- ***CASE 3: MPU C1 Forced Off and C0 in IDLE***

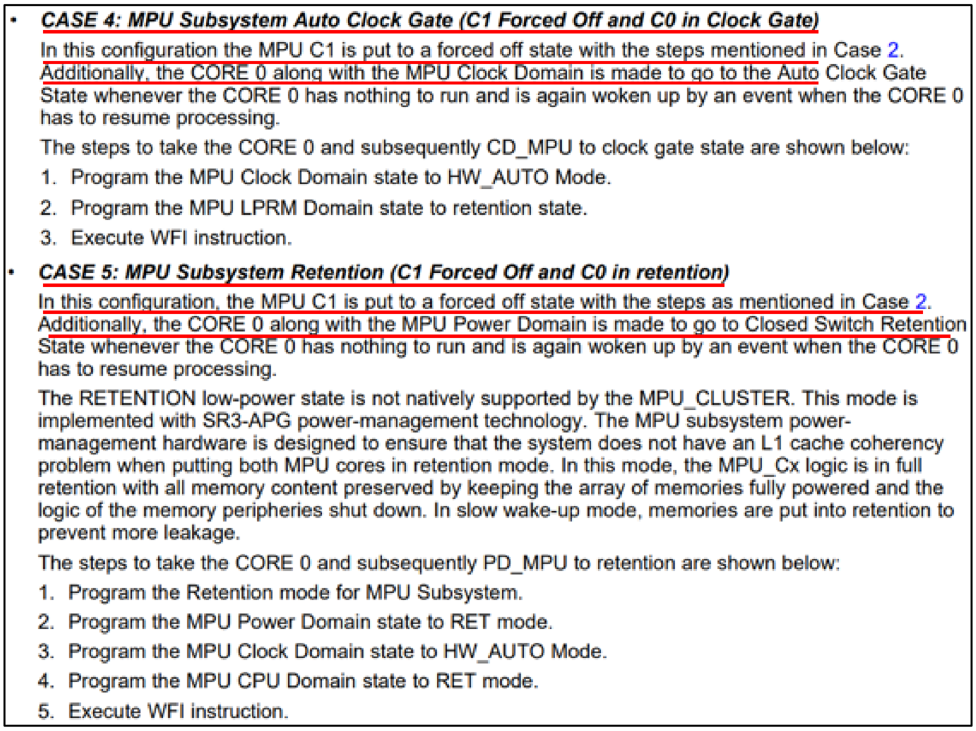
  In this configuration, the MPU C1 is put to a forced off state with the steps as mentioned in Case 2. Additionally the CORE 0 is made to go to IDLE state whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

  The steps to take the CORE 0 to IDLE are shown below:

  1. Program the MPU LPRM Domain state to the desired power state.
  2. Execute WFI instruction.

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Pages 20 and 21)



- ***CASE 4: MPU Subsystem Auto Clock Gate (C1 Forced Off and C0 in Clock Gate)***

  In this configuration the MPU C1 is put to a forced off state with the steps mentioned in Case 2. Additionally, the CORE 0 along with the MPU Clock Domain is made to go to the Auto Clock Gate State whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

  The steps to take the CORE 0 and subsequently CD_MPU to clock gate state are shown below:

  1. Program the MPU Clock Domain state to HW_AUTO Mode.
  2. Program the MPU LPRM Domain state to retention state.
  3. Execute WFI instruction.

- ***CASE 5: MPU Subsystem Retention (C1 Forced Off and C0 in retention)***

  In this configuration, the MPU C1 is put to a forced off state with the steps as mentioned in Case 2. Additionally, the CORE 0 along with the MPU Power Domain is made to go to Closed Switch Retention State whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

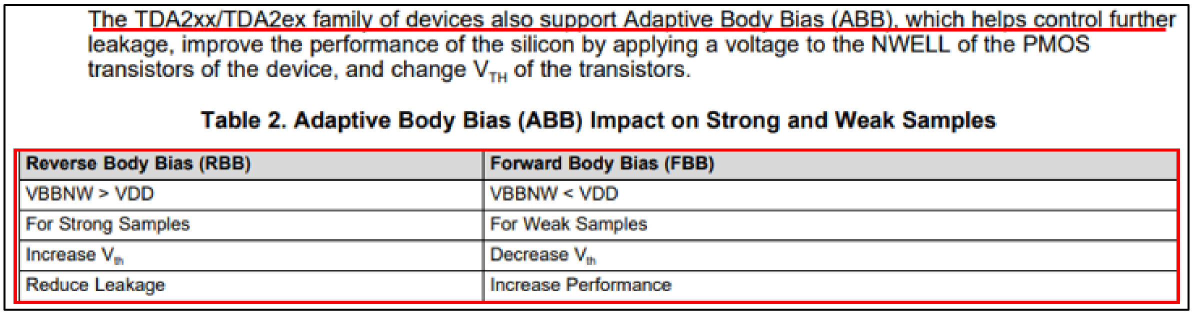
  The RETENTION low-power state is not natively supported by the MPU_CLUSTER. This mode is implemented with SR3-APG power-management technology. The MPU subsystem power-management hardware is designed to ensure that the system does not have an L1 cache coherency problem when putting both MPU cores in retention mode. In this mode, the MPU_Cx logic is in full retention with all memory content preserved by keeping the array of memories fully powered and the logic of the memory peripheries shut down. In slow wake-up mode, memories are put into retention to prevent more leakage.

  The steps to take the CORE 0 and subsequently PD_MPU to retention are shown below:

  1. Program the Retention mode for MPU Subsystem.
  2. Program the MPU Power Domain state to RET mode.
  3. Program the MPU Clock Domain state to HW_AUTO Mode.
  4. Program the MPU CPU Domain state to RET mode.
  5. Execute WFI instruction.

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 23)



The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

**Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples**

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)

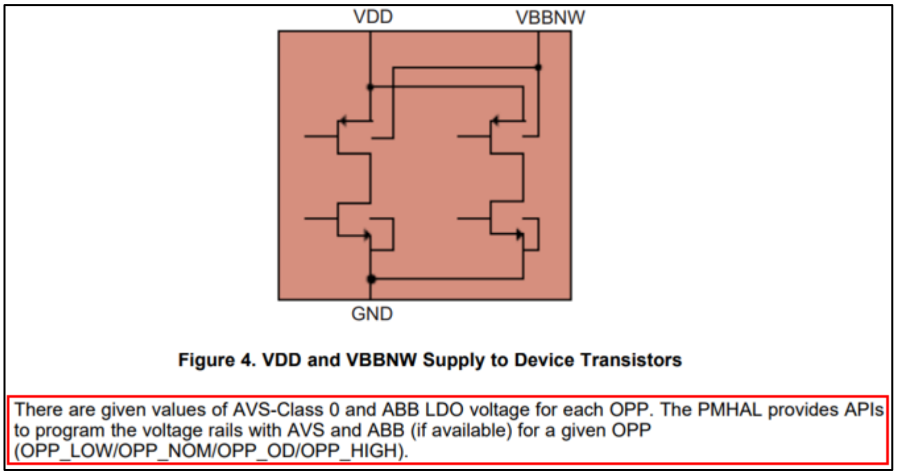
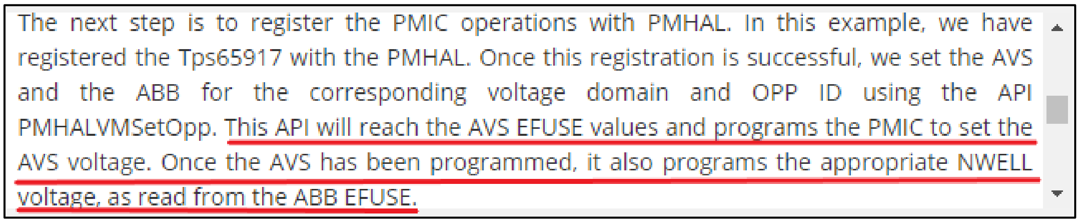


**Figure 4. VDD and VBBNW Supply to Device Transistors**

There are given values of AVS-Class 0 and ABB LDO voltage for each OPP. The PMHAL provides APIs to program the voltage rails with AVS and ABB (if available) for a given OPP (OPP_LOW/OPP_NOM/OPP_OD/OPP_HIGH).

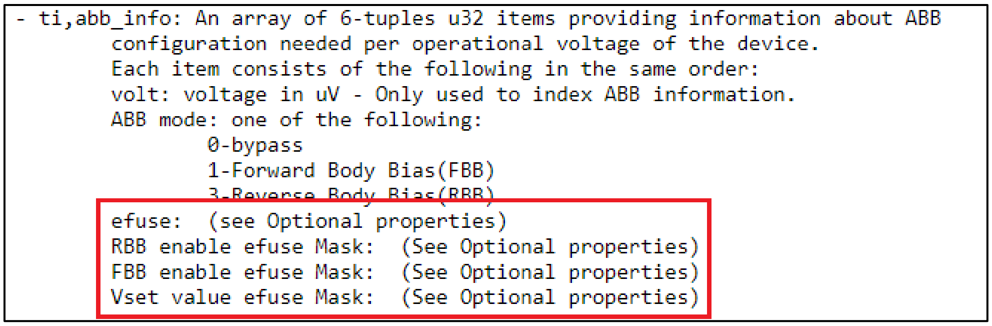
**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F



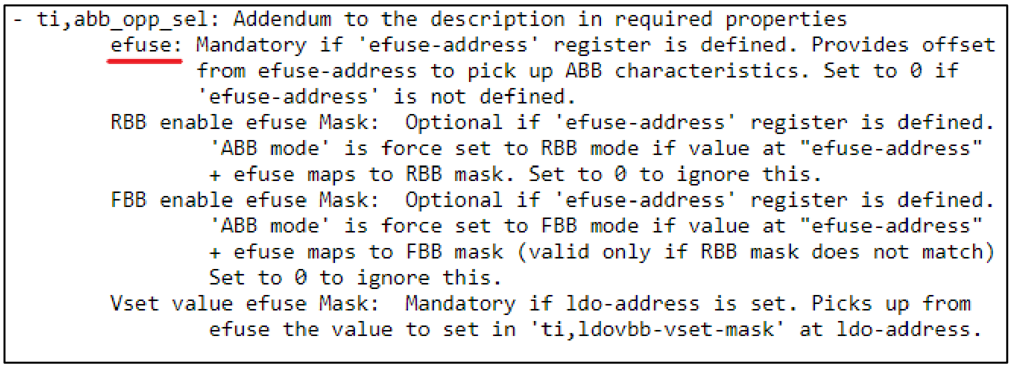
The next step is to register the PMIC operations with PMHAL. In this example, we have registered the Tps65917 with the PMHAL. Once this registration is successful, we set the AVS and the ABB for the corresponding voltage domain and OPP ID using the API PMHALVMSetOpp. This API will reach the AVS EFUSE values and programs the PMIC to set the AVS voltage. Once the AVS has been programmed, it also programs the appropriate NWELL voltage, as read from the ABB EFUSE.

**Source:** https://training.ti.com/introduction-adas-power-management-software (at 17:07)



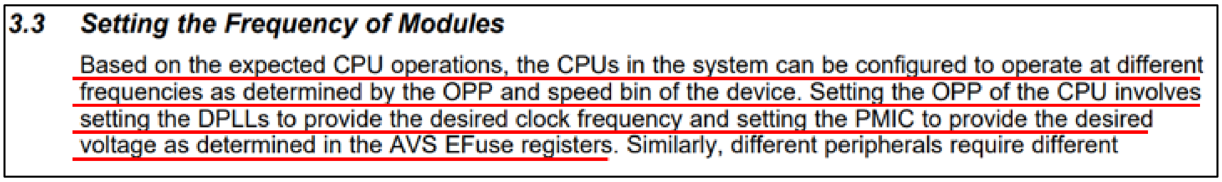
```
- ti,abb_info: An array of 6-tuples u32 items providing information about ABB
        configuration needed per operational voltage of the device.
        Each item consists of the following in the same order:
        volt: voltage in uV - Only used to index ABB information.
        ABB mode: one of the following:
                0-bypass
                1-Forward Body Bias(FBB)
                3-Reverse Body Bias(RBB)
        efuse:  (see Optional properties)
        RBB enable efuse Mask:  (See Optional properties)
        FBB enable efuse Mask:  (See Optional properties)
        Vset value efuse Mask:  (See Optional properties)
```

**Source:** https://www.kernel.org/doc/Documentation/devicetree/bindings/regulator/ti-abb-regulator.txt



```
- ti,abb_opp_sel: Addendum to the description in required properties
        efuse: Mandatory if 'efuse-address' register is defined. Provides offset
               from efuse-address to pick up ABB characteristics. Set to 0 if
               'efuse-address' is not defined.
        RBB enable efuse Mask:  Optional if 'efuse-address' register is defined.
                'ABB mode' is force set to RBB mode if value at "efuse-address"
                + efuse maps to RBB mask. Set to 0 to ignore this.
        FBB enable efuse Mask:  Optional if 'efuse-address' register is defined.
                'ABB mode' is force set to FBB mode if value at "efuse-address"
                + efuse maps to FBB mask (valid only if RBB mask does not match)
                Set to 0 to ignore this.
        Vset value efuse Mask:  Mandatory if ldo-address is set. Picks up from
                efuse the value to set in 'ti,ldovbb-vset-mask' at ldo-address.
```

**Source:** https://www.kernel.org/doc/Documentation/devicetree/bindings/regulator/ti-abb-regulator.txt

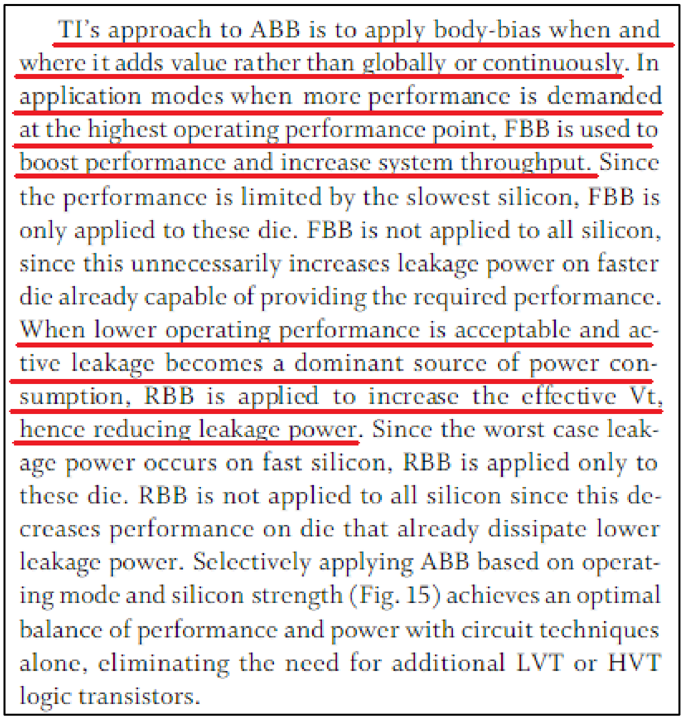


### 3.3 Setting the Frequency of Modules

Based on the expected CPU operations, the CPUs in the system can be configured to operate at different frequencies as determined by the OPP and speed bin of the device. Setting the OPP of the CPU involves setting the DPLLs to provide the desired clock frequency and setting the PMIC to provide the desired voltage as determined in the AVS EFuse registers. Similarly, different peripherals require different

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 9)



TI's approach to ABB is to apply body-bias when and where it adds value rather than globally or continuously. In application modes when more performance is demanded at the highest operating performance point, FBB is used to boost performance and increase system throughput. Since the performance is limited by the slowest silicon, FBB is only applied to these die. FBB is not applied to all silicon, since this unnecessarily increases leakage power on faster die already capable of providing the required performance. When lower operating performance is acceptable and active leakage becomes a dominant source of power consumption, RBB is applied to increase the effective Vt, hence reducing leakage power. Since the worst case leakage power occurs on fast silicon, RBB is applied only to these die. RBB is not applied to all silicon since this decreases performance on die that already dissipate lower leakage power. Selectively applying ABB based on operating mode and silicon strength (Fig. 15) achieves an optimal balance of performance and power with circuit techniques alone, eliminating the need for additional LVT or HVT logic transistors.

**Source:**

https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

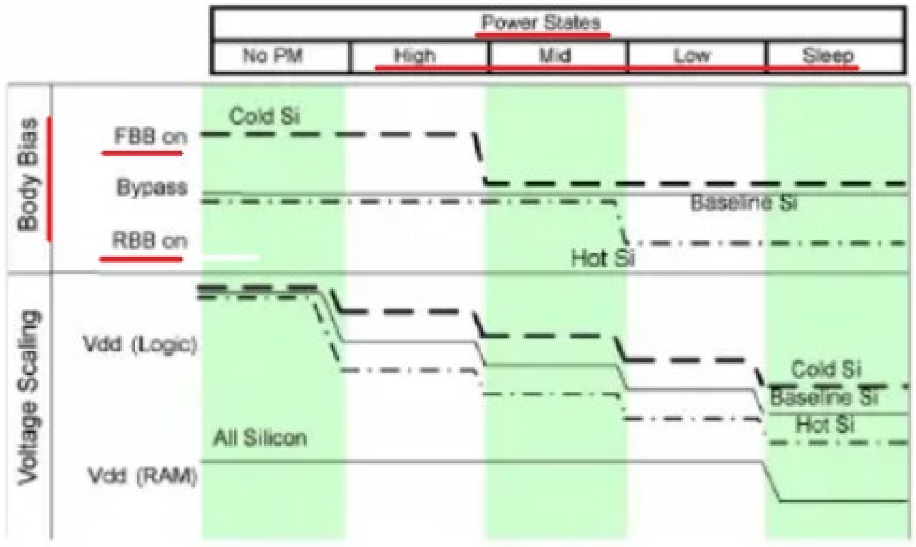
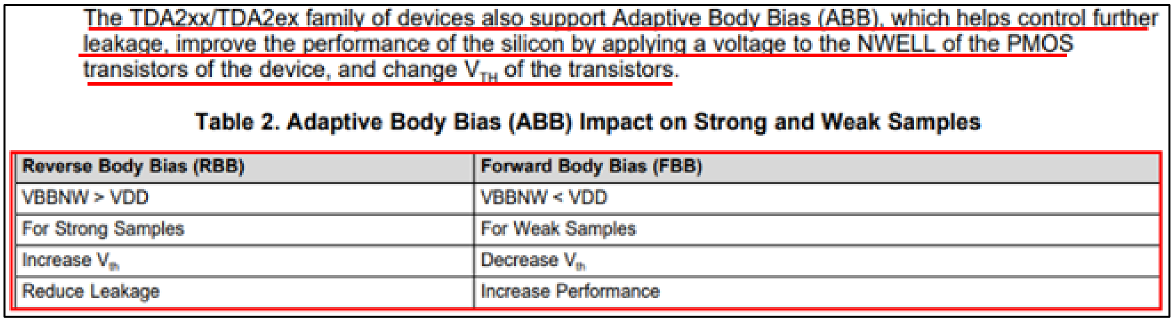


**Fig. 15.** ***Adaptivebody-biasand adaptive voltage scaling configuration with mode and process. Hot and cold refer the process strength or speed (fast and slow).***

**Source:**
https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

26. TI's TDA2x System-on-Chip devices include a voltage supply to generate a body biasing voltage for circuitry of said computer system according to said body biasing voltage information

27. For example, TI's TDA2x System-on-Chip devices include a dual core processor module. The devices support Adaptive Body Bias (ABB). The ABB feature allows application of forward body bias to improve performance and reverse body bias to reduce leakage in the device. A programmed reverse body bias voltage is supplied by the ABB LDO ("voltage supply") to VBBNW pin of the transistors of the device's microprocessor. For example, to reduce leakage in low power states or low operating performance points (OPP Low), the reverse body bias voltage value ("body biasing voltage information") is accessed and applied to the device.



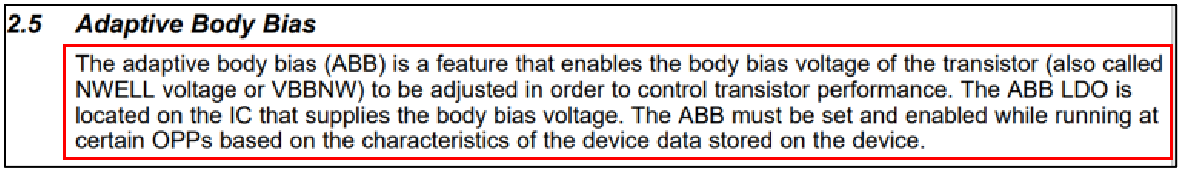
The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

**Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples**

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)



### 2.5 *Adaptive Body Bias*

The adaptive body bias (ABB) is a feature that enables the body bias voltage of the transistor (also called NWELL voltage or VBBNW) to be adjusted in order to control transistor performance. The ABB LDO is located on the IC that supplies the body bias voltage. The ABB must be set and enabled while running at certain OPPs based on the characteristics of the device data stored on the device.

**Source:**

https://www.ti.com/lit/an/slva646/slva646.pdf?ts=1597323545336&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)

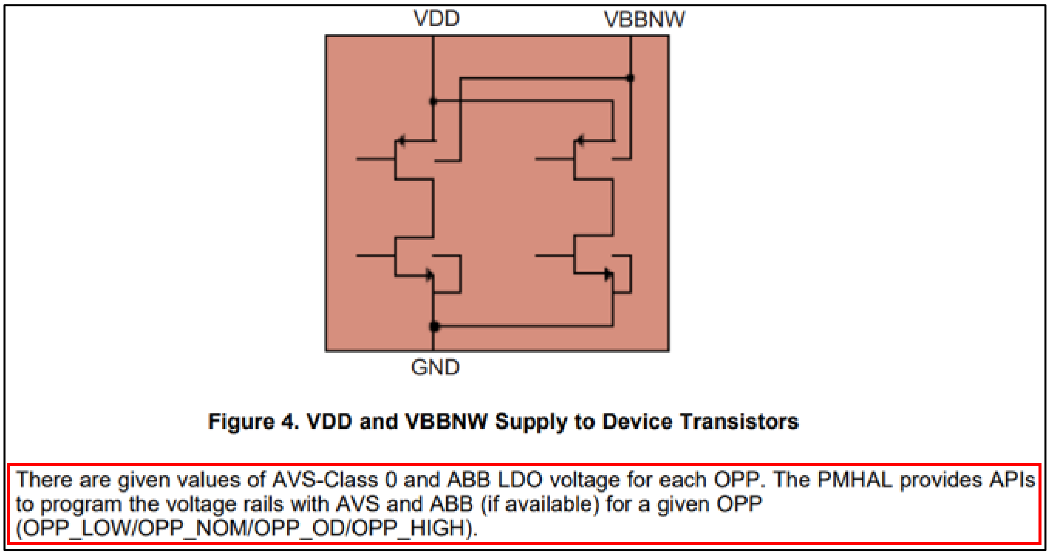
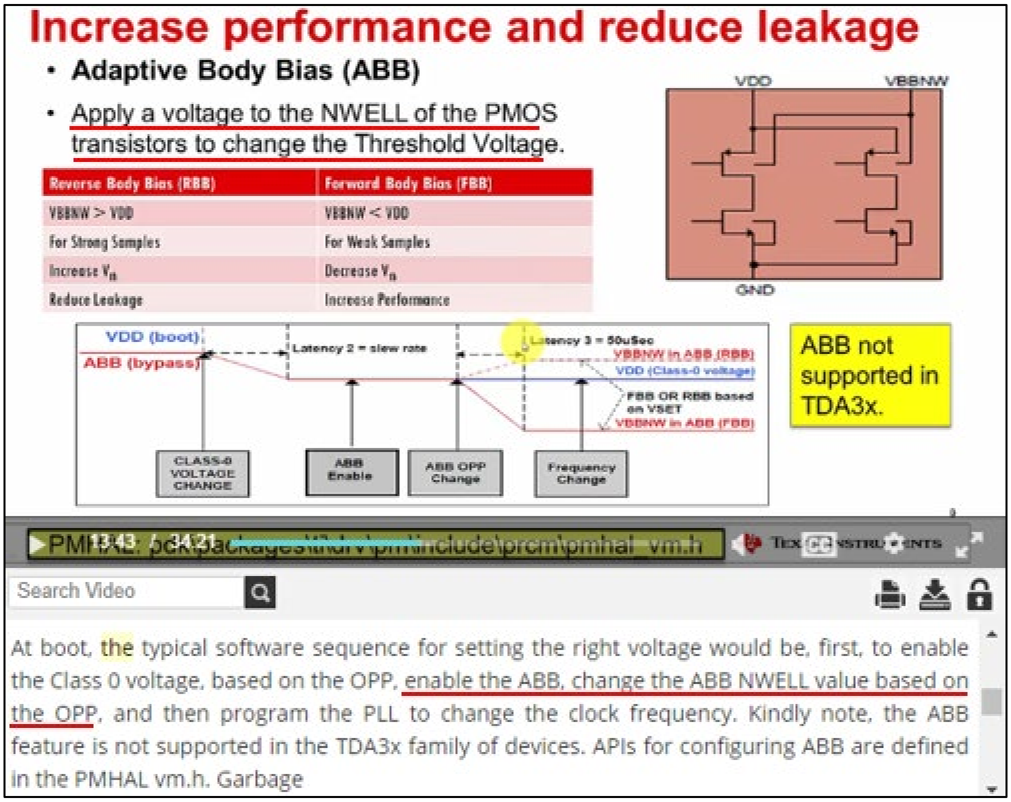


**Figure 4. VDD and VBBNW Supply to Device Transistors**

There are given values of AVS-Class 0 and ABB LDO voltage for each OPP. The PMHAL provides APIs to program the voltage rails with AVS and ABB (if available) for a given OPP (OPP_LOW/OPP_NOM/OPP_OD/OPP_HIGH).

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



## Increase performance and reduce leakage

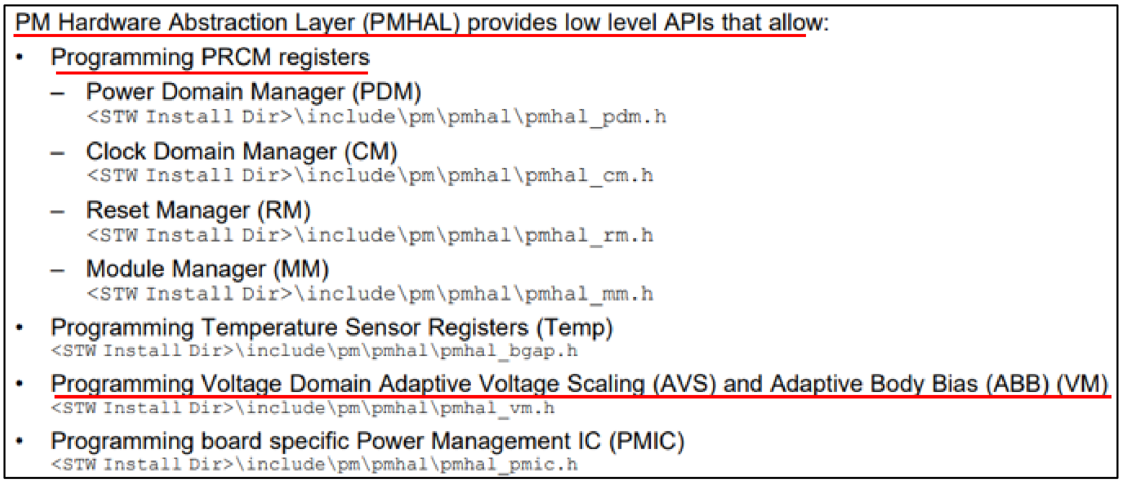
- **Adaptive Body Bias (ABB)**
- Apply a voltage to the NWELL of the PMOS transistors to change the Threshold Voltage.

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

ABB not supported in TDA3x.

At boot, the typical software sequence for setting the right voltage would be, first, to enable the Class 0 voltage, based on the OPP, enable the ABB, change the ABB NWELL value based on the OPP, and then program the PLL to change the clock frequency. Kindly note, the ABB feature is not supported in the TDA3x family of devices. APIs for configuring ABB are defined in the PMHAL vm.h. Garbage

**Source:** https://training.ti.com/introduction-adas-power-management-software?cu=1137175 (at 13:43)

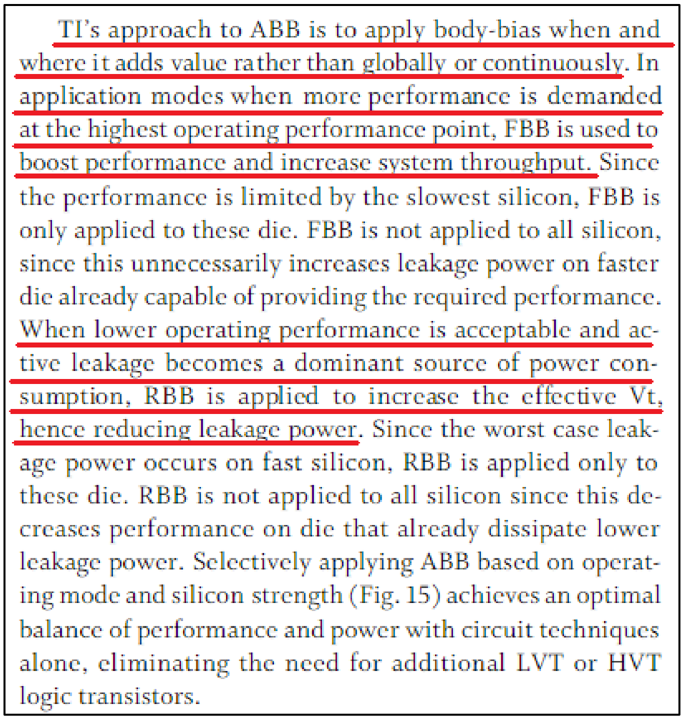


PM Hardware Abstraction Layer (PMHAL) provides low level APIs that allow:

- Programming PRCM registers
  - Power Domain Manager (PDM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_pdm.h`
  - Clock Domain Manager (CM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_cm.h`
  - Reset Manager (RM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_rm.h`
  - Module Manager (MM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_mm.h`
- Programming Temperature Sensor Registers (Temp)
  `<STW Install Dir>\include\pm\pmhal\pmhal_bgap.h`
- Programming Voltage Domain Adaptive Voltage Scaling (AVS) and Adaptive Body Bias (ABB) (VM)
  `<STW Install Dir>\include\pm\pmhal\pmhal_vm.h`
- Programming board specific Power Management IC (PMIC)
  `<STW Install Dir>\include\pm\pmhal\pmhal_pmic.h`

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)



TI's approach to ABB is to apply body-bias when and where it adds value rather than globally or continuously. In application modes when more performance is demanded at the highest operating performance point, FBB is used to boost performance and increase system throughput. Since the performance is limited by the slowest silicon, FBB is only applied to these die. FBB is not applied to all silicon, since this unnecessarily increases leakage power on faster die already capable of providing the required performance. When lower operating performance is acceptable and active leakage becomes a dominant source of power consumption, RBB is applied to increase the effective Vt, hence reducing leakage power. Since the worst case leakage power occurs on fast silicon, RBB is applied only to these die. RBB is not applied to all silicon since this decreases performance on die that already dissipate lower leakage power. Selectively applying ABB based on operating mode and silicon strength (Fig. 15) achieves an optimal balance of performance and power with circuit techniques alone, eliminating the need for additional LVT or HVT logic transistors.

**Source:**

https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

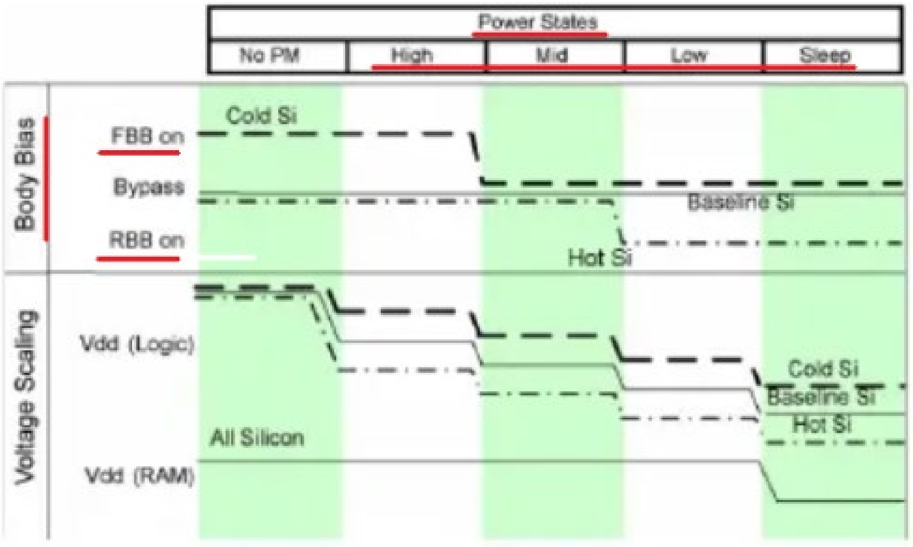


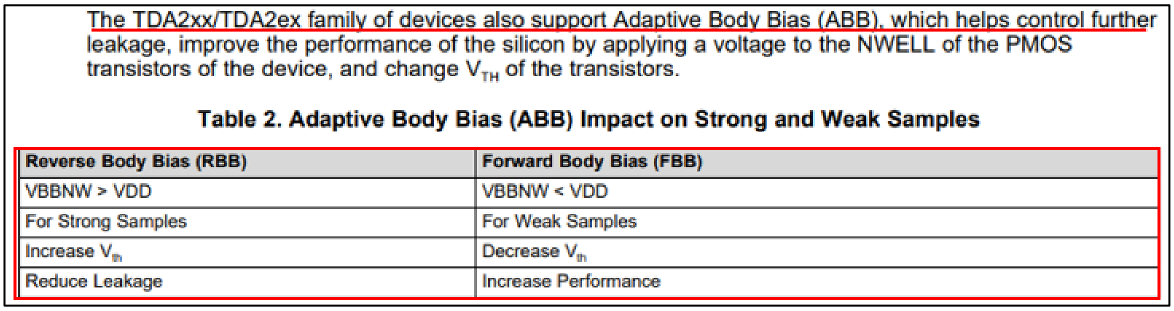
***Fig. 15. Adaptivebody-biasand adaptive voltage scaling configuration with mode and process. Hot and cold refer the process strength or speed (fast and slow).***

**Source:**
https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

28. TI's TDA2x System-on-Chip devices include circuitry to selectively couple the body biasing voltage information from one of the plurality of data storage locations corresponding to one of the plurality of power modes to the voltage supply.

29. For example, TI's TDA2x System-on-Chip devices include a dual core processor module. The devices support Adaptive Body Bias (ABB). A body bias voltage is applied to the transistors of the device using ABB. ABB voltage values are available for each OPP (Operating Performance Point), and therefore, for each power state. The ABB voltage values corresponding to the different power states are stored in Efuse registers ("data storage locations"). Further, the body bias voltage value is supplied by the ABB LDO ("voltage supply") to the VBBNW pin of the transistors of the device's microprocessor. For example, to reduce leakage in low power states or low operating performance points (OPP Low), the reverse body bias voltage value

("body biasing voltage information") is accessed and applied to the device. Accordingly, the devices include circuitry for selectively coupling the body bias voltage information stored in the registers to the ABB LDO voltage supply.

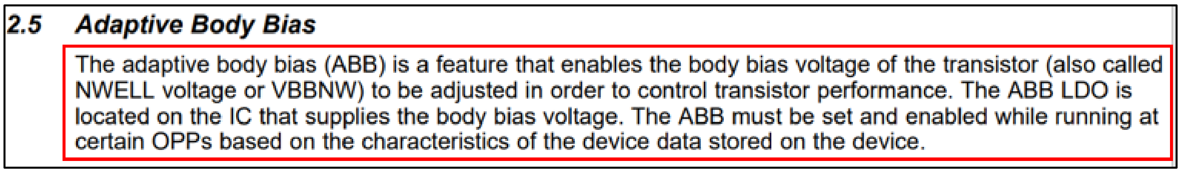


The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

**Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples**

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:**
https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)



**2.5 *Adaptive Body Bias***

The adaptive body bias (ABB) is a feature that enables the body bias voltage of the transistor (also called NWELL voltage or VBBNW) to be adjusted in order to control transistor performance. The ABB LDO is located on the IC that supplies the body bias voltage. The ABB must be set and enabled while running at certain OPPs based on the characteristics of the device data stored on the device.

**Source:**
https://www.ti.com/lit/an/slva646/slva646.pdf?ts=1597323545336&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)

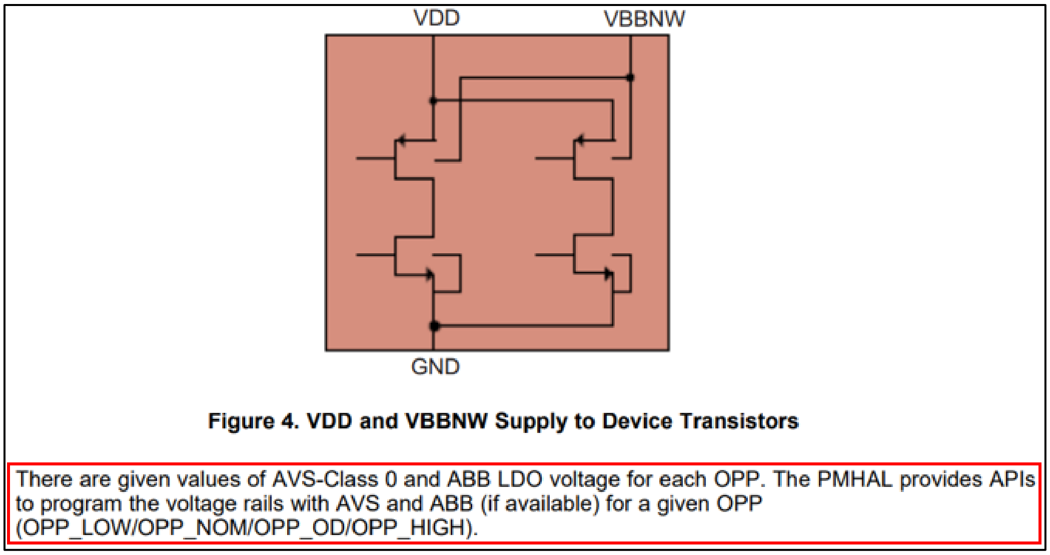
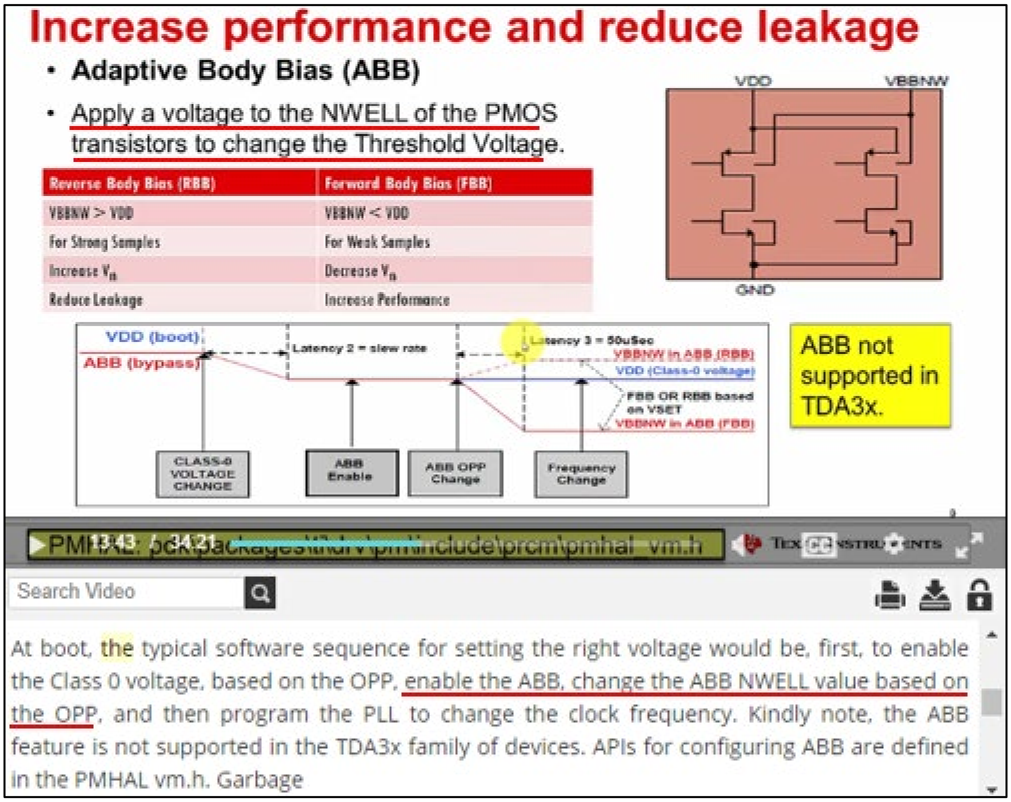

VDD
VBBNW
GND

**Figure 4. VDD and VBBNW Supply to Device Transistors**

There are given values of AVS-Class 0 and ABB LDO voltage for each OPP. The PMHAL provides APIs to program the voltage rails with AVS and ABB (if available) for a given OPP (OPP_LOW/OPP_NOM/OPP_OD/OPP_HIGH).

**Source:**
https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



## Increase performance and reduce leakage

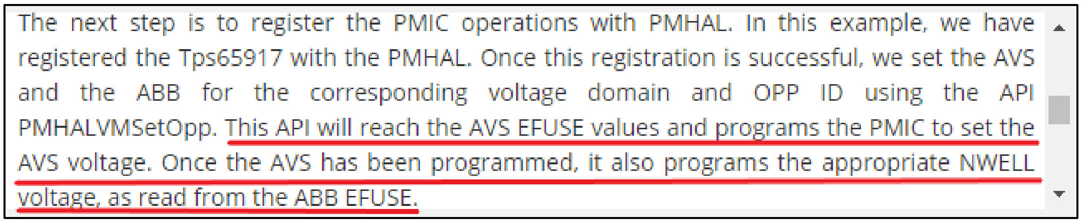
- **Adaptive Body Bias (ABB)**
- Apply a voltage to the NWELL of the PMOS transistors to change the Threshold Voltage.

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

ABB not supported in TDA3x.

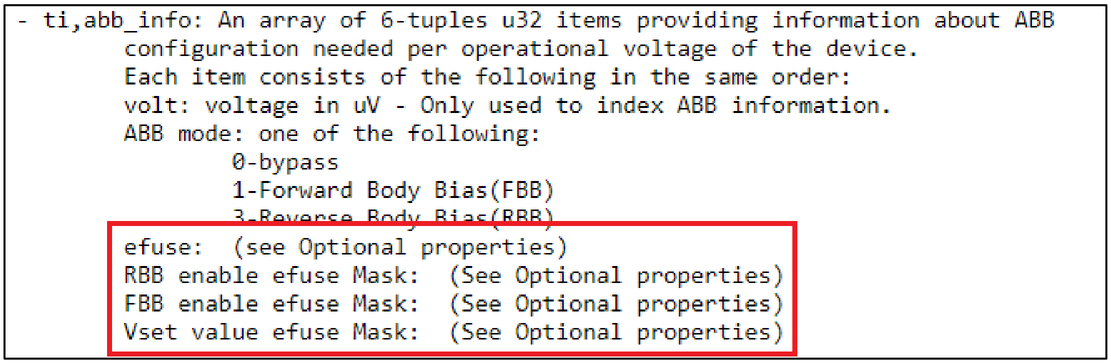
At boot, the typical software sequence for setting the right voltage would be, first, to enable the Class 0 voltage, based on the OPP, enable the ABB, change the ABB NWELL value based on the OPP, and then program the PLL to change the clock frequency. Kindly note, the ABB feature is not supported in the TDA3x family of devices. APIs for configuring ABB are defined in the PMHAL vm.h. Garbage

**Source:** https://training.ti.com/introduction-adas-power-management-software?cu=1137175 (at 13:43)



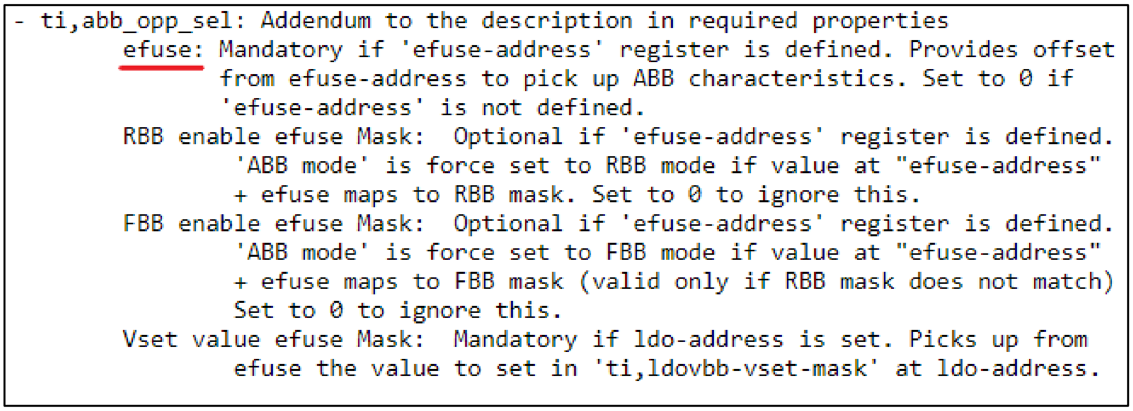
The next step is to register the PMIC operations with PMHAL. In this example, we have registered the Tps65917 with the PMHAL. Once this registration is successful, we set the AVS and the ABB for the corresponding voltage domain and OPP ID using the API PMHALVMSetOpp. This API will reach the AVS EFUSE values and programs the PMIC to set the AVS voltage. Once the AVS has been programmed, it also programs the appropriate NWELL voltage, as read from the ABB EFUSE.

**Source:** https://training.ti.com/introduction-adas-power-management-software (at 17:07)



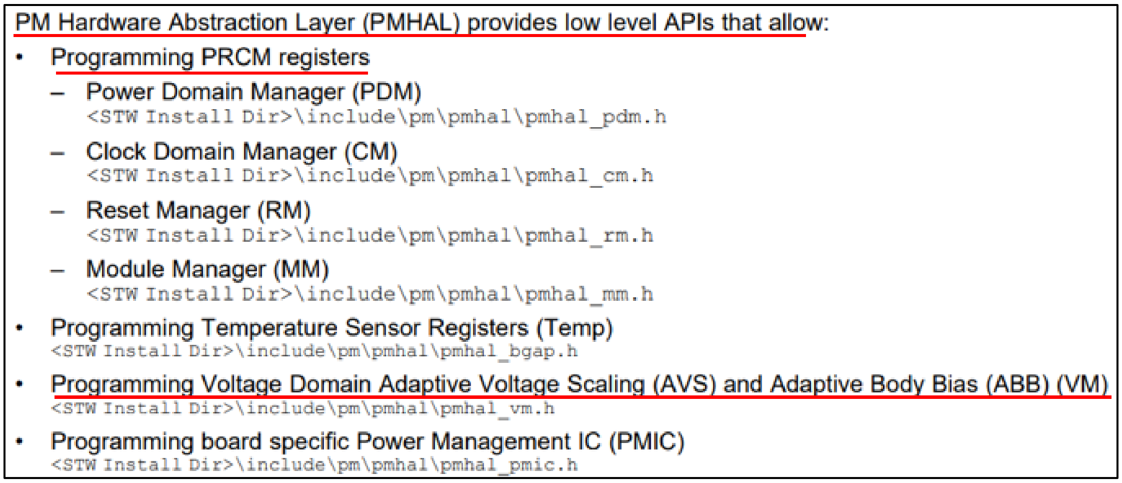
```
- ti,abb_info: An array of 6-tuples u32 items providing information about ABB
        configuration needed per operational voltage of the device.
        Each item consists of the following in the same order:
        volt: voltage in uV - Only used to index ABB information.
        ABB mode: one of the following:
                0-bypass
                1-Forward Body Bias(FBB)
                3-Reverse Body Bias(RBB)
        efuse:  (see Optional properties)
        RBB enable efuse Mask:  (See Optional properties)
        FBB enable efuse Mask:  (See Optional properties)
        Vset value efuse Mask:  (See Optional properties)
```

**Source:** https://www.kernel.org/doc/Documentation/devicetree/bindings/regulator/ti-abb-regulator.txt



```
- ti,abb_opp_sel: Addendum to the description in required properties
        efuse: Mandatory if 'efuse-address' register is defined. Provides offset
               from efuse-address to pick up ABB characteristics. Set to 0 if
               'efuse-address' is not defined.
        RBB enable efuse Mask:  Optional if 'efuse-address' register is defined.
                'ABB mode' is force set to RBB mode if value at "efuse-address"
                + efuse maps to RBB mask. Set to 0 to ignore this.
        FBB enable efuse Mask:  Optional if 'efuse-address' register is defined.
                'ABB mode' is force set to FBB mode if value at "efuse-address"
                + efuse maps to FBB mask (valid only if RBB mask does not match)
                Set to 0 to ignore this.
        Vset value efuse Mask:  Mandatory if ldo-address is set. Picks up from
                efuse the value to set in 'ti,ldovbb-vset-mask' at ldo-address.
```

**Source:** https://www.kernel.org/doc/Documentation/devicetree/bindings/regulator/ti-abb-regulator.txt

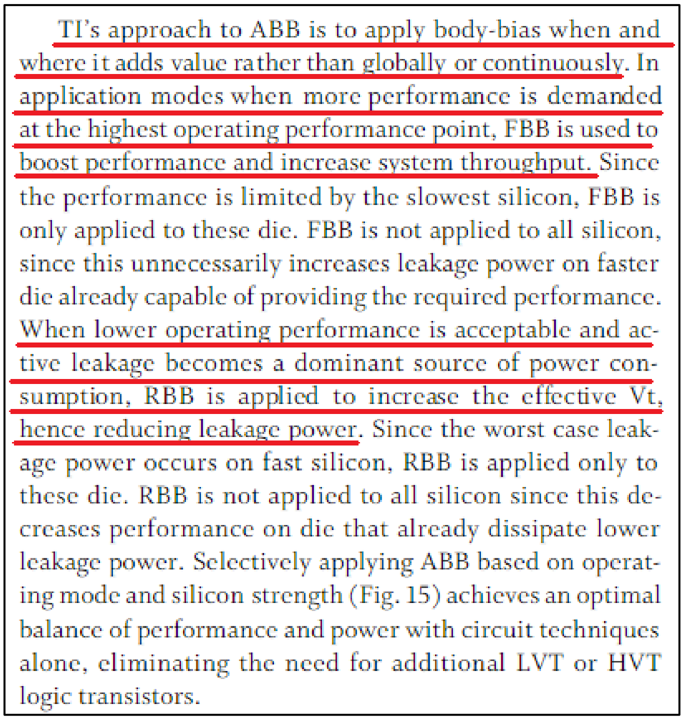


PM Hardware Abstraction Layer (PMHAL) provides low level APIs that allow:

- Programming PRCM registers
  - Power Domain Manager (PDM)
    <STW Install Dir>\include\pm\pmhal\pmhal_pdm.h
  - Clock Domain Manager (CM)
    <STW Install Dir>\include\pm\pmhal\pmhal_cm.h
  - Reset Manager (RM)
    <STW Install Dir>\include\pm\pmhal\pmhal_rm.h
  - Module Manager (MM)
    <STW Install Dir>\include\pm\pmhal\pmhal_mm.h
- Programming Temperature Sensor Registers (Temp)
  <STW Install Dir>\include\pm\pmhal\pmhal_bgap.h
- Programming Voltage Domain Adaptive Voltage Scaling (AVS) and Adaptive Body Bias (ABB) (VM)
  <STW Install Dir>\include\pm\pmhal\pmhal_vm.h
- Programming board specific Power Management IC (PMIC)
  <STW Install Dir>\include\pm\pmhal\pmhal_pmic.h

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)



TI's approach to ABB is to apply body-bias when and where it adds value rather than globally or continuously. In application modes when more performance is demanded at the highest operating performance point, FBB is used to boost performance and increase system throughput. Since the performance is limited by the slowest silicon, FBB is only applied to these die. FBB is not applied to all silicon, since this unnecessarily increases leakage power on faster die already capable of providing the required performance. When lower operating performance is acceptable and active leakage becomes a dominant source of power consumption, RBB is applied to increase the effective Vt, hence reducing leakage power. Since the worst case leakage power occurs on fast silicon, RBB is applied only to these die. RBB is not applied to all silicon since this decreases performance on die that already dissipate lower leakage power. Selectively applying ABB based on operating mode and silicon strength (Fig. 15) achieves an optimal balance of performance and power with circuit techniques alone, eliminating the need for additional LVT or HVT logic transistors.

**Source:**

https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

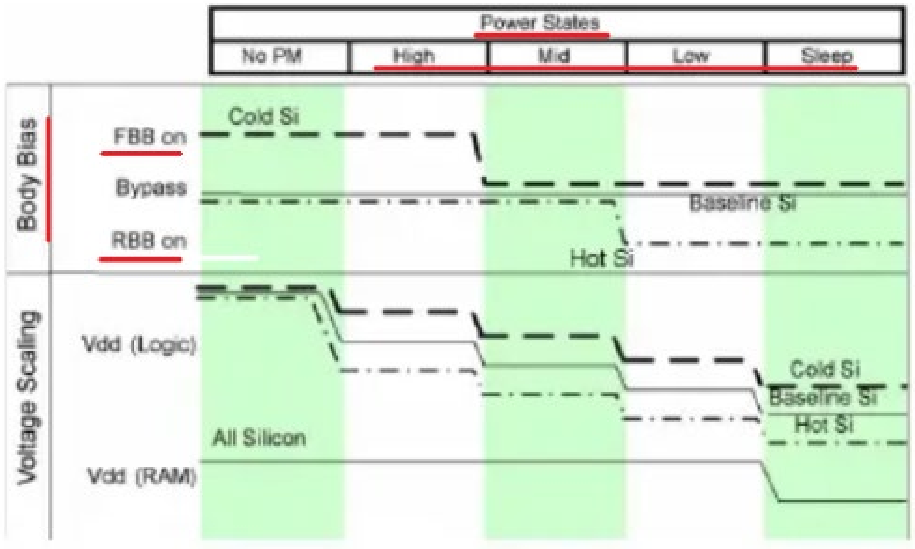
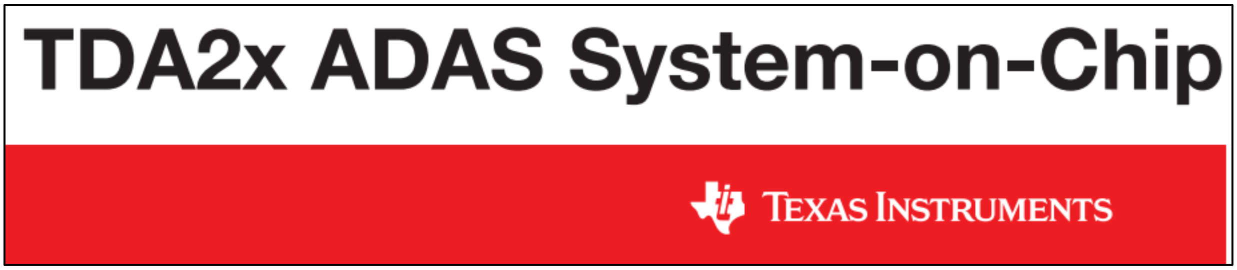


**Fig. 15.** ***Adaptivebody-biasand adaptive voltage scaling configuration with mode and process. Hot and cold refer the process strength or speed (fast and slow).***

**Source:**
https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

30. TI has infringed the ’504 Patent by using the accused products and thereby practicing a method for determining a body biasing voltage applied to a microprocessor.

31. For example, TI’s TDA2x System-on-Chip devices include a dual core processor module (“microprocessor”). The devices support Adaptive Body Bias (ABB), which can apply a forward or reverse body bias voltage to the transistors of the device’s microprocessor using ABB.

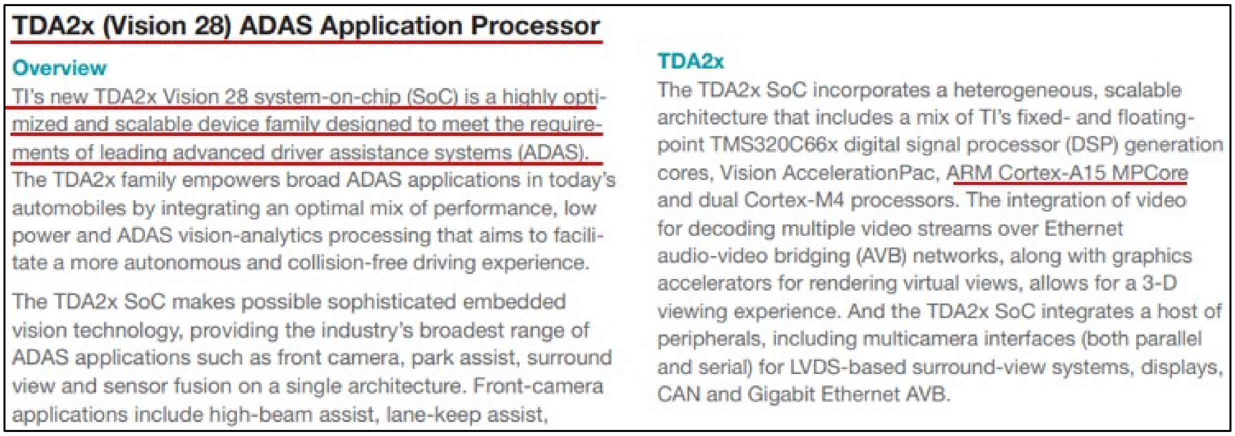


TDA2x ADAS System-on-Chip

Texas Instruments

**Source:**

https://www.ti.com/lit/ml/sprt681/sprt681.pdf?ts=1597386693056&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 1)



TDA2x (Vision 28) ADAS Application Processor

Overview

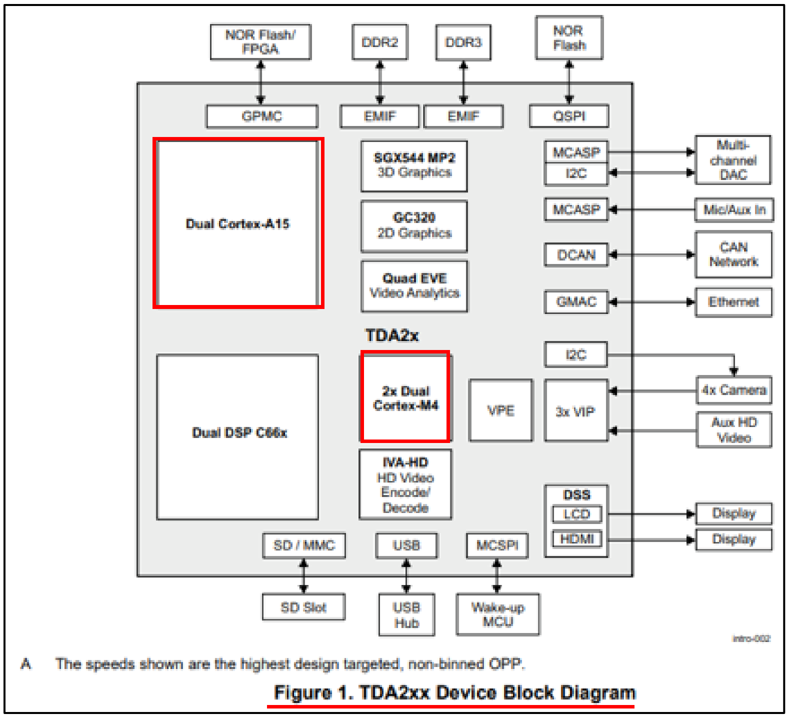
TI's new TDA2x Vision 28 system-on-chip (SoC) is a highly optimized and scalable device family designed to meet the requirements of leading advanced driver assistance systems (ADAS). The TDA2x family empowers broad ADAS applications in today's automobiles by integrating an optimal mix of performance, low power and ADAS vision-analytics processing that aims to facilitate a more autonomous and collision-free driving experience.

The TDA2x SoC makes possible sophisticated embedded vision technology, providing the industry's broadest range of ADAS applications such as front camera, park assist, surround view and sensor fusion on a single architecture. Front-camera applications include high-beam assist, lane-keep assist,

TDA2x

The TDA2x SoC incorporates a heterogeneous, scalable architecture that includes a mix of TI's fixed- and floating-point TMS320C66x digital signal processor (DSP) generation cores, Vision AccelerationPac, ARM Cortex-A15 MPCore and dual Cortex-M4 processors. The integration of video for decoding multiple video streams over Ethernet audio-video bridging (AVB) networks, along with graphics accelerators for rendering virtual views, allows for a 3-D viewing experience. And the TDA2x SoC integrates a host of peripherals, including multicamera interfaces (both parallel and serial) for LVDS-based surround-view systems, displays, CAN and Gigabit Ethernet AVB.

**Source:** https://m.eet.com/media/1241324/slyy044.pdf (Page 5)

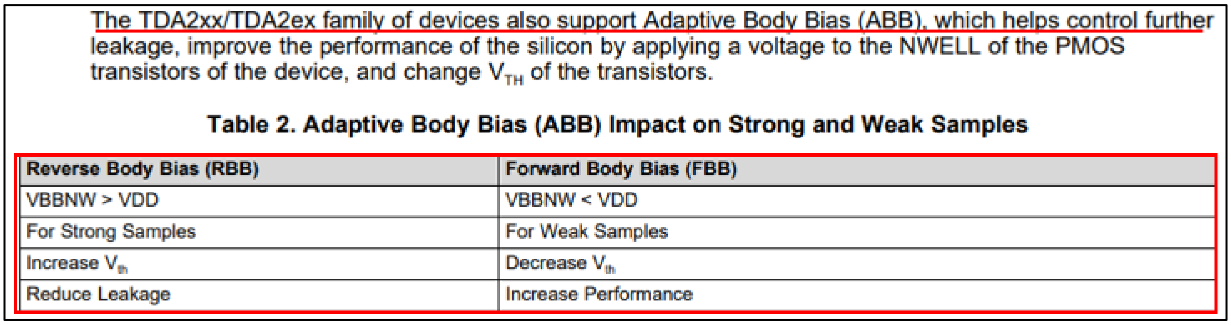


A The speeds shown are the highest design targeted, non-binned OPP.

Figure 1. TDA2xx Device Block Diagram

**Source:**

https://www.ti.com/lit/an/sprac21a/sprac21a.pdf?ts=1597386703137&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

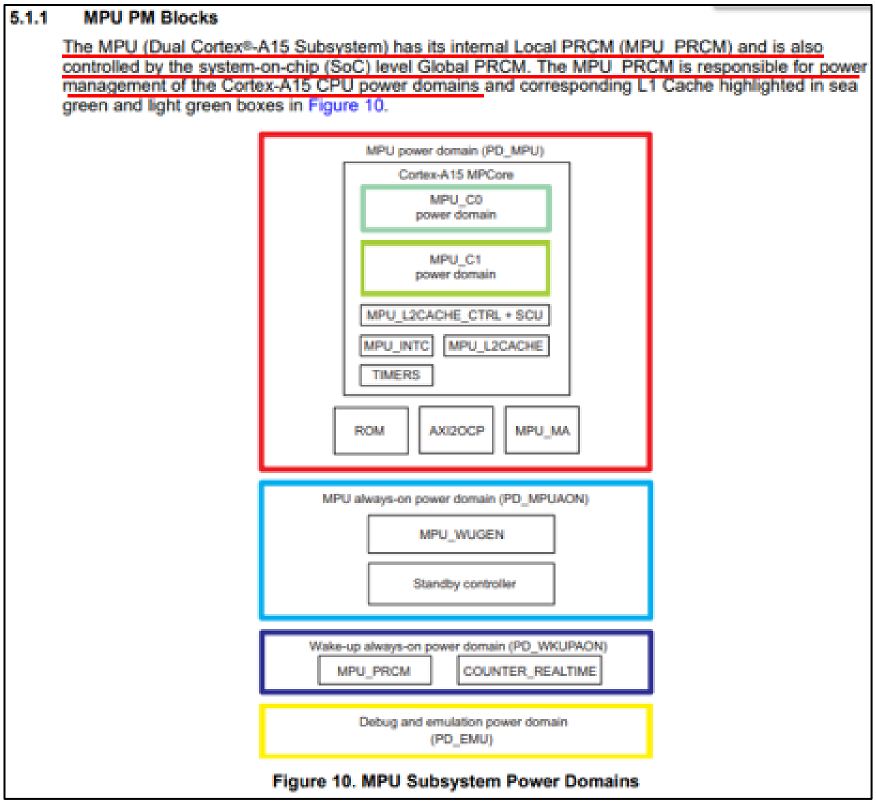
**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)

32. The method practiced using the accused products comprises receiving a command to change to a different power condition of a computer system comprising the microprocessor. The power condition comprises a different microprocessor clock frequency and/or a different microprocessor operating voltage.

33. For example, TI's TDA2x System-on-Chip devices include a dual core processor module. Further, the System-on-Chip ("computer system") includes an MPU_PRCM module for power management of the processor ("microprocessor"). The TDA2x family of devices supports multiple power states (e.g., a power state in which both cores are on, a power state in which one core is forced off, a power state in which one core is forced off and the other core is in retention, etc.).

34. The desired power state can be achieved through programming—that is, a command can be received by the device to change from one power state to a different power state. For example, the system can receive a command to change from a power state in which both cores are on to a power state in which one core is forced off and the other core is in retention ("said different power condition"). A power state changes based on the processor's performance, and each power state is associated with a different clock frequency and/or a different voltage.



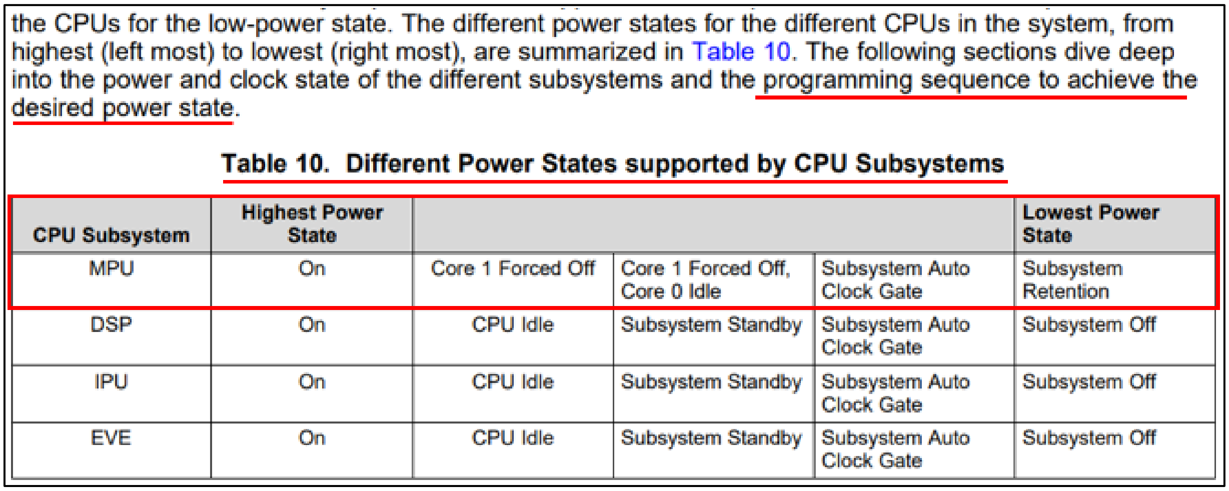
**5.1.1 MPU PM Blocks**

The MPU (Dual Cortex®-A15 Subsystem) has its internal Local PRCM (MPU_PRCM) and is also controlled by the system-on-chip (SoC) level Global PRCM. The MPU_PRCM is responsible for power management of the Cortex-A15 CPU power domains and corresponding L1 Cache highlighted in sea green and light green boxes in Figure 10.

**Figure 10. MPU Subsystem Power Domains**

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 19)



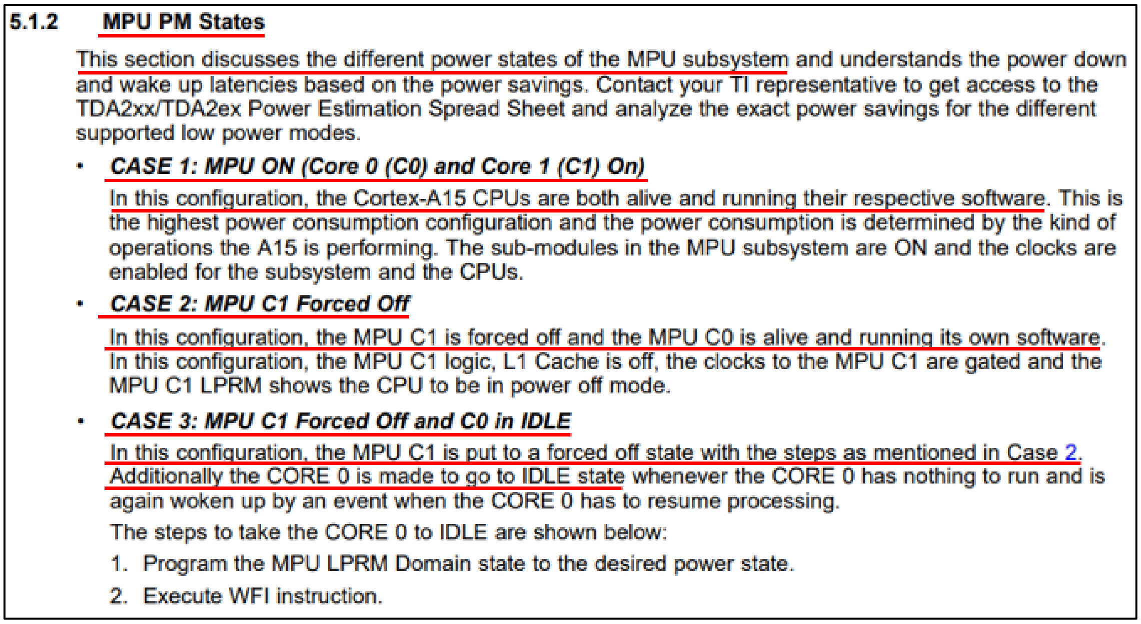
the CPUs for the low-power state. The different power states for the different CPUs in the system, from highest (left most) to lowest (right most), are summarized in Table 10. The following sections dive deep into the power and clock state of the different subsystems and the programming sequence to achieve the desired power state.

**Table 10. Different Power States supported by CPU Subsystems**

| CPU Subsystem | Highest Power State | | | | Lowest Power State |
|---|---|---|---|---|---|
| MPU | On | Core 1 Forced Off | Core 1 Forced Off, Core 0 Idle | Subsystem Auto Clock Gate | Subsystem Retention |
| DSP | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| IPU | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| EVE | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 18)



5.1.2 **MPU PM States**

This section discusses the different power states of the MPU subsystem and understands the power down and wake up latencies based on the power savings. Contact your TI representative to get access to the TDA2xx/TDA2ex Power Estimation Spread Sheet and analyze the exact power savings for the different supported low power modes.

- ***CASE 1: MPU ON (Core 0 (C0) and Core 1 (C1) On)***

  In this configuration, the Cortex-A15 CPUs are both alive and running their respective software. This is the highest power consumption configuration and the power consumption is determined by the kind of operations the A15 is performing. The sub-modules in the MPU subsystem are ON and the clocks are enabled for the subsystem and the CPUs.

- ***CASE 2: MPU C1 Forced Off***

  In this configuration, the MPU C1 is forced off and the MPU C0 is alive and running its own software. In this configuration, the MPU C1 logic, L1 Cache is off, the clocks to the MPU C1 are gated and the MPU C1 LPRM shows the CPU to be in power off mode.

- ***CASE 3: MPU C1 Forced Off and C0 in IDLE***

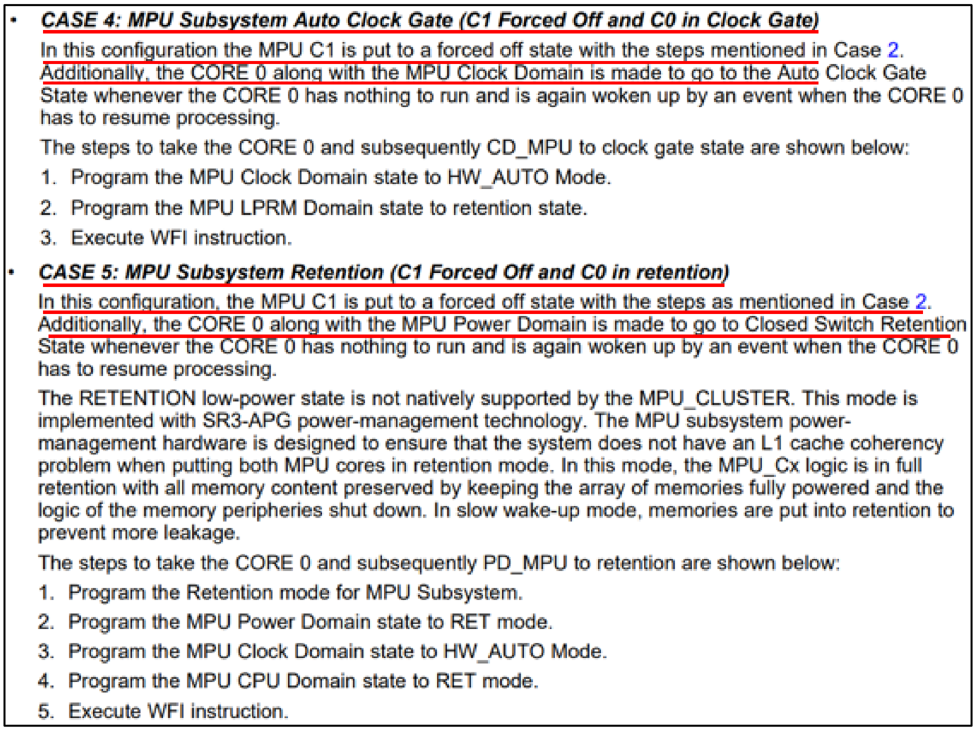
  In this configuration, the MPU C1 is put to a forced off state with the steps as mentioned in Case 2. Additionally the CORE 0 is made to go to IDLE state whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

  The steps to take the CORE 0 to IDLE are shown below:

  1. Program the MPU LPRM Domain state to the desired power state.
  2. Execute WFI instruction.

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Pages 20 and 21)



- ***CASE 4: MPU Subsystem Auto Clock Gate (C1 Forced Off and C0 in Clock Gate)***

  In this configuration the MPU C1 is put to a forced off state with the steps mentioned in Case 2. Additionally, the CORE 0 along with the MPU Clock Domain is made to go to the Auto Clock Gate State whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

  The steps to take the CORE 0 and subsequently CD_MPU to clock gate state are shown below:

  1. Program the MPU Clock Domain state to HW_AUTO Mode.
  2. Program the MPU LPRM Domain state to retention state.
  3. Execute WFI instruction.

- ***CASE 5: MPU Subsystem Retention (C1 Forced Off and C0 in retention)***

  In this configuration, the MPU C1 is put to a forced off state with the steps as mentioned in Case 2. Additionally, the CORE 0 along with the MPU Power Domain is made to go to Closed Switch Retention State whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

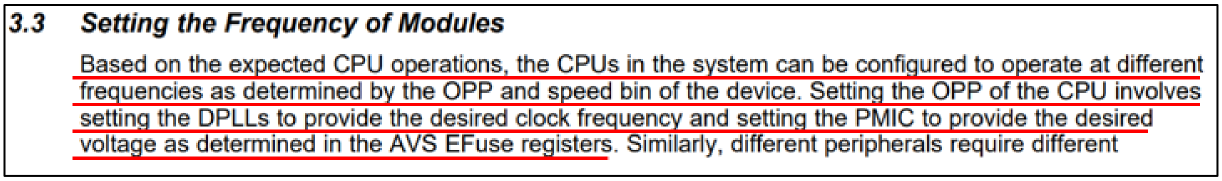
  The RETENTION low-power state is not natively supported by the MPU_CLUSTER. This mode is implemented with SR3-APG power-management technology. The MPU subsystem power-management hardware is designed to ensure that the system does not have an L1 cache coherency problem when putting both MPU cores in retention mode. In this mode, the MPU_Cx logic is in full retention with all memory content preserved by keeping the array of memories fully powered and the logic of the memory peripheries shut down. In slow wake-up mode, memories are put into retention to prevent more leakage.

  The steps to take the CORE 0 and subsequently PD_MPU to retention are shown below:

  1. Program the Retention mode for MPU Subsystem.
  2. Program the MPU Power Domain state to RET mode.
  3. Program the MPU Clock Domain state to HW_AUTO Mode.
  4. Program the MPU CPU Domain state to RET mode.
  5. Execute WFI instruction.

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 23)



**3.3 Setting the Frequency of Modules**

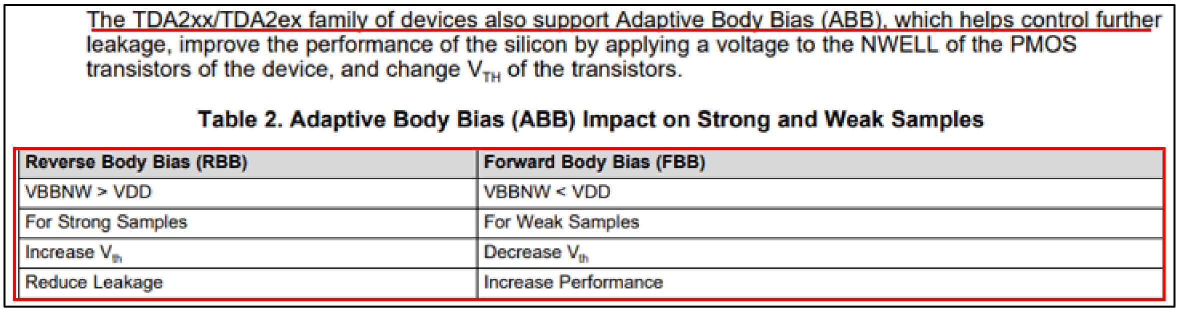
Based on the expected CPU operations, the CPUs in the system can be configured to operate at different frequencies as determined by the OPP and speed bin of the device. Setting the OPP of the CPU involves setting the DPLLs to provide the desired clock frequency and setting the PMIC to provide the desired voltage as determined in the AVS EFuse registers. Similarly, different peripherals require different

**Source:**
https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 9)

35. The method practiced using the accused products further comprises accessing body biasing voltage information corresponding to the power condition.

36. For example, TI's TDA2x System-on-Chip devices support Adaptive Body Bias (ABB). ABB voltage values are given for each OPP (Operating Performance Point), which is the operating condition for the device and is defined by the voltage and frequency applied to the processor. Further, the TDA2x family of devices supports multiple power states (e.g., a power state in which both cores are on, a power state in which one core is forced off, a power state in which one core is forced off and other core is in retention, etc.). An OPP is established based on the power state (i.e., operation) of the CPU.

37. Accordingly, the TDA2x family of devices stores the ABB voltage values associated with each OPP and corresponding power state. The ABB feature allows application of forward body bias to improve performance and reverse body bias to reduce leakage in the device. To reduce leakage in low power states or low operating performance points (OPP Low), the reverse body bias voltage value ("body biasing voltage information") is accessed and applied to the device. This occurs during one of the low power states of the device, such as during the power state in which one core is forced off and the other core is in retention ("said power condition").



The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

**Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples**

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)

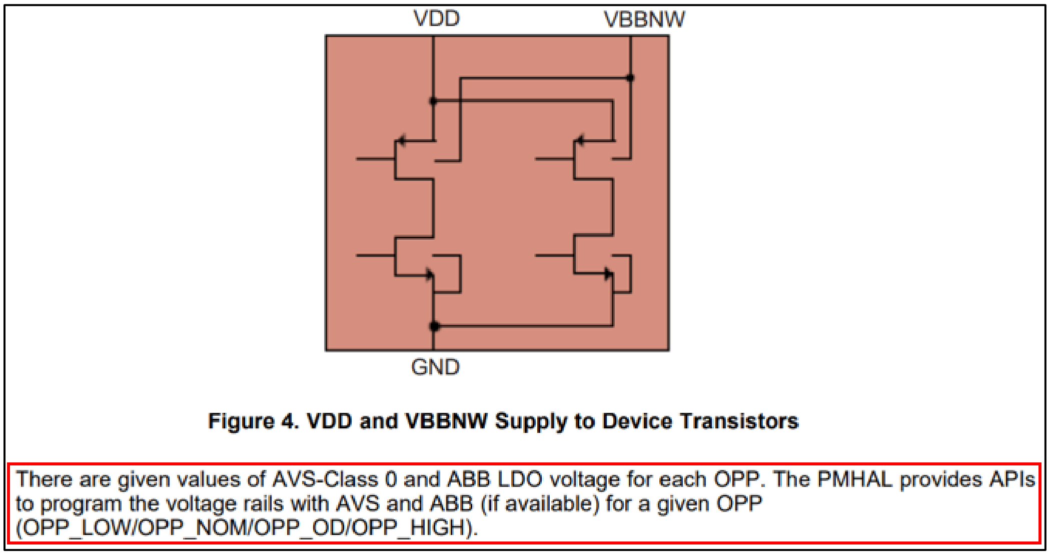
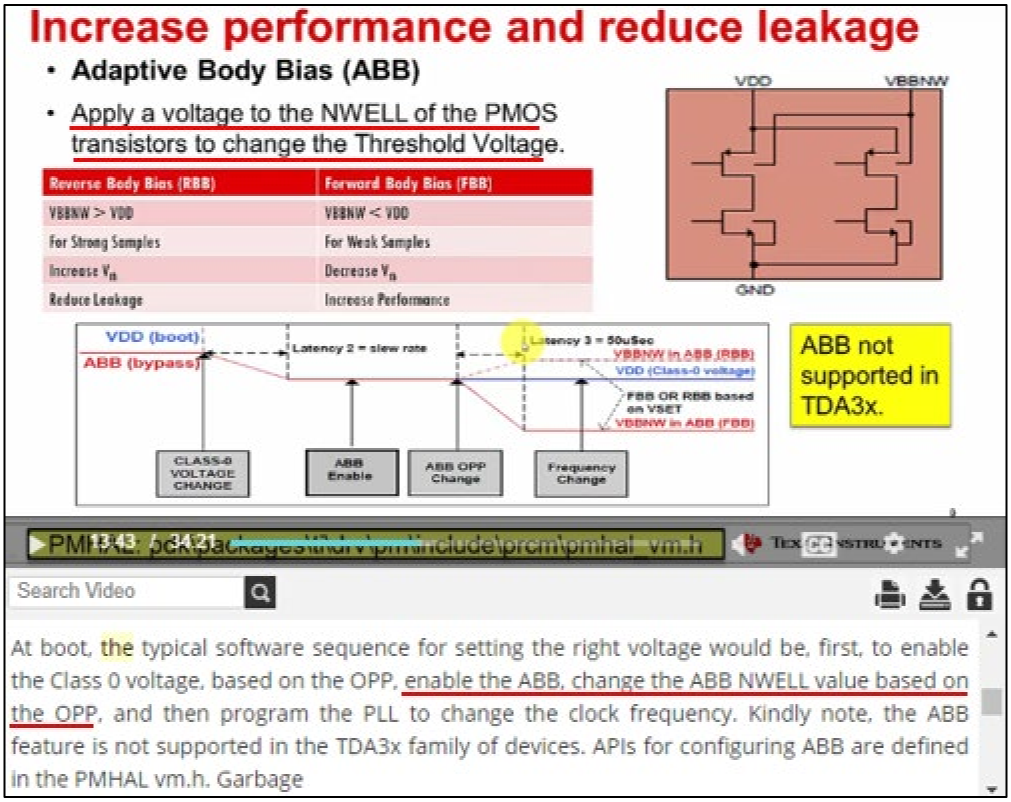


**Figure 4. VDD and VBBNW Supply to Device Transistors**

There are given values of AVS-Class 0 and ABB LDO voltage for each OPP. The PMHAL provides APIs to program the voltage rails with AVS and ABB (if available) for a given OPP (OPP_LOW/OPP_NOM/OPP_OD/OPP_HIGH).

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



## Increase performance and reduce leakage

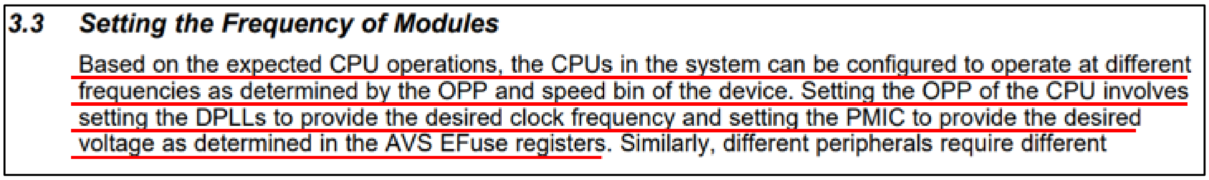
- **Adaptive Body Bias (ABB)**
- Apply a voltage to the NWELL of the PMOS transistors to change the Threshold Voltage.

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

ABB not supported in TDA3x.

At boot, the typical software sequence for setting the right voltage would be, first, to enable the Class 0 voltage, based on the OPP, enable the ABB, change the ABB NWELL value based on the OPP, and then program the PLL to change the clock frequency. Kindly note, the ABB feature is not supported in the TDA3x family of devices. APIs for configuring ABB are defined in the PMHAL vm.h. Garbage

**Source:** https://training.ti.com/introduction-adas-power-management-software?cu=1137175 (at 13:43)



### 3.3 *Setting the Frequency of Modules*

Based on the expected CPU operations, the CPUs in the system can be configured to operate at different frequencies as determined by the OPP and speed bin of the device. Setting the OPP of the CPU involves setting the DPLLs to provide the desired clock frequency and setting the PMIC to provide the desired voltage as determined in the AVS EFuse registers. Similarly, different peripherals require different

**Source:** https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 9)

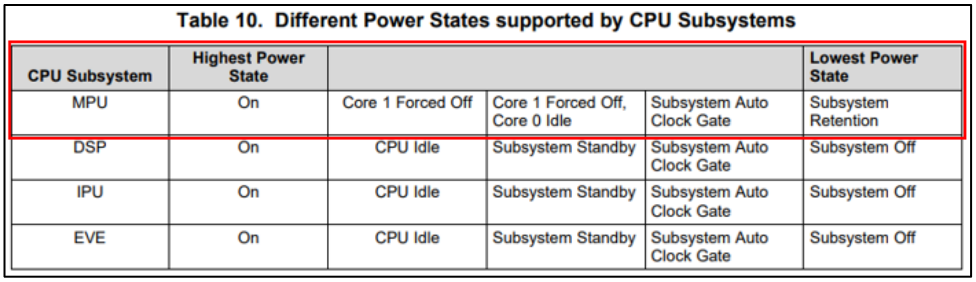
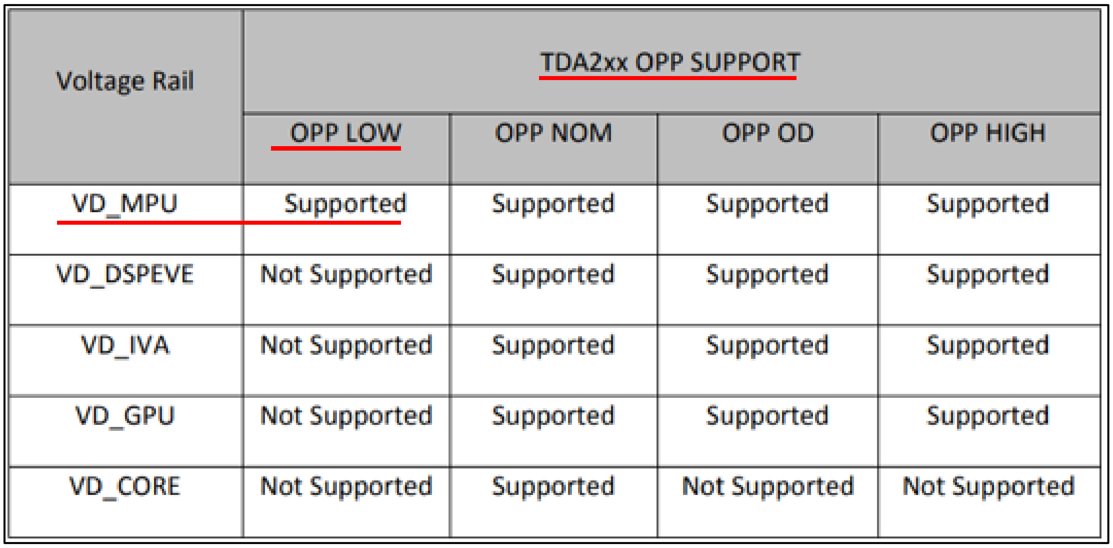


**Table 10. Different Power States supported by CPU Subsystems**

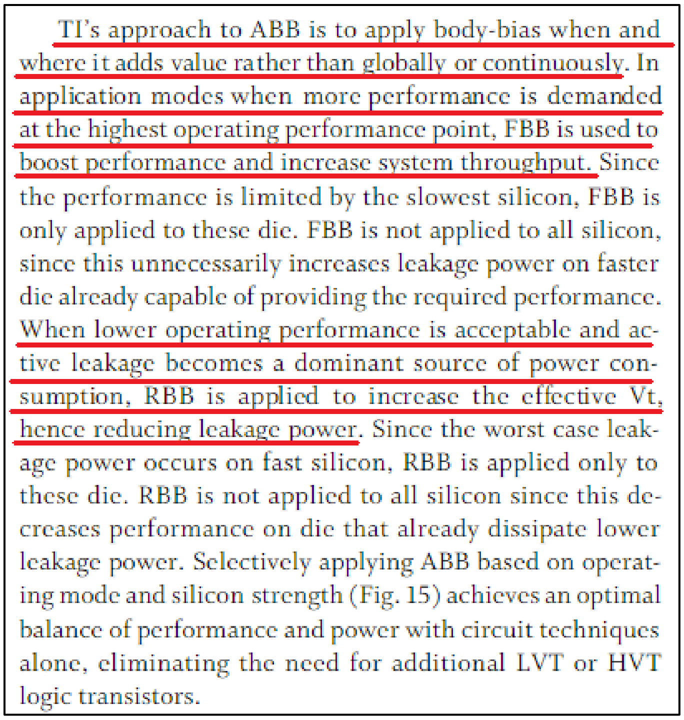
| CPU Subsystem | Highest Power State | | | | Lowest Power State |
|---|---|---|---|---|---|
| MPU | On | Core 1 Forced Off | Core 1 Forced Off, Core 0 Idle | Subsystem Auto Clock Gate | Subsystem Retention |
| DSP | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| IPU | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| EVE | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |

**Source:** https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 18)



| Voltage Rail | TDA2xx OPP SUPPORT | | | |
|---|---|---|---|---|
| | OPP LOW | OPP NOM | OPP OD | OPP HIGH |
| VD_MPU | Supported | Supported | Supported | Supported |
| VD_DSPEVE | Not Supported | Supported | Supported | Supported |
| VD_IVA | Not Supported | Supported | Supported | Supported |
| VD_GPU | Not Supported | Supported | Supported | Supported |
| VD_CORE | Not Supported | Supported | Not Supported | Not Supported |

**Source:** https://processors.wiki.ti.com/images/b/b7/TDA_SBL_UserGuide.pdf (Page 23)



TI's approach to ABB is to apply body-bias when and where it adds value rather than globally or continuously. In application modes when more performance is demanded at the highest operating performance point, FBB is used to boost performance and increase system throughput. Since the performance is limited by the slowest silicon, FBB is only applied to these die. FBB is not applied to all silicon, since this unnecessarily increases leakage power on faster die already capable of providing the required performance. When lower operating performance is acceptable and active leakage becomes a dominant source of power consumption, RBB is applied to increase the effective Vt, hence reducing leakage power. Since the worst case leakage power occurs on fast silicon, RBB is applied only to these die. RBB is not applied to all silicon since this decreases performance on die that already dissipate lower leakage power. Selectively applying ABB based on operating mode and silicon strength (Fig. 15) achieves an optimal balance of performance and power with circuit techniques alone, eliminating the need for additional LVT or HVT logic transistors.

**Source:**

https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

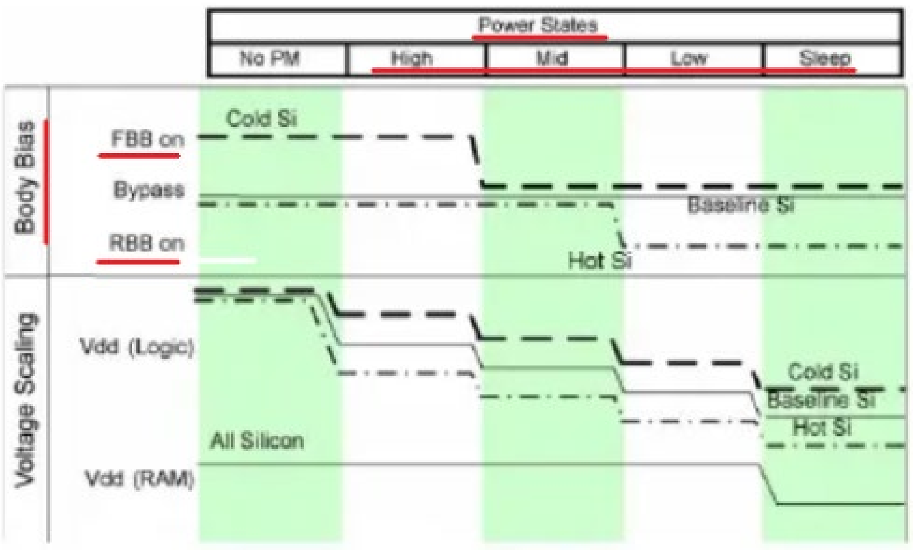


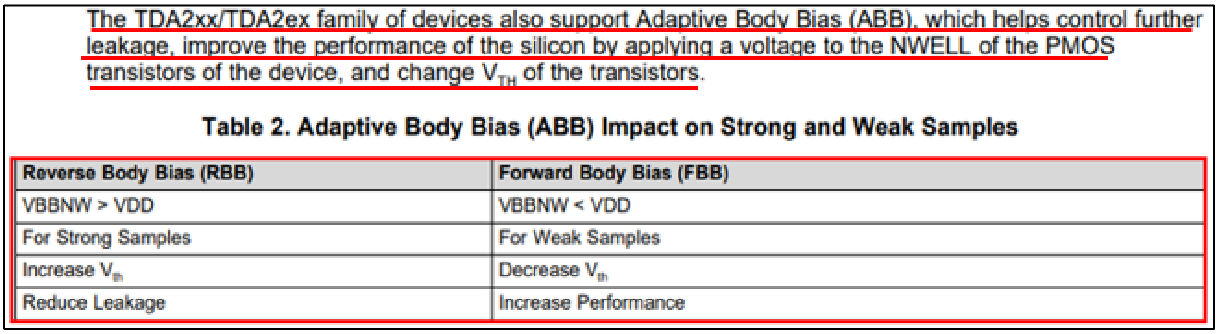
**Fig. 15.** ***Adaptivebody-biasand adaptive voltage scaling configuration with mode and process. Hot and cold refer the process strength or speed (fast and slow).***

**Source:**
https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

38. The method practiced using the accused products further comprises commanding a voltage supply coupled to a body terminal of the microprocessor to generate a voltage corresponding to the body biasing voltage information corresponding to the power condition.

39. For example, TI’s TDA2x System-on-Chip devices include a dual core processor module (“microprocessor”). The devices support Adaptive Body Bias (ABB), which allows application of forward body bias to increase performance and reverse body bias to reduce leakage in the device. The ABB voltage value can be programmed—that is, a command can be generated by the device to apply a body bias voltage value. Further, the programmed reverse body bias voltage value is supplied by the ABB LDO (“voltage supply”) to the VBBNW pin (“body terminal”) of the transistors of the device’s microprocessor.

40. To reduce leakage in low power states or low operating performance points (OPP Low), the reverse body bias voltage value ("body biasing voltage information") is accessed and applied to the device. This occurs during one of the low power states of the device, such as the power state in which one core is forced off and the other core is in retention ("said power condition"). Accordingly, the device can command the ABB LDO, which is coupled to the VBBNW pin, to supply a reverse body bias voltage corresponding to the low power state, i.e., the power state in which one core is forced off and other core is in retention ("said power condition").

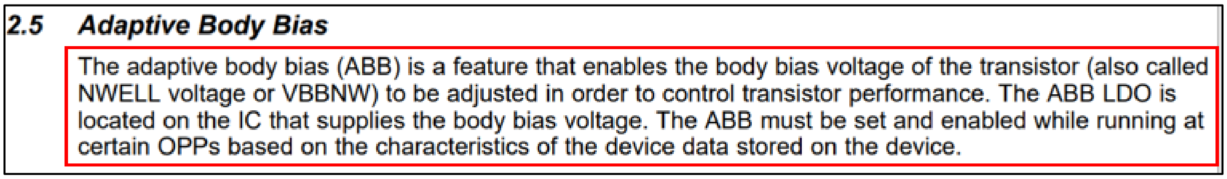


The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:**
https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)



2.5 *Adaptive Body Bias*

The adaptive body bias (ABB) is a feature that enables the body bias voltage of the transistor (also called NWELL voltage or VBBNW) to be adjusted in order to control transistor performance. The ABB LDO is located on the IC that supplies the body bias voltage. The ABB must be set and enabled while running at certain OPPs based on the characteristics of the device data stored on the device.

**Source:**
https://www.ti.com/lit/an/slva646/slva646.pdf?ts=1597323545336&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)

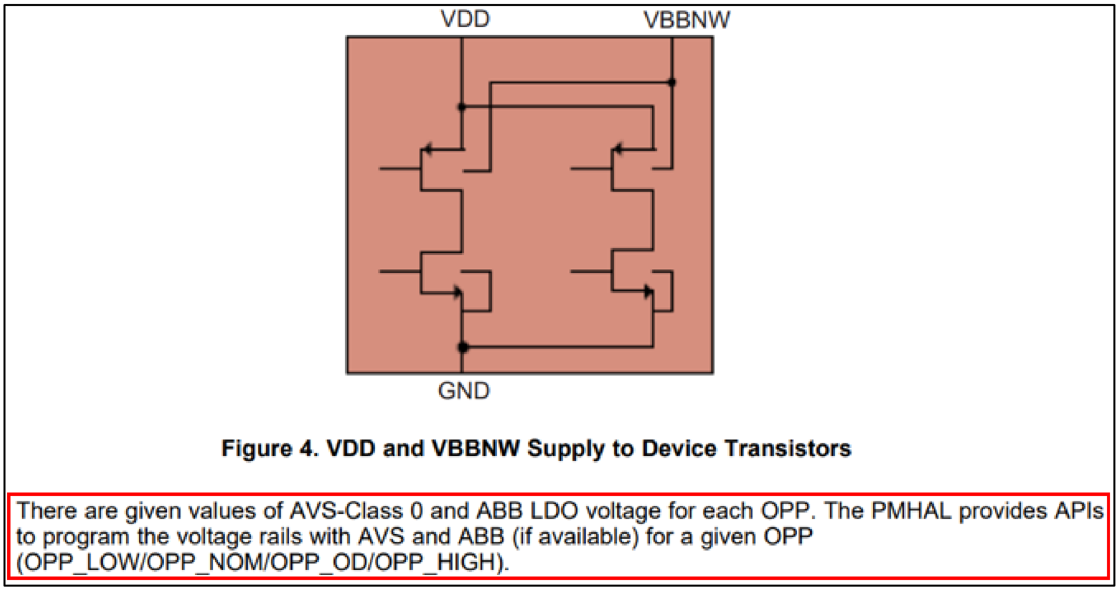
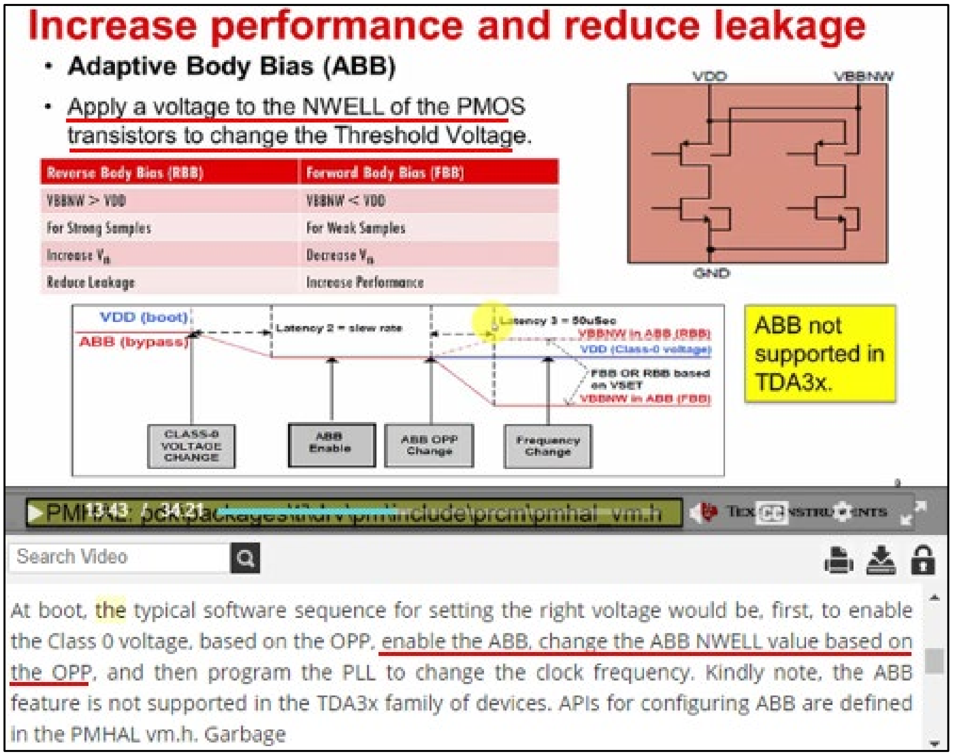


**Figure 4. VDD and VBBNW Supply to Device Transistors**

There are given values of AVS-Class 0 and ABB LDO voltage for each OPP. The PMHAL provides APIs to program the voltage rails with AVS and ABB (if available) for a given OPP (OPP_LOW/OPP_NOM/OPP_OD/OPP_HIGH).

**Source:** https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



Increase performance and reduce leakage

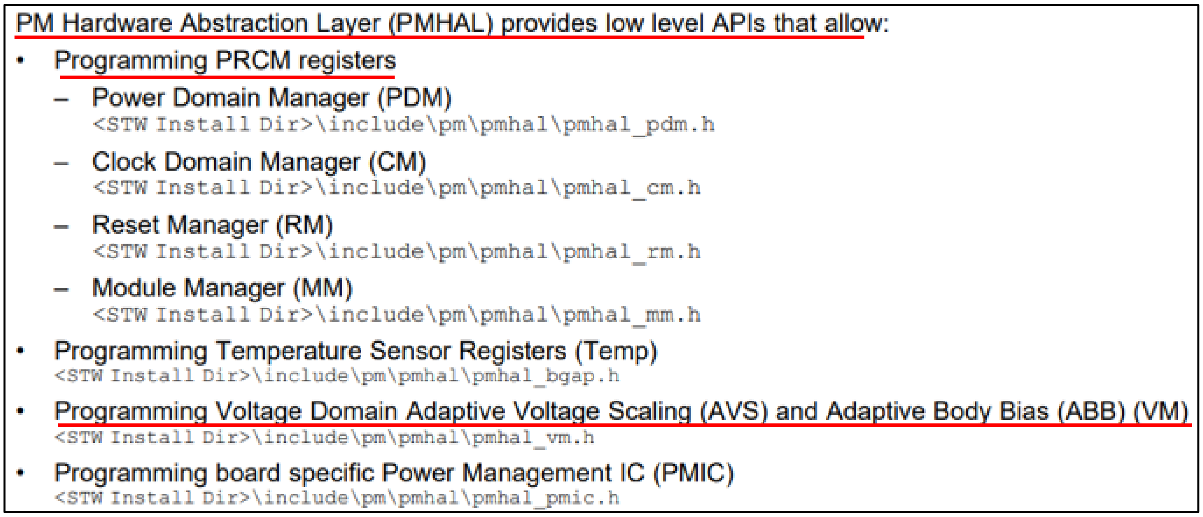
- Adaptive Body Bias (ABB)
- Apply a voltage to the NWELL of the PMOS transistors to change the Threshold Voltage.

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

ABB not supported in TDA3x.

At boot, the typical software sequence for setting the right voltage would be, first, to enable the Class 0 voltage, based on the OPP, enable the ABB, change the ABB NWELL value based on the OPP, and then program the PLL to change the clock frequency. Kindly note, the ABB feature is not supported in the TDA3x family of devices. APIs for configuring ABB are defined in the PMHAL vm.h. Garbage

**Source:** https://training.ti.com/introduction-adas-power-management-software?cu=1137175 (at 13:43)

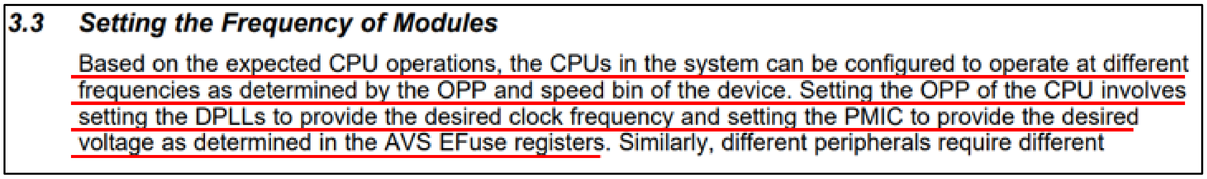


PM Hardware Abstraction Layer (PMHAL) provides low level APIs that allow:

- Programming PRCM registers
  - Power Domain Manager (PDM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_pdm.h`
  - Clock Domain Manager (CM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_cm.h`
  - Reset Manager (RM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_rm.h`
  - Module Manager (MM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_mm.h`
- Programming Temperature Sensor Registers (Temp)
  `<STW Install Dir>\include\pm\pmhal\pmhal_bgap.h`
- Programming Voltage Domain Adaptive Voltage Scaling (AVS) and Adaptive Body Bias (ABB) (VM)
  `<STW Install Dir>\include\pm\pmhal\pmhal_vm.h`
- Programming board specific Power Management IC (PMIC)
  `<STW Install Dir>\include\pm\pmhal\pmhal_pmic.h`

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)



### *3.3 Setting the Frequency of Modules*

Based on the expected CPU operations, the CPUs in the system can be configured to operate at different frequencies as determined by the OPP and speed bin of the device. Setting the OPP of the CPU involves setting the DPLLs to provide the desired clock frequency and setting the PMIC to provide the desired voltage as determined in the AVS EFuse registers. Similarly, different peripherals require different

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 9)

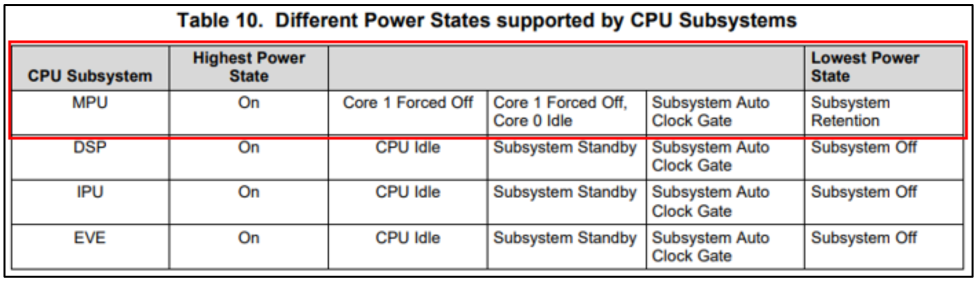
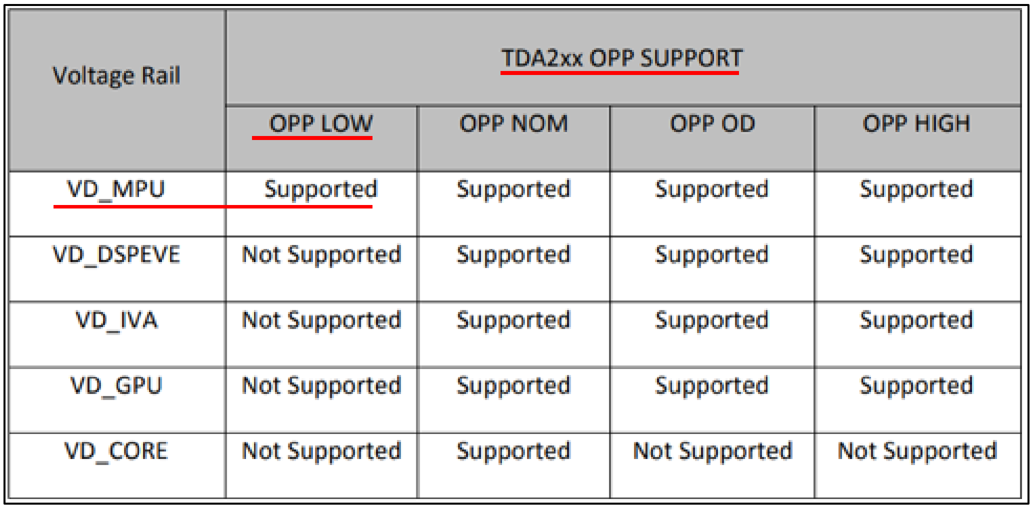


**Table 10. Different Power States supported by CPU Subsystems**

| CPU Subsystem | Highest Power State | | | | Lowest Power State |
|---|---|---|---|---|---|
| MPU | On | Core 1 Forced Off | Core 1 Forced Off, Core 0 Idle | Subsystem Auto Clock Gate | Subsystem Retention |
| DSP | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| IPU | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| EVE | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |

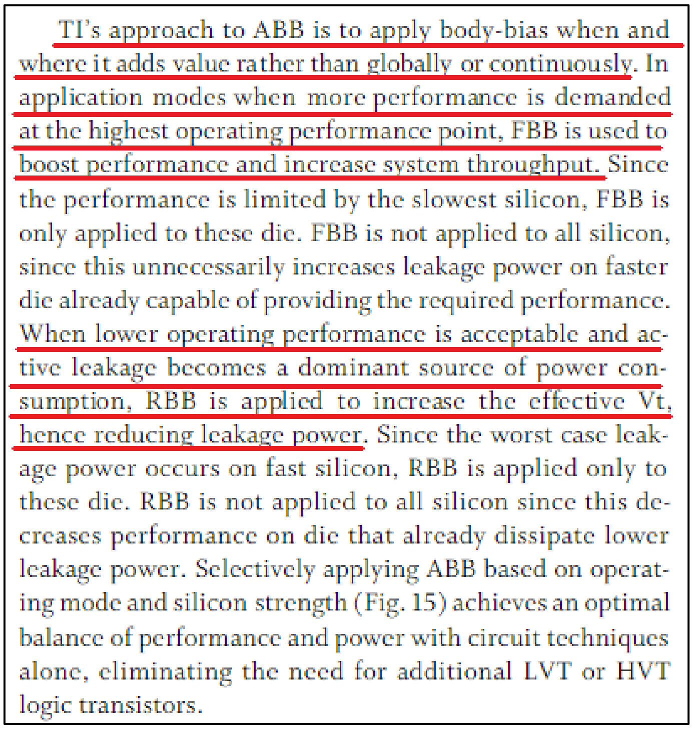
**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 18)



| Voltage Rail | TDA2xx OPP SUPPORT | | | |
|---|---|---|---|---|
| | OPP LOW | OPP NOM | OPP OD | OPP HIGH |
| VD_MPU | Supported | Supported | Supported | Supported |
| VD_DSPEVE | Not Supported | Supported | Supported | Supported |
| VD_IVA | Not Supported | Supported | Supported | Supported |
| VD_GPU | Not Supported | Supported | Supported | Supported |
| VD_CORE | Not Supported | Supported | Not Supported | Not Supported |

**Source:** https://processors.wiki.ti.com/images/b/b7/TDA_SBL_UserGuide.pdf (Page 23)



TI's approach to ABB is to apply body-bias when and where it adds value rather than globally or continuously. In application modes when more performance is demanded at the highest operating performance point, FBB is used to boost performance and increase system throughput. Since the performance is limited by the slowest silicon, FBB is only applied to these die. FBB is not applied to all silicon, since this unnecessarily increases leakage power on faster die already capable of providing the required performance. When lower operating performance is acceptable and active leakage becomes a dominant source of power consumption, RBB is applied to increase the effective Vt, hence reducing leakage power. Since the worst case leakage power occurs on fast silicon, RBB is applied only to these die. RBB is not applied to all silicon since this decreases performance on die that already dissipate lower leakage power. Selectively applying ABB based on operating mode and silicon strength (Fig. 15) achieves an optimal balance of performance and power with circuit techniques alone, eliminating the need for additional LVT or HVT logic transistors.

**Source:**

https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

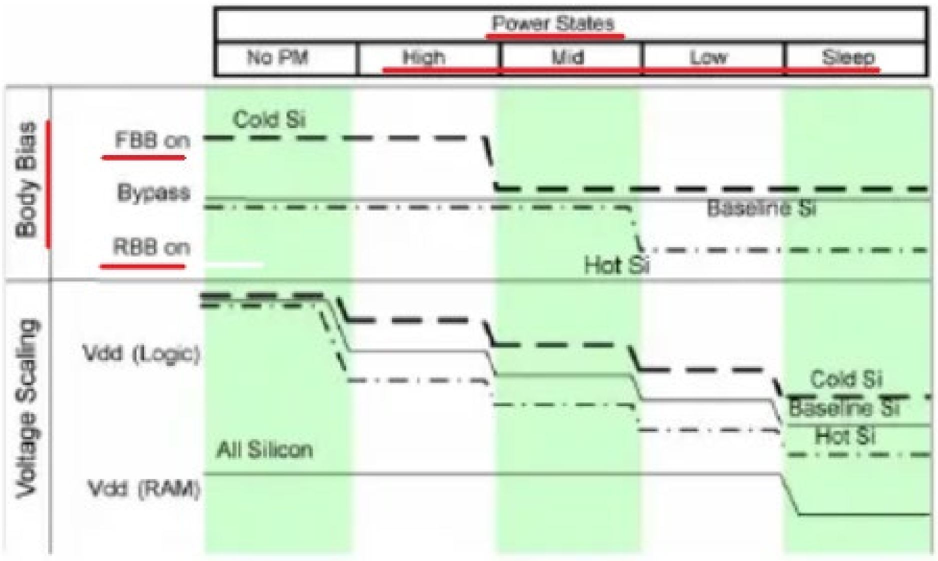


**Fig. 15.** ***Adaptivebody-biasand adaptive voltage scaling configuration with mode and process. Hot and cold refer the process strength or speed (fast and slow).***

**Source:**
https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

41. TI has had knowledge of the ’504 Patent at least as of the date when it was notified of the filing of this action.

42. Liberty Patents has been damaged as a result of the infringing conduct by TI alleged above. Thus, TI is liable to Liberty Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

43. Liberty Patents and/or its predecessors-in-interest have satisfied all statutory obligations required to collect pre-filing damages for the full period allowed by law for infringement of the ’504 Patent.

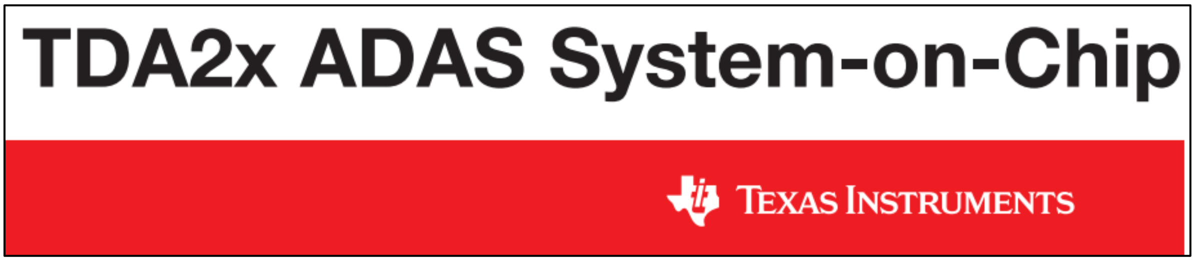
## COUNT II

## DIRECT INFRINGEMENT OF U.S. PATENT NO. 8,127,156

44. On February 28, 2012, U.S. Patent No. 8,127,156 ("the '156 Patent") was duly and legally issued by the United States Patent and Trademark Office for an invention entitled "Systems and Methods for Control of Integrated Circuits Comprising Body Biasing Systems."

45. Liberty Patents is the owner of the '156 Patent, with all substantive rights in and to that patent, including the sole and exclusive right to prosecute this action and enforce the '156 Patent against infringers, and to collect damages for all relevant times.

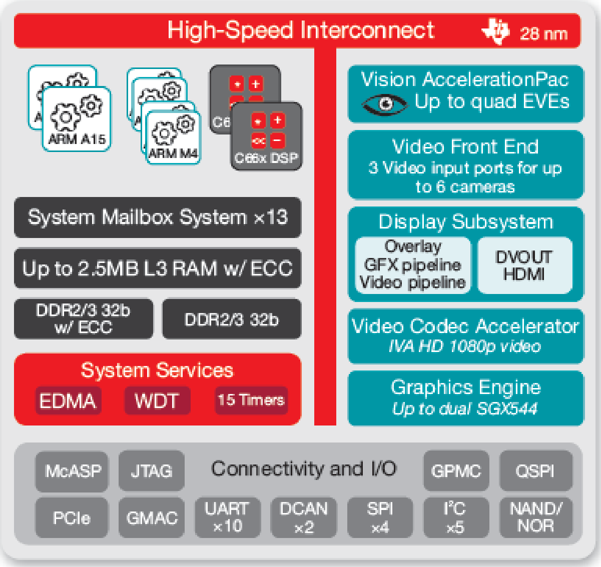
46. TI made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, TI's TDA2x System-on-Chip devices and other products[7] that are capable of applying body biasing voltages during a power mode of a processor or other integrated circuit ("accused products"):



TDA2x ADAS System-on-Chip

TEXAS INSTRUMENTS

[7] *See, e.g.*, TI AM5706, AM5708, AM5716, AM5718, AM5718-HIREL, AM5726, AM5728, AM5729, AM5746, AM5748, AM5749, DRA710, DRA712, DRA714, DRA716, DRA718, DRA722, DRA724, DRA725, DRA726, DRA744, DRA745, DRA746, DRA74P, DRA750, DRA756, DRA75P, DRA76P, DRA77P, DRA790, DRA791, DRA793, DRA797, OMAP4430, OMAP4460, OMAP4470, OMAP5430, OMAP5432, TDA2E, TDA2EG-17, TDA2HF, TDA2HG, TDA2HV, TDA2LF, TDA2P-ABZ, TDA2P-ACD, TDA2SA, TDA2SG, TDA2SX, TMDXIDK5718, TMDSEVM572X, TMDSIDK572, TMDSIDK574, D3-3P-ADAS-DK, D3-3P-TDAX-DK, DRA71XEVM, DRA72XEVM, DRA79XEVM, 703664-1001, 705851-1001, 705852-1001, DRA76XP-DRA77XP-TDA2PX-ACD-CPUBOARD, J6EVM5777, OMAP5432-EVM, TDA2EXEVM, TDA2PXEVM, EVM5777BG-03-00-00, EVM5777G-03-40-00, EVM5777VISION-V2-0, etc.



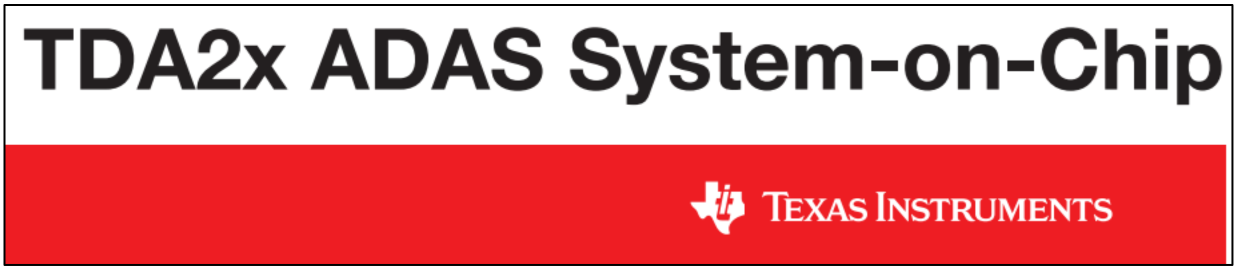
▲ *Figure 1. Block diagram for TDA2x SoC*

**Source:**
https://www.ti.com/lit/ml/sprt681/sprt681.pdf?ts=1597386693056&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 1)

47. By doing so, TI has directly infringed (literally and/or under the doctrine of equivalents) at least Claims 1 and 9 of the '156 Patent. TI's infringement in this regard is ongoing.

48. TI's TDA2x System-on-Chip devices are exemplary accused products. The devices include a computer system.

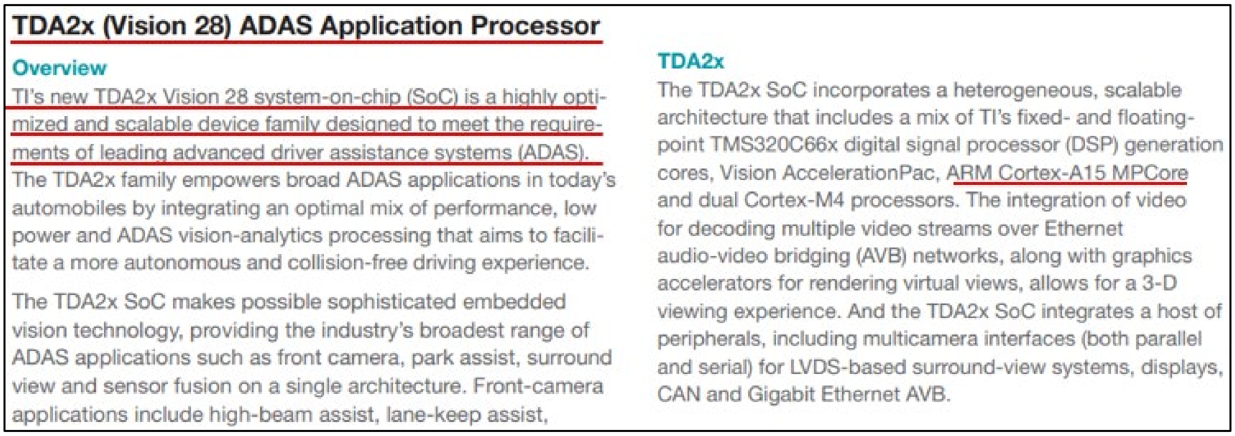
49. For example, TI's TDA2x System-on-Chip devices include a dual core processor module along with different memory elements and peripherals ("computer system"). The devices support Adaptive Body Bias (ABB), which can apply a forward or reverse body bias voltage to the transistors of the device using ABB.



TDA2x ADAS System-on-Chip

TEXAS INSTRUMENTS

**Source:** https://www.ti.com/lit/ml/sprt681/sprt681.pdf?ts=1597386693056&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 1)



TDA2x (Vision 28) ADAS Application Processor

Overview

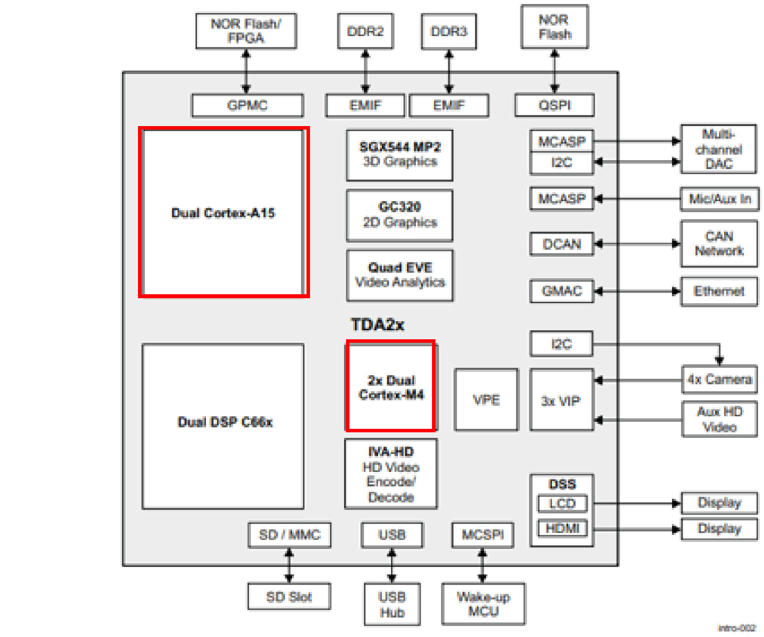
TI's new TDA2x Vision 28 system-on-chip (SoC) is a highly optimized and scalable device family designed to meet the requirements of leading advanced driver assistance systems (ADAS). The TDA2x family empowers broad ADAS applications in today's automobiles by integrating an optimal mix of performance, low power and ADAS vision-analytics processing that aims to facilitate a more autonomous and collision-free driving experience.

The TDA2x SoC makes possible sophisticated embedded vision technology, providing the industry's broadest range of ADAS applications such as front camera, park assist, surround view and sensor fusion on a single architecture. Front-camera applications include high-beam assist, lane-keep assist,

TDA2x

The TDA2x SoC incorporates a heterogeneous, scalable architecture that includes a mix of TI's fixed- and floating-point TMS320C66x digital signal processor (DSP) generation cores, Vision AccelerationPac, ARM Cortex-A15 MPCore and dual Cortex-M4 processors. The integration of video for decoding multiple video streams over Ethernet audio-video bridging (AVB) networks, along with graphics accelerators for rendering virtual views, allows for a 3-D viewing experience. And the TDA2x SoC integrates a host of peripherals, including multicamera interfaces (both parallel and serial) for LVDS-based surround-view systems, displays, CAN and Gigabit Ethernet AVB.

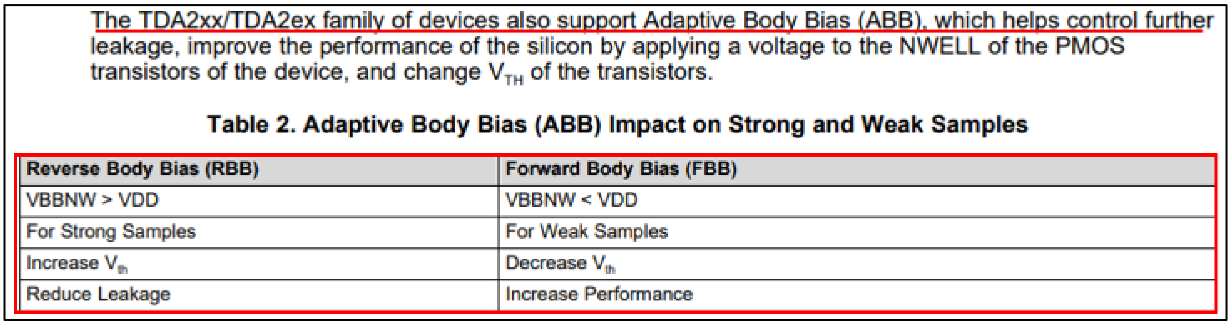
**Source:** https://m.eet.com/media/1241324/slyy044.pdf (Page 5)



A The speeds shown are the highest design targeted, non-binned OPP.

Figure 1. TDA2xx Device Block Diagram

**Source:**
https://www.ti.com/lit/an/sprac21a/sprac21a.pdf?ts=1597386703137&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

**Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples**

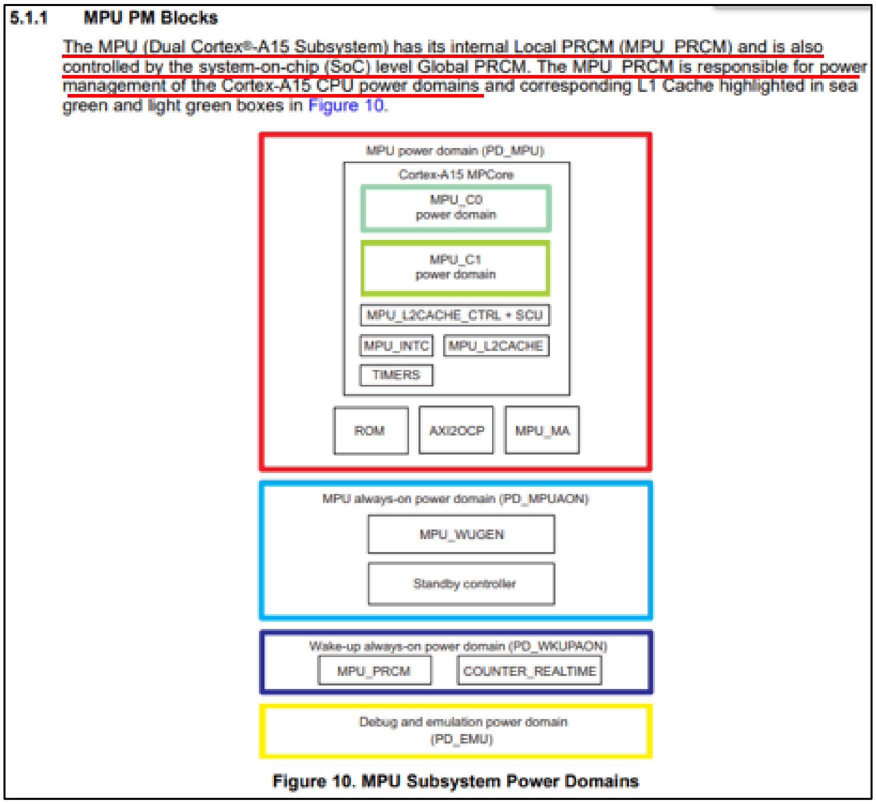
| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:**
https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)

50. TI's TDA2x System-on-Chip devices include a plurality of data storage locations comprising body biasing voltage information corresponding to a plurality of power modes of the computer system.

51. For example, TI's TDA2x System-on-Chip devices include a dual core processor module. Further, the System-on-Chip includes an MPU_PRCM module for power management of the processor. The TDA2x family of devices supports multiple power states (e.g., a power state in which both cores are on, a power state in which one core is forced off, a power state in which one core is forced off and the other core is in retention, etc.).

52. The devices support Adaptive Body Bias (ABB). ABB voltage values are available for each OPP (Operating Performance Point), and therefore, for each power state. The OPP is the operating condition for the device and is defined by the voltage and frequency applied to the processor. The ABB voltage values ("body biasing voltage information") corresponding to the different power states ("plurality of power modes") are stored in Efuse registers ("data storage locations").



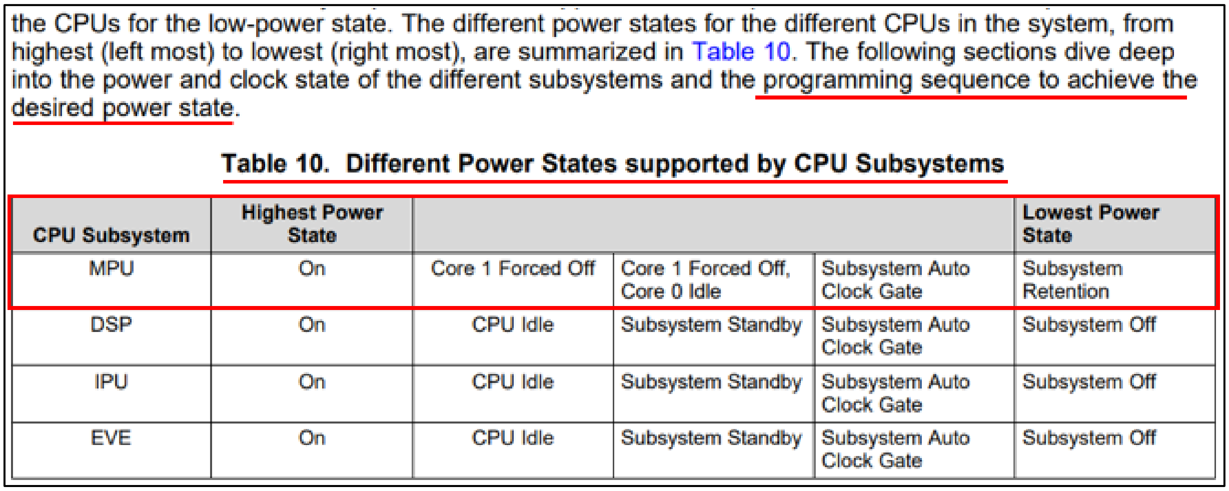
5.1.1 **MPU PM Blocks**

The MPU (Dual Cortex®-A15 Subsystem) has its internal Local PRCM (MPU_PRCM) and is also controlled by the system-on-chip (SoC) level Global PRCM. The MPU_PRCM is responsible for power management of the Cortex-A15 CPU power domains and corresponding L1 Cache highlighted in sea green and light green boxes in Figure 10.

**Figure 10. MPU Subsystem Power Domains**

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 19)



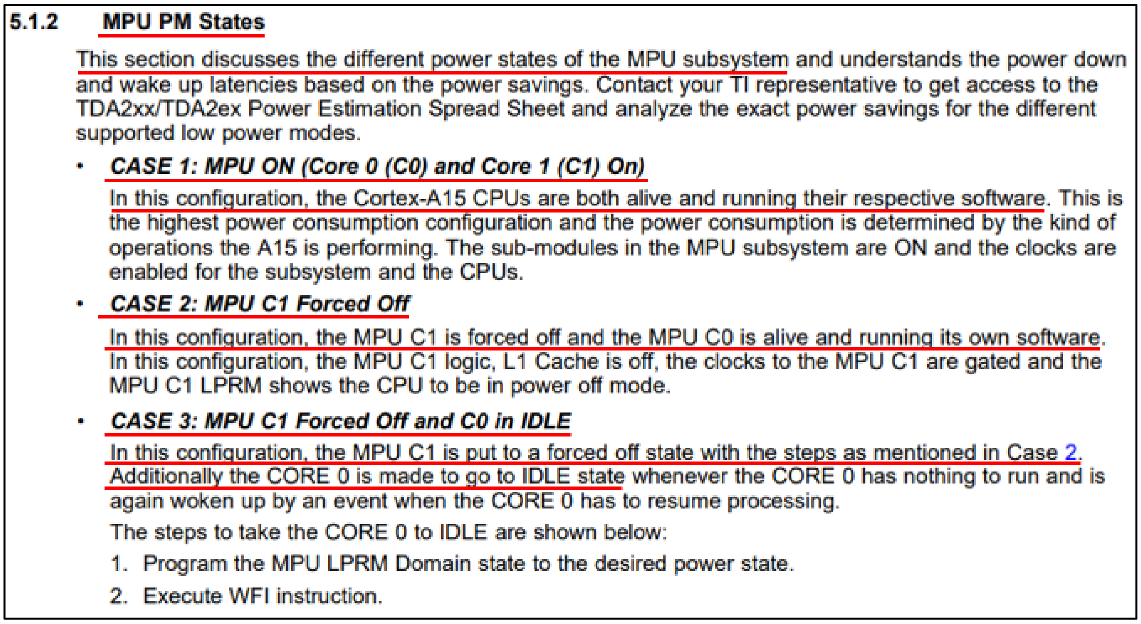
the CPUs for the low-power state. The different power states for the different CPUs in the system, from highest (left most) to lowest (right most), are summarized in Table 10. The following sections dive deep into the power and clock state of the different subsystems and the programming sequence to achieve the desired power state.

**Table 10. Different Power States supported by CPU Subsystems**

| CPU Subsystem | Highest Power State | | | | Lowest Power State |
|---|---|---|---|---|---|
| MPU | On | Core 1 Forced Off | Core 1 Forced Off, Core 0 Idle | Subsystem Auto Clock Gate | Subsystem Retention |
| DSP | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| IPU | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| EVE | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 18)



### 5.1.2 MPU PM States

This section discusses the different power states of the MPU subsystem and understands the power down and wake up latencies based on the power savings. Contact your TI representative to get access to the TDA2xx/TDA2ex Power Estimation Spread Sheet and analyze the exact power savings for the different supported low power modes.

- ***CASE 1: MPU ON (Core 0 (C0) and Core 1 (C1) On)***

  In this configuration, the Cortex-A15 CPUs are both alive and running their respective software. This is the highest power consumption configuration and the power consumption is determined by the kind of operations the A15 is performing. The sub-modules in the MPU subsystem are ON and the clocks are enabled for the subsystem and the CPUs.

- ***CASE 2: MPU C1 Forced Off***

  In this configuration, the MPU C1 is forced off and the MPU C0 is alive and running its own software. In this configuration, the MPU C1 logic, L1 Cache is off, the clocks to the MPU C1 are gated and the MPU C1 LPRM shows the CPU to be in power off mode.

- ***CASE 3: MPU C1 Forced Off and C0 in IDLE***

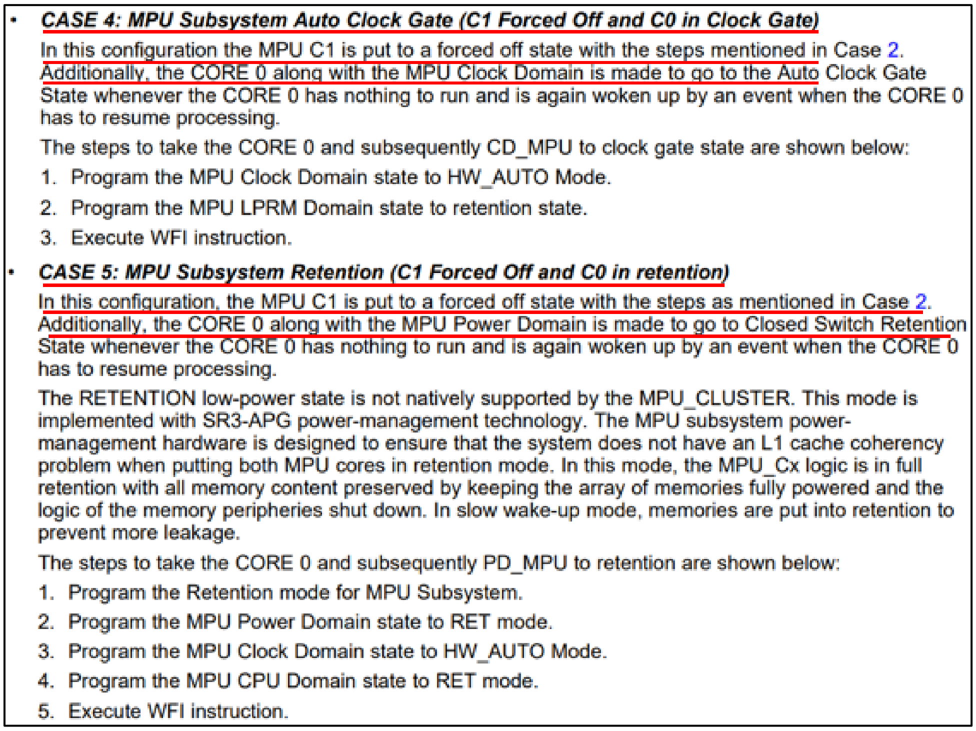
  In this configuration, the MPU C1 is put to a forced off state with the steps as mentioned in Case 2. Additionally the CORE 0 is made to go to IDLE state whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

  The steps to take the CORE 0 to IDLE are shown below:

  1. Program the MPU LPRM Domain state to the desired power state.
  2. Execute WFI instruction.

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Pages 20 and 21)



- ***CASE 4: MPU Subsystem Auto Clock Gate (C1 Forced Off and C0 in Clock Gate)***

  In this configuration the MPU C1 is put to a forced off state with the steps mentioned in Case 2. Additionally, the CORE 0 along with the MPU Clock Domain is made to go to the Auto Clock Gate State whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

  The steps to take the CORE 0 and subsequently CD_MPU to clock gate state are shown below:

  1. Program the MPU Clock Domain state to HW_AUTO Mode.
  2. Program the MPU LPRM Domain state to retention state.
  3. Execute WFI instruction.

- ***CASE 5: MPU Subsystem Retention (C1 Forced Off and C0 in retention)***

  In this configuration, the MPU C1 is put to a forced off state with the steps as mentioned in Case 2. Additionally, the CORE 0 along with the MPU Power Domain is made to go to Closed Switch Retention State whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

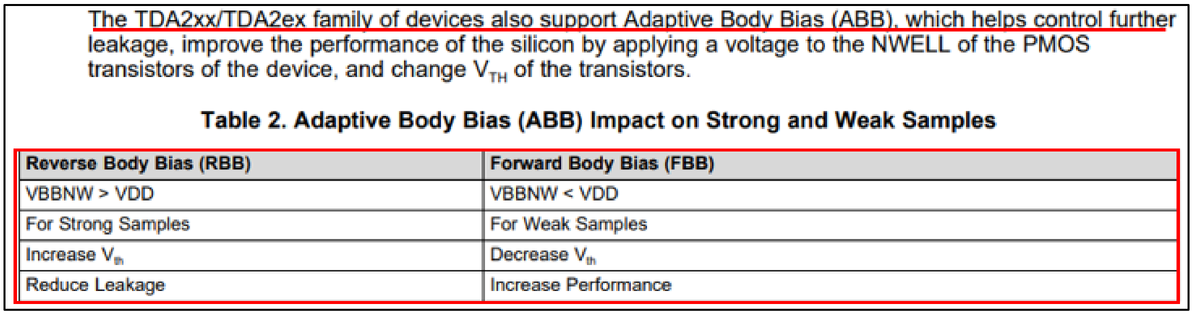
  The RETENTION low-power state is not natively supported by the MPU_CLUSTER. This mode is implemented with SR3-APG power-management technology. The MPU subsystem power-management hardware is designed to ensure that the system does not have an L1 cache coherency problem when putting both MPU cores in retention mode. In this mode, the MPU_Cx logic is in full retention with all memory content preserved by keeping the array of memories fully powered and the logic of the memory peripheries shut down. In slow wake-up mode, memories are put into retention to prevent more leakage.

  The steps to take the CORE 0 and subsequently PD_MPU to retention are shown below:

  1. Program the Retention mode for MPU Subsystem.
  2. Program the MPU Power Domain state to RET mode.
  3. Program the MPU Clock Domain state to HW_AUTO Mode.
  4. Program the MPU CPU Domain state to RET mode.
  5. Execute WFI instruction.

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 23)



The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

**Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples**

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)

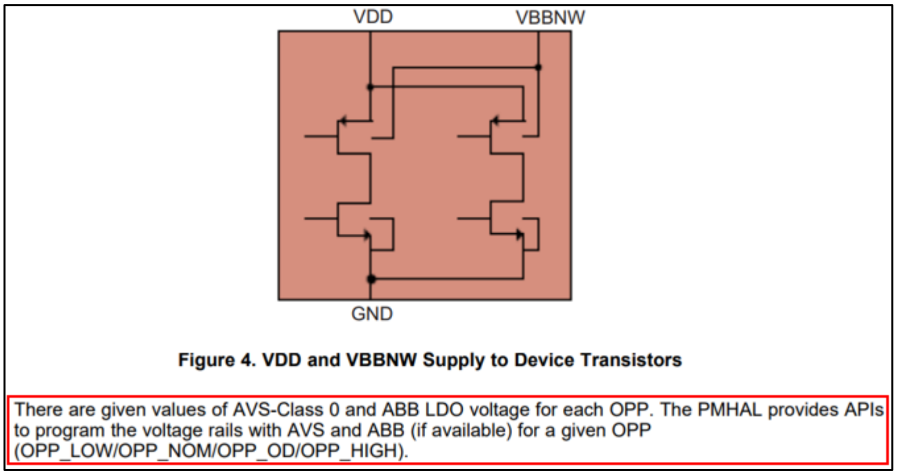
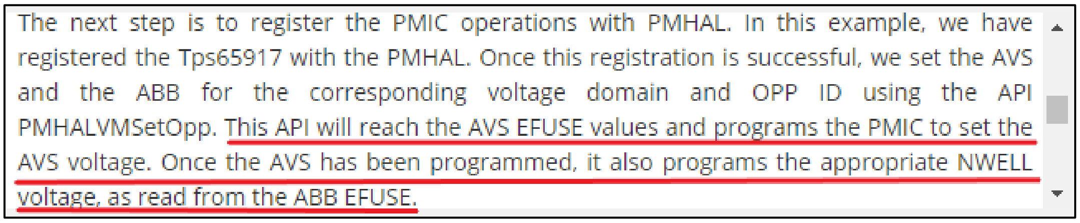


**Figure 4. VDD and VBBNW Supply to Device Transistors**

There are given values of AVS-Class 0 and ABB LDO voltage for each OPP. The PMHAL provides APIs to program the voltage rails with AVS and ABB (if available) for a given OPP (OPP_LOW/OPP_NOM/OPP_OD/OPP_HIGH).

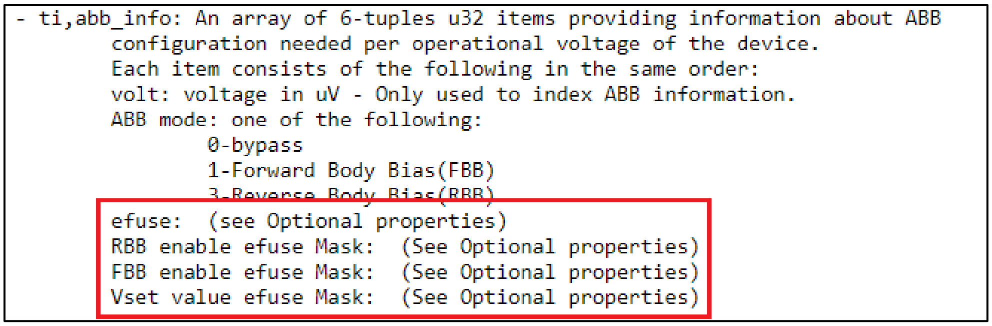
**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F



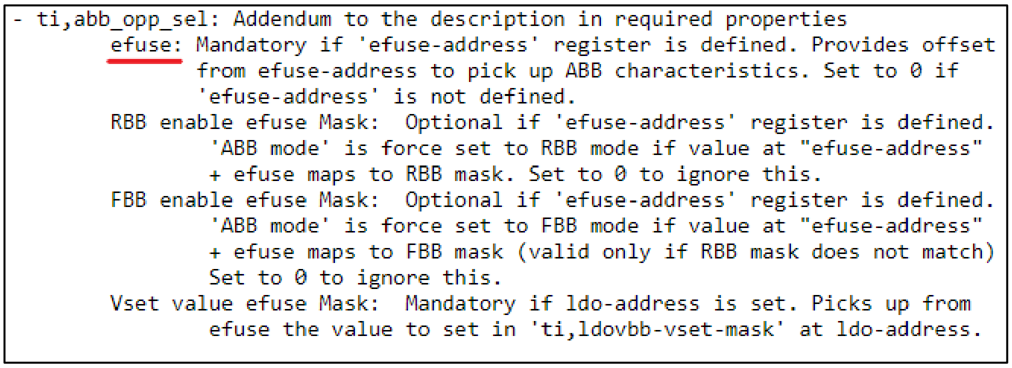
The next step is to register the PMIC operations with PMHAL. In this example, we have registered the Tps65917 with the PMHAL. Once this registration is successful, we set the AVS and the ABB for the corresponding voltage domain and OPP ID using the API PMHALVMSetOpp. This API will reach the AVS EFUSE values and programs the PMIC to set the AVS voltage. Once the AVS has been programmed, it also programs the appropriate NWELL voltage, as read from the ABB EFUSE.

**Source:** https://training.ti.com/introduction-adas-power-management-software (at 17:07)



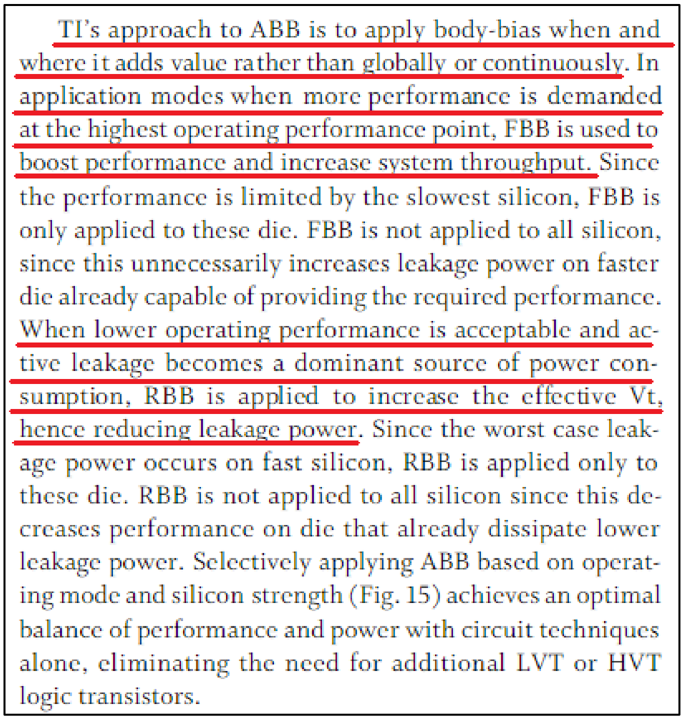
```
- ti,abb_info: An array of 6-tuples u32 items providing information about ABB
        configuration needed per operational voltage of the device.
        Each item consists of the following in the same order:
        volt: voltage in uV - Only used to index ABB information.
        ABB mode: one of the following:
                0-bypass
                1-Forward Body Bias(FBB)
                3-Reverse Body Bias(RBB)
        efuse:  (see Optional properties)
        RBB enable efuse Mask:  (See Optional properties)
        FBB enable efuse Mask:  (See Optional properties)
        Vset value efuse Mask:  (See Optional properties)
```

**Source:** https://www.kernel.org/doc/Documentation/devicetree/bindings/regulator/ti-abb-regulator.txt



```
- ti,abb_opp_sel: Addendum to the description in required properties
        efuse: Mandatory if 'efuse-address' register is defined. Provides offset
               from efuse-address to pick up ABB characteristics. Set to 0 if
               'efuse-address' is not defined.
        RBB enable efuse Mask:  Optional if 'efuse-address' register is defined.
                'ABB mode' is force set to RBB mode if value at "efuse-address"
                + efuse maps to RBB mask. Set to 0 to ignore this.
        FBB enable efuse Mask:  Optional if 'efuse-address' register is defined.
                'ABB mode' is force set to FBB mode if value at "efuse-address"
                + efuse maps to FBB mask (valid only if RBB mask does not match)
                Set to 0 to ignore this.
        Vset value efuse Mask:  Mandatory if ldo-address is set. Picks up from
                efuse the value to set in 'ti,ldovbb-vset-mask' at ldo-address.
```

**Source:** https://www.kernel.org/doc/Documentation/devicetree/bindings/regulator/ti-abb-regulator.txt



TI's approach to ABB is to apply body-bias when and where it adds value rather than globally or continuously. In application modes when more performance is demanded at the highest operating performance point, FBB is used to boost performance and increase system throughput. Since the performance is limited by the slowest silicon, FBB is only applied to these die. FBB is not applied to all silicon, since this unnecessarily increases leakage power on faster die already capable of providing the required performance. When lower operating performance is acceptable and active leakage becomes a dominant source of power consumption, RBB is applied to increase the effective Vt, hence reducing leakage power. Since the worst case leakage power occurs on fast silicon, RBB is applied only to these die. RBB is not applied to all silicon since this decreases performance on die that already dissipate lower leakage power. Selectively applying ABB based on operating mode and silicon strength (Fig. 15) achieves an optimal balance of performance and power with circuit techniques alone, eliminating the need for additional LVT or HVT logic transistors.

**Source:**

https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

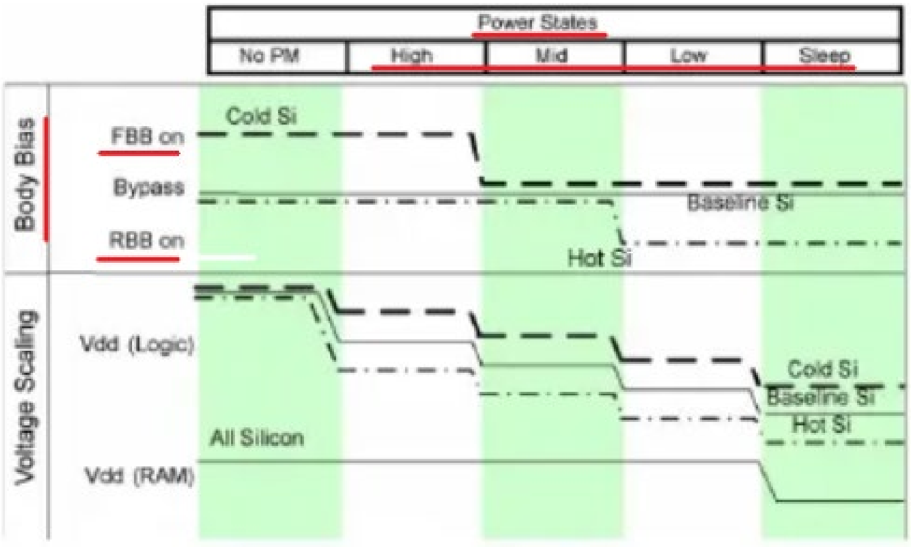


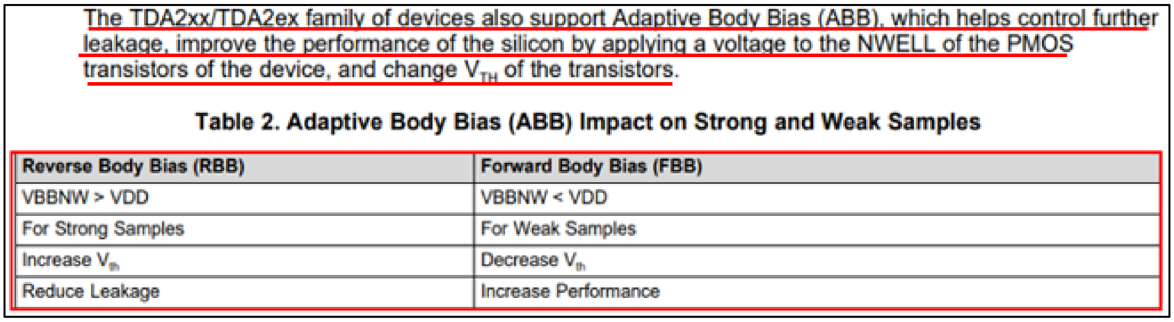
**Fig. 15.** ***Adaptivebody-biasand adaptive voltage scaling configuration with mode and process. Hot and cold refer the process strength or speed (fast and slow).***

**Source:**

https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

53. TI's TDA2x System-on-Chip devices include a voltage supply to generate a body biasing voltage for circuitry of the computer system according to the body biasing voltage information.

54. For example, TI's TDA2x System-on-Chip devices include a dual core processor module. The devices support Adaptive Body Bias (ABB). The ABB feature allows application of forward body bias to improve performance and reverse body bias to reduce leakage in the device. A programmed reverse body bias voltage is supplied by the ABB LDO ("voltage supply") to VBBNW pin of the transistors of the device's microprocessor. To reduce leakage in low power states or low operating performance points (OPP Low), the reverse body bias voltage value ("body biasing voltage information") from the Efuse registers is accessed and applied to the device.



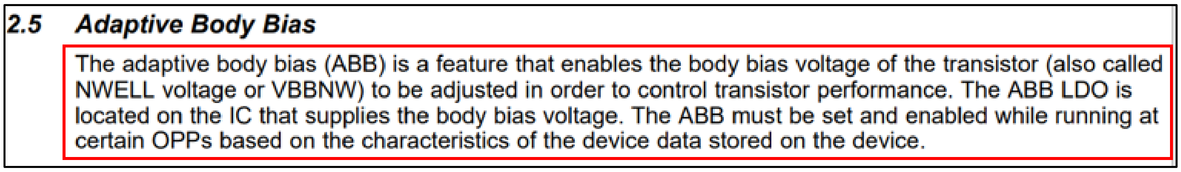
The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

**Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples**

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)



### 2.5 Adaptive Body Bias

The adaptive body bias (ABB) is a feature that enables the body bias voltage of the transistor (also called NWELL voltage or VBBNW) to be adjusted in order to control transistor performance. The ABB LDO is located on the IC that supplies the body bias voltage. The ABB must be set and enabled while running at certain OPPs based on the characteristics of the device data stored on the device.

**Source:**

https://www.ti.com/lit/an/slva646/slva646.pdf?ts=1597323545336&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)

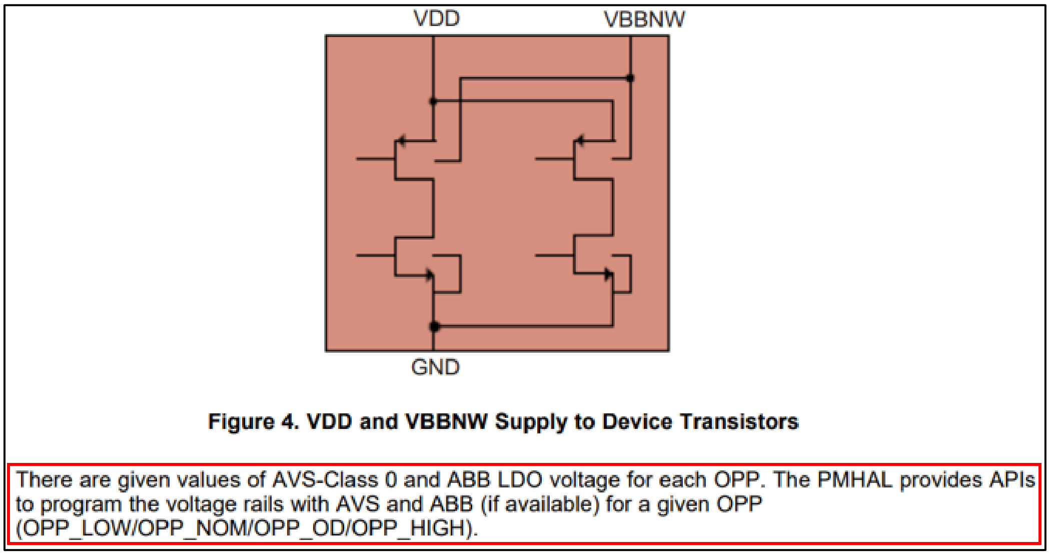
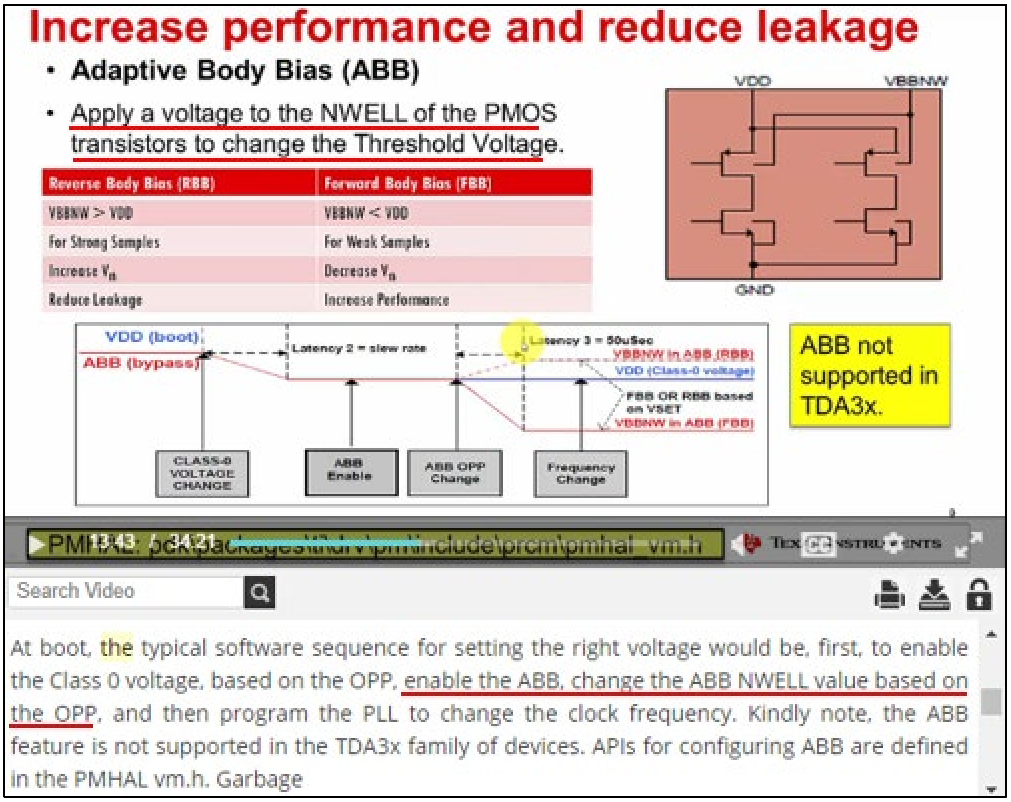


**Figure 4. VDD and VBBNW Supply to Device Transistors**

There are given values of AVS-Class 0 and ABB LDO voltage for each OPP. The PMHAL provides APIs to program the voltage rails with AVS and ABB (if available) for a given OPP (OPP_LOW/OPP_NOM/OPP_OD/OPP_HIGH).

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



## Increase performance and reduce leakage

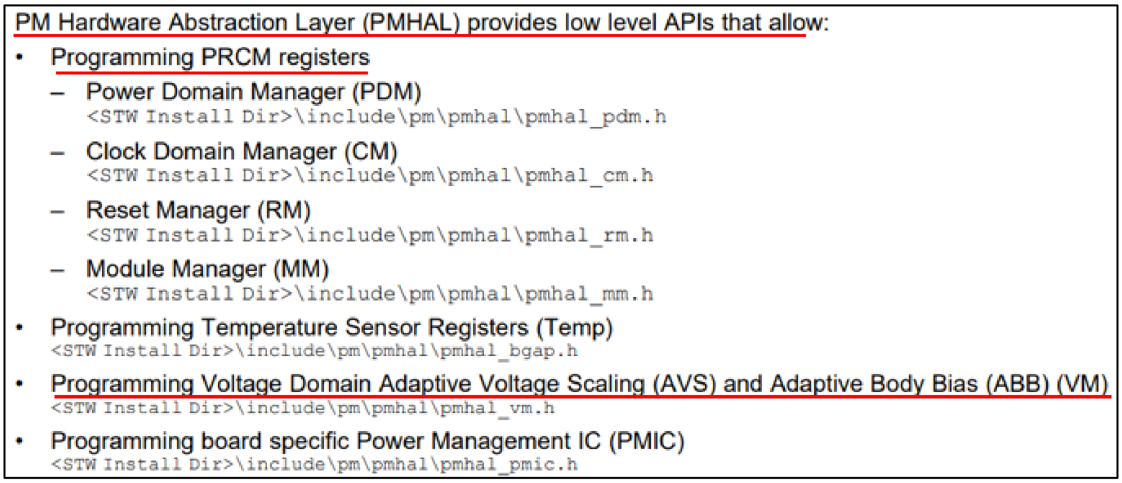
- **Adaptive Body Bias (ABB)**
- Apply a voltage to the NWELL of the PMOS transistors to change the Threshold Voltage.

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

ABB not supported in TDA3x.

At boot, the typical software sequence for setting the right voltage would be, first, to enable the Class 0 voltage, based on the OPP, enable the ABB, change the ABB NWELL value based on the OPP, and then program the PLL to change the clock frequency. Kindly note, the ABB feature is not supported in the TDA3x family of devices. APIs for configuring ABB are defined in the PMHAL vm.h. Garbage

**Source:** https://training.ti.com/introduction-adas-power-management-software?cu=1137175 (at 13:43)

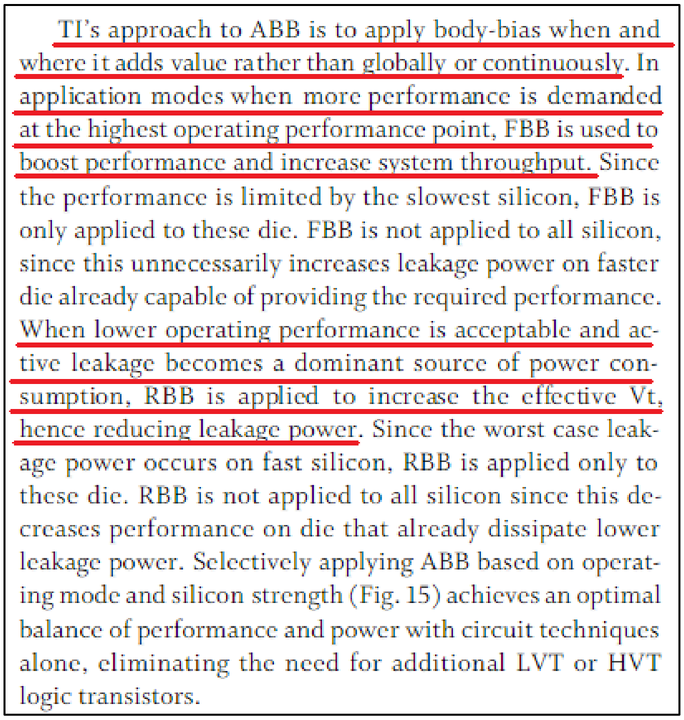


PM Hardware Abstraction Layer (PMHAL) provides low level APIs that allow:

- Programming PRCM registers
  - Power Domain Manager (PDM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_pdm.h`
  - Clock Domain Manager (CM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_cm.h`
  - Reset Manager (RM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_rm.h`
  - Module Manager (MM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_mm.h`
- Programming Temperature Sensor Registers (Temp)
  `<STW Install Dir>\include\pm\pmhal\pmhal_bgap.h`
- Programming Voltage Domain Adaptive Voltage Scaling (AVS) and Adaptive Body Bias (ABB) (VM)
  `<STW Install Dir>\include\pm\pmhal\pmhal_vm.h`
- Programming board specific Power Management IC (PMIC)
  `<STW Install Dir>\include\pm\pmhal\pmhal_pmic.h`

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)



TI's approach to ABB is to apply body-bias when and where it adds value rather than globally or continuously. In application modes when more performance is demanded at the highest operating performance point, FBB is used to boost performance and increase system throughput. Since the performance is limited by the slowest silicon, FBB is only applied to these die. FBB is not applied to all silicon, since this unnecessarily increases leakage power on faster die already capable of providing the required performance. When lower operating performance is acceptable and active leakage becomes a dominant source of power consumption, RBB is applied to increase the effective Vt, hence reducing leakage power. Since the worst case leakage power occurs on fast silicon, RBB is applied only to these die. RBB is not applied to all silicon since this decreases performance on die that already dissipate lower leakage power. Selectively applying ABB based on operating mode and silicon strength (Fig. 15) achieves an optimal balance of performance and power with circuit techniques alone, eliminating the need for additional LVT or HVT logic transistors.

**Source:**

https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

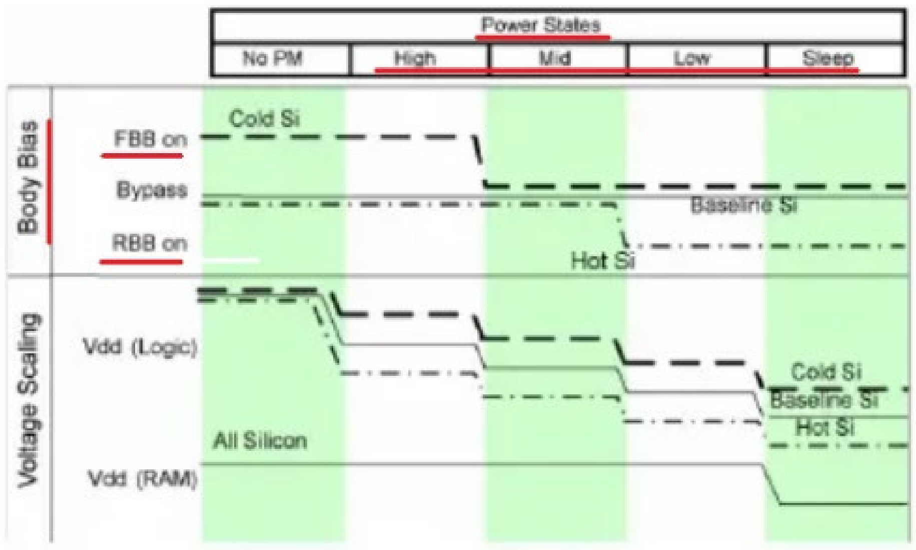


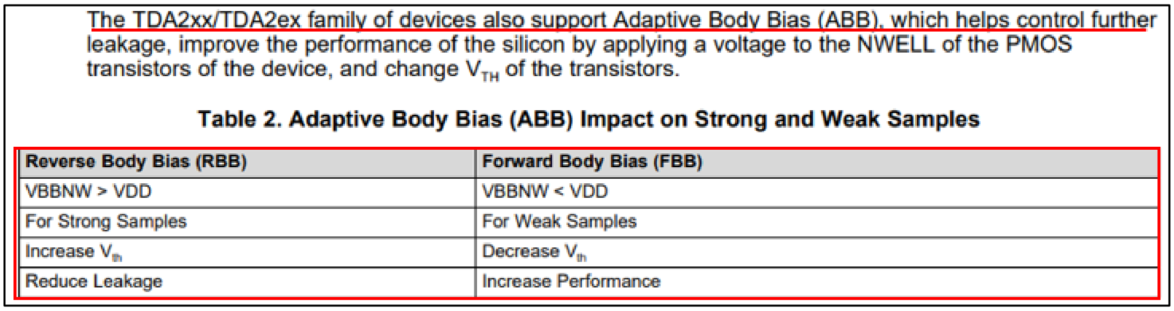
**Fig. 15.** ***Adaptivebody-biasand adaptive voltage scaling configuration with mode and process. Hot and cold refer the process strength or speed (fast and slow).***

**Source:**
https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

55. TI's TDA2x System-on-Chip devices include circuitry to selectively couple the body biasing voltage information from one of the plurality of data storage locations corresponding to one of the plurality of power modes to the voltage supply.

56. For example, TI's TDA2x System-on-Chip devices include a dual core processor module. The devices support Adaptive Body Bias (ABB). A body bias voltage is applied to the transistors of the device using ABB. ABB voltage values are available for each OPP (Operating Performance Point), and therefore, for each power state. The ABB voltage values corresponding to the different power states are stored in Efuse registers ("data storage locations"). Further, the body bias voltage value is supplied by the ABB LDO ("voltage supply") to the VBBNW pin of the transistors of the device's microprocessor. For example, to reduce leakage in low power states or low operating performance points (OPP Low), the reverse body bias voltage value

("body biasing voltage information") is accessed and applied to the device. Accordingly, the devices include circuitry for selectively coupling the body bias voltage information stored in the registers to the ABB LDO voltage supply.

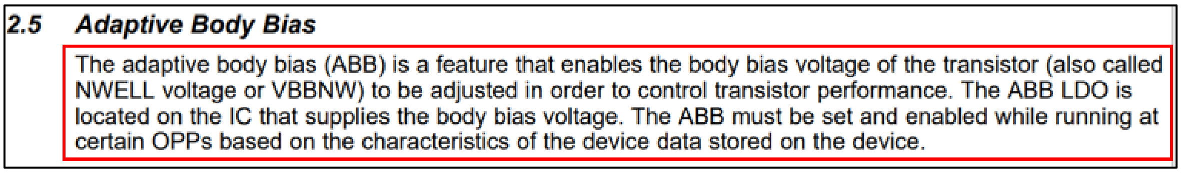


The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

**Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples**

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:**
https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)



**2.5 *Adaptive Body Bias***

The adaptive body bias (ABB) is a feature that enables the body bias voltage of the transistor (also called NWELL voltage or VBBNW) to be adjusted in order to control transistor performance. The ABB LDO is located on the IC that supplies the body bias voltage. The ABB must be set and enabled while running at certain OPPs based on the characteristics of the device data stored on the device.

**Source:**
https://www.ti.com/lit/an/slva646/slva646.pdf?ts=1597323545336&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)

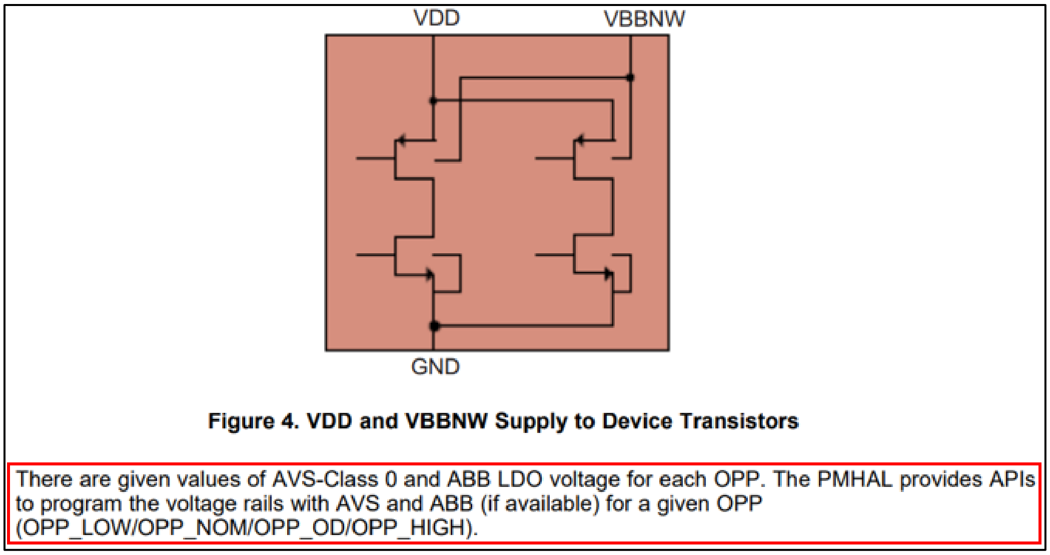
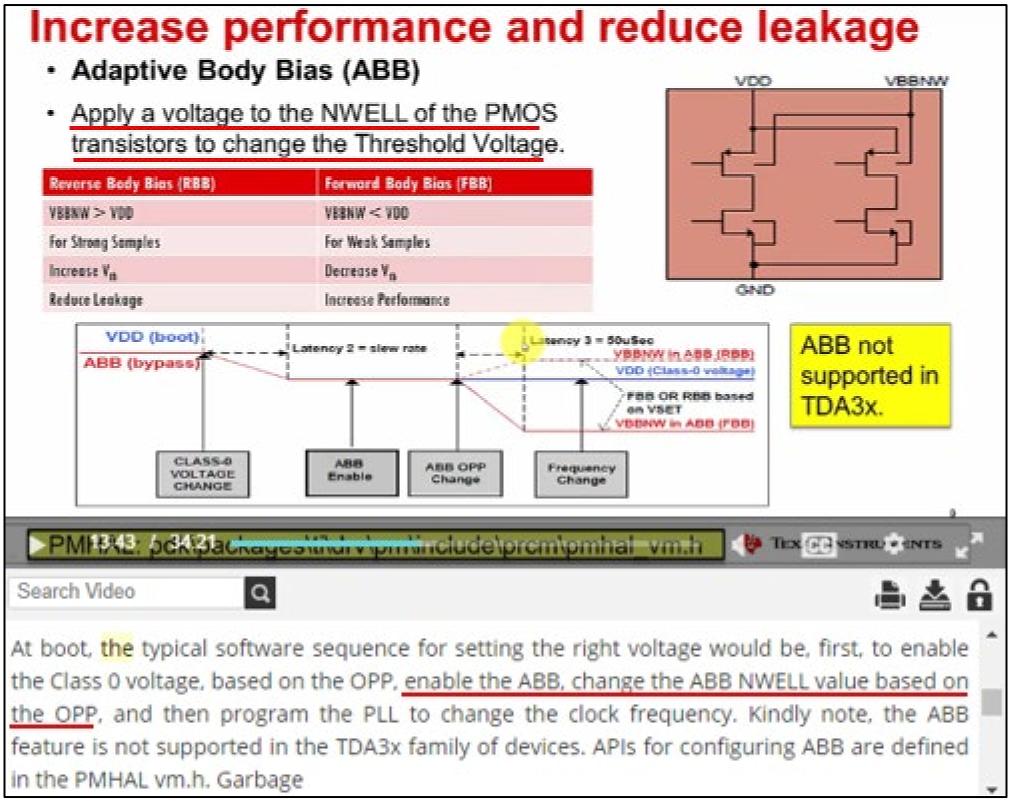


**Figure 4. VDD and VBBNW Supply to Device Transistors**

There are given values of AVS-Class 0 and ABB LDO voltage for each OPP. The PMHAL provides APIs to program the voltage rails with AVS and ABB (if available) for a given OPP (OPP_LOW/OPP_NOM/OPP_OD/OPP_HIGH).

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



## Increase performance and reduce leakage

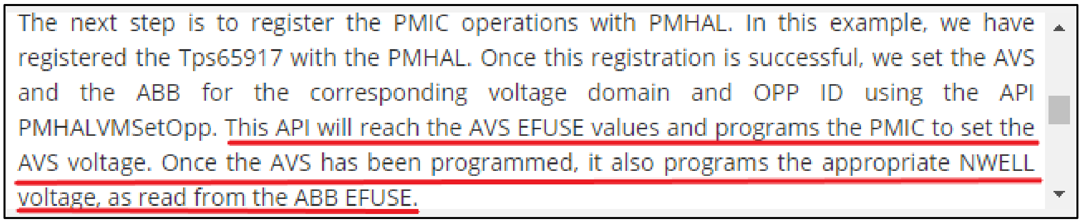
- **Adaptive Body Bias (ABB)**
- Apply a voltage to the NWELL of the PMOS transistors to change the Threshold Voltage.

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

ABB not supported in TDA3x.

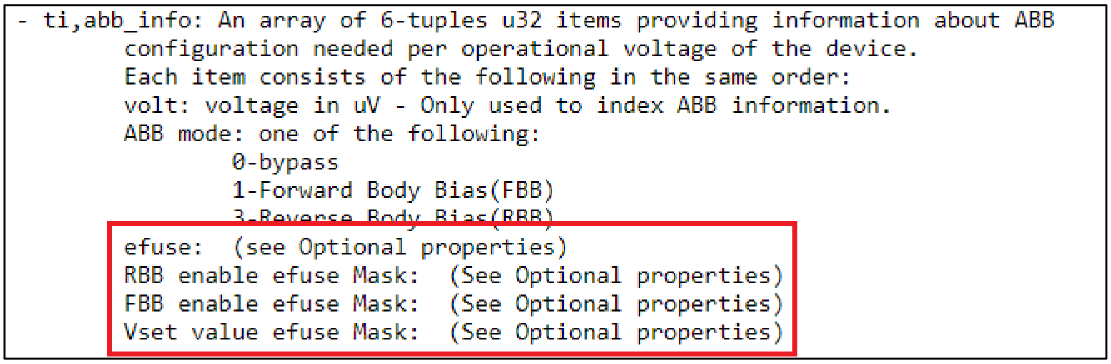
At boot, the typical software sequence for setting the right voltage would be, first, to enable the Class 0 voltage, based on the OPP, enable the ABB, change the ABB NWELL value based on the OPP, and then program the PLL to change the clock frequency. Kindly note, the ABB feature is not supported in the TDA3x family of devices. APIs for configuring ABB are defined in the PMHAL vm.h. Garbage

**Source:** https://training.ti.com/introduction-adas-power-management-software?cu=1137175 (at 13:43)



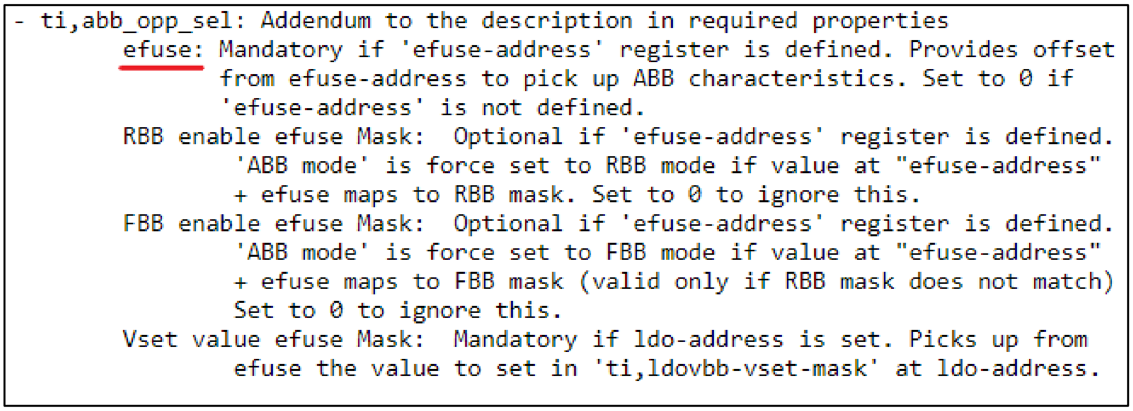
The next step is to register the PMIC operations with PMHAL. In this example, we have registered the Tps65917 with the PMHAL. Once this registration is successful, we set the AVS and the ABB for the corresponding voltage domain and OPP ID using the API PMHALVMSetOpp. This API will reach the AVS EFUSE values and programs the PMIC to set the AVS voltage. Once the AVS has been programmed, it also programs the appropriate NWELL voltage, as read from the ABB EFUSE.

**Source:** https://training.ti.com/introduction-adas-power-management-software (at 17:07)



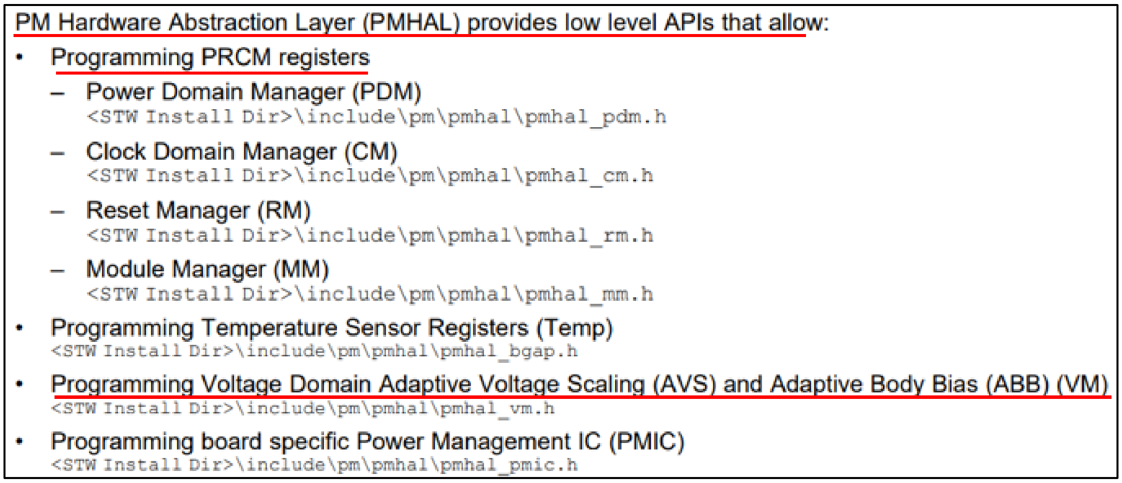
```
- ti,abb_info: An array of 6-tuples u32 items providing information about ABB
        configuration needed per operational voltage of the device.
        Each item consists of the following in the same order:
        volt: voltage in uV - Only used to index ABB information.
        ABB mode: one of the following:
                0-bypass
                1-Forward Body Bias(FBB)
                3-Reverse Body Bias(RBB)
        efuse:  (see Optional properties)
        RBB enable efuse Mask:  (See Optional properties)
        FBB enable efuse Mask:  (See Optional properties)
        Vset value efuse Mask:  (See Optional properties)
```

**Source:** https://www.kernel.org/doc/Documentation/devicetree/bindings/regulator/ti-abb-regulator.txt



```
- ti,abb_opp_sel: Addendum to the description in required properties
        efuse: Mandatory if 'efuse-address' register is defined. Provides offset
               from efuse-address to pick up ABB characteristics. Set to 0 if
               'efuse-address' is not defined.
        RBB enable efuse Mask:  Optional if 'efuse-address' register is defined.
                'ABB mode' is force set to RBB mode if value at "efuse-address"
                + efuse maps to RBB mask. Set to 0 to ignore this.
        FBB enable efuse Mask:  Optional if 'efuse-address' register is defined.
                'ABB mode' is force set to FBB mode if value at "efuse-address"
                + efuse maps to FBB mask (valid only if RBB mask does not match)
                Set to 0 to ignore this.
        Vset value efuse Mask:  Mandatory if ldo-address is set. Picks up from
                efuse the value to set in 'ti,ldovbb-vset-mask' at ldo-address.
```

**Source:** https://www.kernel.org/doc/Documentation/devicetree/bindings/regulator/ti-abb-regulator.txt

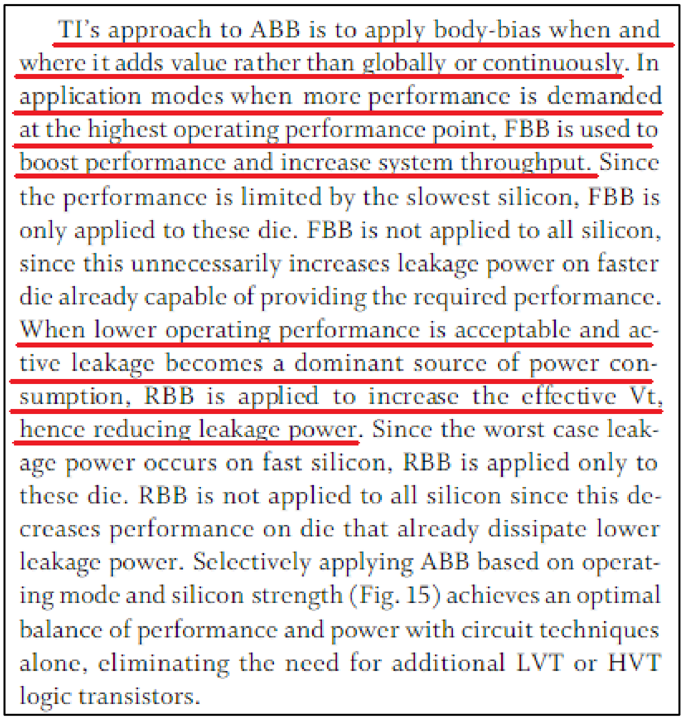


PM Hardware Abstraction Layer (PMHAL) provides low level APIs that allow:

- Programming PRCM registers
  - Power Domain Manager (PDM)
    <STW Install Dir>\include\pm\pmhal\pmhal_pdm.h
  - Clock Domain Manager (CM)
    <STW Install Dir>\include\pm\pmhal\pmhal_cm.h
  - Reset Manager (RM)
    <STW Install Dir>\include\pm\pmhal\pmhal_rm.h
  - Module Manager (MM)
    <STW Install Dir>\include\pm\pmhal\pmhal_mm.h
- Programming Temperature Sensor Registers (Temp)
  <STW Install Dir>\include\pm\pmhal\pmhal_bgap.h
- Programming Voltage Domain Adaptive Voltage Scaling (AVS) and Adaptive Body Bias (ABB) (VM)
  <STW Install Dir>\include\pm\pmhal\pmhal_vm.h
- Programming board specific Power Management IC (PMIC)
  <STW Install Dir>\include\pm\pmhal\pmhal_pmic.h

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)



TI's approach to ABB is to apply body-bias when and where it adds value rather than globally or continuously. In application modes when more performance is demanded at the highest operating performance point, FBB is used to boost performance and increase system throughput. Since the performance is limited by the slowest silicon, FBB is only applied to these die. FBB is not applied to all silicon, since this unnecessarily increases leakage power on faster die already capable of providing the required performance. When lower operating performance is acceptable and active leakage becomes a dominant source of power consumption, RBB is applied to increase the effective Vt, hence reducing leakage power. Since the worst case leakage power occurs on fast silicon, RBB is applied only to these die. RBB is not applied to all silicon since this decreases performance on die that already dissipate lower leakage power. Selectively applying ABB based on operating mode and silicon strength (Fig. 15) achieves an optimal balance of performance and power with circuit techniques alone, eliminating the need for additional LVT or HVT logic transistors.

**Source:**

https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

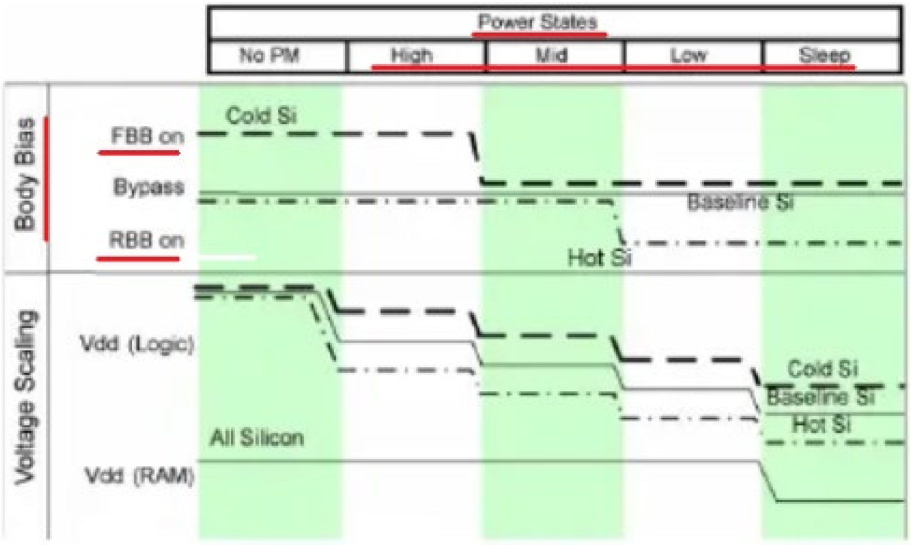


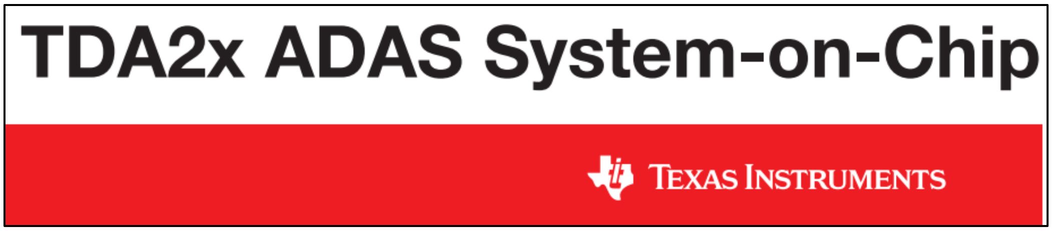
**Fig. 15.** ***Adaptivebody-biasand adaptive voltage scaling configuration with mode and process. Hot and cold refer the process strength or speed (fast and slow).***

**Source:**

https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

57. The computer system of TI's TDA2x System-on-Chip devices does not execute instructions in at least one of the plurality of power modes.

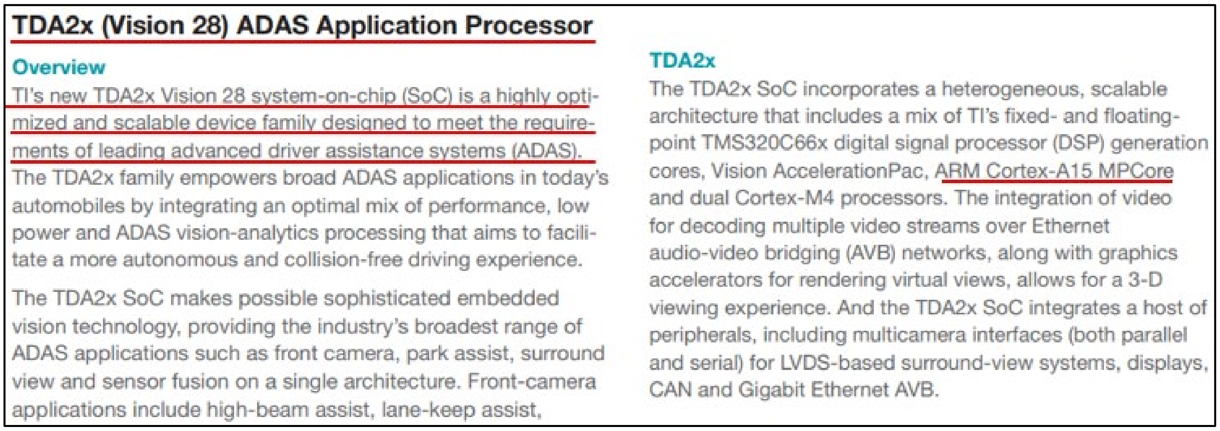
58. For example, TI's TDA2x System-on-Chip devices include a dual core processor module. The devices support Adaptive Body Bias (ABB). ABB voltage values are available for each OPP (Operating Performance Point). The devices support the lowest power mode in which one core is forced off and the other core is in retention (i.e., subsystem retention). During the subsystem retention mode, the computer system is unable to execute instructions and is in OPP_Low mode.



TDA2x ADAS System-on-Chip

Texas Instruments

**Source:** https://www.ti.com/lit/ml/sprt681/sprt681.pdf?ts=1597386693056&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 1)



TDA2x (Vision 28) ADAS Application Processor

Overview

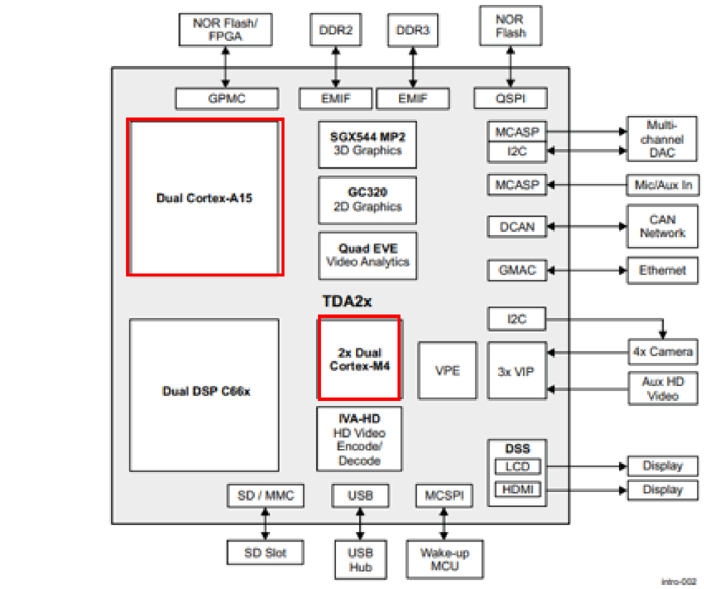
TI's new TDA2x Vision 28 system-on-chip (SoC) is a highly optimized and scalable device family designed to meet the requirements of leading advanced driver assistance systems (ADAS). The TDA2x family empowers broad ADAS applications in today's automobiles by integrating an optimal mix of performance, low power and ADAS vision-analytics processing that aims to facilitate a more autonomous and collision-free driving experience.

The TDA2x SoC makes possible sophisticated embedded vision technology, providing the industry's broadest range of ADAS applications such as front camera, park assist, surround view and sensor fusion on a single architecture. Front-camera applications include high-beam assist, lane-keep assist,

TDA2x

The TDA2x SoC incorporates a heterogeneous, scalable architecture that includes a mix of TI's fixed- and floating-point TMS320C66x digital signal processor (DSP) generation cores, Vision AccelerationPac, ARM Cortex-A15 MPCore and dual Cortex-M4 processors. The integration of video for decoding multiple video streams over Ethernet audio-video bridging (AVB) networks, along with graphics accelerators for rendering virtual views, allows for a 3-D viewing experience. And the TDA2x SoC integrates a host of peripherals, including multicamera interfaces (both parallel and serial) for LVDS-based surround-view systems, displays, CAN and Gigabit Ethernet AVB.

**Source:** https://m.eet.com/media/1241324/slyy044.pdf (Page 5)

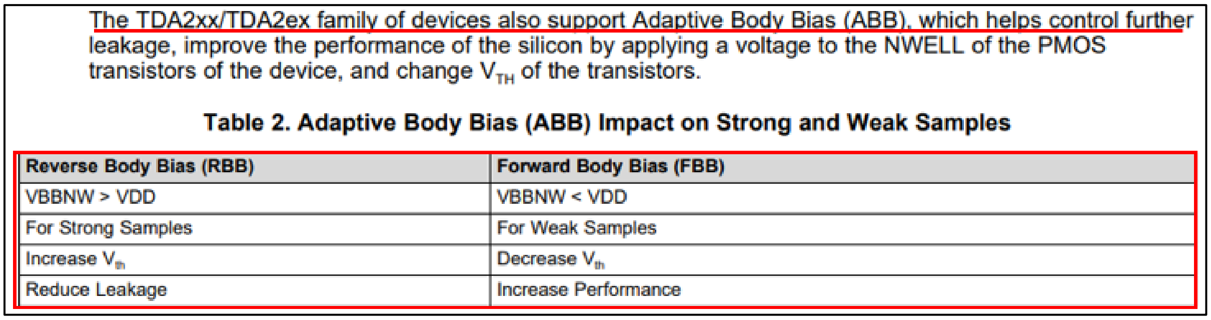


A The speeds shown are the highest design targeted, non-binned OPP.

Figure 1. TDA2xx Device Block Diagram

**Source:**

https://www.ti.com/lit/an/sprac21a/sprac21a.pdf?ts=1597386703137&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



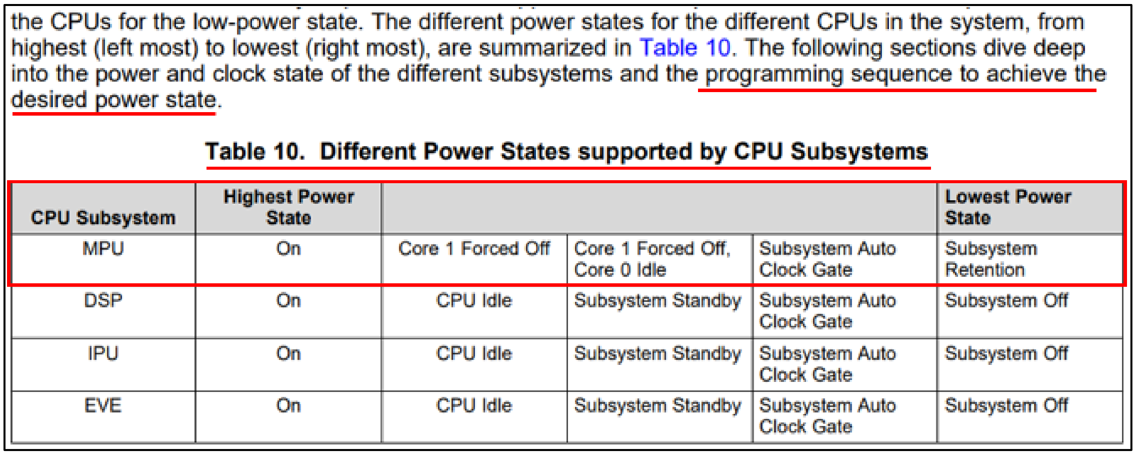
The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)



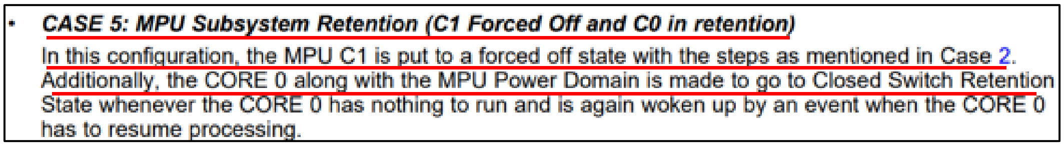
the CPUs for the low-power state. The different power states for the different CPUs in the system, from highest (left most) to lowest (right most), are summarized in Table 10. The following sections dive deep into the power and clock state of the different subsystems and the programming sequence to achieve the desired power state.

Table 10. Different Power States supported by CPU Subsystems

| CPU Subsystem | Highest Power State | | | | Lowest Power State |
|---|---|---|---|---|---|
| MPU | On | Core 1 Forced Off | Core 1 Forced Off, Core 0 Idle | Subsystem Auto Clock Gate | Subsystem Retention |
| DSP | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| IPU | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| EVE | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 18)



- ***CASE 5: MPU Subsystem Retention (C1 Forced Off and C0 in retention)***
  In this configuration, the MPU C1 is put to a forced off state with the steps as mentioned in Case 2. Additionally, the CORE 0 along with the MPU Power Domain is made to go to Closed Switch Retention State whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 23)

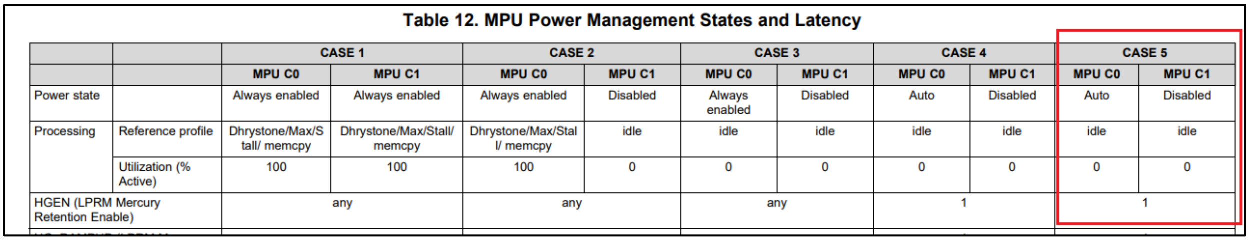
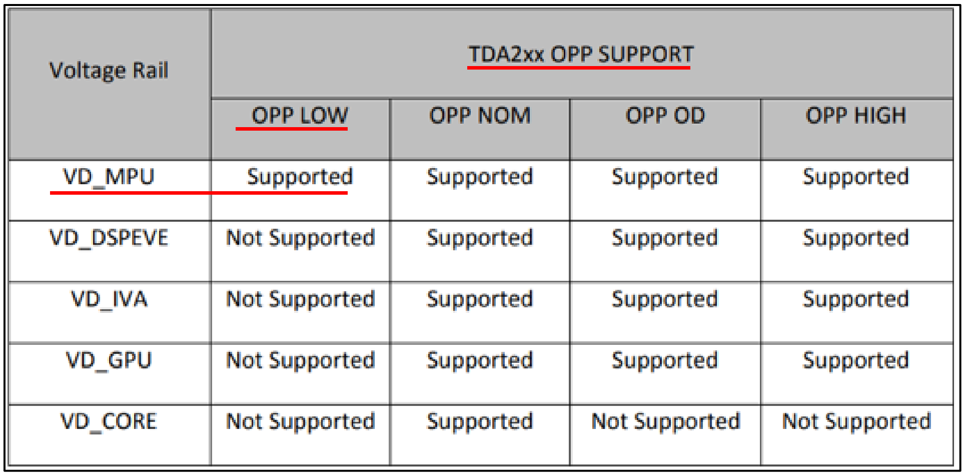


Table 12. MPU Power Management States and Latency

| | | CASE 1 | | CASE 2 | | CASE 3 | | CASE 4 | | CASE 5 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MPU C0 | MPU C1 | MPU C0 | MPU C1 | MPU C0 | MPU C1 | MPU C0 | MPU C1 | MPU C0 | MPU C1 |
| Power state | | Always enabled | Always enabled | Always enabled | Disabled | Always enabled | Disabled | Auto | Disabled | Auto | Disabled |
| Processing | Reference profile | Dhrystone/Max/Stall/ memcpy | Dhrystone/Max/Stall/ memcpy | Dhrystone/Max/Stall/ memcpy | idle | idle | idle | idle | idle | idle | idle |
| | Utilization (% Active) | 100 | 100 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| HGEN (LPRM Mercury Retention Enable) | | any | | any | | any | | 1 | | 1 | |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



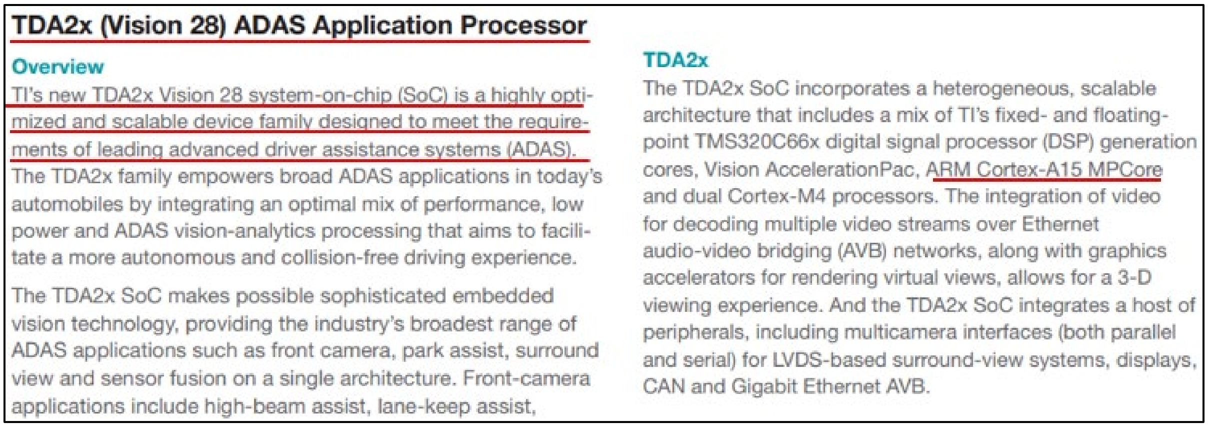
| Voltage Rail | TDA2xx OPP SUPPORT | | | |
|---|---|---|---|---|
| | OPP LOW | OPP NOM | OPP OD | OPP HIGH |
| VD_MPU | Supported | Supported | Supported | Supported |
| VD_DSPEVE | Not Supported | Supported | Supported | Supported |
| VD_IVA | Not Supported | Supported | Supported | Supported |
| VD_GPU | Not Supported | Supported | Supported | Supported |
| VD_CORE | Not Supported | Supported | Not Supported | Not Supported |

**Source:** https://processors.wiki.ti.com/images/b/b7/TDA_SBL_UserGuide.pdf (Page 23)

59. TI has infringed the '156 Patent by using the accused products and thereby practicing a method that comprises determining a desirable power condition, of a set of power conditions, of a computer system comprising a microprocessor, wherein the set of power conditions comprises a power down state.

60. For example, TI's TDA2x System-on-Chip devices include a dual core processor module. Further, the System-on-Chip ("computer system") includes an MPU_PRCM module for power management of the processor ("microprocessor"). The TDA2x family of devices supports multiple power states or "a set of power conditions" (e.g., a power state in which both cores are on, a power state in which one core is forced off, a power state in which one core is forced off and the other core is in retention' etc.).

61. The desired power state can be achieved through programming—that is, a command can be received by the device to change from one power state to a different power state. For example, the system can receive a command to change from a power state in which both cores are on to a power state in which one core is forced off and the other core is in retention. The power state in which one core is forced off and the other core is in retention is a low-power state ("power down state").



## TDA2x (Vision 28) ADAS Application Processor

### Overview

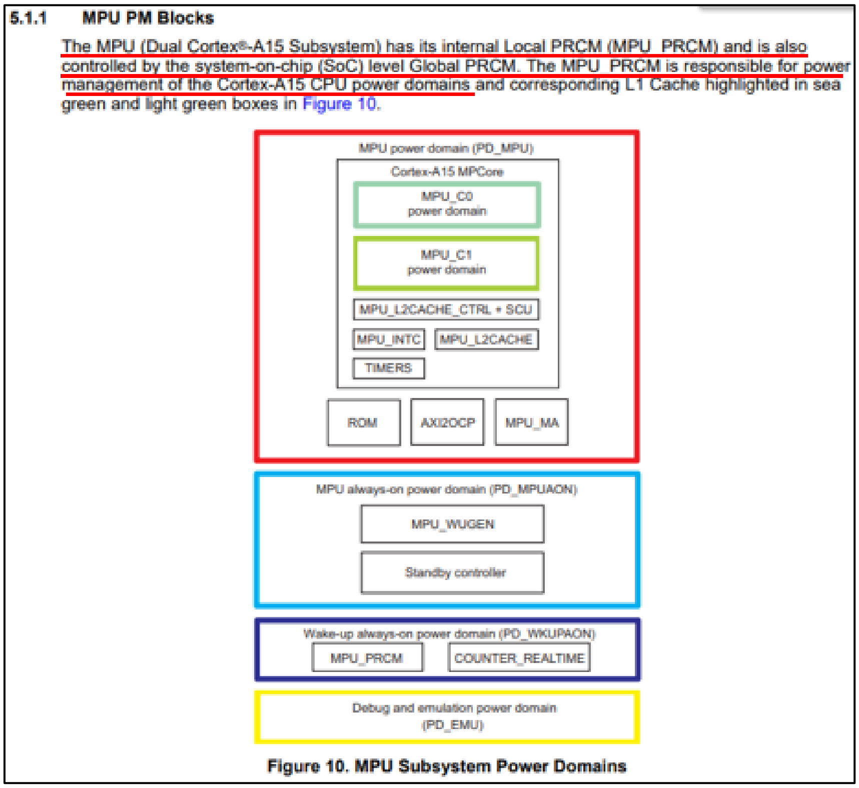
TI's new TDA2x Vision 28 system-on-chip (SoC) is a highly optimized and scalable device family designed to meet the requirements of leading advanced driver assistance systems (ADAS). The TDA2x family empowers broad ADAS applications in today's automobiles by integrating an optimal mix of performance, low power and ADAS vision-analytics processing that aims to facilitate a more autonomous and collision-free driving experience.

The TDA2x SoC makes possible sophisticated embedded vision technology, providing the industry's broadest range of ADAS applications such as front camera, park assist, surround view and sensor fusion on a single architecture. Front-camera applications include high-beam assist, lane-keep assist,

### TDA2x

The TDA2x SoC incorporates a heterogeneous, scalable architecture that includes a mix of TI's fixed- and floating-point TMS320C66x digital signal processor (DSP) generation cores, Vision AccelerationPac, ARM Cortex-A15 MPCore and dual Cortex-M4 processors. The integration of video for decoding multiple video streams over Ethernet audio-video bridging (AVB) networks, along with graphics accelerators for rendering virtual views, allows for a 3-D viewing experience. And the TDA2x SoC integrates a host of peripherals, including multicamera interfaces (both parallel and serial) for LVDS-based surround-view systems, displays, CAN and Gigabit Ethernet AVB.

**Source:** https://m.eet.com/media/1241324/slyy044.pdf (Page 5)



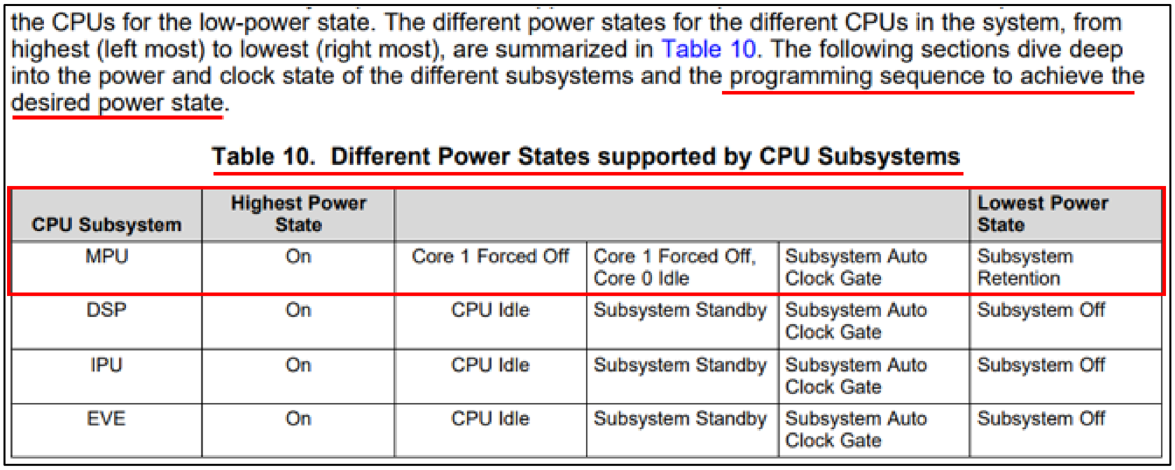
**5.1.1 MPU PM Blocks**

The MPU (Dual Cortex®-A15 Subsystem) has its internal Local PRCM (MPU_PRCM) and is also controlled by the system-on-chip (SoC) level Global PRCM. The MPU_PRCM is responsible for power management of the Cortex-A15 CPU power domains and corresponding L1 Cache highlighted in sea green and light green boxes in Figure 10.

**Figure 10. MPU Subsystem Power Domains**

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 19)



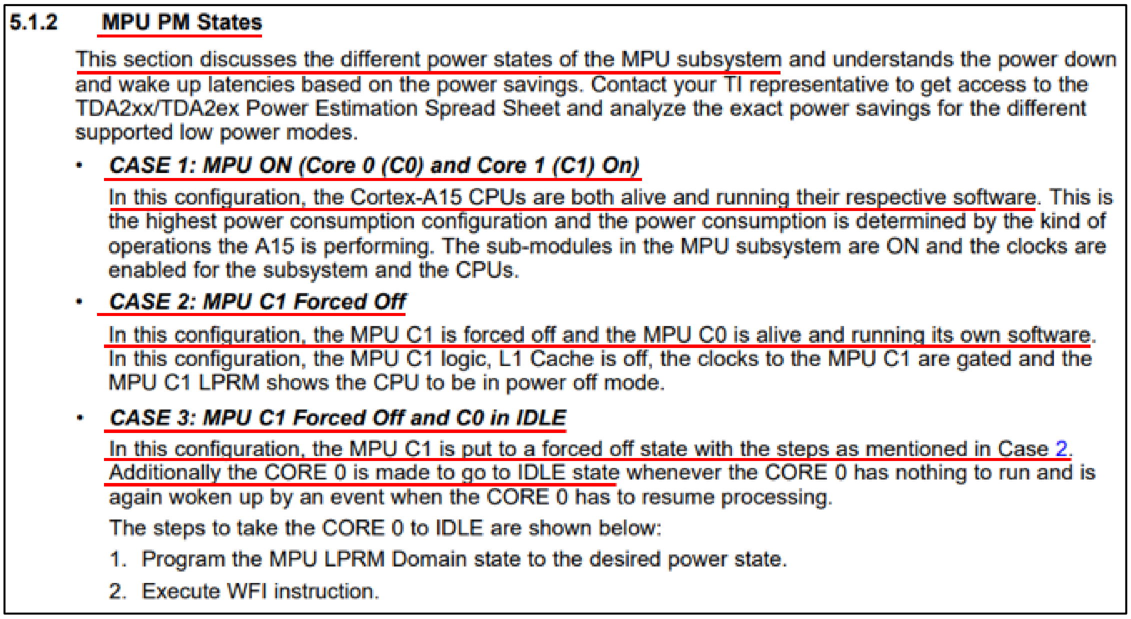
the CPUs for the low-power state. The different power states for the different CPUs in the system, from highest (left most) to lowest (right most), are summarized in Table 10. The following sections dive deep into the power and clock state of the different subsystems and the programming sequence to achieve the desired power state.

**Table 10. Different Power States supported by CPU Subsystems**

| CPU Subsystem | Highest Power State | | | | Lowest Power State |
|---|---|---|---|---|---|
| MPU | On | Core 1 Forced Off | Core 1 Forced Off, Core 0 Idle | Subsystem Auto Clock Gate | Subsystem Retention |
| DSP | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| IPU | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| EVE | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 18)



### 5.1.2 MPU PM States

This section discusses the different power states of the MPU subsystem and understands the power down and wake up latencies based on the power savings. Contact your TI representative to get access to the TDA2xx/TDA2ex Power Estimation Spread Sheet and analyze the exact power savings for the different supported low power modes.

- ***CASE 1: MPU ON (Core 0 (C0) and Core 1 (C1) On)***

  In this configuration, the Cortex-A15 CPUs are both alive and running their respective software. This is the highest power consumption configuration and the power consumption is determined by the kind of operations the A15 is performing. The sub-modules in the MPU subsystem are ON and the clocks are enabled for the subsystem and the CPUs.

- ***CASE 2: MPU C1 Forced Off***

  In this configuration, the MPU C1 is forced off and the MPU C0 is alive and running its own software. In this configuration, the MPU C1 logic, L1 Cache is off, the clocks to the MPU C1 are gated and the MPU C1 LPRM shows the CPU to be in power off mode.

- ***CASE 3: MPU C1 Forced Off and C0 in IDLE***

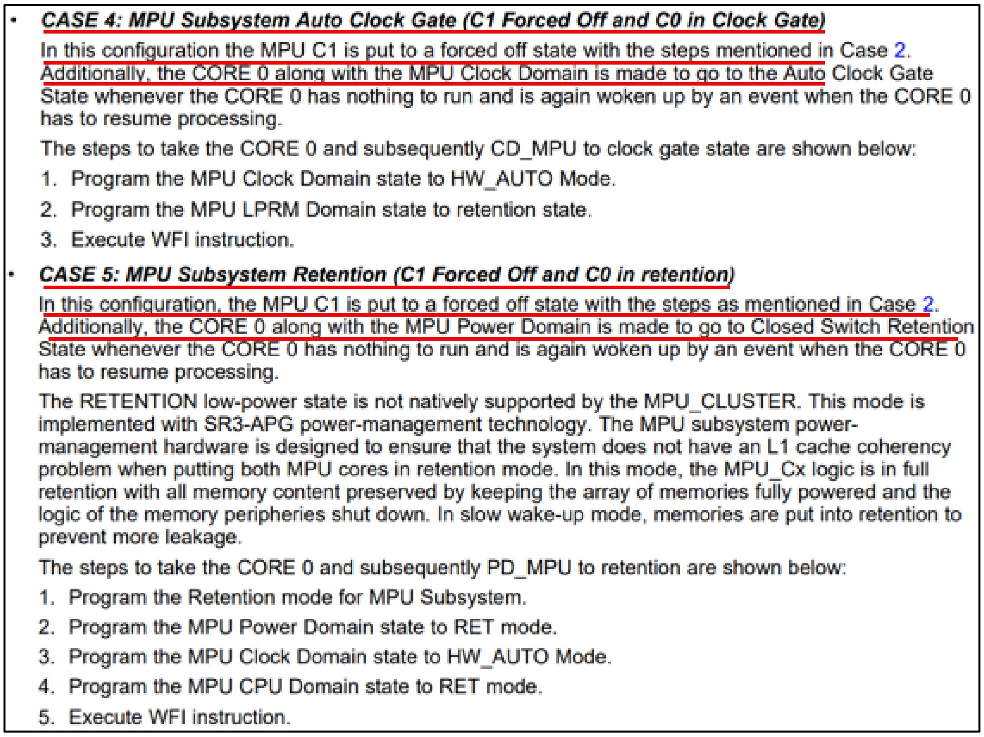
  In this configuration, the MPU C1 is put to a forced off state with the steps as mentioned in Case 2. Additionally the CORE 0 is made to go to IDLE state whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

  The steps to take the CORE 0 to IDLE are shown below:

  1. Program the MPU LPRM Domain state to the desired power state.
  2. Execute WFI instruction.

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Pages 20 and 21)



- ***CASE 4: MPU Subsystem Auto Clock Gate (C1 Forced Off and C0 in Clock Gate)***

  In this configuration the MPU C1 is put to a forced off state with the steps mentioned in Case 2. Additionally, the CORE 0 along with the MPU Clock Domain is made to go to the Auto Clock Gate State whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

  The steps to take the CORE 0 and subsequently CD_MPU to clock gate state are shown below:

  1. Program the MPU Clock Domain state to HW_AUTO Mode.
  2. Program the MPU LPRM Domain state to retention state.
  3. Execute WFI instruction.

- ***CASE 5: MPU Subsystem Retention (C1 Forced Off and C0 in retention)***

  In this configuration, the MPU C1 is put to a forced off state with the steps as mentioned in Case 2. Additionally, the CORE 0 along with the MPU Power Domain is made to go to Closed Switch Retention State whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

  The RETENTION low-power state is not natively supported by the MPU_CLUSTER. This mode is implemented with SR3-APG power-management technology. The MPU subsystem power-management hardware is designed to ensure that the system does not have an L1 cache coherency problem when putting both MPU cores in retention mode. In this mode, the MPU_Cx logic is in full retention with all memory content preserved by keeping the array of memories fully powered and the logic of the memory peripheries shut down. In slow wake-up mode, memories are put into retention to prevent more leakage.

  The steps to take the CORE 0 and subsequently PD_MPU to retention are shown below:

  1. Program the Retention mode for MPU Subsystem.
  2. Program the MPU Power Domain state to RET mode.
  3. Program the MPU Clock Domain state to HW_AUTO Mode.
  4. Program the MPU CPU Domain state to RET mode.
  5. Execute WFI instruction.

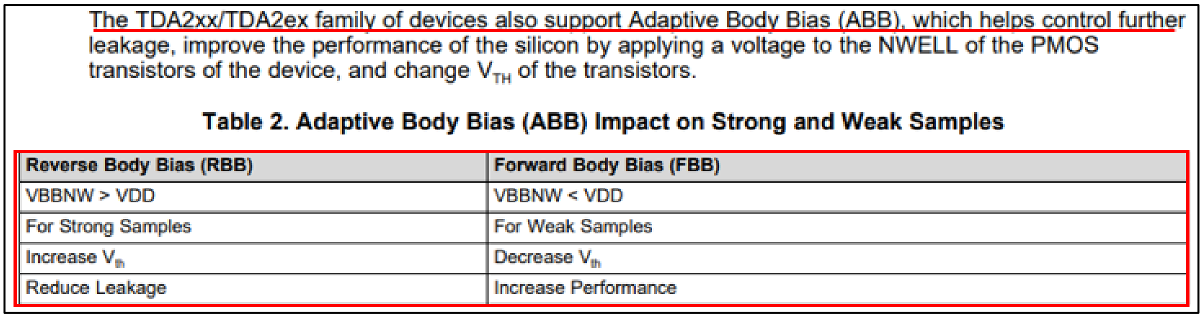
**Source:**
https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 23)

62. The method practiced using the accused products further comprises accessing body biasing voltage information corresponding to the power condition.

63. For example, TI's TDA2x System-on-Chip devices support Adaptive Body Bias (ABB). ABB voltage values are given for each OPP (Operating Performance Point), which is the operating condition for the device and is defined by the voltage and frequency applied to the processor. Further, the TDA2x family of devices supports multiple power states (e.g., a power state in which both cores are on, a power state in which one core is forced off, a power state in which one core is forced off and other core is in retention, etc.). An OPP is established based on the power state (i.e., operation) of the CPU.

64. Accordingly, the TDA2x family of devices stores the ABB voltage values associated with each OPP and corresponding power state. The ABB feature allows application

of forward body bias to improve performance and reverse body bias to reduce leakage in the device. To reduce leakage in low power states or low operating performance points (OPP Low), the reverse body bias voltage value ("body biasing voltage information") is accessed and applied to the device. This occurs during one of the low power states of the device, such as during the power state in which one core is forced off and the other core is in retention ("said power condition").



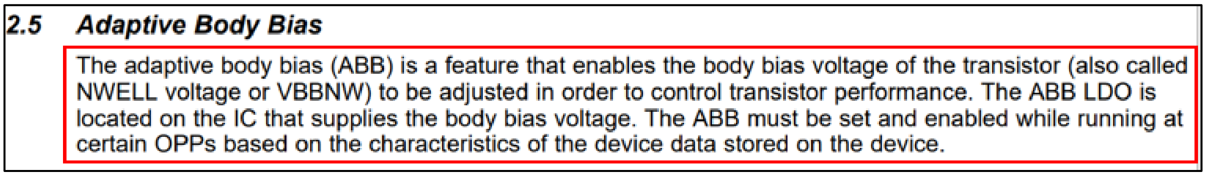
The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

**Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples**

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)



**2.5 *Adaptive Body Bias***

The adaptive body bias (ABB) is a feature that enables the body bias voltage of the transistor (also called NWELL voltage or VBBNW) to be adjusted in order to control transistor performance. The ABB LDO is located on the IC that supplies the body bias voltage. The ABB must be set and enabled while running at certain OPPs based on the characteristics of the device data stored on the device.

**Source:**

https://www.ti.com/lit/an/slva646/slva646.pdf?ts=1597323545336&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)

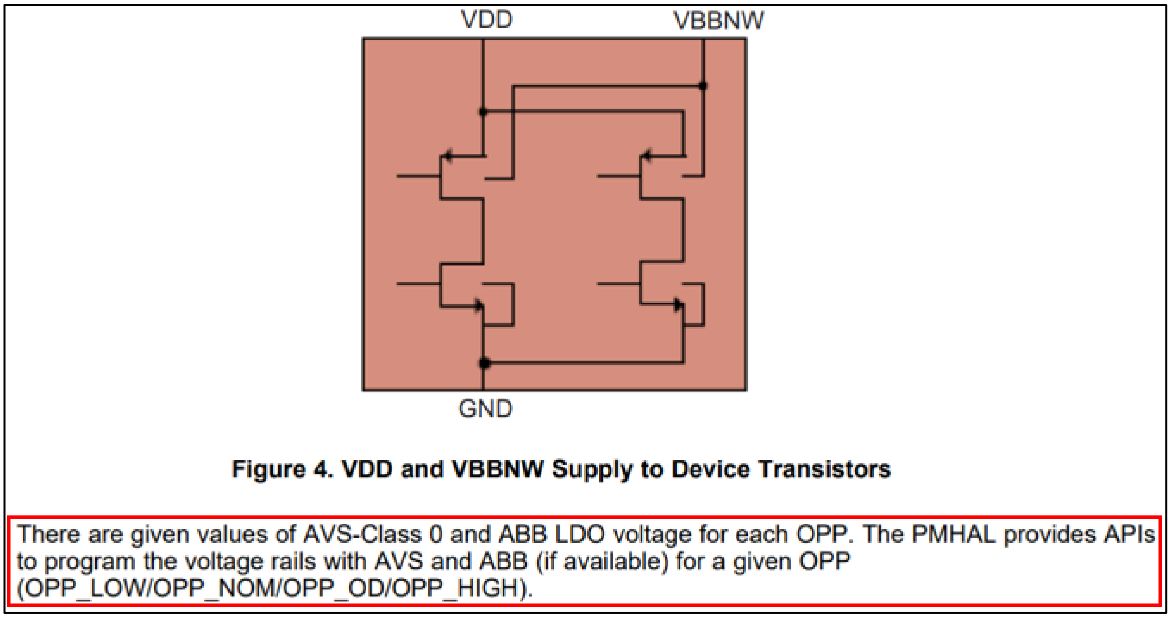
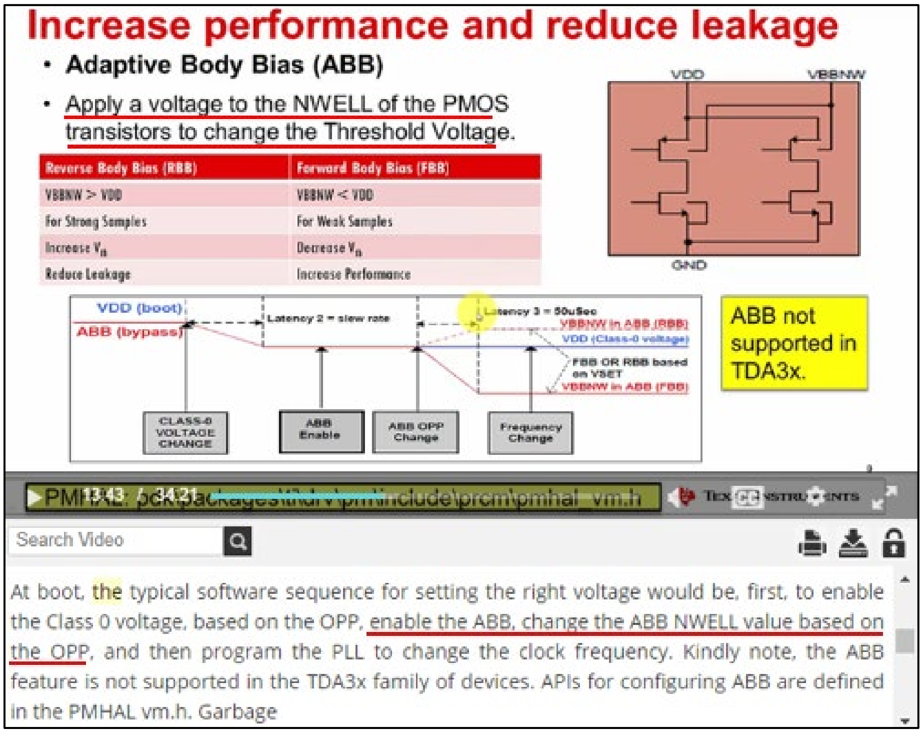


**Figure 4. VDD and VBBNW Supply to Device Transistors**

There are given values of AVS-Class 0 and ABB LDO voltage for each OPP. The PMHAL provides APIs to program the voltage rails with AVS and ABB (if available) for a given OPP (OPP_LOW/OPP_NOM/OPP_OD/OPP_HIGH).

**Source:** https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



Increase performance and reduce leakage

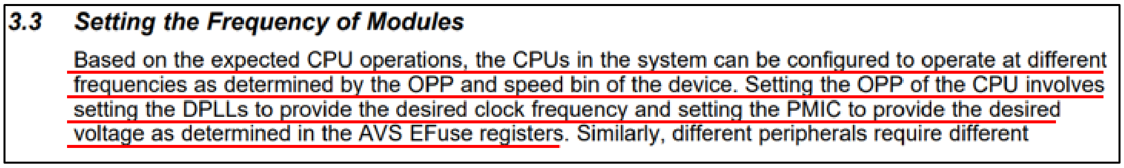
- Adaptive Body Bias (ABB)
- Apply a voltage to the NWELL of the PMOS transistors to change the Threshold Voltage.

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

ABB not supported in TDA3x.

At boot, the typical software sequence for setting the right voltage would be, first, to enable the Class 0 voltage, based on the OPP, enable the ABB, change the ABB NWELL value based on the OPP, and then program the PLL to change the clock frequency. Kindly note, the ABB feature is not supported in the TDA3x family of devices. APIs for configuring ABB are defined in the PMHAL vm.h. Garbage

**Source:** https://training.ti.com/introduction-adas-power-management-software?cu=1137175 (at 13:43)



***3.3 Setting the Frequency of Modules***

Based on the expected CPU operations, the CPUs in the system can be configured to operate at different frequencies as determined by the OPP and speed bin of the device. Setting the OPP of the CPU involves setting the DPLLs to provide the desired clock frequency and setting the PMIC to provide the desired voltage as determined in the AVS EFuse registers. Similarly, different peripherals require different

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 9)

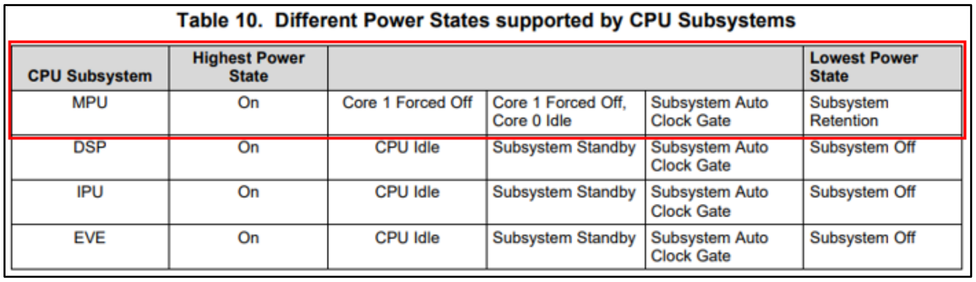
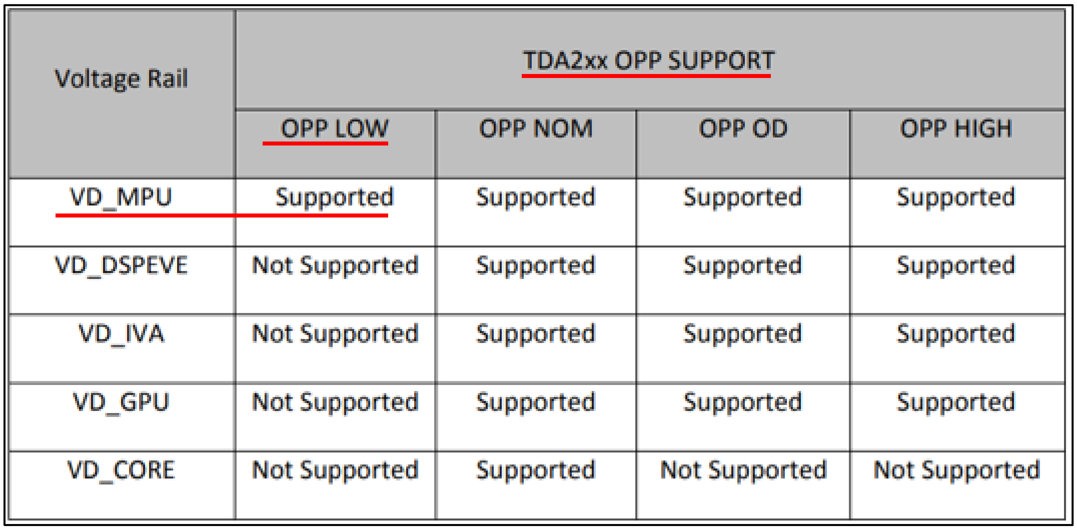


**Table 10. Different Power States supported by CPU Subsystems**

| CPU Subsystem | Highest Power State | | | | Lowest Power State |
|---|---|---|---|---|---|
| MPU | On | Core 1 Forced Off | Core 1 Forced Off, Core 0 Idle | Subsystem Auto Clock Gate | Subsystem Retention |
| DSP | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| IPU | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| EVE | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |

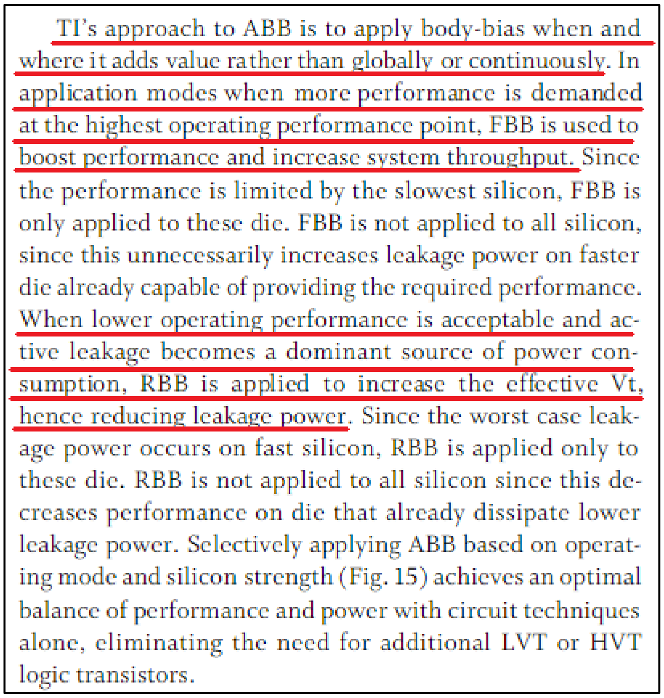
**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 18)



| Voltage Rail | TDA2xx OPP SUPPORT | | | |
|---|---|---|---|---|
| | OPP LOW | OPP NOM | OPP OD | OPP HIGH |
| VD_MPU | Supported | Supported | Supported | Supported |
| VD_DSPEVE | Not Supported | Supported | Supported | Supported |
| VD_IVA | Not Supported | Supported | Supported | Supported |
| VD_GPU | Not Supported | Supported | Supported | Supported |
| VD_CORE | Not Supported | Supported | Not Supported | Not Supported |

**Source:** https://processors.wiki.ti.com/images/b/b7/TDA_SBL_UserGuide.pdf (Page 23)



TI's approach to ABB is to apply body-bias when and where it adds value rather than globally or continuously. In application modes when more performance is demanded at the highest operating performance point, FBB is used to boost performance and increase system throughput. Since the performance is limited by the slowest silicon, FBB is only applied to these die. FBB is not applied to all silicon, since this unnecessarily increases leakage power on faster die already capable of providing the required performance. When lower operating performance is acceptable and active leakage becomes a dominant source of power consumption, RBB is applied to increase the effective Vt, hence reducing leakage power. Since the worst case leakage power occurs on fast silicon, RBB is applied only to these die. RBB is not applied to all silicon since this decreases performance on die that already dissipate lower leakage power. Selectively applying ABB based on operating mode and silicon strength (Fig. 15) achieves an optimal balance of performance and power with circuit techniques alone, eliminating the need for additional LVT or HVT logic transistors.

**Source:**

https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

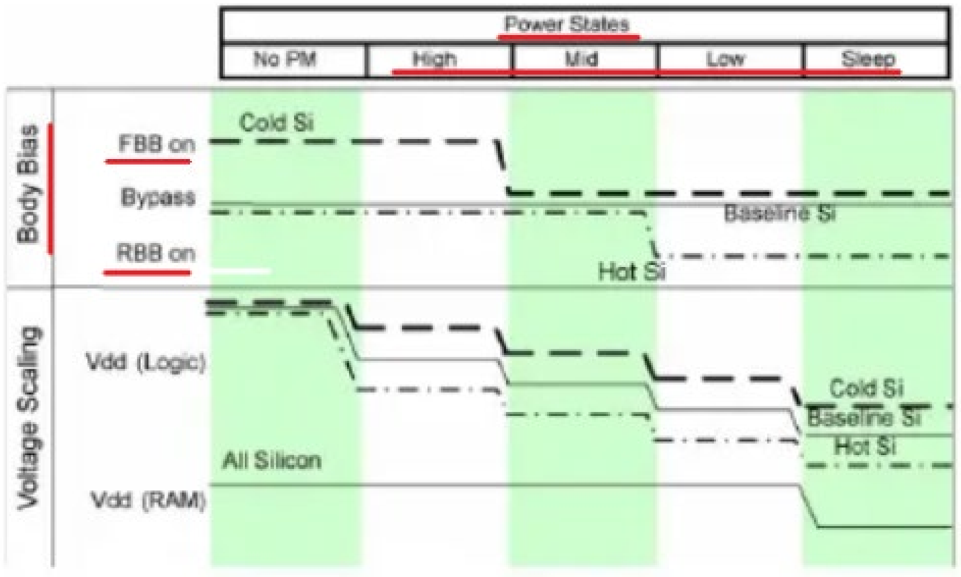


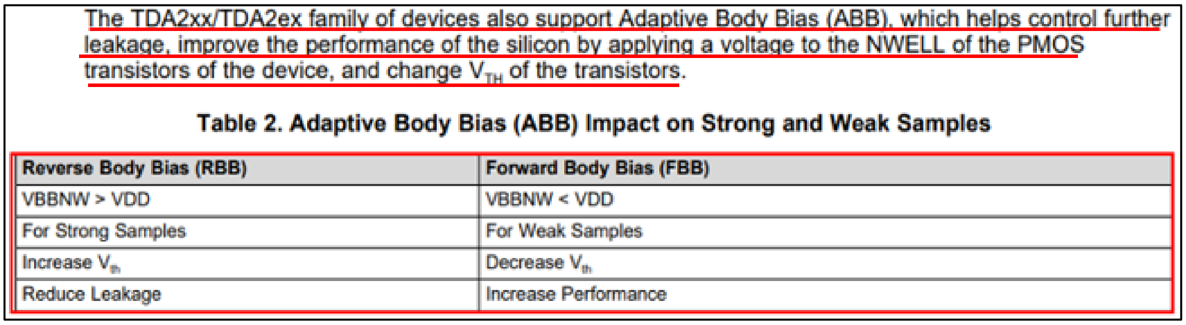
***Fig. 15. Adaptivebody-biasand adaptive voltage scaling configuration with mode and process. Hot and cold refer the process strength or speed (fast and slow).***

**Source:**
https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

65. The method practiced using the accused products further comprises commanding a voltage supply coupled to a body terminal of the microprocessor to generate a voltage corresponding to the body biasing voltage information corresponding to the power condition.

66. For example, TI's TDA2x System-on-Chip devices include a dual core processor module ("microprocessor"). The devices support Adaptive Body Bias (ABB), which allows application of forward body bias to increase performance and reverse body bias to reduce leakage in the device. The ABB voltage value can be programmed—that is, a command can be generated by the device to apply a body bias voltage value. Further, the programmed reverse body bias voltage value is supplied by the ABB LDO ("voltage supply") to the VBBNW pin ("body terminal") of the transistors of the device's microprocessor.

67. To reduce leakage in low power states or low operating performance points (OPP Low), the reverse body bias voltage value ("body biasing voltage information") is accessed and applied to the device. This occurs during one of the low power states of the device, such as the power state in which one core is forced off and the other core is in retention ("said power condition"). Accordingly, the device can command the ABB LDO, which is coupled to the VBBNW pin, to supply a reverse body bias voltage corresponding to the low power state, i.e., the power state in which one core is forced off and other core is in retention ("said power condition").

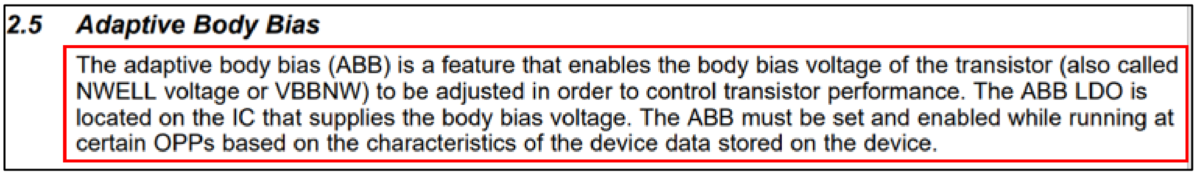


The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:** https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)



**2.5 *Adaptive Body Bias***

The adaptive body bias (ABB) is a feature that enables the body bias voltage of the transistor (also called NWELL voltage or VBBNW) to be adjusted in order to control transistor performance. The ABB LDO is located on the IC that supplies the body bias voltage. The ABB must be set and enabled while running at certain OPPs based on the characteristics of the device data stored on the device.

**Source:** https://www.ti.com/lit/an/slva646/slva646.pdf?ts=1597323545336&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)

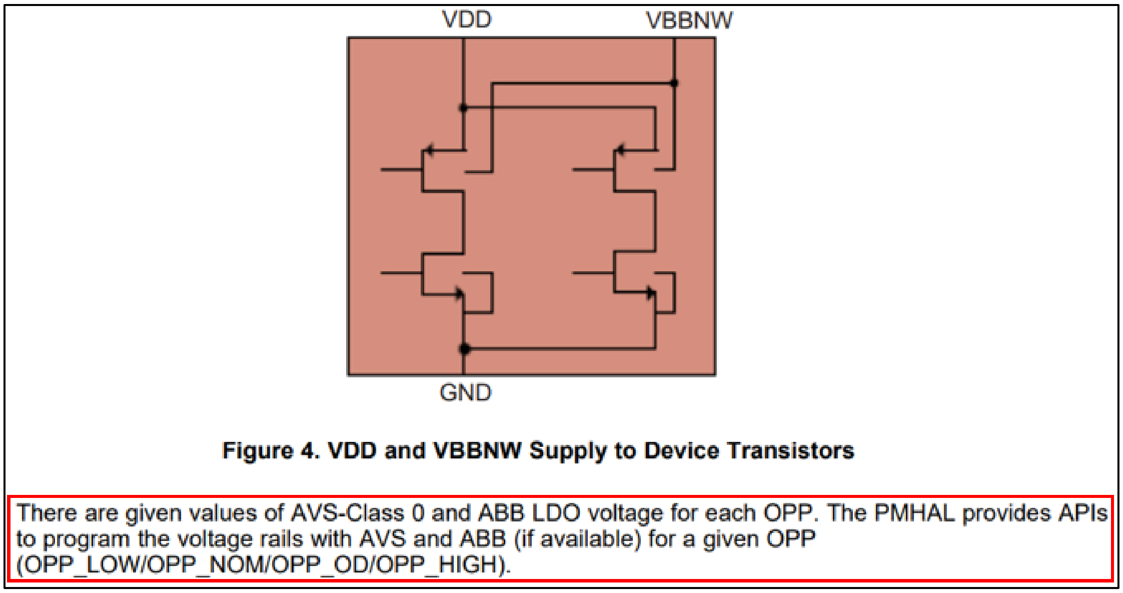
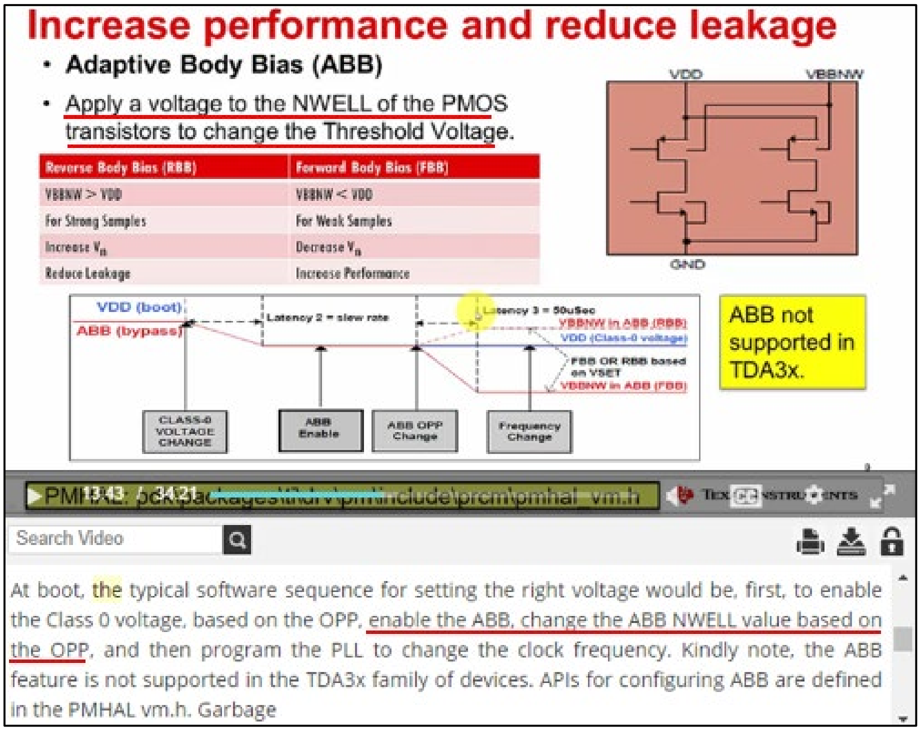


**Figure 4. VDD and VBBNW Supply to Device Transistors**

There are given values of AVS-Class 0 and ABB LDO voltage for each OPP. The PMHAL provides APIs to program the voltage rails with AVS and ABB (if available) for a given OPP (OPP_LOW/OPP_NOM/OPP_OD/OPP_HIGH).

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



## Increase performance and reduce leakage

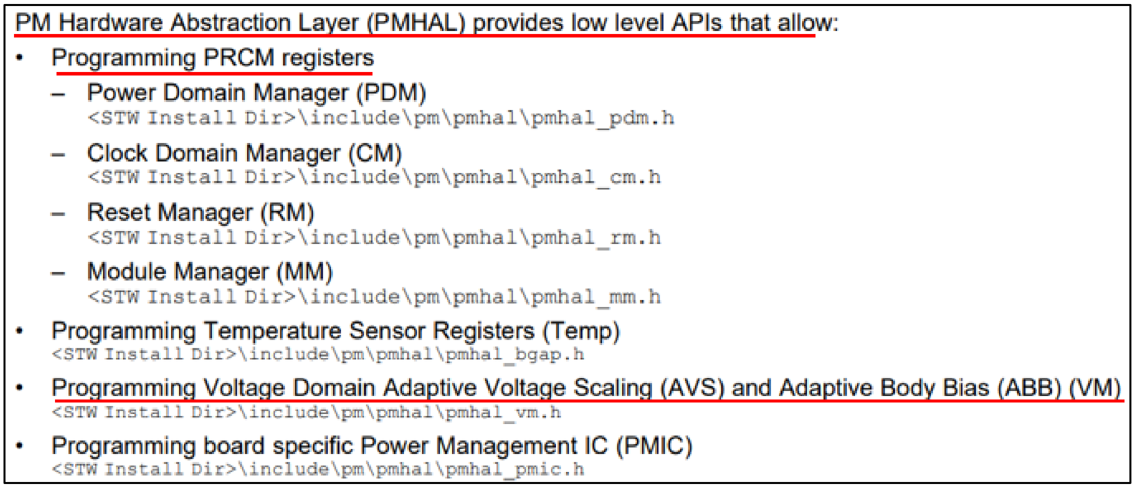
- **Adaptive Body Bias (ABB)**
- Apply a voltage to the NWELL of the PMOS transistors to change the Threshold Voltage.

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

ABB not supported in TDA3x.

At boot, the typical software sequence for setting the right voltage would be, first, to enable the Class 0 voltage, based on the OPP, enable the ABB, change the ABB NWELL value based on the OPP, and then program the PLL to change the clock frequency. Kindly note, the ABB feature is not supported in the TDA3x family of devices. APIs for configuring ABB are defined in the PMHAL vm.h. Garbage

**Source:** https://training.ti.com/introduction-adas-power-management-software?cu=1137175 (at 13:43)

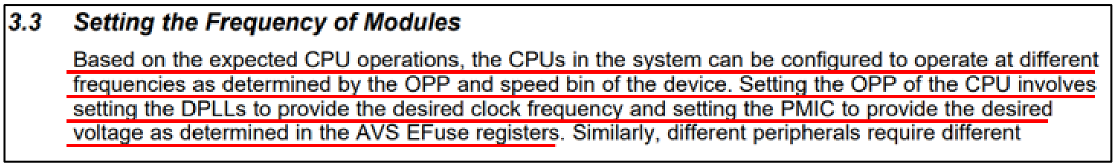


PM Hardware Abstraction Layer (PMHAL) provides low level APIs that allow:

- Programming PRCM registers
  - Power Domain Manager (PDM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_pdm.h`
  - Clock Domain Manager (CM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_cm.h`
  - Reset Manager (RM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_rm.h`
  - Module Manager (MM)
    `<STW Install Dir>\include\pm\pmhal\pmhal_mm.h`
- Programming Temperature Sensor Registers (Temp)
  `<STW Install Dir>\include\pm\pmhal\pmhal_bgap.h`
- Programming Voltage Domain Adaptive Voltage Scaling (AVS) and Adaptive Body Bias (ABB) (VM)
  `<STW Install Dir>\include\pm\pmhal\pmhal_vm.h`
- Programming board specific Power Management IC (PMIC)
  `<STW Install Dir>\include\pm\pmhal\pmhal_pmic.h`

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)



### 3.3 *Setting the Frequency of Modules*

Based on the expected CPU operations, the CPUs in the system can be configured to operate at different frequencies as determined by the OPP and speed bin of the device. Setting the OPP of the CPU involves setting the DPLLs to provide the desired clock frequency and setting the PMIC to provide the desired voltage as determined in the AVS EFuse registers. Similarly, different peripherals require different

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 9)

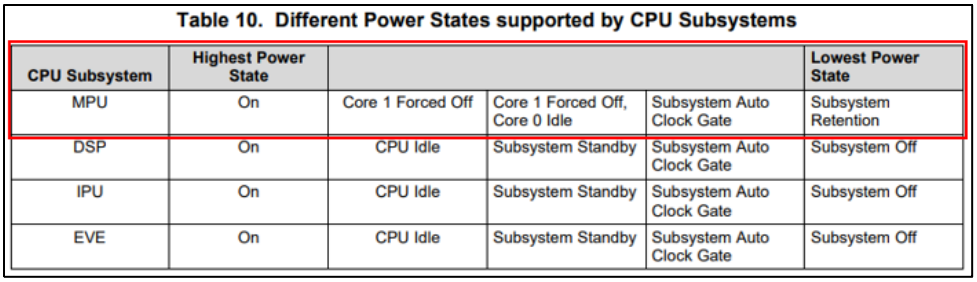
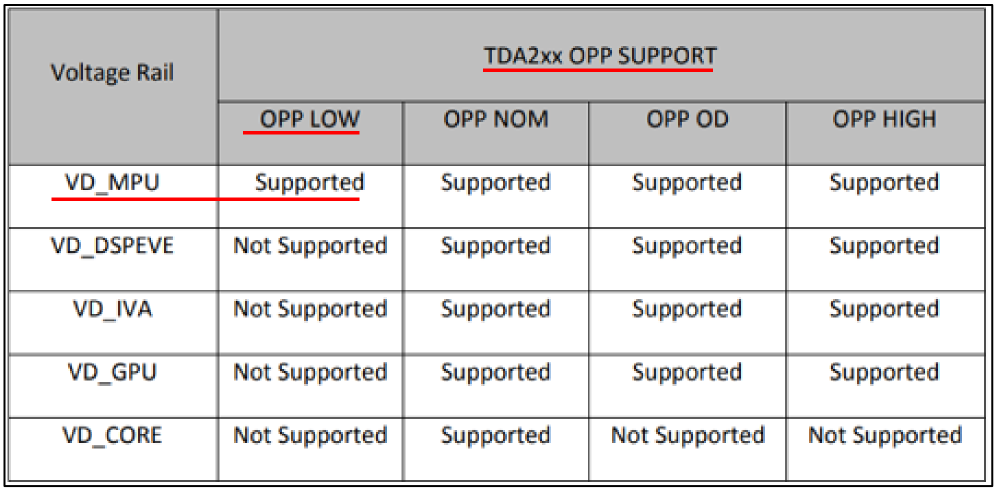


**Table 10. Different Power States supported by CPU Subsystems**

| CPU Subsystem | Highest Power State | | | | Lowest Power State |
|---|---|---|---|---|---|
| MPU | On | Core 1 Forced Off | Core 1 Forced Off, Core 0 Idle | Subsystem Auto Clock Gate | Subsystem Retention |
| DSP | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| IPU | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| EVE | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |

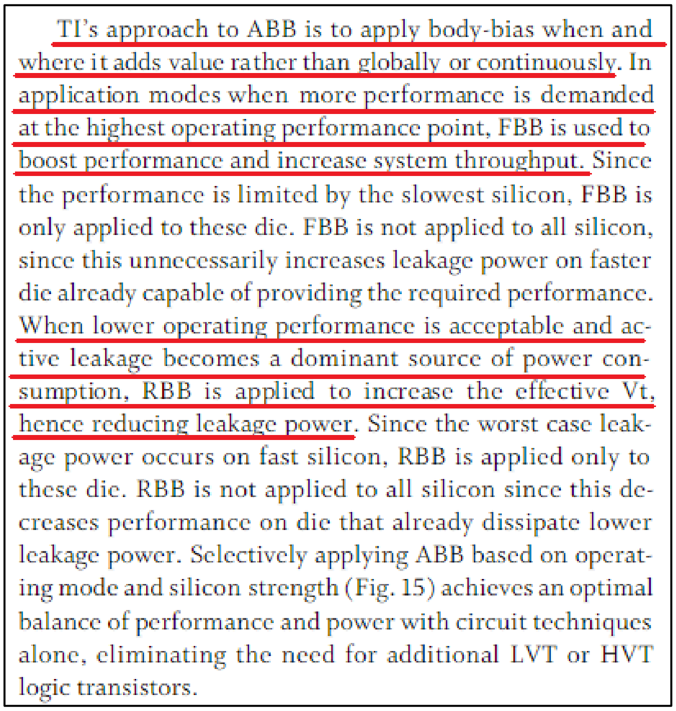
**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 18)



| Voltage Rail | TDA2xx OPP SUPPORT | | | |
|---|---|---|---|---|
| | OPP LOW | OPP NOM | OPP OD | OPP HIGH |
| VD_MPU | Supported | Supported | Supported | Supported |
| VD_DSPEVE | Not Supported | Supported | Supported | Supported |
| VD_IVA | Not Supported | Supported | Supported | Supported |
| VD_GPU | Not Supported | Supported | Supported | Supported |
| VD_CORE | Not Supported | Supported | Not Supported | Not Supported |

**Source:** https://processors.wiki.ti.com/images/b/b7/TDA_SBL_UserGuide.pdf (Page 23)



TI's approach to ABB is to apply body-bias when and where it adds value rather than globally or continuously. In application modes when more performance is demanded at the highest operating performance point, FBB is used to boost performance and increase system throughput. Since the performance is limited by the slowest silicon, FBB is only applied to these die. FBB is not applied to all silicon, since this unnecessarily increases leakage power on faster die already capable of providing the required performance. When lower operating performance is acceptable and active leakage becomes a dominant source of power consumption, RBB is applied to increase the effective Vt, hence reducing leakage power. Since the worst case leakage power occurs on fast silicon, RBB is applied only to these die. RBB is not applied to all silicon since this decreases performance on die that already dissipate lower leakage power. Selectively applying ABB based on operating mode and silicon strength (Fig. 15) achieves an optimal balance of performance and power with circuit techniques alone, eliminating the need for additional LVT or HVT logic transistors.

**Source:**

https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

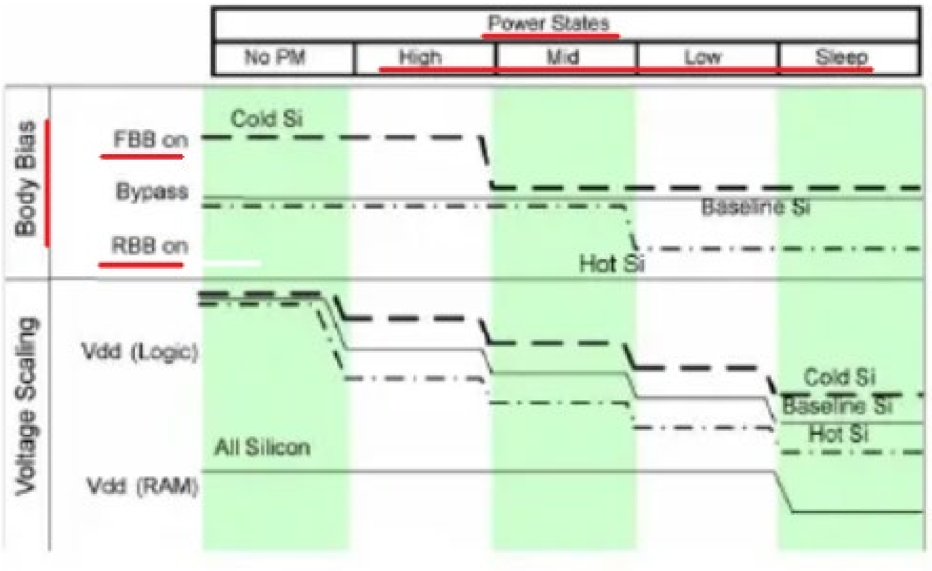


**Fig. 15.** ***Adaptivebody-biasand adaptive voltage scaling configuration with mode and process. Hot and cold refer the process strength or speed (fast and slow).***

**Source:**
https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

68. TI has had knowledge of the ’156 Patent at least as of the date when it was notified of the filing of this action.

69. Liberty Patents has been damaged as a result of the infringing conduct by TI alleged above. Thus, TI is liable to Liberty Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

70. Liberty Patents and/or its predecessors-in-interest have satisfied all statutory obligations required to collect pre-filing damages for the full period allowed by law for infringement of the ’156 Patent.

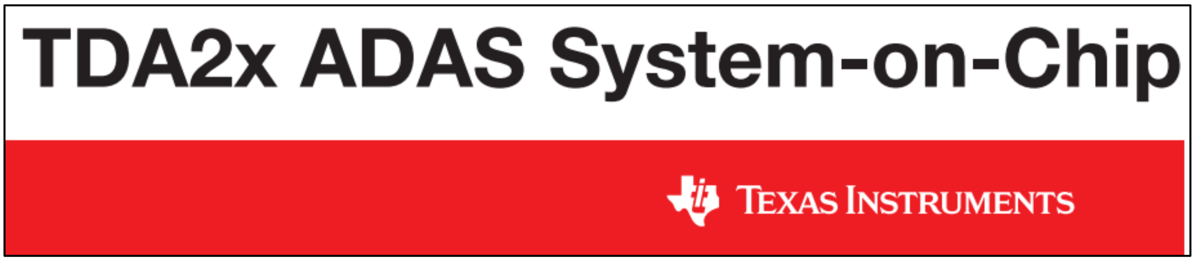
## COUNT III

## DIRECT INFRINGEMENT OF U.S. PATENT NO. 8,458,496

71. On June 4, 2013, U.S. Patent No. 8,458,496 ("the '496 Patent") was duly and legally issued by the United States Patent and Trademark Office for an invention entitled "Systems and Methods for Control of Integrated Circuits Comprising Body Biasing Systems."

72. Liberty Patents is the owner of the '496 Patent, with all substantive rights in and to that patent, including the sole and exclusive right to prosecute this action and enforce the '496 Patent against infringers, and to collect damages for all relevant times.

73. TI made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, TI's TDA2x System-on-Chip devices and other products[8] that are capable of applying body biasing voltages during a power mode of a processor or other integrated circuit ("accused products"):



TDA2x ADAS System-on-Chip

Texas Instruments

[8] *See, e.g.*, TI AM5706, AM5708, AM5716, AM5718, AM5718-HIREL, AM5726, AM5728, AM5729, AM5746, AM5748, AM5749, DRA710, DRA712, DRA714, DRA716, DRA718, DRA722, DRA724, DRA725, DRA726, DRA744, DRA745, DRA746, DRA74P, DRA750, DRA756, DRA75P, DRA76P, DRA77P, DRA790, DRA791, DRA793, DRA797, OMAP4430, OMAP4460, OMAP4470, OMAP5430, OMAP5432, TDA2E, TDA2EG-17, TDA2HF, TDA2HG, TDA2HV, TDA2LF, TDA2P-ABZ, TDA2P-ACD, TDA2SA, TDA2SG, TDA2SX, TMDXIDK5718, TMDSEVM572X, TMDSIDK572, TMDSIDK574, D3-3P-ADAS-DK, D3-3P-TDAX-DK, DRA71XEVM, DRA72XEVM, DRA79XEVM, 703664-1001, 705851-1001, 705852-1001, DRA76XP-DRA77XP-TDA2PX-ACD-CPUBOARD, J6EVM5777, OMAP5432-EVM, TDA2EXEVM, TDA2PXEVM, EVM5777BG-03-00-00, EVM5777G-03-40-00, EVM5777VISION-V2-0, etc.

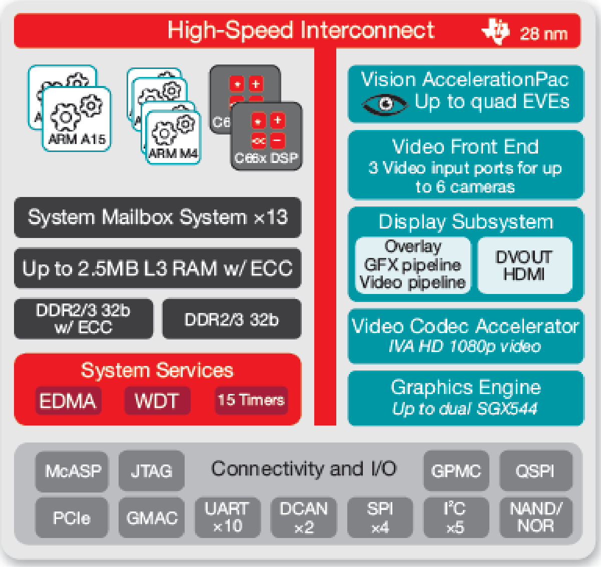


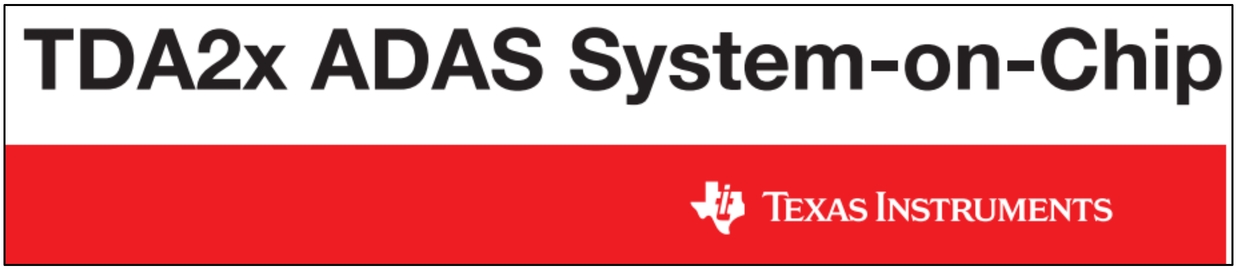
*Figure 1. Block diagram for TDA2x SoC*

**Source:**
https://www.ti.com/lit/ml/sprt681/sprt681.pdf?ts=1597386693056&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 1)

74. By doing so, TI has directly infringed (literally and/or under the doctrine of equivalents) at least Claims 1 and 9 of the '496 Patent. TI's infringement in this regard is ongoing.

75. TI's TDA2x System-on-Chip devices are exemplary accused products. The devices include a computer system

76. For example, TI's TDA2x System-on-Chip devices include a dual core processor module along with different memory elements and peripherals ("computer system"). The devices support Adaptive Body Bias (ABB), which can apply a forward or reverse body bias voltage to the transistors of the device using ABB.

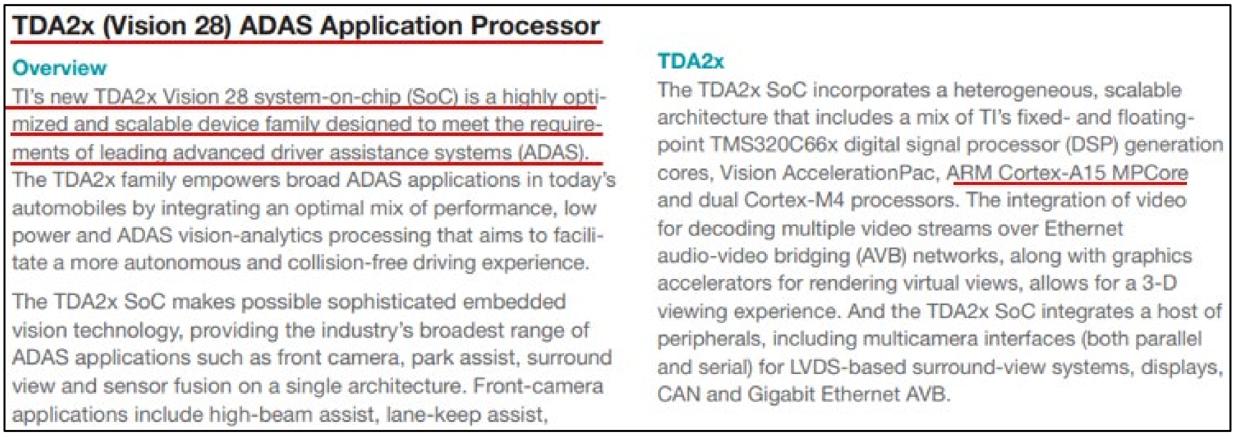


# TDA2x ADAS System-on-Chip

Texas Instruments

**Source:**

https://www.ti.com/lit/ml/sprt681/sprt681.pdf?ts=1597386693056&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 1)



## TDA2x (Vision 28) ADAS Application Processor

### Overview

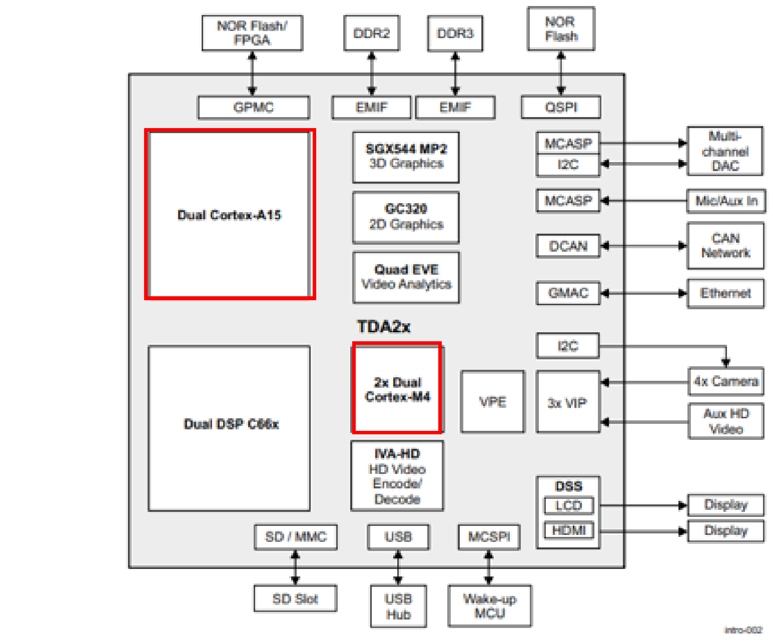
TI's new TDA2x Vision 28 system-on-chip (SoC) is a highly optimized and scalable device family designed to meet the requirements of leading advanced driver assistance systems (ADAS). The TDA2x family empowers broad ADAS applications in today's automobiles by integrating an optimal mix of performance, low power and ADAS vision-analytics processing that aims to facilitate a more autonomous and collision-free driving experience.

The TDA2x SoC makes possible sophisticated embedded vision technology, providing the industry's broadest range of ADAS applications such as front camera, park assist, surround view and sensor fusion on a single architecture. Front-camera applications include high-beam assist, lane-keep assist,

### TDA2x

The TDA2x SoC incorporates a heterogeneous, scalable architecture that includes a mix of TI's fixed- and floating-point TMS320C66x digital signal processor (DSP) generation cores, Vision AccelerationPac, ARM Cortex-A15 MPCore and dual Cortex-M4 processors. The integration of video for decoding multiple video streams over Ethernet audio-video bridging (AVB) networks, along with graphics accelerators for rendering virtual views, allows for a 3-D viewing experience. And the TDA2x SoC integrates a host of peripherals, including multicamera interfaces (both parallel and serial) for LVDS-based surround-view systems, displays, CAN and Gigabit Ethernet AVB.

**Source:** https://m.eet.com/media/1241324/slyy044.pdf (Page 5)

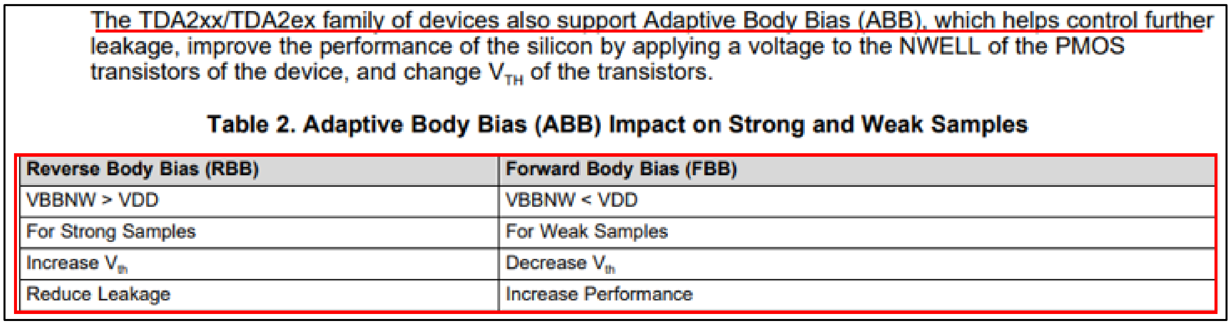


A The speeds shown are the highest design targeted, non-binned OPP.

Figure 1. TDA2xx Device Block Diagram

**Source:**

https://www.ti.com/lit/an/sprac21a/sprac21a.pdf?ts=1597386703137&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples

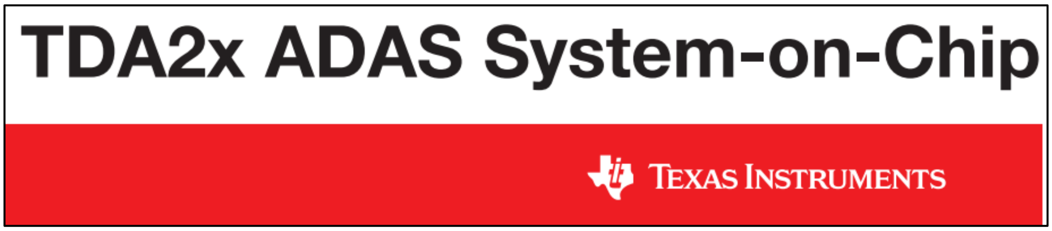
| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)

77. TI's TDA2x System-on-Chip devices include first circuitry configured to implement a plurality of power modes of the computer system, wherein the plurality of power modes comprises a mode in which the computer system does not execute instructions.

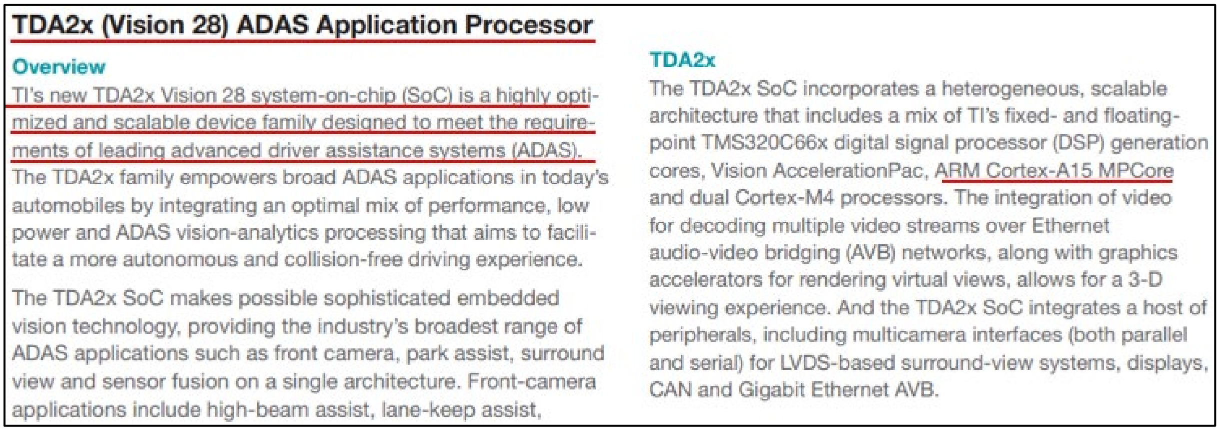
78. For example, TI's TDA2x System-on-Chip devices include a dual core processor module. Further, the System-on-Chip includes an MPU_PRCM module for power management of the processor. The TDA2x family of devices supports multiple power states (e.g., a power state in which both cores are on, a power state in which one core is forced off, a power state in which one core is forced off and the other core is in retention, etc.). The desired power state can be achieved by programming. The power state in which one core is forced off and the other core is in retention is a low-power state (i.e., subsystem retention) ("mode in which said computer system does not execute instructions"). During the subsystem retention mode, the computer system is unable to execute instructions.



TDA2x ADAS System-on-Chip

Texas Instruments

**Source:** https://www.ti.com/lit/ml/sprt681/sprt681.pdf?ts=1597386693056&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 1)



**TDA2x (Vision 28) ADAS Application Processor**

**Overview**

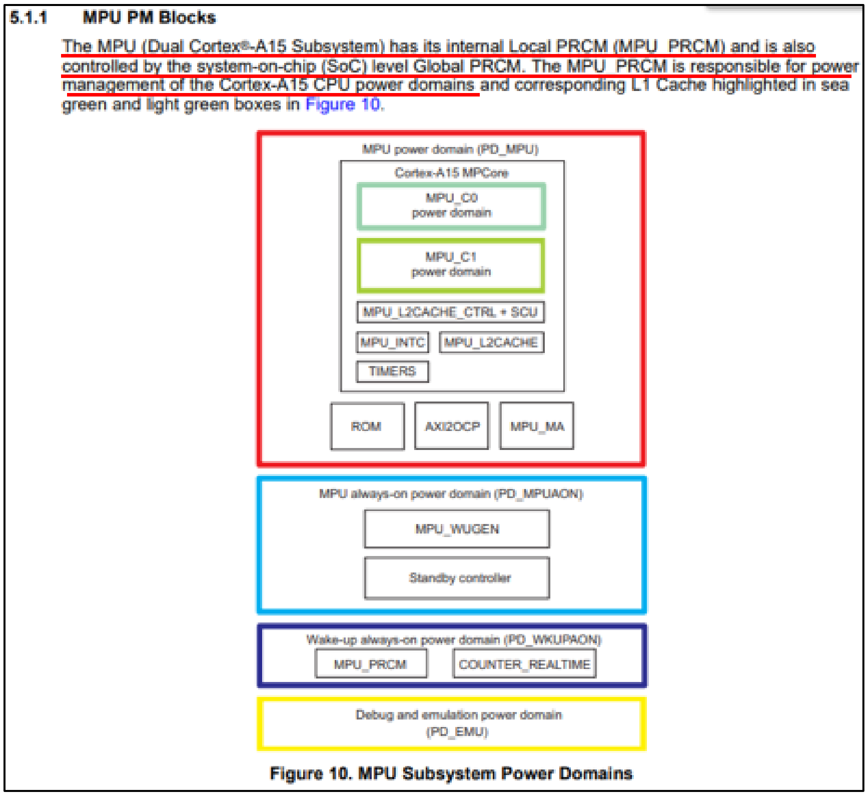
TI's new TDA2x Vision 28 system-on-chip (SoC) is a highly optimized and scalable device family designed to meet the requirements of leading advanced driver assistance systems (ADAS). The TDA2x family empowers broad ADAS applications in today's automobiles by integrating an optimal mix of performance, low power and ADAS vision-analytics processing that aims to facilitate a more autonomous and collision-free driving experience.

The TDA2x SoC makes possible sophisticated embedded vision technology, providing the industry's broadest range of ADAS applications such as front camera, park assist, surround view and sensor fusion on a single architecture. Front-camera applications include high-beam assist, lane-keep assist,

**TDA2x**

The TDA2x SoC incorporates a heterogeneous, scalable architecture that includes a mix of TI's fixed- and floating-point TMS320C66x digital signal processor (DSP) generation cores, Vision AccelerationPac, ARM Cortex-A15 MPCore and dual Cortex-M4 processors. The integration of video for decoding multiple video streams over Ethernet audio-video bridging (AVB) networks, along with graphics accelerators for rendering virtual views, allows for a 3-D viewing experience. And the TDA2x SoC integrates a host of peripherals, including multicamera interfaces (both parallel and serial) for LVDS-based surround-view systems, displays, CAN and Gigabit Ethernet AVB.

**Source:** https://m.eet.com/media/1241324/slyy044.pdf (Page 5)



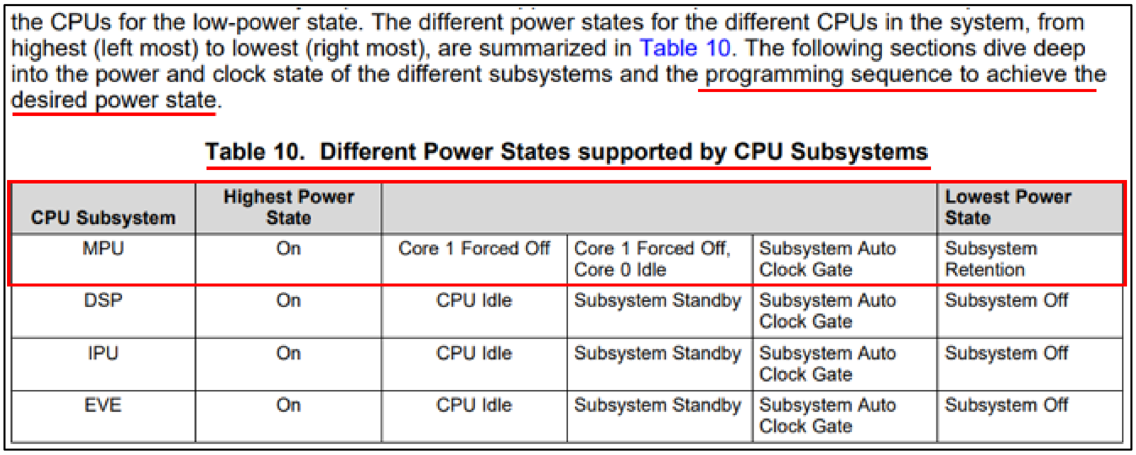
5.1.1 **MPU PM Blocks**

The MPU (Dual Cortex®-A15 Subsystem) has its internal Local PRCM (MPU_PRCM) and is also controlled by the system-on-chip (SoC) level Global PRCM. The MPU_PRCM is responsible for power management of the Cortex-A15 CPU power domains and corresponding L1 Cache highlighted in sea green and light green boxes in Figure 10.

**Figure 10. MPU Subsystem Power Domains**

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 19)



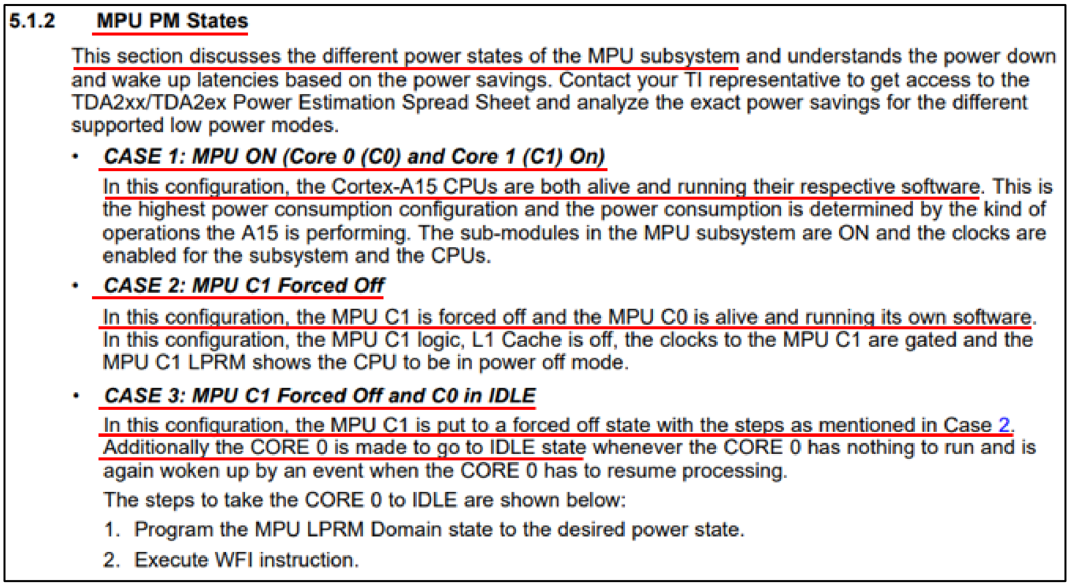
the CPUs for the low-power state. The different power states for the different CPUs in the system, from highest (left most) to lowest (right most), are summarized in Table 10. The following sections dive deep into the power and clock state of the different subsystems and the programming sequence to achieve the desired power state.

**Table 10. Different Power States supported by CPU Subsystems**

| CPU Subsystem | Highest Power State | | | | Lowest Power State |
|---|---|---|---|---|---|
| MPU | On | Core 1 Forced Off | Core 1 Forced Off, Core 0 Idle | Subsystem Auto Clock Gate | Subsystem Retention |
| DSP | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| IPU | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| EVE | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 18)



**5.1.2 MPU PM States**

This section discusses the different power states of the MPU subsystem and understands the power down and wake up latencies based on the power savings. Contact your TI representative to get access to the TDA2xx/TDA2ex Power Estimation Spread Sheet and analyze the exact power savings for the different supported low power modes.

- ***CASE 1: MPU ON (Core 0 (C0) and Core 1 (C1) On)***

  In this configuration, the Cortex-A15 CPUs are both alive and running their respective software. This is the highest power consumption configuration and the power consumption is determined by the kind of operations the A15 is performing. The sub-modules in the MPU subsystem are ON and the clocks are enabled for the subsystem and the CPUs.

- ***CASE 2: MPU C1 Forced Off***

  In this configuration, the MPU C1 is forced off and the MPU C0 is alive and running its own software. In this configuration, the MPU C1 logic, L1 Cache is off, the clocks to the MPU C1 are gated and the MPU C1 LPRM shows the CPU to be in power off mode.

- ***CASE 3: MPU C1 Forced Off and C0 in IDLE***

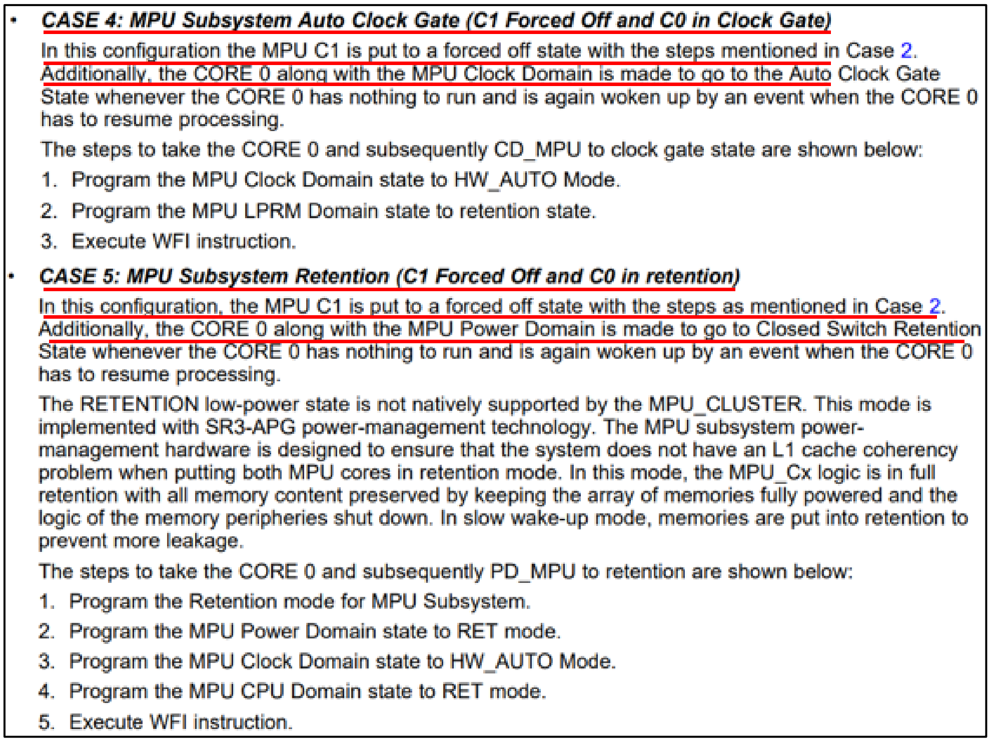
  In this configuration, the MPU C1 is put to a forced off state with the steps as mentioned in Case 2. Additionally the CORE 0 is made to go to IDLE state whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

  The steps to take the CORE 0 to IDLE are shown below:

  1. Program the MPU LPRM Domain state to the desired power state.
  2. Execute WFI instruction.

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Pages 20 and 21)



- ***CASE 4: MPU Subsystem Auto Clock Gate (C1 Forced Off and C0 in Clock Gate)***

  In this configuration the MPU C1 is put to a forced off state with the steps mentioned in Case 2. Additionally, the CORE 0 along with the MPU Clock Domain is made to go to the Auto Clock Gate State whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

  The steps to take the CORE 0 and subsequently CD_MPU to clock gate state are shown below:

  1. Program the MPU Clock Domain state to HW_AUTO Mode.
  2. Program the MPU LPRM Domain state to retention state.
  3. Execute WFI instruction.

- ***CASE 5: MPU Subsystem Retention (C1 Forced Off and C0 in retention)***

  In this configuration, the MPU C1 is put to a forced off state with the steps as mentioned in Case 2. Additionally, the CORE 0 along with the MPU Power Domain is made to go to Closed Switch Retention State whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

  The RETENTION low-power state is not natively supported by the MPU_CLUSTER. This mode is implemented with SR3-APG power-management technology. The MPU subsystem power-management hardware is designed to ensure that the system does not have an L1 cache coherency problem when putting both MPU cores in retention mode. In this mode, the MPU_Cx logic is in full retention with all memory content preserved by keeping the array of memories fully powered and the logic of the memory peripheries shut down. In slow wake-up mode, memories are put into retention to prevent more leakage.

  The steps to take the CORE 0 and subsequently PD_MPU to retention are shown below:

  1. Program the Retention mode for MPU Subsystem.
  2. Program the MPU Power Domain state to RET mode.
  3. Program the MPU Clock Domain state to HW_AUTO Mode.
  4. Program the MPU CPU Domain state to RET mode.
  5. Execute WFI instruction.

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 23)

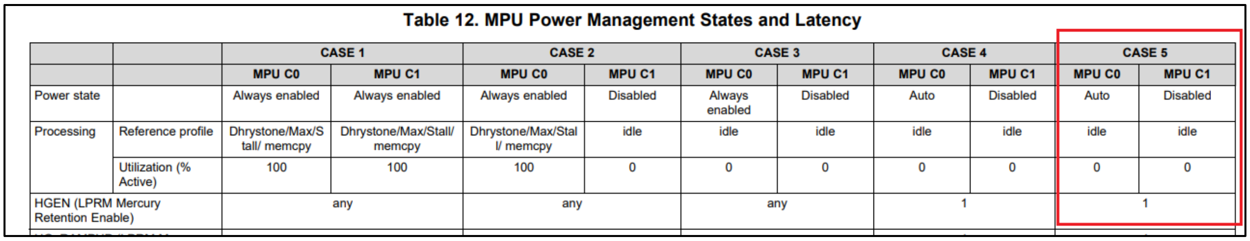


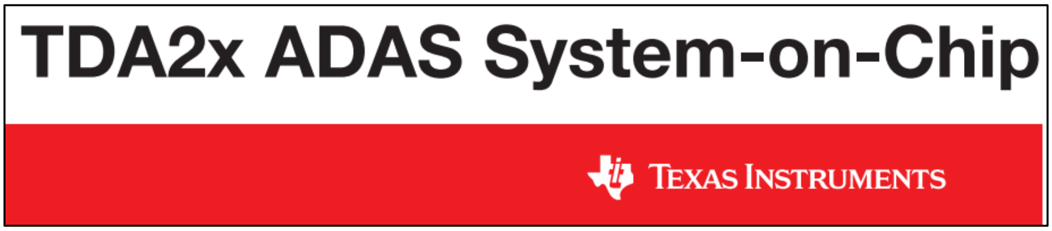
Table 12. MPU Power Management States and Latency

| | | CASE 1 | | CASE 2 | | CASE 3 | | CASE 4 | | CASE 5 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MPU C0 | MPU C1 | MPU C0 | MPU C1 | MPU C0 | MPU C1 | MPU C0 | MPU C1 | MPU C0 | MPU C1 |
| Power state | | Always enabled | Always enabled | Always enabled | Disabled | Always enabled | Disabled | Auto | Disabled | Auto | Disabled |
| Processing | Reference profile | Dhrystone/Max/Stall/ memcpy | Dhrystone/Max/Stall/ memcpy | Dhrystone/Max/Stall/ memcpy | idle | idle | idle | idle | idle | idle | idle |
| | Utilization (% Active) | 100 | 100 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| HGEN (LPRM Mercury Retention Enable) | | any | | any | | any | | 1 | | 1 | |

**Source:**
https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)

79. The first circuitry of TI's TDA2x System-on-Chip devices is further configured to decrease static power consumption of said mode by application of a plurality of body biasing voltages to second circuitry of the computer system.

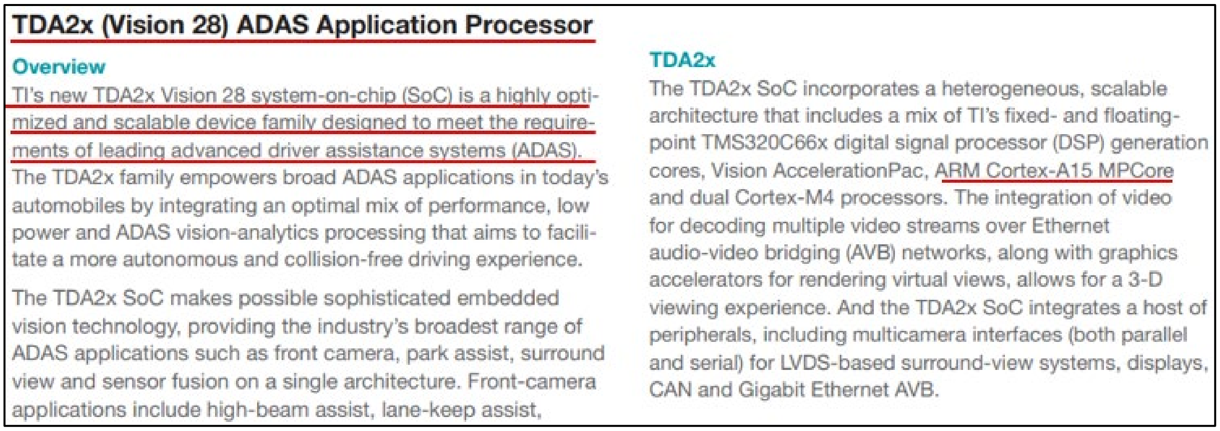
80. For example, TI's TDA2x System-on-Chip devices include a dual core processor module. The devices support Adaptive Body Bias (ABB). ABB voltage values are available for each OPP (Operating Performance Point). The devices support the lowest power mode in which one core is forced off and the other core is in retention (i.e., subsystem retention). During the subsystem retention mode ("said mode"), the computer system is unable to execute instructions and is in OPP_Low mode. An ABB voltage is also applied to the transistors during the subsystem retention mode. The static power consumption during subsystem retention mode is decreased by applying Reverse Body Bias (RBB) voltage to each processor ("second circuitry") of the system.



TDA2x ADAS System-on-Chip

Texas Instruments

**Source:**
https://www.ti.com/lit/ml/sprt681/sprt681.pdf?ts=1597386693056&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 1)



## TDA2x (Vision 28) ADAS Application Processor

### Overview

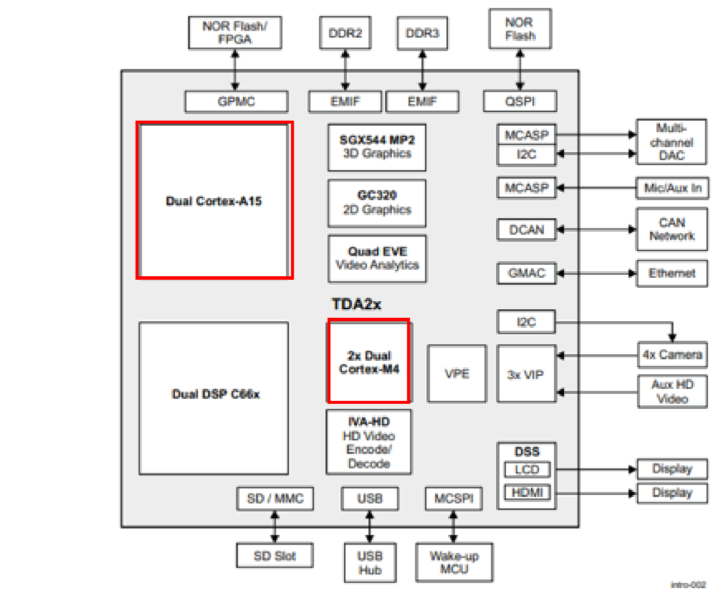
TI's new TDA2x Vision 28 system-on-chip (SoC) is a highly optimized and scalable device family designed to meet the requirements of leading advanced driver assistance systems (ADAS). The TDA2x family empowers broad ADAS applications in today's automobiles by integrating an optimal mix of performance, low power and ADAS vision-analytics processing that aims to facilitate a more autonomous and collision-free driving experience.

The TDA2x SoC makes possible sophisticated embedded vision technology, providing the industry's broadest range of ADAS applications such as front camera, park assist, surround view and sensor fusion on a single architecture. Front-camera applications include high-beam assist, lane-keep assist,

### TDA2x

The TDA2x SoC incorporates a heterogeneous, scalable architecture that includes a mix of TI's fixed- and floating-point TMS320C66x digital signal processor (DSP) generation cores, Vision AccelerationPac, ARM Cortex-A15 MPCore and dual Cortex-M4 processors. The integration of video for decoding multiple video streams over Ethernet audio-video bridging (AVB) networks, along with graphics accelerators for rendering virtual views, allows for a 3-D viewing experience. And the TDA2x SoC integrates a host of peripherals, including multicamera interfaces (both parallel and serial) for LVDS-based surround-view systems, displays, CAN and Gigabit Ethernet AVB.

**Source:** https://m.eet.com/media/1241324/slyy044.pdf (Page 5)

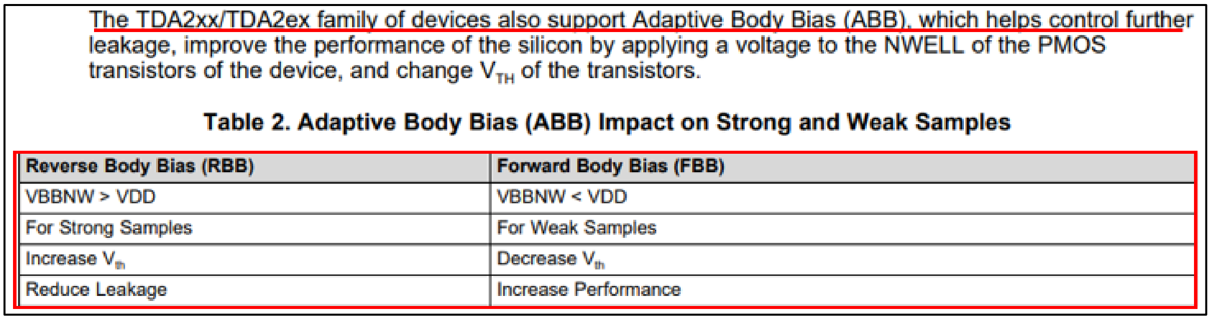


A The speeds shown are the highest design targeted, non-binned OPP.

**Figure 1. TDA2xx Device Block Diagram**

**Source:**

https://www.ti.com/lit/an/sprac21a/sprac21a.pdf?ts=1597386703137&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



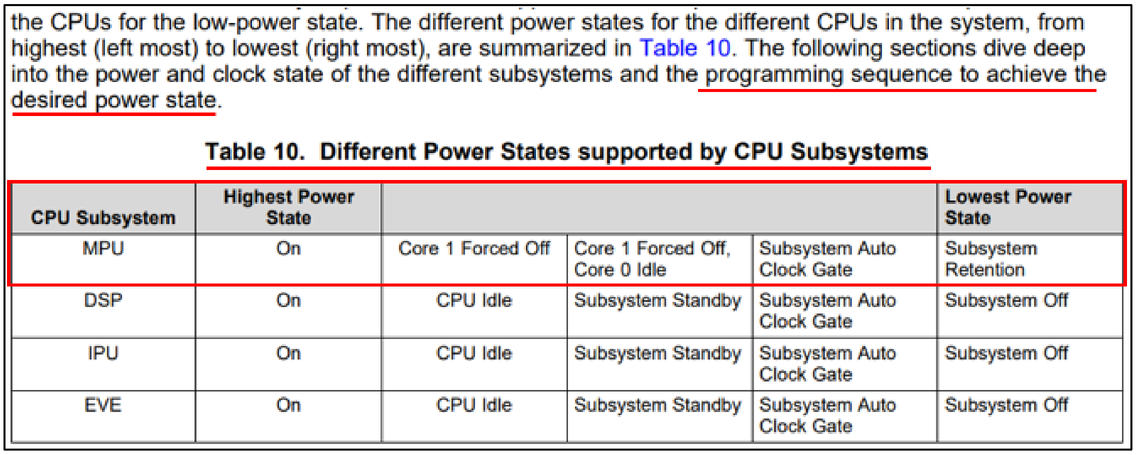
The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

**Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples**

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)



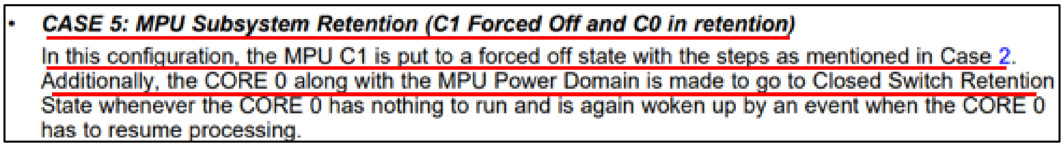
the CPUs for the low-power state. The different power states for the different CPUs in the system, from highest (left most) to lowest (right most), are summarized in Table 10. The following sections dive deep into the power and clock state of the different subsystems and the programming sequence to achieve the desired power state.

**Table 10. Different Power States supported by CPU Subsystems**

| CPU Subsystem | Highest Power State | | | | Lowest Power State |
|---|---|---|---|---|---|
| MPU | On | Core 1 Forced Off | Core 1 Forced Off, Core 0 Idle | Subsystem Auto Clock Gate | Subsystem Retention |
| DSP | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| IPU | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| EVE | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 18)



- ***CASE 5: MPU Subsystem Retention (C1 Forced Off and C0 in retention)***

  In this configuration, the MPU C1 is put to a forced off state with the steps as mentioned in Case 2. Additionally, the CORE 0 along with the MPU Power Domain is made to go to Closed Switch Retention State whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 23)

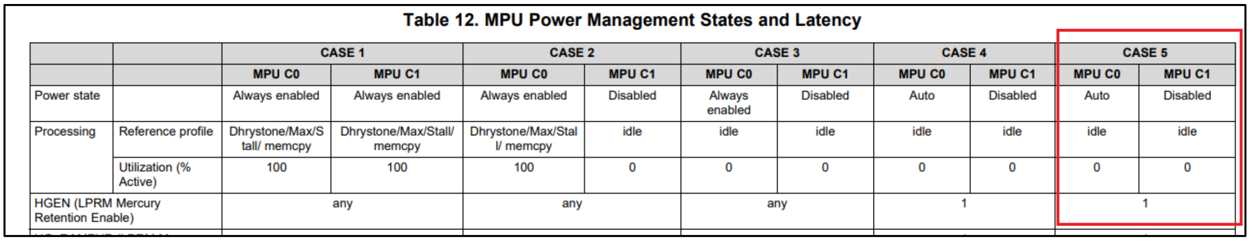
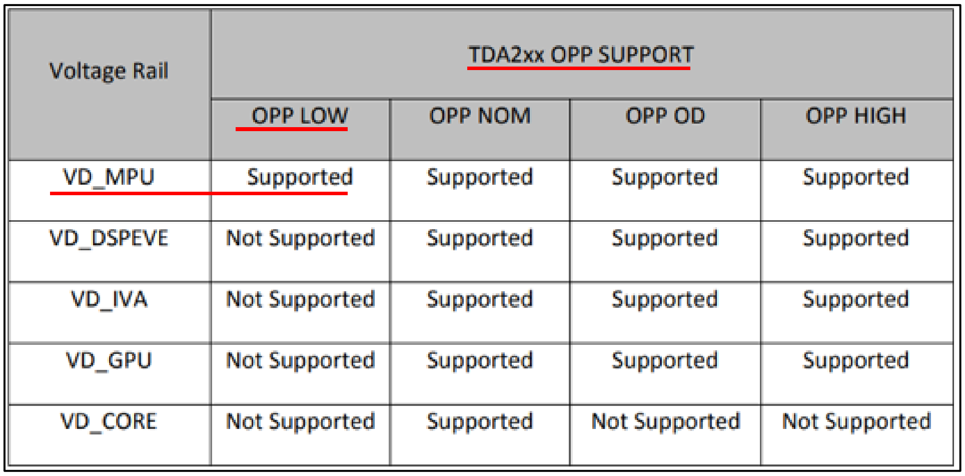


Table 12. MPU Power Management States and Latency

| | | CASE 1 | | CASE 2 | | CASE 3 | | CASE 4 | | CASE 5 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MPU C0 | MPU C1 | MPU C0 | MPU C1 | MPU C0 | MPU C1 | MPU C0 | MPU C1 | MPU C0 | MPU C1 |
| Power state | | Always enabled | Always enabled | Always enabled | Disabled | Always enabled | Disabled | Auto | Disabled | Auto | Disabled |
| Processing | Reference profile | Dhrystone/Max/Stall/ memcpy | Dhrystone/Max/Stall/ memcpy | Dhrystone/Max/Stall/ memcpy | idle | idle | idle | idle | idle | idle | idle |
| | Utilization (% Active) | 100 | 100 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| HGEN (LPRM Mercury Retention Enable) | | any | | any | | any | | 1 | | 1 | |

**Source:**
https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



| Voltage Rail | TDA2xx OPP SUPPORT | | | |
|---|---|---|---|---|
| | OPP LOW | OPP NOM | OPP OD | OPP HIGH |
| VD_MPU | Supported | Supported | Supported | Supported |
| VD_DSPEVE | Not Supported | Supported | Supported | Supported |
| VD_IVA | Not Supported | Supported | Supported | Supported |
| VD_GPU | Not Supported | Supported | Supported | Supported |
| VD_CORE | Not Supported | Supported | Not Supported | Not Supported |

**Source:** https://processors.wiki.ti.com/images/b/b7/TDA_SBL_UserGuide.pdf (Page 23)

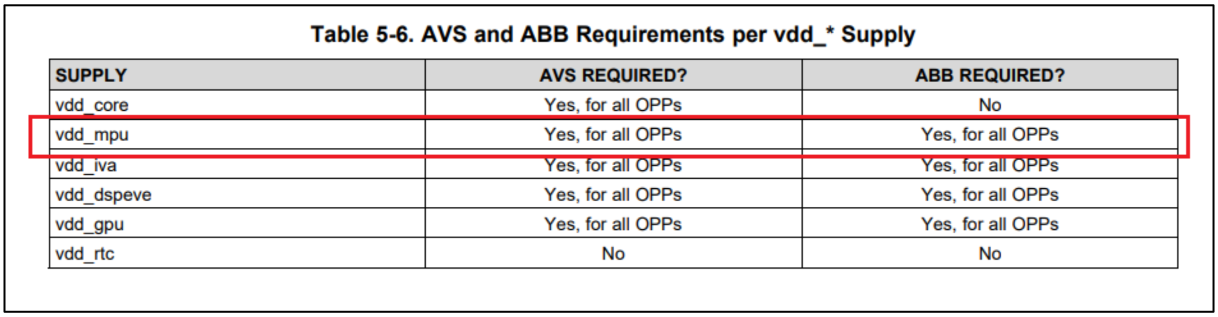
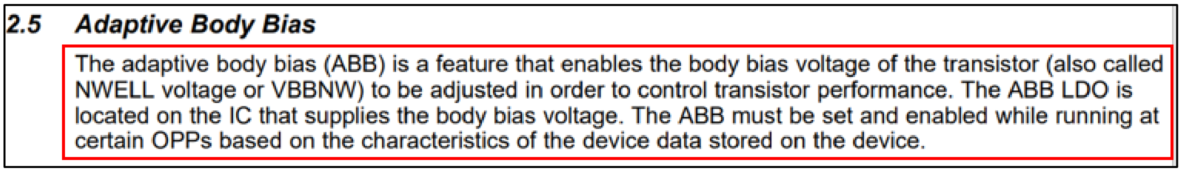


Table 5-6. AVS and ABB Requirements per vdd_* Supply

| SUPPLY | AVS REQUIRED? | ABB REQUIRED? |
|---|---|---|
| vdd_core | Yes, for all OPPs | No |
| vdd_mpu | Yes, for all OPPs | Yes, for all OPPs |
| vdd_iva | Yes, for all OPPs | Yes, for all OPPs |
| vdd_dspeve | Yes, for all OPPs | Yes, for all OPPs |
| vdd_gpu | Yes, for all OPPs | Yes, for all OPPs |
| vdd_rtc | No | No |

**Source:** TDA2x ADAS Applications Processor 23mm Package (ABC Package) Silicon Revision 2.0 datasheet (Rev. F) (Page 142)



**2.5 *Adaptive Body Bias***

The adaptive body bias (ABB) is a feature that enables the body bias voltage of the transistor (also called NWELL voltage or VBBNW) to be adjusted in order to control transistor performance. The ABB LDO is located on the IC that supplies the body bias voltage. The ABB must be set and enabled while running at certain OPPs based on the characteristics of the device data stored on the device.

**Source:**

https://www.ti.com/lit/an/slva646/slva646.pdf?ts=1597323545336&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)

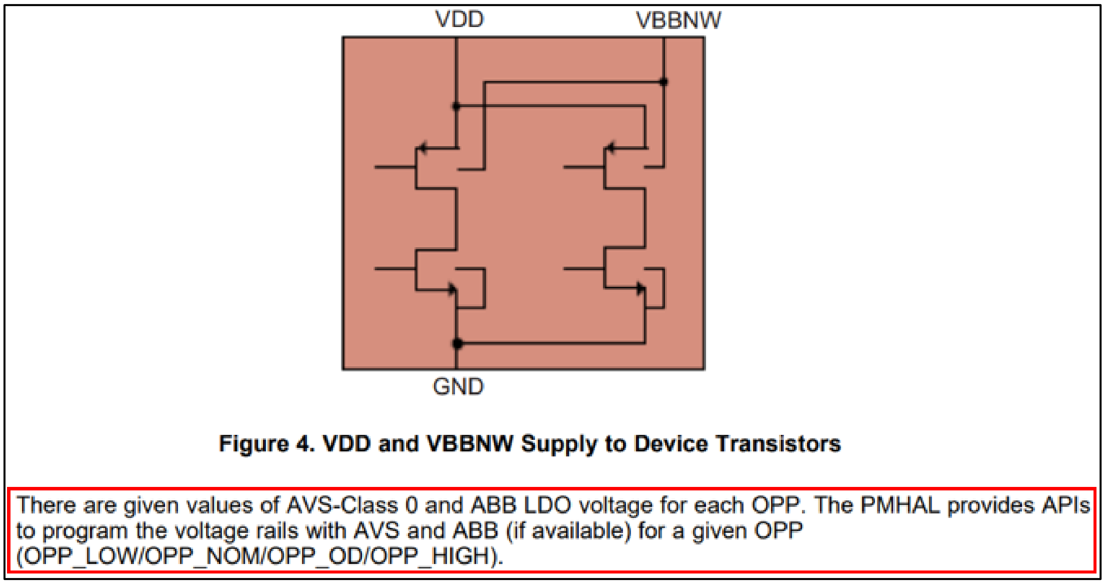


**Figure 4. VDD and VBBNW Supply to Device Transistors**

There are given values of AVS-Class 0 and ABB LDO voltage for each OPP. The PMHAL provides APIs to program the voltage rails with AVS and ABB (if available) for a given OPP (OPP_LOW/OPP_NOM/OPP_OD/OPP_HIGH).

**Source:**

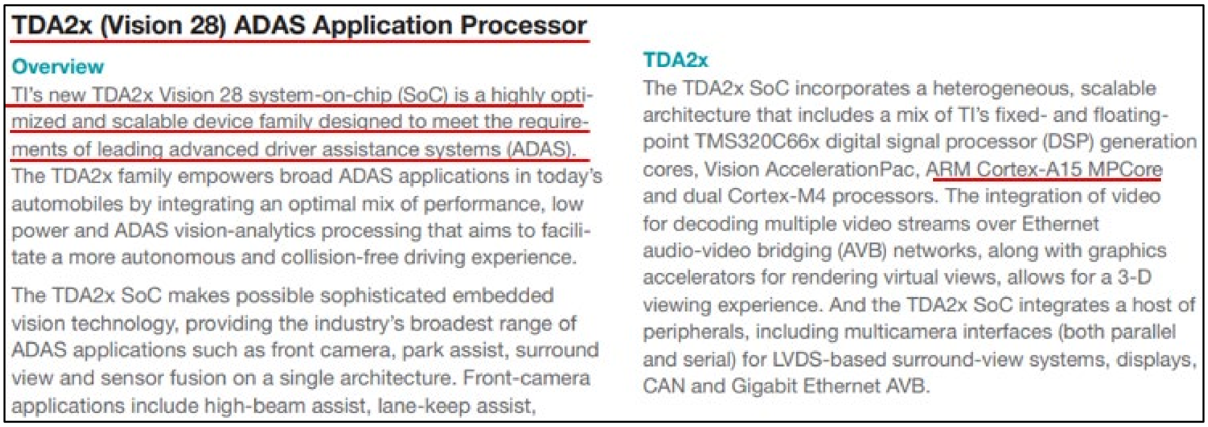
https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)

81. TI's TDA2x System-on-Chip devices include means for determining a particular power condition, of a set of power conditions, of a computer system comprising a microprocessor, wherein the set of power conditions comprises a power down state.

82. For example, TI's TDA2x System-on-Chip devices include a dual core processor module. Further, the System-on-Chip ("computer system") includes an MPU_PRCM module for power management of the processor ("microprocessor"). The TDA2x family of devices supports multiple power states or "a set of power conditions" (e.g., a power state in which both cores are

on, a power state in which one core is forced off, a power state in which one core is forced off and the other core is in retention' etc.).

83. The desired power state can be achieved through programming—that is, a command can be received by the device to change from one power state to a different power state. For example, the system can receive a command to change from a power state in which both cores are on to a power state in which one core is forced off and the other core is in retention. The power state in which one core is forced off and the other core is in retention is a low-power state ("power down state").



### TDA2x (Vision 28) ADAS Application Processor

#### Overview

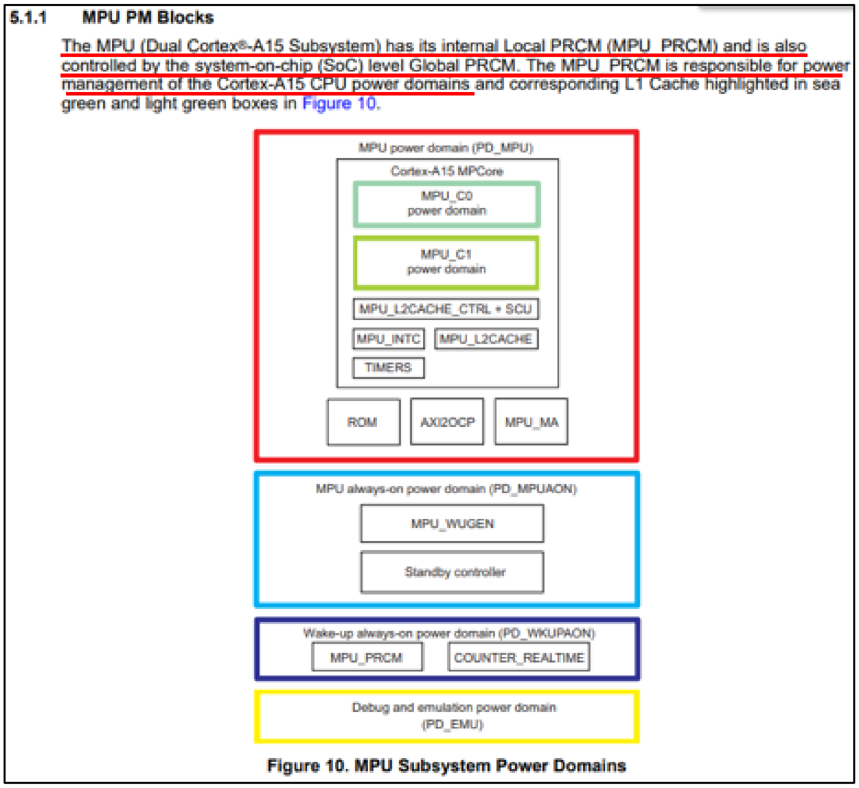
TI's new TDA2x Vision 28 system-on-chip (SoC) is a highly optimized and scalable device family designed to meet the requirements of leading advanced driver assistance systems (ADAS). The TDA2x family empowers broad ADAS applications in today's automobiles by integrating an optimal mix of performance, low power and ADAS vision-analytics processing that aims to facilitate a more autonomous and collision-free driving experience.

The TDA2x SoC makes possible sophisticated embedded vision technology, providing the industry's broadest range of ADAS applications such as front camera, park assist, surround view and sensor fusion on a single architecture. Front-camera applications include high-beam assist, lane-keep assist,

#### TDA2x

The TDA2x SoC incorporates a heterogeneous, scalable architecture that includes a mix of TI's fixed- and floating-point TMS320C66x digital signal processor (DSP) generation cores, Vision AccelerationPac, ARM Cortex-A15 MPCore and dual Cortex-M4 processors. The integration of video for decoding multiple video streams over Ethernet audio-video bridging (AVB) networks, along with graphics accelerators for rendering virtual views, allows for a 3-D viewing experience. And the TDA2x SoC integrates a host of peripherals, including multicamera interfaces (both parallel and serial) for LVDS-based surround-view systems, displays, CAN and Gigabit Ethernet AVB.

**Source:** https://m.eet.com/media/1241324/slyy044.pdf (Page 5)



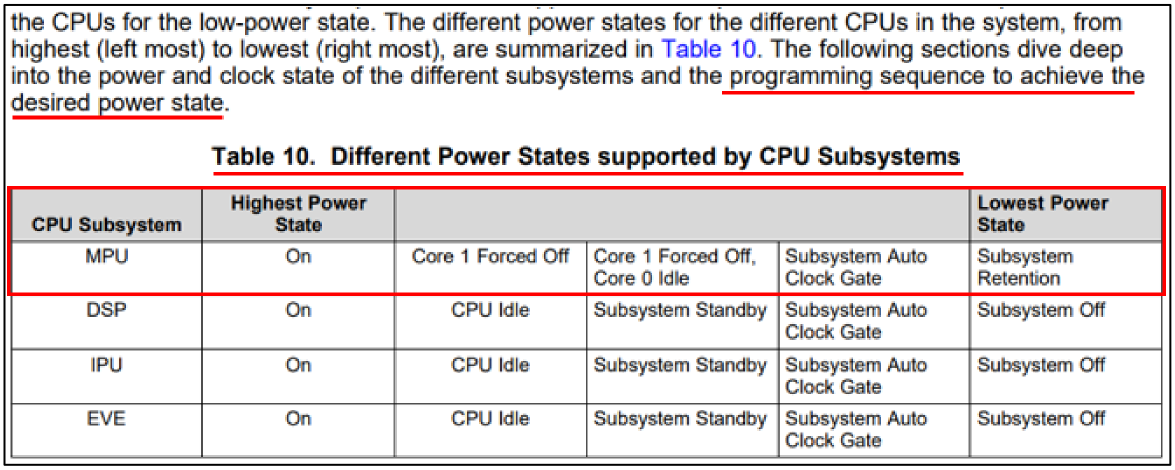
5.1.1 **MPU PM Blocks**

The MPU (Dual Cortex®-A15 Subsystem) has its internal Local PRCM (MPU_PRCM) and is also controlled by the system-on-chip (SoC) level Global PRCM. The MPU_PRCM is responsible for power management of the Cortex-A15 CPU power domains and corresponding L1 Cache highlighted in sea green and light green boxes in Figure 10.

**Figure 10. MPU Subsystem Power Domains**

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 19)



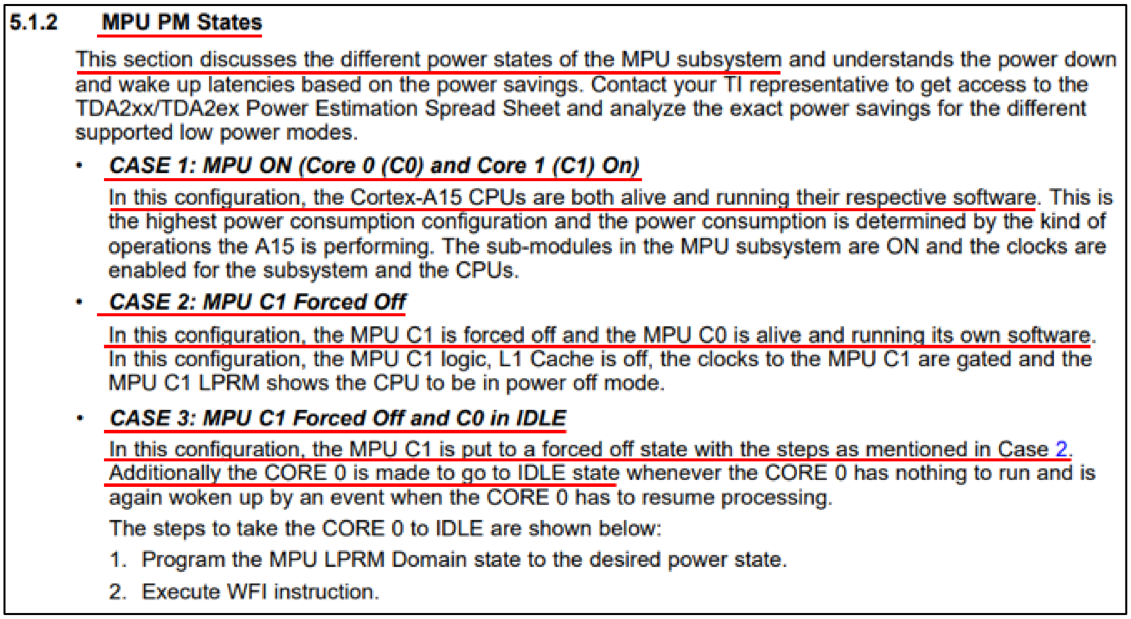
the CPUs for the low-power state. The different power states for the different CPUs in the system, from highest (left most) to lowest (right most), are summarized in Table 10. The following sections dive deep into the power and clock state of the different subsystems and the programming sequence to achieve the desired power state.

**Table 10. Different Power States supported by CPU Subsystems**

| CPU Subsystem | Highest Power State | | | | Lowest Power State |
|---|---|---|---|---|---|
| MPU | On | Core 1 Forced Off | Core 1 Forced Off, Core 0 Idle | Subsystem Auto Clock Gate | Subsystem Retention |
| DSP | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| IPU | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| EVE | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 18)



### 5.1.2 MPU PM States

This section discusses the different power states of the MPU subsystem and understands the power down and wake up latencies based on the power savings. Contact your TI representative to get access to the TDA2xx/TDA2ex Power Estimation Spread Sheet and analyze the exact power savings for the different supported low power modes.

- ***CASE 1: MPU ON (Core 0 (C0) and Core 1 (C1) On)***

  In this configuration, the Cortex-A15 CPUs are both alive and running their respective software. This is the highest power consumption configuration and the power consumption is determined by the kind of operations the A15 is performing. The sub-modules in the MPU subsystem are ON and the clocks are enabled for the subsystem and the CPUs.

- ***CASE 2: MPU C1 Forced Off***

  In this configuration, the MPU C1 is forced off and the MPU C0 is alive and running its own software. In this configuration, the MPU C1 logic, L1 Cache is off, the clocks to the MPU C1 are gated and the MPU C1 LPRM shows the CPU to be in power off mode.

- ***CASE 3: MPU C1 Forced Off and C0 in IDLE***

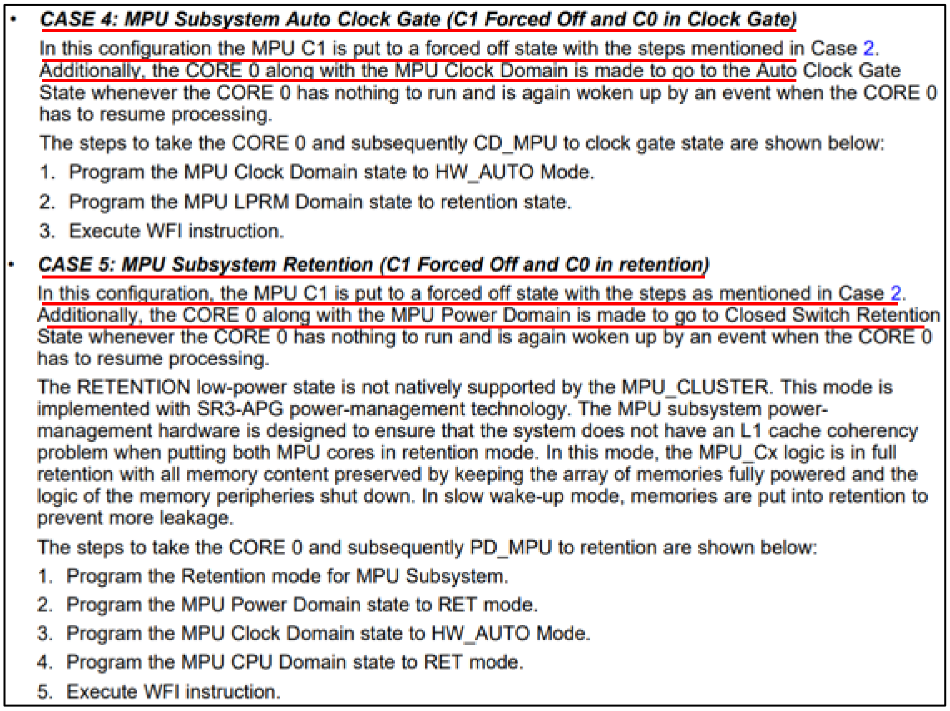
  In this configuration, the MPU C1 is put to a forced off state with the steps as mentioned in Case 2. Additionally the CORE 0 is made to go to IDLE state whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

  The steps to take the CORE 0 to IDLE are shown below:

  1. Program the MPU LPRM Domain state to the desired power state.
  2. Execute WFI instruction.

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Pages 20 and 21)



- ***CASE 4: MPU Subsystem Auto Clock Gate (C1 Forced Off and C0 in Clock Gate)***

  In this configuration the MPU C1 is put to a forced off state with the steps mentioned in Case 2. Additionally, the CORE 0 along with the MPU Clock Domain is made to go to the Auto Clock Gate State whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

  The steps to take the CORE 0 and subsequently CD_MPU to clock gate state are shown below:

  1. Program the MPU Clock Domain state to HW_AUTO Mode.
  2. Program the MPU LPRM Domain state to retention state.
  3. Execute WFI instruction.

- ***CASE 5: MPU Subsystem Retention (C1 Forced Off and C0 in retention)***

  In this configuration, the MPU C1 is put to a forced off state with the steps as mentioned in Case 2. Additionally, the CORE 0 along with the MPU Power Domain is made to go to Closed Switch Retention State whenever the CORE 0 has nothing to run and is again woken up by an event when the CORE 0 has to resume processing.

  The RETENTION low-power state is not natively supported by the MPU_CLUSTER. This mode is implemented with SR3-APG power-management technology. The MPU subsystem power-management hardware is designed to ensure that the system does not have an L1 cache coherency problem when putting both MPU cores in retention mode. In this mode, the MPU_Cx logic is in full retention with all memory content preserved by keeping the array of memories fully powered and the logic of the memory peripheries shut down. In slow wake-up mode, memories are put into retention to prevent more leakage.

  The steps to take the CORE 0 and subsequently PD_MPU to retention are shown below:

  1. Program the Retention mode for MPU Subsystem.
  2. Program the MPU Power Domain state to RET mode.
  3. Program the MPU Clock Domain state to HW_AUTO Mode.
  4. Program the MPU CPU Domain state to RET mode.
  5. Execute WFI instruction.

**Source:**

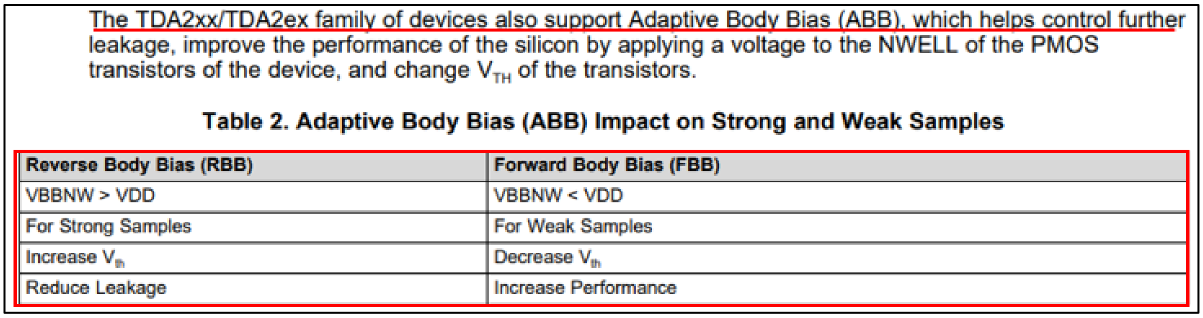
https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 23)

84. TI's TDA2x System-on-Chip devices include means for accessing body biasing voltage information corresponding to the particular power condition.

85. For example, TI's TDA2x System-on-Chip devices support Adaptive Body Bias (ABB). ABB voltage values are given for each OPP (Operating Performance Point), which is the operating condition for the device and is defined by the voltage and frequency applied to the processor. Further, the TDA2x family of devices supports multiple power states (e.g., a power state in which both cores are on, a power state in which one core is forced off, a power state in which one core is forced off and other core is in retention, etc.). An OPP is established based on the power state (i.e., operation) of the CPU.

86. Accordingly, the TDA2x family of devices stores the ABB voltage values associated with each OPP and corresponding power state. The ABB feature allows application of forward body bias to improve performance and reverse body bias to reduce leakage in the device. To reduce leakage in low power states or low operating performance points (OPP Low), the reverse body bias voltage value ("body biasing voltage information") is accessed and applied to the device. This occurs during one of the low power states of the device, such as during the power state in which one core is forced off and the other core is in retention ("said particular power condition").

87. Accordingly, the TDA2x family of devices includes means for accessing the corresponding body biasing voltage information.



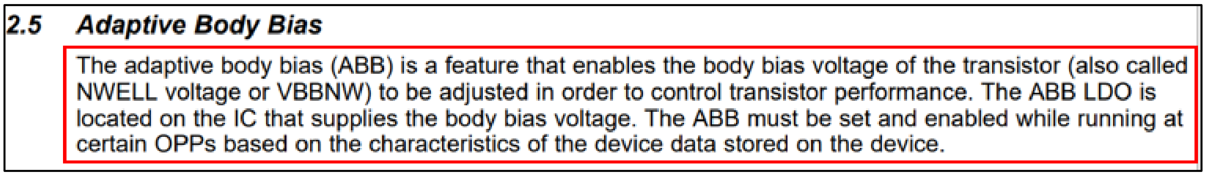
The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

**Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples**

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)



2.5 *Adaptive Body Bias*

The adaptive body bias (ABB) is a feature that enables the body bias voltage of the transistor (also called NWELL voltage or VBBNW) to be adjusted in order to control transistor performance. The ABB LDO is located on the IC that supplies the body bias voltage. The ABB must be set and enabled while running at certain OPPs based on the characteristics of the device data stored on the device.

**Source:**

https://www.ti.com/lit/an/slva646/slva646.pdf?ts=1597323545336&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)

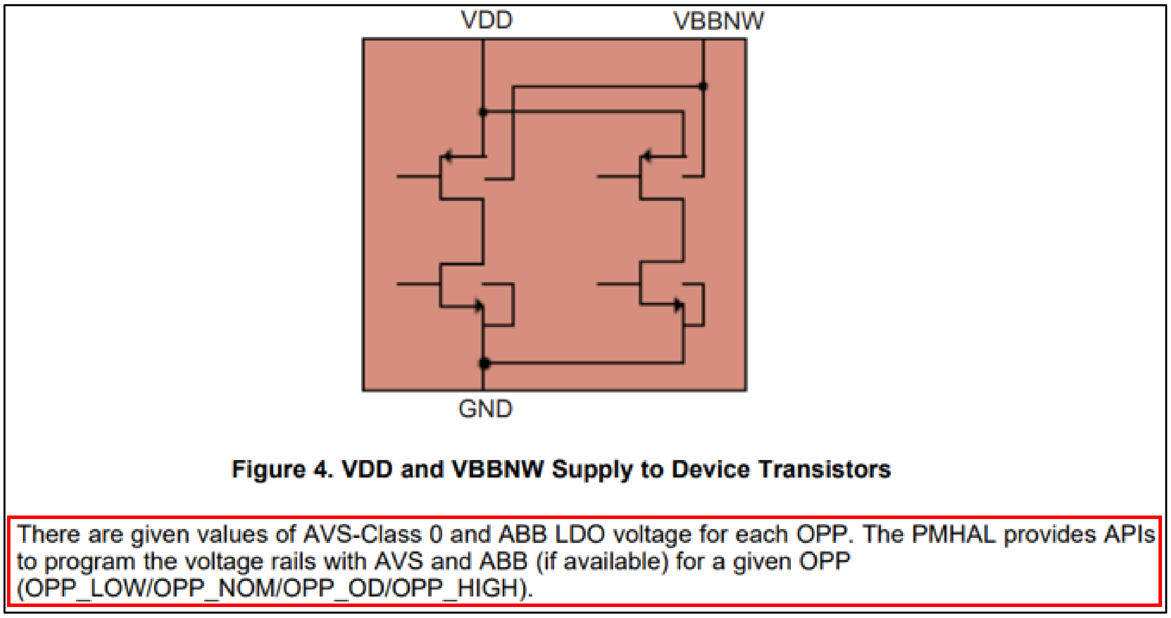
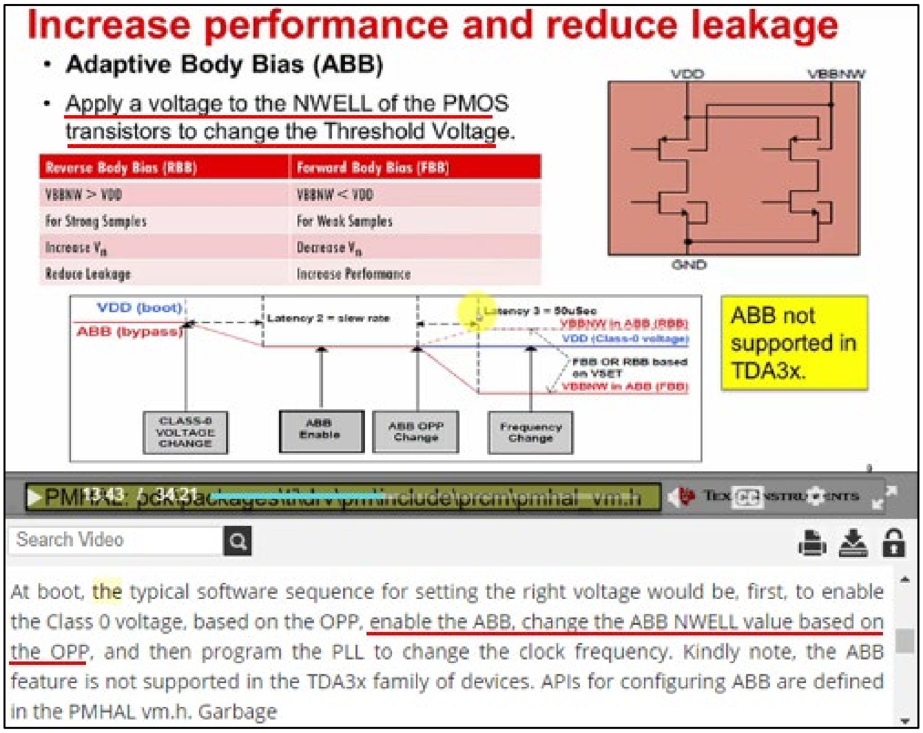


**Figure 4. VDD and VBBNW Supply to Device Transistors**

There are given values of AVS-Class 0 and ABB LDO voltage for each OPP. The PMHAL provides APIs to program the voltage rails with AVS and ABB (if available) for a given OPP (OPP_LOW/OPP_NOM/OPP_OD/OPP_HIGH).

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



Increase performance and reduce leakage

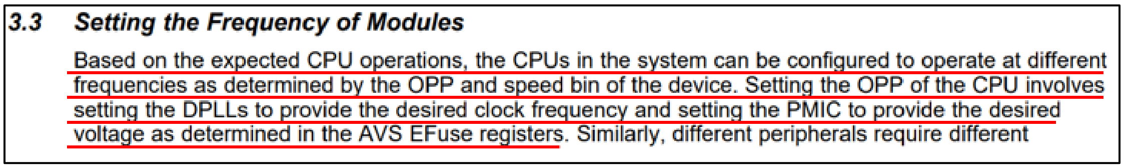
- Adaptive Body Bias (ABB)
- Apply a voltage to the NWELL of the PMOS transistors to change the Threshold Voltage.

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

ABB not supported in TDA3x.

At boot, the typical software sequence for setting the right voltage would be, first, to enable the Class 0 voltage, based on the OPP, enable the ABB, change the ABB NWELL value based on the OPP, and then program the PLL to change the clock frequency. Kindly note, the ABB feature is not supported in the TDA3x family of devices. APIs for configuring ABB are defined in the PMHAL vm.h. Garbage

**Source:** https://training.ti.com/introduction-adas-power-management-software?cu=1137175 (at 13:43)



### 3.3 Setting the Frequency of Modules

Based on the expected CPU operations, the CPUs in the system can be configured to operate at different frequencies as determined by the OPP and speed bin of the device. Setting the OPP of the CPU involves setting the DPLLs to provide the desired clock frequency and setting the PMIC to provide the desired voltage as determined in the AVS EFuse registers. Similarly, different peripherals require different

**Source:** https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 9)

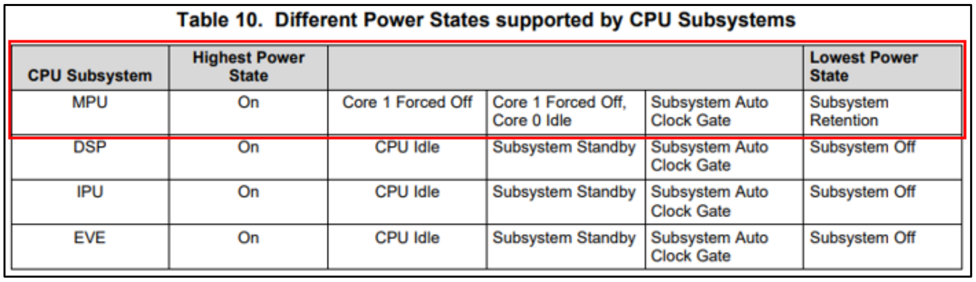
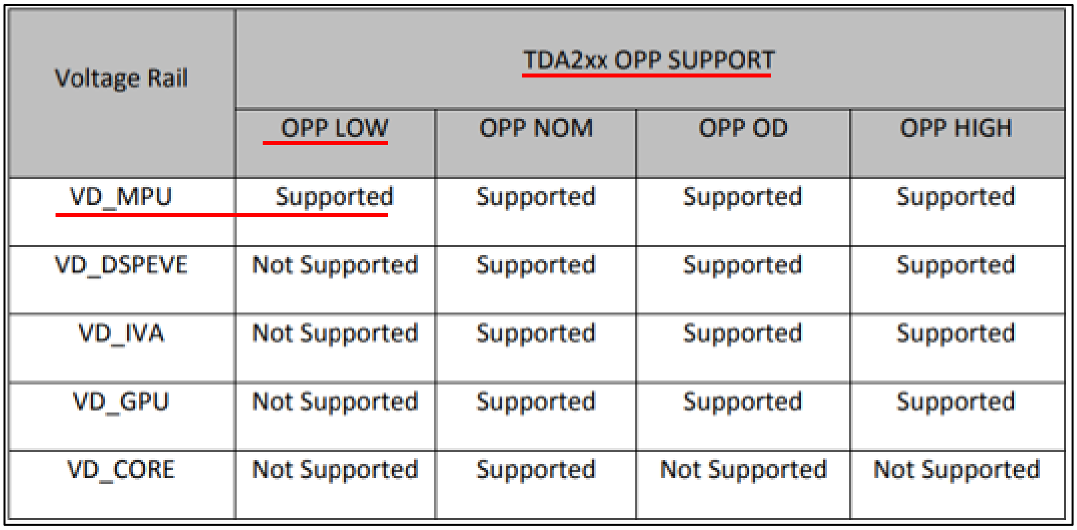


**Table 10. Different Power States supported by CPU Subsystems**

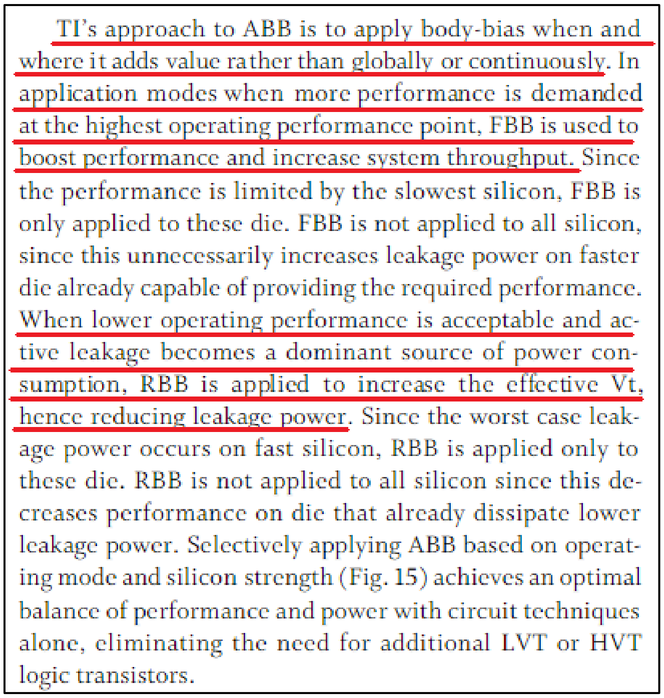
| CPU Subsystem | Highest Power State | | | | Lowest Power State |
|---|---|---|---|---|---|
| MPU | On | Core 1 Forced Off | Core 1 Forced Off, Core 0 Idle | Subsystem Auto Clock Gate | Subsystem Retention |
| DSP | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| IPU | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| EVE | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |

**Source:** https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 18)



| Voltage Rail | TDA2xx OPP SUPPORT | | | |
|---|---|---|---|---|
| | OPP LOW | OPP NOM | OPP OD | OPP HIGH |
| VD_MPU | Supported | Supported | Supported | Supported |
| VD_DSPEVE | Not Supported | Supported | Supported | Supported |
| VD_IVA | Not Supported | Supported | Supported | Supported |
| VD_GPU | Not Supported | Supported | Supported | Supported |
| VD_CORE | Not Supported | Supported | Not Supported | Not Supported |

**Source:** https://processors.wiki.ti.com/images/b/b7/TDA_SBL_UserGuide.pdf (Page 23)



TI's approach to ABB is to apply body-bias when and where it adds value rather than globally or continuously. In application modes when more performance is demanded at the highest operating performance point, FBB is used to boost performance and increase system throughput. Since the performance is limited by the slowest silicon, FBB is only applied to these die. FBB is not applied to all silicon, since this unnecessarily increases leakage power on faster die already capable of providing the required performance. When lower operating performance is acceptable and active leakage becomes a dominant source of power consumption, RBB is applied to increase the effective Vt, hence reducing leakage power. Since the worst case leakage power occurs on fast silicon, RBB is applied only to these die. RBB is not applied to all silicon since this decreases performance on die that already dissipate lower leakage power. Selectively applying ABB based on operating mode and silicon strength (Fig. 15) achieves an optimal balance of performance and power with circuit techniques alone, eliminating the need for additional LVT or HVT logic transistors.

**Source:**

https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

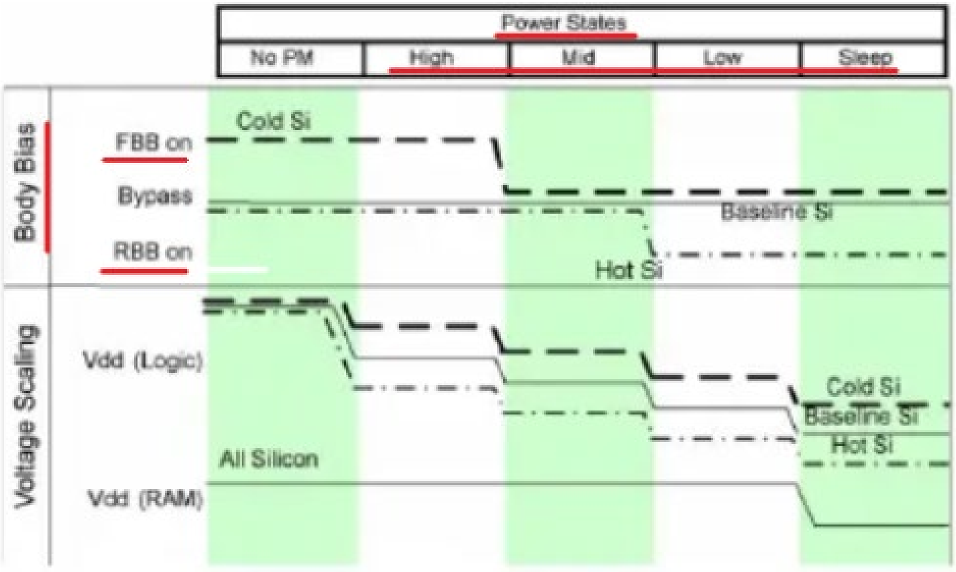


**Fig. 15.** ***Adaptivebody-biasand adaptive voltage scaling configuration with mode and process. Hot and cold refer the process strength or speed (fast and slow).***

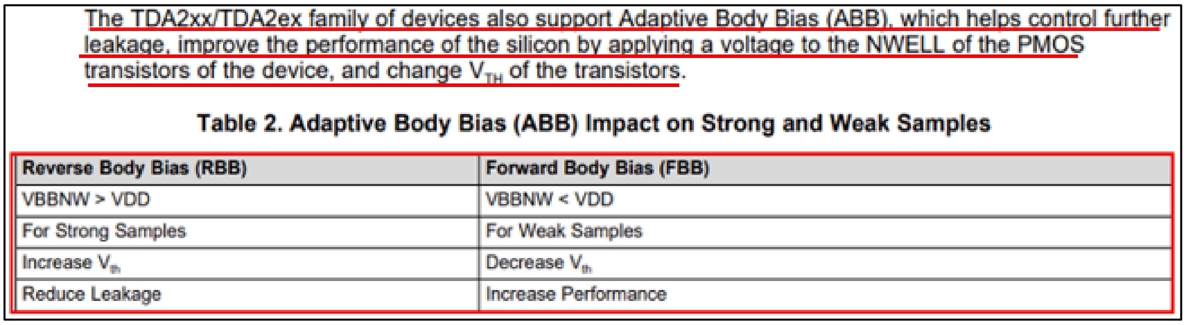
**Source:**

https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

88. TI's TDA2x System-on-Chip devices include means for commanding a voltage supply coupled to a body terminal of the microprocessor to generate a voltage corresponding to the body biasing voltage information corresponding to the particular power condition.

89. For example, TI's TDA2x System-on-Chip devices include a dual core processor module ("microprocessor"). The devices support Adaptive Body Bias (ABB), which allows application of forward body bias to increase performance and reverse body bias to reduce leakage in the device. The ABB voltage value can be programmed—that is, a command can be generated by the device to apply a body bias voltage value. Further, the programmed reverse body bias voltage value is supplied by the ABB LDO ("voltage supply") to the VBBNW pin ("body terminal") of the transistors of the device's microprocessor.

90. To reduce leakage in low power states or low operating performance points (OPP Low), the reverse body bias voltage value ("body biasing voltage information") is accessed and applied to the device. This occurs during one of the low power states of the device, such as the power state in which one core is forced off and the other core is in retention ("said particular power condition"). Accordingly, the device includes means (e.g., programming) that can command the ABB LDO, which is coupled to the VBBNW pin, to supply a reverse body bias voltage corresponding to the low power state, i.e., the power state in which one core is forced off and other core is in retention.

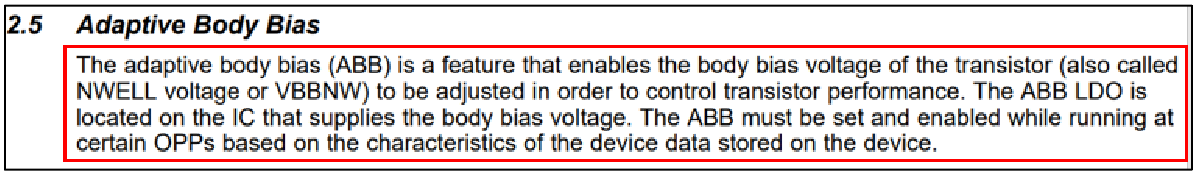


The TDA2xx/TDA2ex family of devices also support Adaptive Body Bias (ABB), which helps control further leakage, improve the performance of the silicon by applying a voltage to the NWELL of the PMOS transistors of the device, and change $V_{TH}$ of the transistors.

**Table 2. Adaptive Body Bias (ABB) Impact on Strong and Weak Samples**

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

**Source:**
https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 5)



**2.5 *Adaptive Body Bias***

The adaptive body bias (ABB) is a feature that enables the body bias voltage of the transistor (also called NWELL voltage or VBBNW) to be adjusted in order to control transistor performance. The ABB LDO is located on the IC that supplies the body bias voltage. The ABB must be set and enabled while running at certain OPPs based on the characteristics of the device data stored on the device.

**Source:**
https://www.ti.com/lit/an/slva646/slva646.pdf?ts=1597323545336&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)

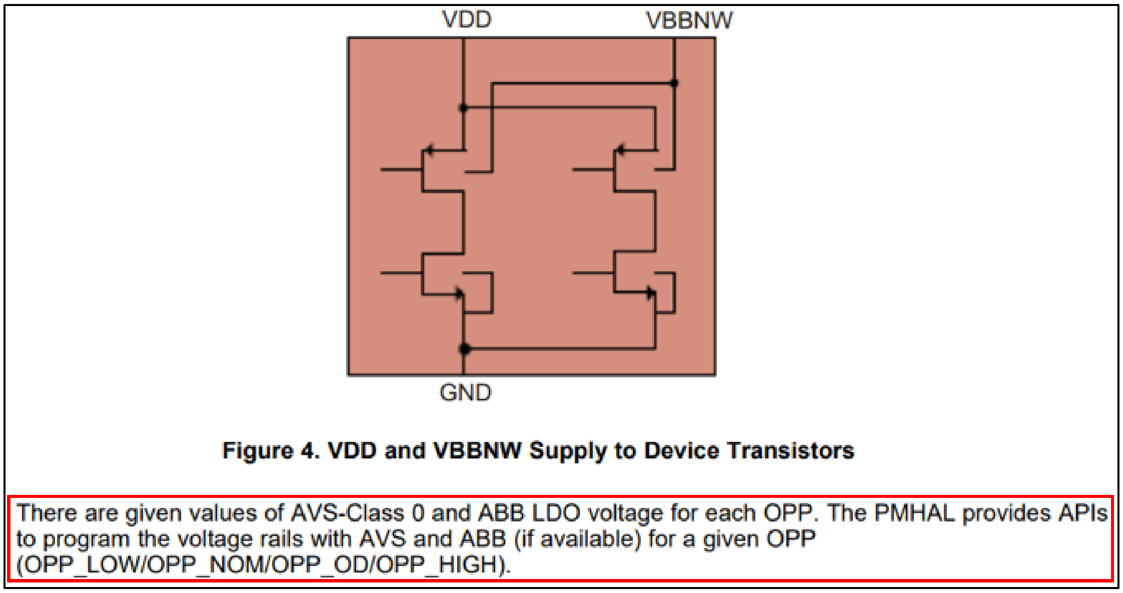
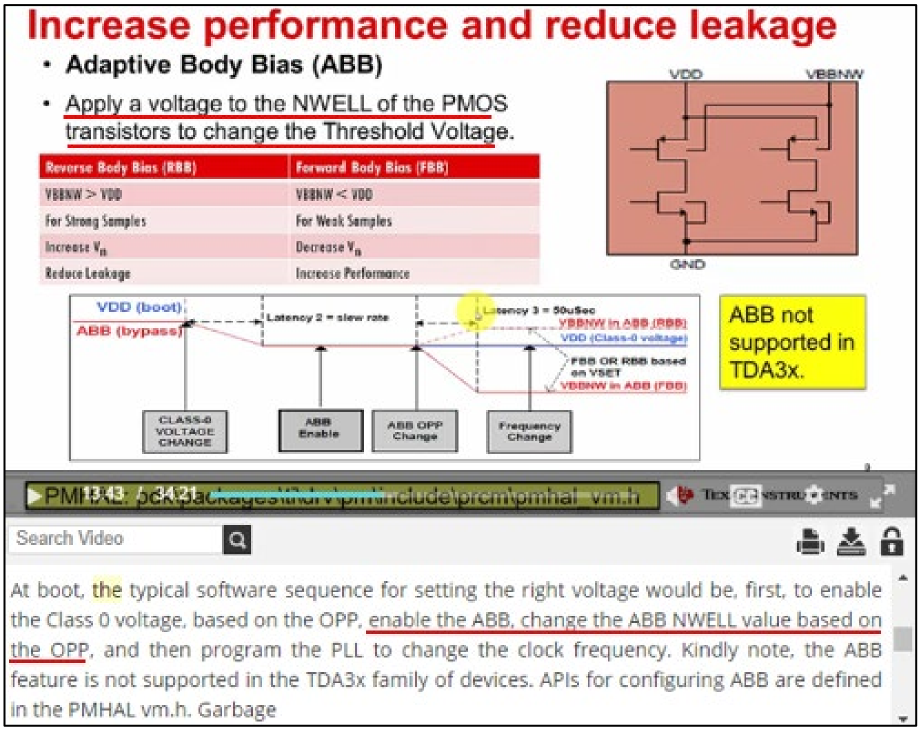


**Figure 4. VDD and VBBNW Supply to Device Transistors**

There are given values of AVS-Class 0 and ABB LDO voltage for each OPP. The PMHAL provides APIs to program the voltage rails with AVS and ABB (if available) for a given OPP (OPP_LOW/OPP_NOM/OPP_OD/OPP_HIGH).

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)



Increase performance and reduce leakage

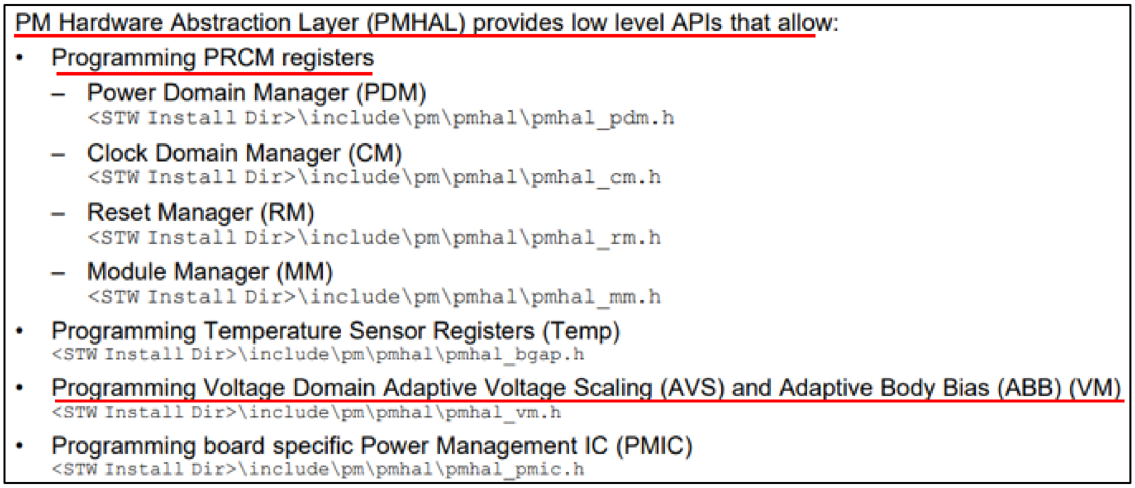
- Adaptive Body Bias (ABB)
- Apply a voltage to the NWELL of the PMOS transistors to change the Threshold Voltage.

| Reverse Body Bias (RBB) | Forward Body Bias (FBB) |
|---|---|
| VBBNW > VDD | VBBNW < VDD |
| For Strong Samples | For Weak Samples |
| Increase $V_{th}$ | Decrease $V_{th}$ |
| Reduce Leakage | Increase Performance |

ABB not supported in TDA3x.

At boot, the typical software sequence for setting the right voltage would be, first, to enable the Class 0 voltage, based on the OPP, enable the ABB, change the ABB NWELL value based on the OPP, and then program the PLL to change the clock frequency. Kindly note, the ABB feature is not supported in the TDA3x family of devices. APIs for configuring ABB are defined in the PMHAL vm.h. Garbage

**Source:** https://training.ti.com/introduction-adas-power-management-software?cu=1137175 (at 13:43)

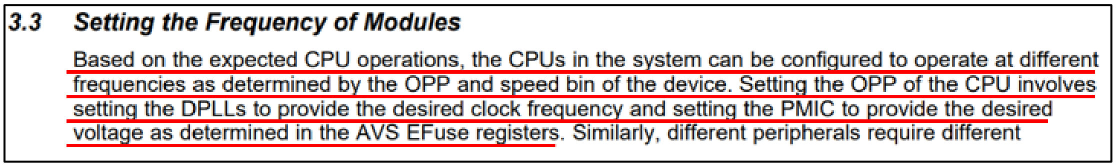


PM Hardware Abstraction Layer (PMHAL) provides low level APIs that allow:

- Programming PRCM registers
  - Power Domain Manager (PDM)
    <STW Install Dir>\include\pm\pmhal\pmhal_pdm.h
  - Clock Domain Manager (CM)
    <STW Install Dir>\include\pm\pmhal\pmhal_cm.h
  - Reset Manager (RM)
    <STW Install Dir>\include\pm\pmhal\pmhal_rm.h
  - Module Manager (MM)
    <STW Install Dir>\include\pm\pmhal\pmhal_mm.h
- Programming Temperature Sensor Registers (Temp)
  <STW Install Dir>\include\pm\pmhal\pmhal_bgap.h
- Programming Voltage Domain Adaptive Voltage Scaling (AVS) and Adaptive Body Bias (ABB) (VM)
  <STW Install Dir>\include\pm\pmhal\pmhal_vm.h
- Programming board specific Power Management IC (PMIC)
  <STW Install Dir>\include\pm\pmhal\pmhal_pmic.h

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 4)



### *3.3 Setting the Frequency of Modules*

Based on the expected CPU operations, the CPUs in the system can be configured to operate at different frequencies as determined by the OPP and speed bin of the device. Setting the OPP of the CPU involves setting the DPLLs to provide the desired clock frequency and setting the PMIC to provide the desired voltage as determined in the AVS EFuse registers. Similarly, different peripherals require different

**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 9)

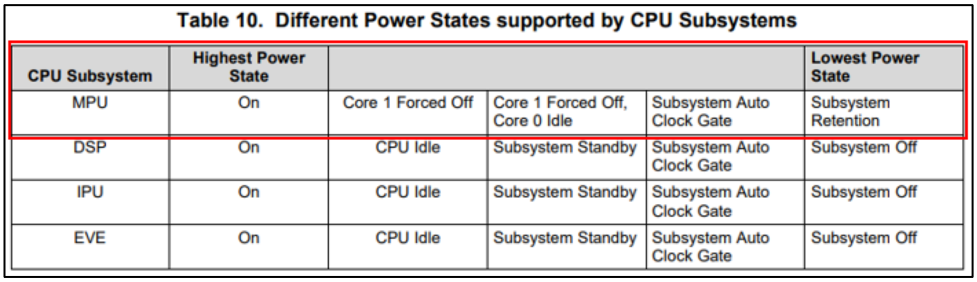
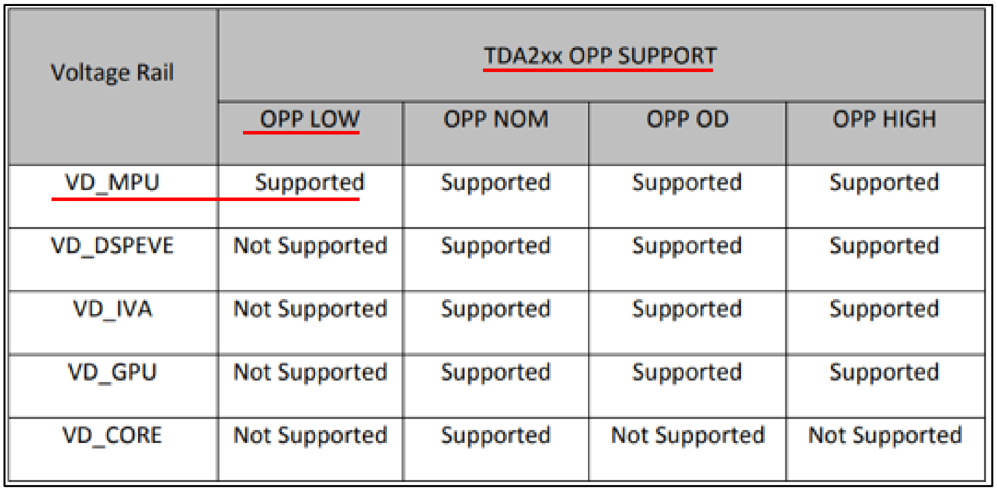


**Table 10. Different Power States supported by CPU Subsystems**

| CPU Subsystem | Highest Power State | | | | Lowest Power State |
|---|---|---|---|---|---|
| MPU | On | Core 1 Forced Off | Core 1 Forced Off, Core 0 Idle | Subsystem Auto Clock Gate | Subsystem Retention |
| DSP | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| IPU | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |
| EVE | On | CPU Idle | Subsystem Standby | Subsystem Auto Clock Gate | Subsystem Off |

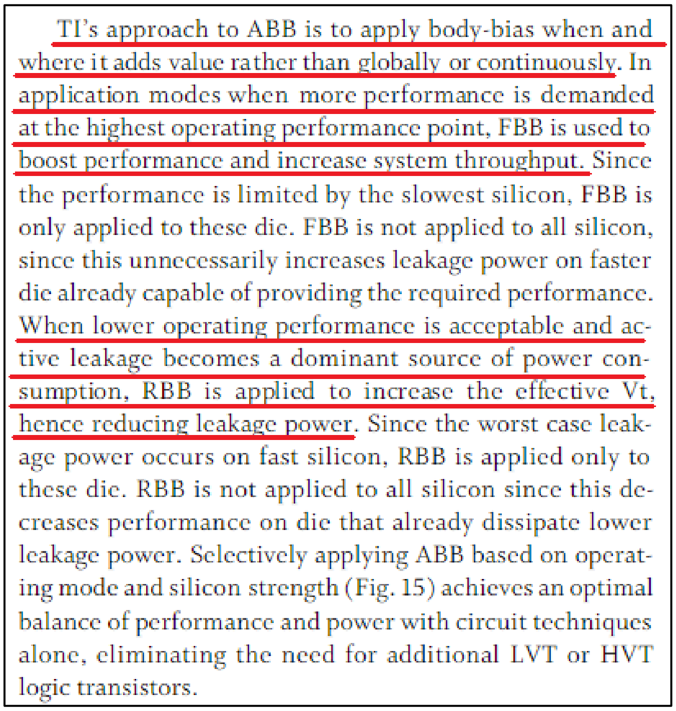
**Source:**

https://www.ti.com/lit/an/sprac22/sprac22.pdf?ts=1597386914341&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 18)



| Voltage Rail | TDA2xx OPP SUPPORT | | | |
|---|---|---|---|---|
| | OPP LOW | OPP NOM | OPP OD | OPP HIGH |
| VD_MPU | Supported | Supported | Supported | Supported |
| VD_DSPEVE | Not Supported | Supported | Supported | Supported |
| VD_IVA | Not Supported | Supported | Supported | Supported |
| VD_GPU | Not Supported | Supported | Supported | Supported |
| VD_CORE | Not Supported | Supported | Not Supported | Not Supported |

**Source:** https://processors.wiki.ti.com/images/b/b7/TDA_SBL_UserGuide.pdf (Page 23)



TI's approach to ABB is to apply body-bias when and where it adds value rather than globally or continuously. In application modes when more performance is demanded at the highest operating performance point, FBB is used to boost performance and increase system throughput. Since the performance is limited by the slowest silicon, FBB is only applied to these die. FBB is not applied to all silicon, since this unnecessarily increases leakage power on faster die already capable of providing the required performance. When lower operating performance is acceptable and active leakage becomes a dominant source of power consumption, RBB is applied to increase the effective Vt, hence reducing leakage power. Since the worst case leakage power occurs on fast silicon, RBB is applied only to these die. RBB is not applied to all silicon since this decreases performance on die that already dissipate lower leakage power. Selectively applying ABB based on operating mode and silicon strength (Fig. 15) achieves an optimal balance of performance and power with circuit techniques alone, eliminating the need for additional LVT or HVT logic transistors.

**Source:**

https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

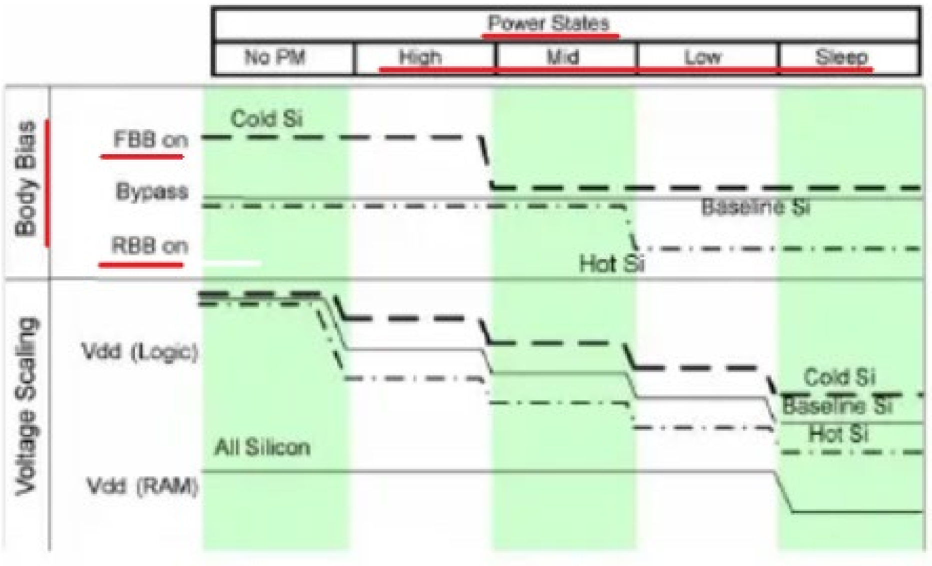


**Fig. 15.** ***Adaptivebody-biasand adaptive voltage scaling configuration with mode and process. Hot and cold refer the process strength or speed (fast and slow).***

**Source:**
https://www.academia.edu/29769936/SmartReflex_Power_and_Performance_Management_Technologies_for_90_nm_65_nm_and_45_nm_Mobile_Application_Processors (Page 152)

91. TI has had knowledge of the ’496 Patent at least as of the date when it was notified of the filing of this action.

92. Liberty Patents has been damaged as a result of the infringing conduct by TI alleged above. Thus, TI is liable to Liberty Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

93. Liberty Patents and/or its predecessors-in-interest have satisfied all statutory obligations required to collect pre-filing damages for the full period allowed by law for infringement of the ’496 Patent.

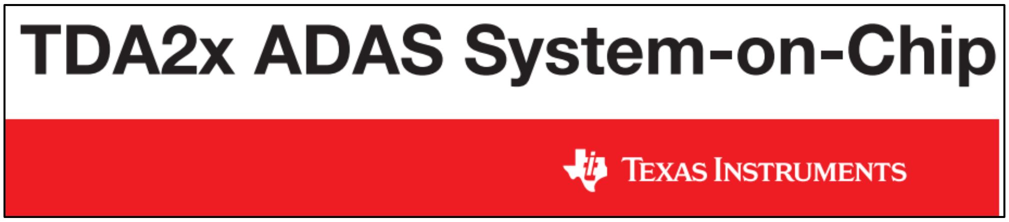
## COUNT IV

## DIRECT INFRINGEMENT OF U.S. PATENT NO. 6,535,959

94. On March 18, 2003, U.S. Patent No. 6,535,959 ("the '959 Patent") was duly and legally issued by the United States Patent and Trademark Office for an invention entitled "Circuit and Method for Reducing Power Consumption in an Instruction Cache."

95. Liberty Patents is the owner of the '959 Patent, with all substantive rights in and to that patent, including the sole and exclusive right to prosecute this action and enforce the '959 Patent against infringers, and to collect damages for all relevant times.

96. TI made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, TI's TDA2x System-on-Chip devices and other products[9] that include processors with the capability to ignore reading the tag field when a sequential instruction is to be loaded (processors such as the ARM Cortex-A72, Cortex-A57, Cortex-A15, Cortex-A9, Cortex-R5, Cortex-R4, ARM11, etc.) ("accused products"):

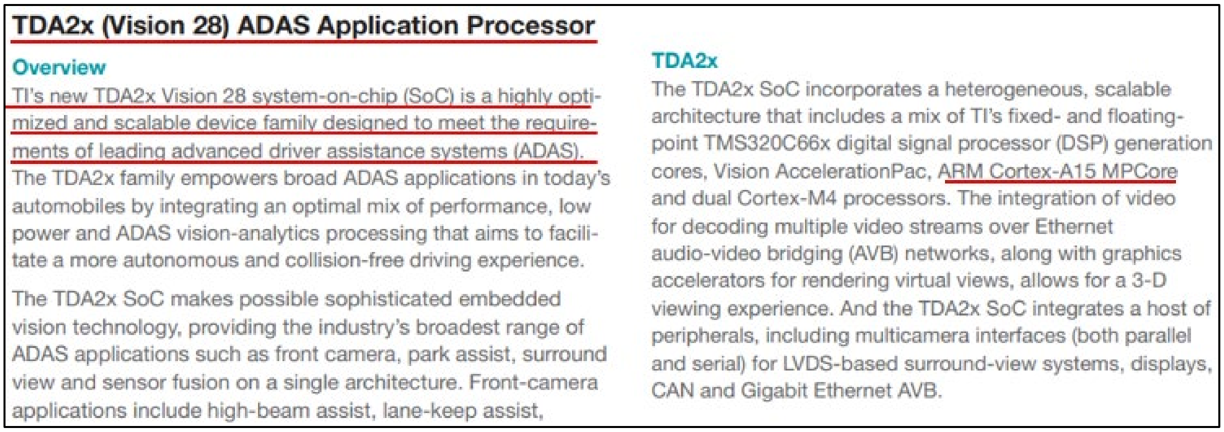


TDA2x ADAS System-on-Chip

TEXAS INSTRUMENTS

**Source:**
https://www.ti.com/lit/ml/sprt681/sprt681.pdf?ts=1597386693056&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 1)

[9] *See, e.g.*, TI OMAP 4470, 66AK2Hx KeyStone Multicore System-on-Chips, DRA829V Jacinto Automotive Processors, etc.



## TDA2x (Vision 28) ADAS Application Processor

### Overview

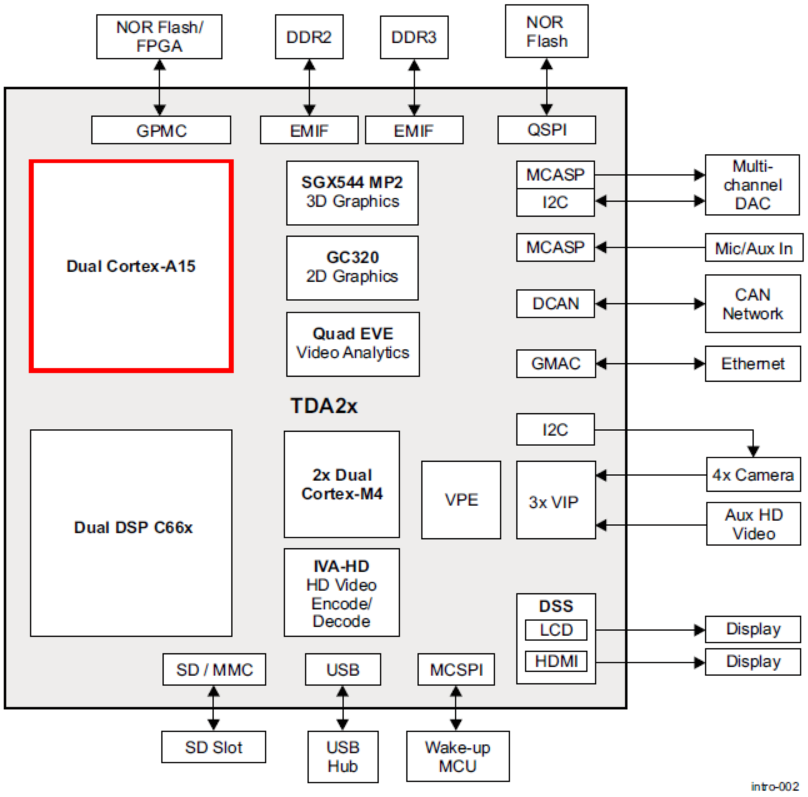
TI's new TDA2x Vision 28 system-on-chip (SoC) is a highly optimized and scalable device family designed to meet the requirements of leading advanced driver assistance systems (ADAS). The TDA2x family empowers broad ADAS applications in today's automobiles by integrating an optimal mix of performance, low power and ADAS vision-analytics processing that aims to facilitate a more autonomous and collision-free driving experience.

The TDA2x SoC makes possible sophisticated embedded vision technology, providing the industry's broadest range of ADAS applications such as front camera, park assist, surround view and sensor fusion on a single architecture. Front-camera applications include high-beam assist, lane-keep assist,

### TDA2x

The TDA2x SoC incorporates a heterogeneous, scalable architecture that includes a mix of TI's fixed- and floating-point TMS320C66x digital signal processor (DSP) generation cores, Vision AccelerationPac, ARM Cortex-A15 MPCore and dual Cortex-M4 processors. The integration of video for decoding multiple video streams over Ethernet audio-video bridging (AVB) networks, along with graphics accelerators for rendering virtual views, allows for a 3-D viewing experience. And the TDA2x SoC integrates a host of peripherals, including multicamera interfaces (both parallel and serial) for LVDS-based surround-view systems, displays, CAN and Gigabit Ethernet AVB.

**Source:** https://m.eet.com/media/1241324/slyy044.pdf (Page 5)



A The speeds shown are the highest design targeted, non-binned OPP.

**Figure 1. TDA2xx Device Block Diagram**

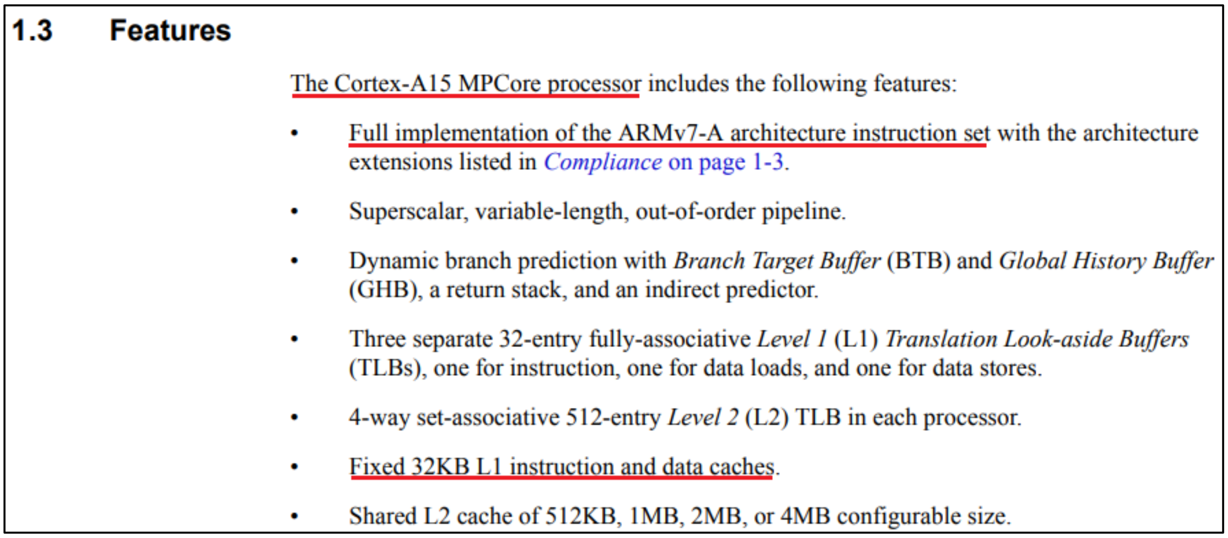
**Source:**

https://www.ti.com/lit/an/sprac21a/sprac21a.pdf?ts=1597386703137&ref_url=https%253A%252F%252Fwww.google.com%252F (Page 6)

97. By doing so, TI has directly infringed (literally and/or under the doctrine of equivalents) at least Claims 1 and 9 of the ’959 Patent. TI’s infringement in this regard is ongoing.

98. TI’s TDA2x System-on-Chip devices are exemplary accused products that include the Cortex-A15 processor. The Cortex-A15 is an exemplary processor that infringes the ’959 Patent.

99. For example, the ARM Cortex-A15 in the accused products is a processor that includes a 2-way set associative instruction cache, which consists of multiple lines or blocks (“plurality of blocks”).



### 1.3 Features

The Cortex-A15 MPCore processor includes the following features:

- Full implementation of the ARMv7-A architecture instruction set with the architecture extensions listed in *Compliance* on page 1-3.
- Superscalar, variable-length, out-of-order pipeline.
- Dynamic branch prediction with *Branch Target Buffer* (BTB) and *Global History Buffer* (GHB), a return stack, and an indirect predictor.
- Three separate 32-entry fully-associative *Level 1* (L1) *Translation Look-aside Buffers* (TLBs), one for instruction, one for data loads, and one for data stores.
- 4-way set-associative 512-entry *Level 2* (L2) TLB in each processor.
- Fixed 32KB L1 instruction and data caches.
- Shared L2 cache of 512KB, 1MB, 2MB, or 4MB configurable size.

**Source:** https://static.docs.arm.com/ddi0438/i/DDI0438.pdf (Page 16)

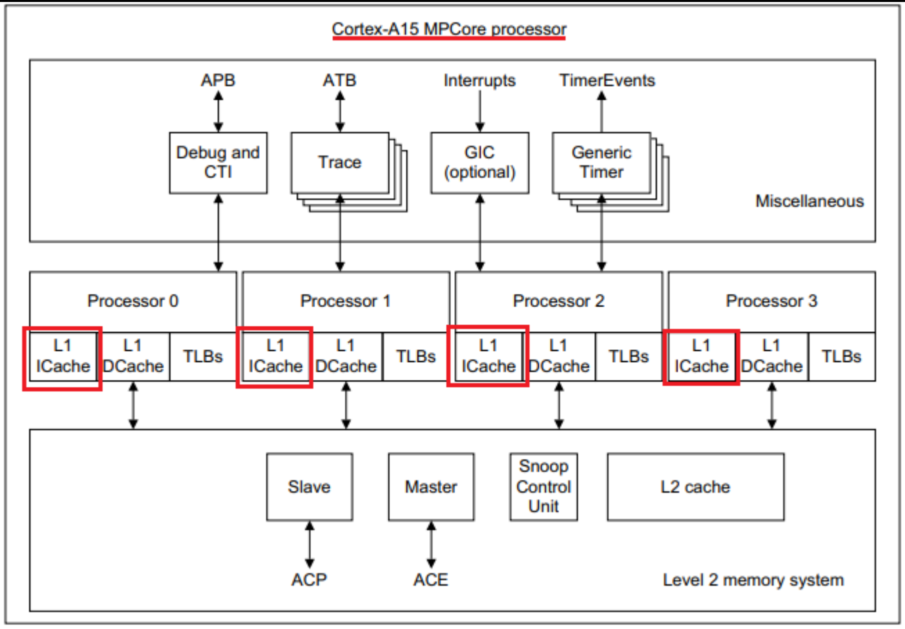
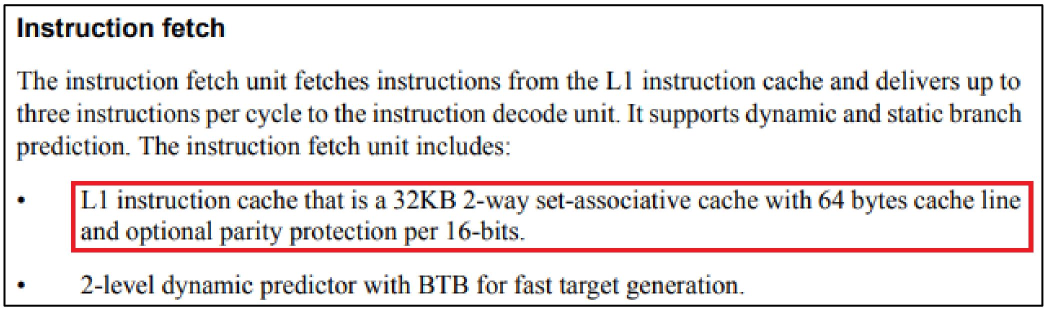


**Figure 1-1 Example multiprocessor configuration**

**Source:** https://static.docs.arm.com/ddi0438/i/DDI0438.pdf (Page 13)

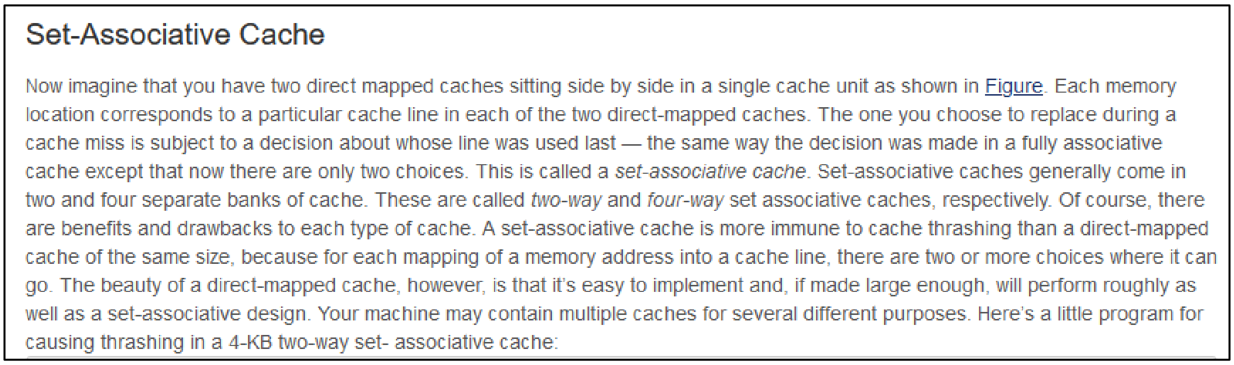


**Instruction fetch**

The instruction fetch unit fetches instructions from the L1 instruction cache and delivers up to three instructions per cycle to the instruction decode unit. It supports dynamic and static branch prediction. The instruction fetch unit includes:

- L1 instruction cache that is a 32KB 2-way set-associative cache with 64 bytes cache line and optional parity protection per 16-bits.
- 2-level dynamic predictor with BTB for fast target generation.

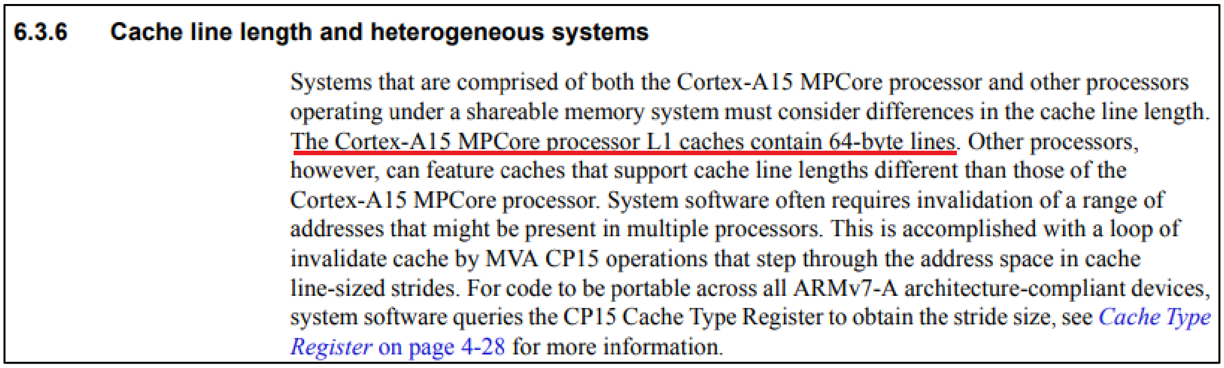
**Source:** https://static.docs.arm.com/ddi0438/i/DDI0438.pdf (Page 28)



### Set-Associative Cache

Now imagine that you have two direct mapped caches sitting side by side in a single cache unit as shown in Figure. Each memory location corresponds to a particular cache line in each of the two direct-mapped caches. The one you choose to replace during a cache miss is subject to a decision about whose line was used last — the same way the decision was made in a fully associative cache except that now there are only two choices. This is called a *set-associative cache*. Set-associative caches generally come in two and four separate banks of cache. These are called *two-way* and *four-way* set associative caches, respectively. Of course, there are benefits and drawbacks to each type of cache. A set-associative cache is more immune to cache thrashing than a direct-mapped cache of the same size, because for each mapping of a memory address into a cache line, there are two or more choices where it can go. The beauty of a direct-mapped cache, however, is that it's easy to implement and, if made large enough, will perform roughly as well as a set-associative design. Your machine may contain multiple caches for several different purposes. Here's a little program for causing thrashing in a 4-KB two-way set- associative cache:

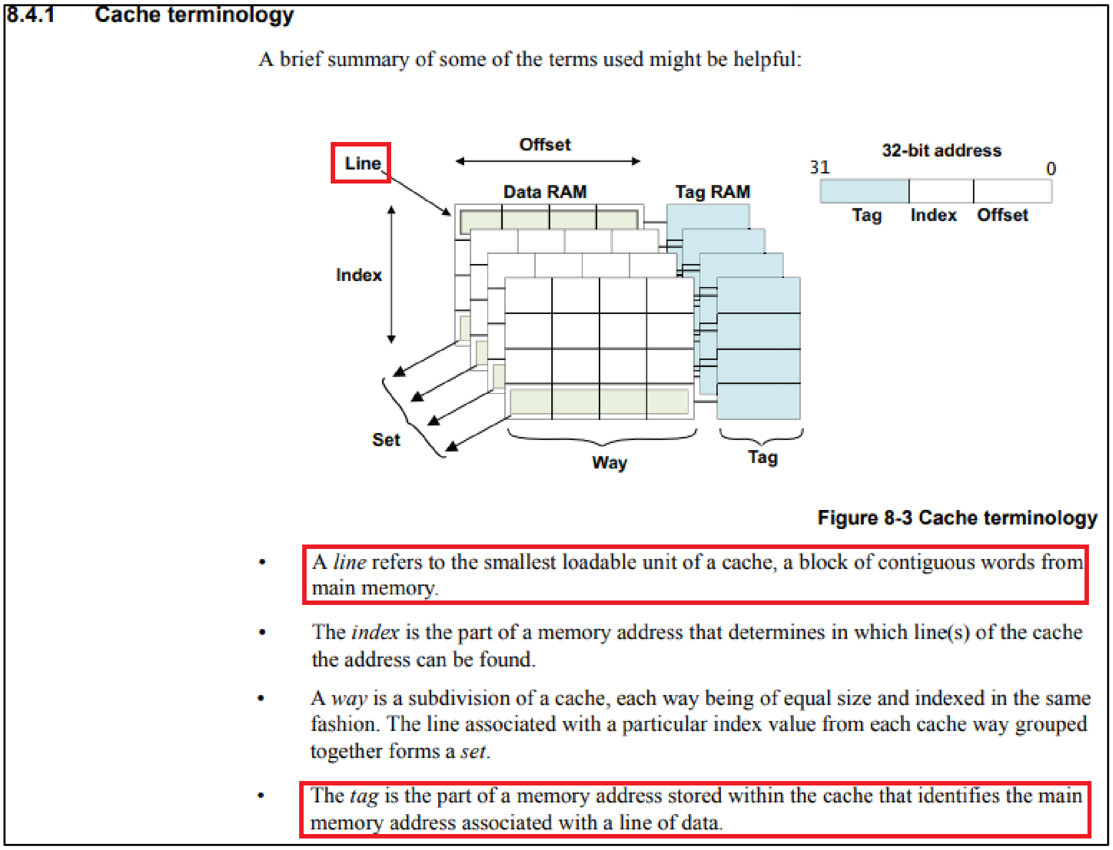
**Source:** https://cnx.org/contents/u4IVVH92@5.2:n6sjy3Ju@3/Cache-Organization



### 6.3.6 Cache line length and heterogeneous systems

Systems that are comprised of both the Cortex-A15 MPCore processor and other processors operating under a shareable memory system must consider differences in the cache line length. The Cortex-A15 MPCore processor L1 caches contain 64-byte lines. Other processors, however, can feature caches that support cache line lengths different than those of the Cortex-A15 MPCore processor. System software often requires invalidation of a range of addresses that might be present in multiple processors. This is accomplished with a loop of invalidate cache by MVA CP15 operations that step through the address space in cache line-sized strides. For code to be portable across all ARMv7-A architecture-compliant devices, system software queries the CP15 Cache Type Register to obtain the stride size, see *Cache Type Register* on page 4-28 for more information.

**Source:** https://static.docs.arm.com/ddi0438/i/DDI0438.pdf (Page 199)



### 8.4.1 Cache terminology

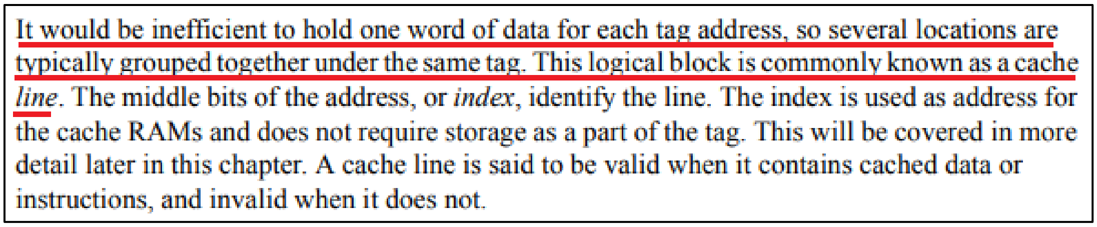
A brief summary of some of the terms used might be helpful:

**Figure 8-3 Cache terminology**

- A *line* refers to the smallest loadable unit of a cache, a block of contiguous words from main memory.
- The *index* is the part of a memory address that determines in which line(s) of the cache the address can be found.
- A *way* is a subdivision of a cache, each way being of equal size and indexed in the same fashion. The line associated with a particular index value from each cache way grouped together forms a *set*.
- The *tag* is the part of a memory address stored within the cache that identifies the main memory address associated with a line of data.

**Source:** https://documentation-service.arm.com/static/5e909fb6c8052b16087625aa?token=

(Pages 112-13)



It would be inefficient to hold one word of data for each tag address, so several locations are typically grouped together under the same tag. This logical block is commonly known as a cache *line*. The middle bits of the address, or *index*, identify the line. The index is used as address for the cache RAMs and does not require storage as a part of the tag. This will be covered in more detail later in this chapter. A cache line is said to be valid when it contains cached data or instructions, and invalid when it does not.

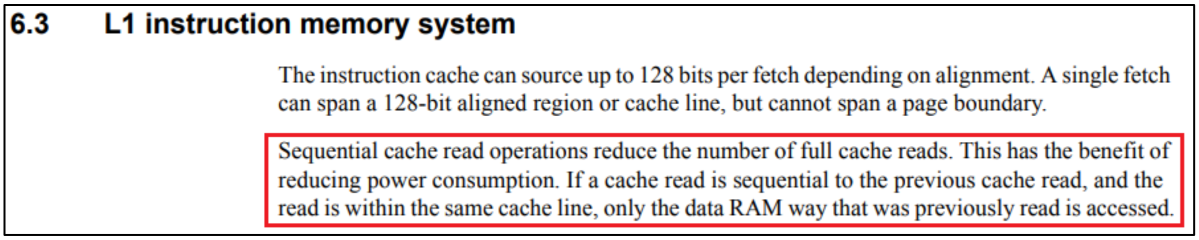
**Source:** https://documentation-service.arm.com/static/5e909fb6c8052b16087625aa?token= (Page 112)

100. The ARM Cortex-A15 processor includes a circuit configured to generate a power reduction signal. The power reduction signal indicates if a subsequent instruction to be fetched is in a same block of the plurality of blocks as a previous instruction fetched from the instruction cache.

101. For example, the ARM Cortex-A15 processor supports a power reduction method that is operational when an instruction is being accessed from the instruction cache. The instruction cache includes multiple cache lines or blocks, and each cache line or block is associated with a tag value. These tag values are stored in the tag RAM. The cache also includes data RAM for storing the instructions.

102. If a sequential ("subsequent") instruction to be read from the instruction cache is in the same cache line or block as the previous instruction, only the data RAM that was previously read is accessed for the instruction, and the tag RAM is *not* accessed.

103. Accordingly, the device, which includes an ARM Cortex-A15, includes a circuit that can generate a signal ("power reduction signal") if a sequential ("subsequent") instruction to be accessed ("fetched") from the cache is identified as being in the same cache line or block.

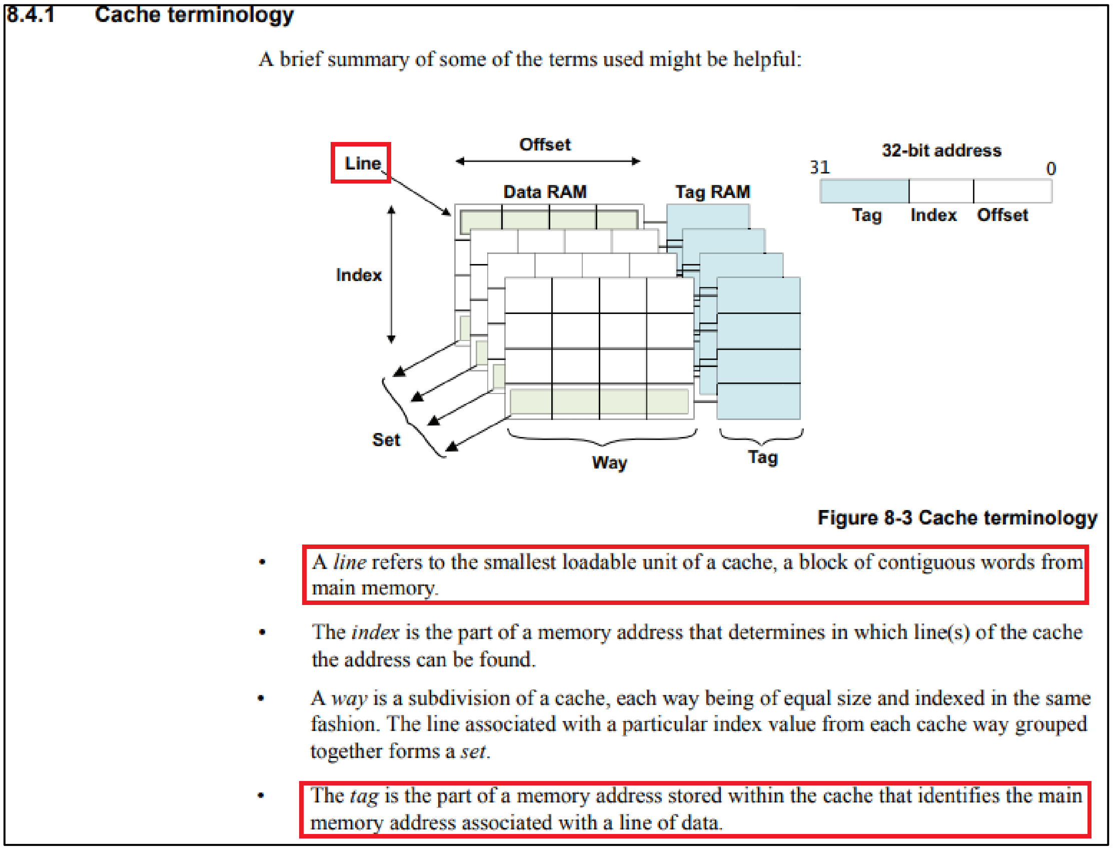


### 6.3 L1 instruction memory system

The instruction cache can source up to 128 bits per fetch depending on alignment. A single fetch can span a 128-bit aligned region or cache line, but cannot span a page boundary.

Sequential cache read operations reduce the number of full cache reads. This has the benefit of reducing power consumption. If a cache read is sequential to the previous cache read, and the read is within the same cache line, only the data RAM way that was previously read is accessed.

**Source:** https://static.docs.arm.com/ddi0438/i/DDI0438.pdf (Page 198)



#### 8.4.1 Cache terminology

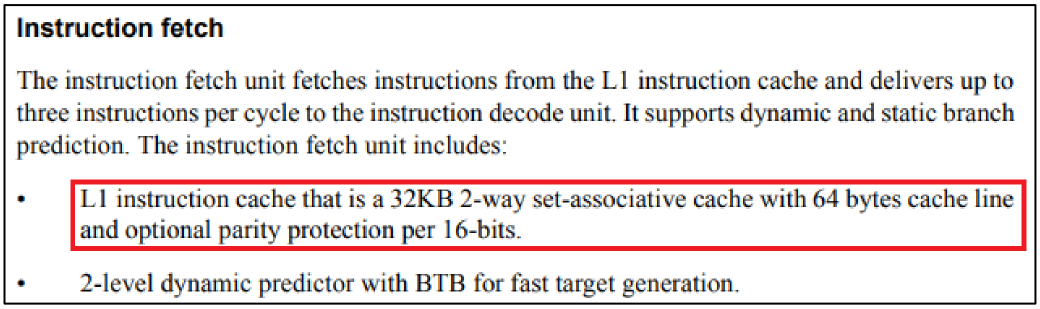
A brief summary of some of the terms used might be helpful:

**Figure 8-3 Cache terminology**

- A *line* refers to the smallest loadable unit of a cache, a block of contiguous words from main memory.
- The *index* is the part of a memory address that determines in which line(s) of the cache the address can be found.
- A *way* is a subdivision of a cache, each way being of equal size and indexed in the same fashion. The line associated with a particular index value from each cache way grouped together forms a *set*.
- The *tag* is the part of a memory address stored within the cache that identifies the main memory address associated with a line of data.

**Source:** https://documentation-service.arm.com/static/5e909fb6c8052b16087625aa?token=

(Pages 112-13)

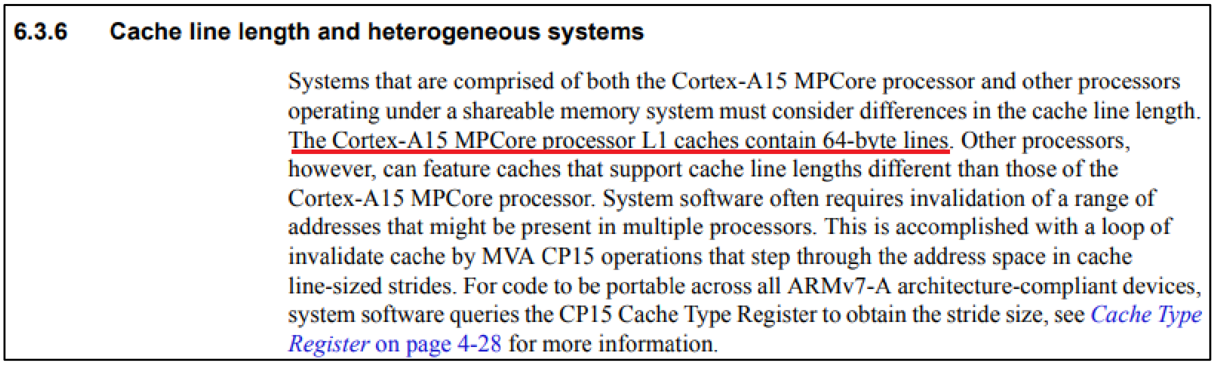


**Instruction fetch**

The instruction fetch unit fetches instructions from the L1 instruction cache and delivers up to three instructions per cycle to the instruction decode unit. It supports dynamic and static branch prediction. The instruction fetch unit includes:

- L1 instruction cache that is a 32KB 2-way set-associative cache with 64 bytes cache line and optional parity protection per 16-bits.
- 2-level dynamic predictor with BTB for fast target generation.

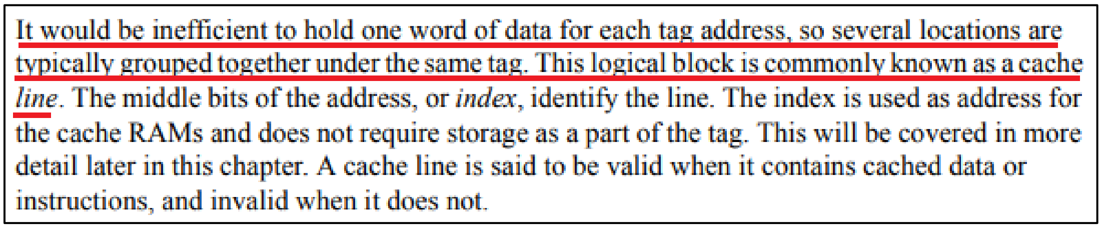
**Source:** https://static.docs.arm.com/ddi0438/i/DDI0438.pdf (Page 28)



**6.3.6 Cache line length and heterogeneous systems**

Systems that are comprised of both the Cortex-A15 MPCore processor and other processors operating under a shareable memory system must consider differences in the cache line length. The Cortex-A15 MPCore processor L1 caches contain 64-byte lines. Other processors, however, can feature caches that support cache line lengths different than those of the Cortex-A15 MPCore processor. System software often requires invalidation of a range of addresses that might be present in multiple processors. This is accomplished with a loop of invalidate cache by MVA CP15 operations that step through the address space in cache line-sized strides. For code to be portable across all ARMv7-A architecture-compliant devices, system software queries the CP15 Cache Type Register to obtain the stride size, see *Cache Type Register* on page 4-28 for more information.

**Source:** https://static.docs.arm.com/ddi0438/i/DDI0438.pdf (Page 199)



It would be inefficient to hold one word of data for each tag address, so several locations are typically grouped together under the same tag. This logical block is commonly known as a cache *line*. The middle bits of the address, or *index*, identify the line. The index is used as address for the cache RAMs and does not require storage as a part of the tag. This will be covered in more detail later in this chapter. A cache line is said to be valid when it contains cached data or instructions, and invalid when it does not.

**Source:** https://documentation-service.arm.com/static/5e909fb6c8052b16087625aa?token= (Page 112)

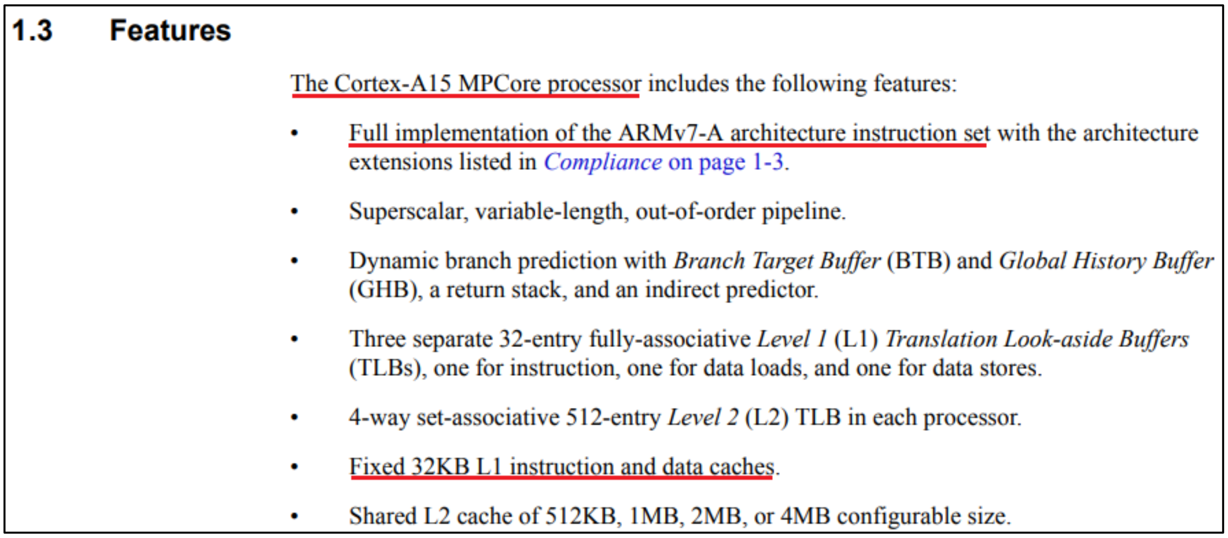
104. TI has infringed the ’959 Patent by using the accused products, which include an ARM Cortex-A15 processor, and thereby practicing a method for reducing power consumed in an instruction cache having a plurality of blocks.

105. The method practiced using the accused products comprises the step of generating a power reduction signal to indicate if a subsequent instruction to be fetched is in a same block of the plurality of blocks as a previous instruction fetched from the instruction cache.

106. For example, the ARM Cortex-A15 processor includes a 2-way set associative instruction cache, which consists of multiple cache lines or blocks (“plurality of blocks”). Each cache line or block consists of data (i.e., instructions) and an associated tag value. These tag values are stored in the tag RAM and the data (i.e., instructions) is stored in data RAM. Contiguous instructions from main memory are copied and stored into a single cache line and are associated with a single tag value.

107. The Cortex-A15 processor is capable of reducing power consumption through the sequential cache read process. While accessing the instruction cache, the sequential ("subsequent") instruction that is to be read can be in the same cache line or block as the previously-read instruction. In such cases, only the data RAM that was previously read is accessed, and the tag RAM is *not* accessed because the subsequent instruction is from the same cache line or block and has the same tag RAM value as the previous instruction.

108. Accordingly, the processor is capable of generating a signal ("power reduction signal") to indicate whether the sequential ("subsequent") instruction to be accessed is in the same cache line or block as the previous instruction accessed from the instruction cache.



**1.3 Features**

The Cortex-A15 MPCore processor includes the following features:

- Full implementation of the ARMv7-A architecture instruction set with the architecture extensions listed in *Compliance* on page 1-3.
- Superscalar, variable-length, out-of-order pipeline.
- Dynamic branch prediction with *Branch Target Buffer* (BTB) and *Global History Buffer* (GHB), a return stack, and an indirect predictor.
- Three separate 32-entry fully-associative *Level 1* (L1) *Translation Look-aside Buffers* (TLBs), one for instruction, one for data loads, and one for data stores.
- 4-way set-associative 512-entry *Level 2* (L2) TLB in each processor.
- Fixed 32KB L1 instruction and data caches.
- Shared L2 cache of 512KB, 1MB, 2MB, or 4MB configurable size.

**Source:** https://static.docs.arm.com/ddi0438/i/DDI0438.pdf (Page 16)

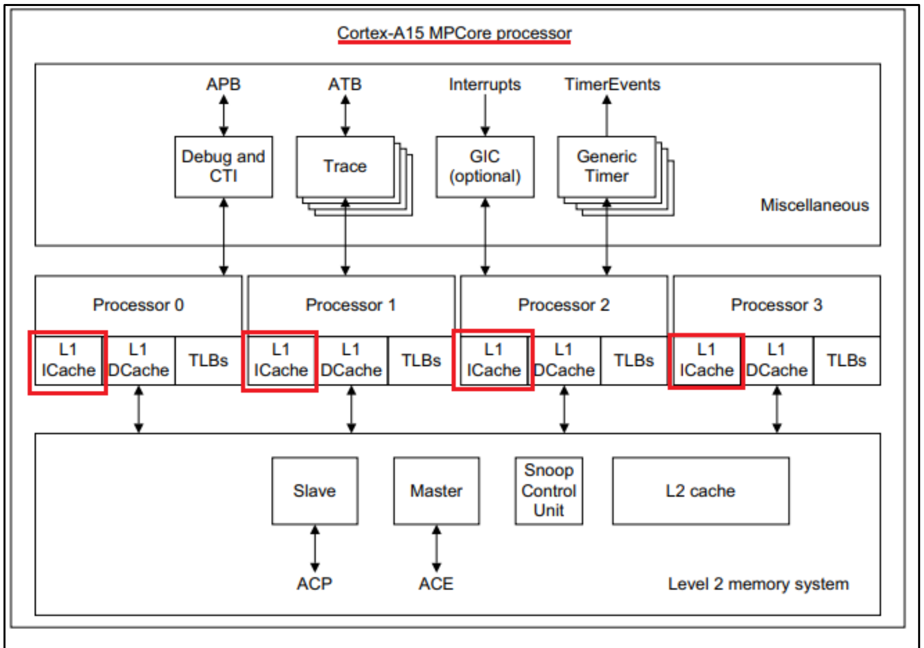
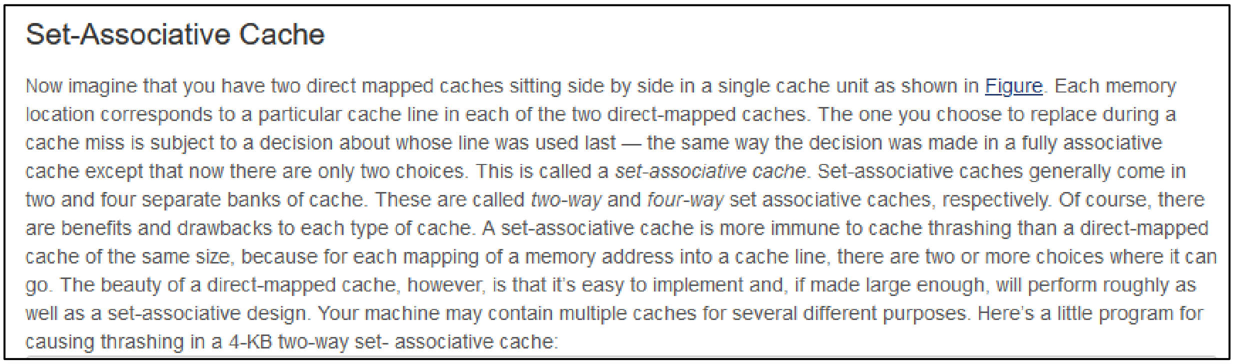


**Figure 1-1 Example multiprocessor configuration**

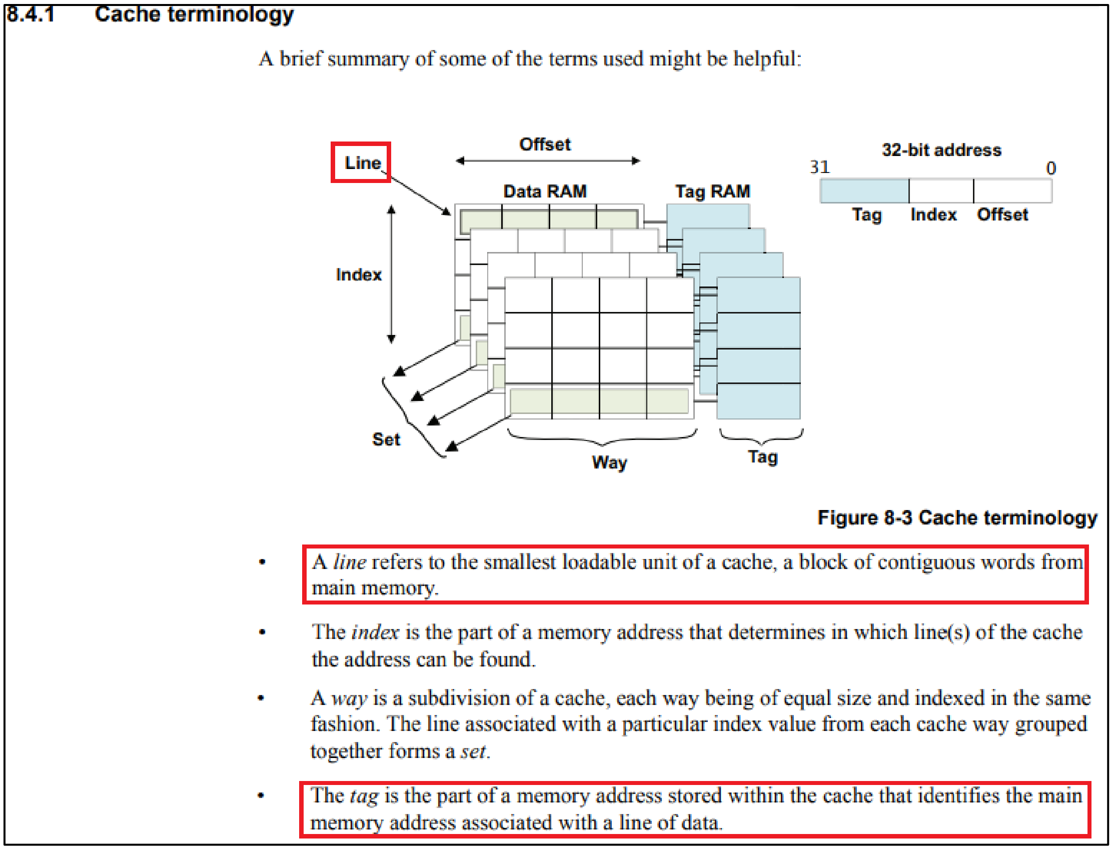
**Source:** https://static.docs.arm.com/ddi0438/i/DDI0438.pdf (Page 13)



## Set-Associative Cache

Now imagine that you have two direct mapped caches sitting side by side in a single cache unit as shown in Figure. Each memory location corresponds to a particular cache line in each of the two direct-mapped caches. The one you choose to replace during a cache miss is subject to a decision about whose line was used last — the same way the decision was made in a fully associative cache except that now there are only two choices. This is called a *set-associative cache*. Set-associative caches generally come in two and four separate banks of cache. These are called *two-way* and *four-way* set associative caches, respectively. Of course, there are benefits and drawbacks to each type of cache. A set-associative cache is more immune to cache thrashing than a direct-mapped cache of the same size, because for each mapping of a memory address into a cache line, there are two or more choices where it can go. The beauty of a direct-mapped cache, however, is that it's easy to implement and, if made large enough, will perform roughly as well as a set-associative design. Your machine may contain multiple caches for several different purposes. Here's a little program for causing thrashing in a 4-KB two-way set- associative cache:

**Source:** https://cnx.org/contents/u4IVVH92@5.2:n6sjy3Ju@3/Cache-Organization



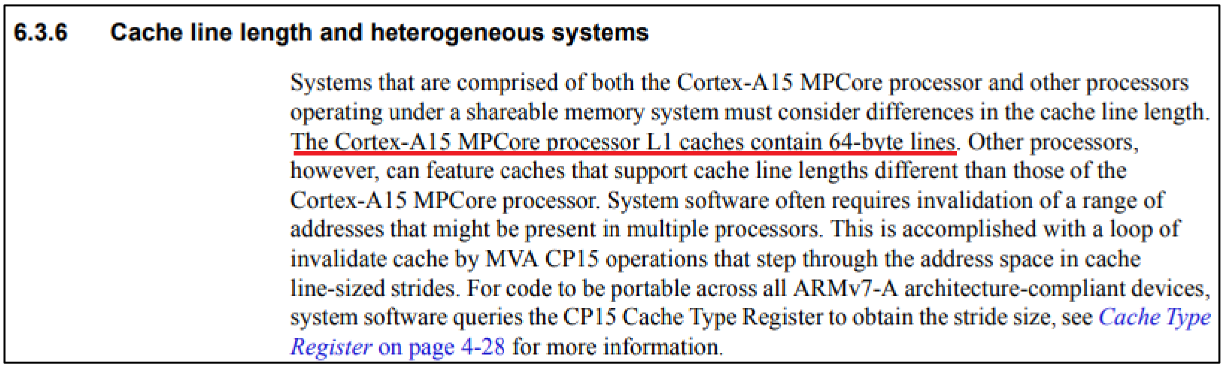
### 8.4.1 Cache terminology

A brief summary of some of the terms used might be helpful:

Figure 8-3 Cache terminology

- A *line* refers to the smallest loadable unit of a cache, a block of contiguous words from main memory.
- The *index* is the part of a memory address that determines in which line(s) of the cache the address can be found.
- A *way* is a subdivision of a cache, each way being of equal size and indexed in the same fashion. The line associated with a particular index value from each cache way grouped together forms a *set*.
- The *tag* is the part of a memory address stored within the cache that identifies the main memory address associated with a line of data.

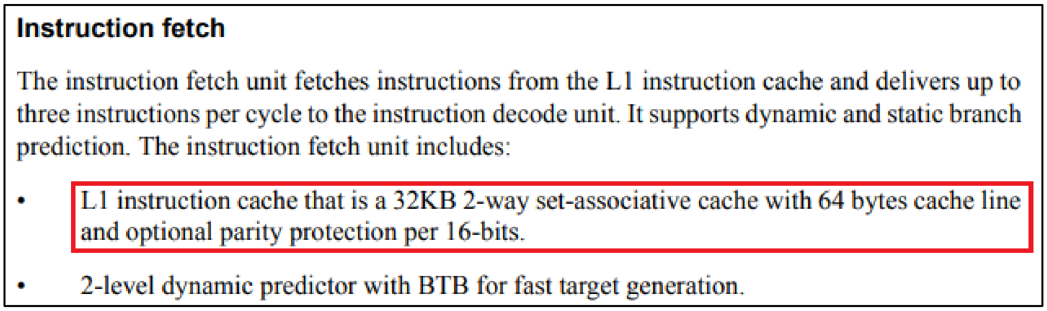
**Source:** https://documentation-service.arm.com/static/5e909fb6c8052b16087625aa?token= (Pages 112-13)



### 6.3.6 Cache line length and heterogeneous systems

Systems that are comprised of both the Cortex-A15 MPCore processor and other processors operating under a shareable memory system must consider differences in the cache line length. The Cortex-A15 MPCore processor L1 caches contain 64-byte lines. Other processors, however, can feature caches that support cache line lengths different than those of the Cortex-A15 MPCore processor. System software often requires invalidation of a range of addresses that might be present in multiple processors. This is accomplished with a loop of invalidate cache by MVA CP15 operations that step through the address space in cache line-sized strides. For code to be portable across all ARMv7-A architecture-compliant devices, system software queries the CP15 Cache Type Register to obtain the stride size, see *Cache Type Register* on page 4-28 for more information.

**Source:** https://static.docs.arm.com/ddi0438/i/DDI0438.pdf (Page 199)

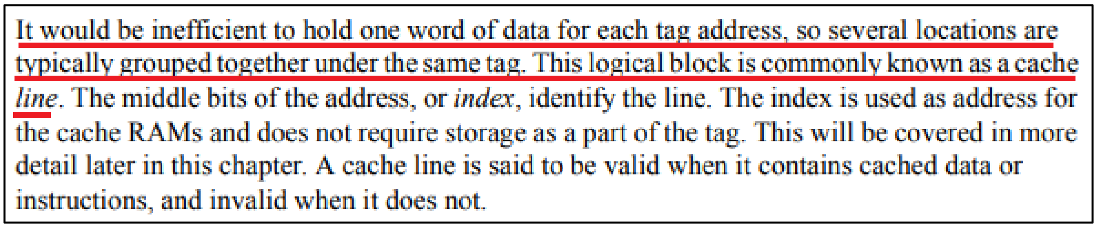


**Instruction fetch**

The instruction fetch unit fetches instructions from the L1 instruction cache and delivers up to three instructions per cycle to the instruction decode unit. It supports dynamic and static branch prediction. The instruction fetch unit includes:

- L1 instruction cache that is a 32KB 2-way set-associative cache with 64 bytes cache line and optional parity protection per 16-bits.
- 2-level dynamic predictor with BTB for fast target generation.

**Source:** https://static.docs.arm.com/ddi0438/i/DDI0438.pdf (Page 28)



It would be inefficient to hold one word of data for each tag address, so several locations are typically grouped together under the same tag. This logical block is commonly known as a cache *line*. The middle bits of the address, or *index*, identify the line. The index is used as address for the cache RAMs and does not require storage as a part of the tag. This will be covered in more detail later in this chapter. A cache line is said to be valid when it contains cached data or instructions, and invalid when it does not.

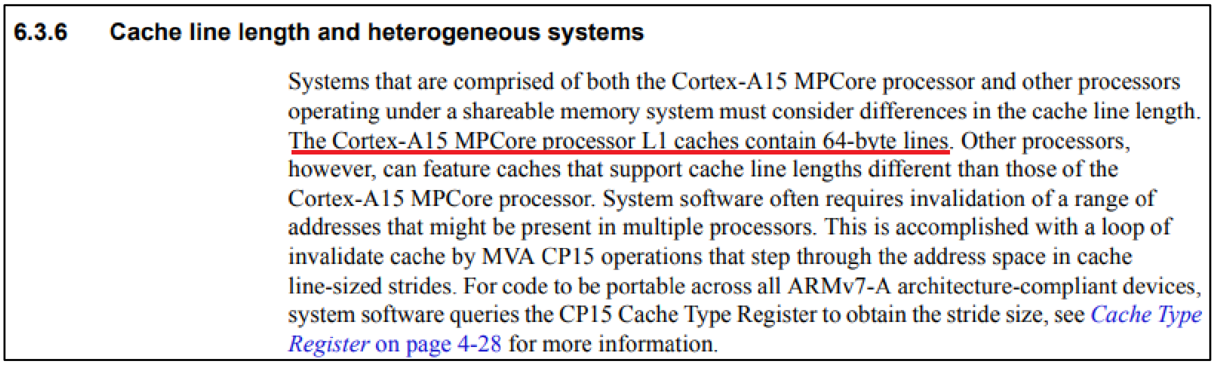
**Source:** https://documentation-service.arm.com/static/5e909fb6c8052b16087625aa?token= (Page 112)

109. The method practiced using the accused products further comprises the step of enabling the instruction cache for reading the subsequent instruction from the instruction cache when the power reduction signal is in a first logical state.

110. For example, the processor is capable of generating a signal (“power reduction signal”) that indicates whether the sequential (“subsequent”) instruction to be accessed is in the same cache line or block as the previous instruction. The processor can implement a method based on this signal (“power reduction signal”) to enable the data RAM way that was previously read.

111. The Cortex-A15 processor is capable of generating signals that have two logical states: logic 1 (high) and logic 0 (low). Specifically, the processor can generate a signal (“power reduction signal”) whose first logical state, e.g., logic 1 (high), indicates that the sequential (“subsequent”) instruction is in the same cache line or block as the previous instruction. Based on this signal (“power reduction signal”), the processor enables the data RAM way that was

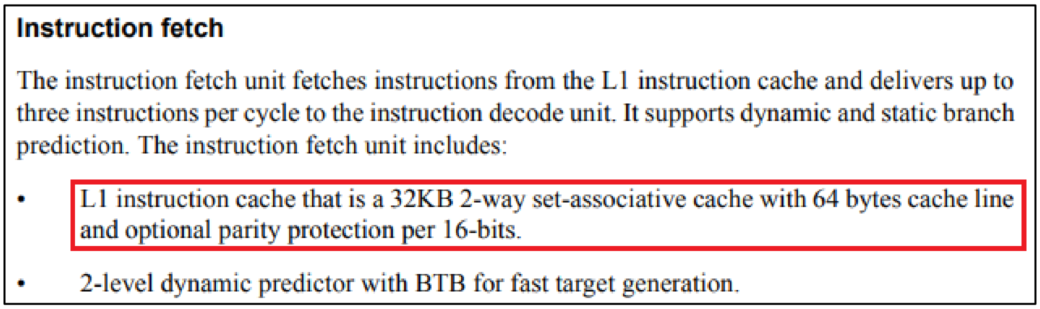
previously read, and the subsequent instruction is read. Accordingly, the subsequent instruction is read when the power reduction signal is in a first logical state, e.g., logic 1 (high).



**6.3.6 Cache line length and heterogeneous systems**

Systems that are comprised of both the Cortex-A15 MPCore processor and other processors operating under a shareable memory system must consider differences in the cache line length. The Cortex-A15 MPCore processor L1 caches contain 64-byte lines. Other processors, however, can feature caches that support cache line lengths different than those of the Cortex-A15 MPCore processor. System software often requires invalidation of a range of addresses that might be present in multiple processors. This is accomplished with a loop of invalidate cache by MVA CP15 operations that step through the address space in cache line-sized strides. For code to be portable across all ARMv7-A architecture-compliant devices, system software queries the CP15 Cache Type Register to obtain the stride size, see *Cache Type Register* on page 4-28 for more information.

**Source:** https://static.docs.arm.com/ddi0438/i/DDI0438.pdf (Page 199)

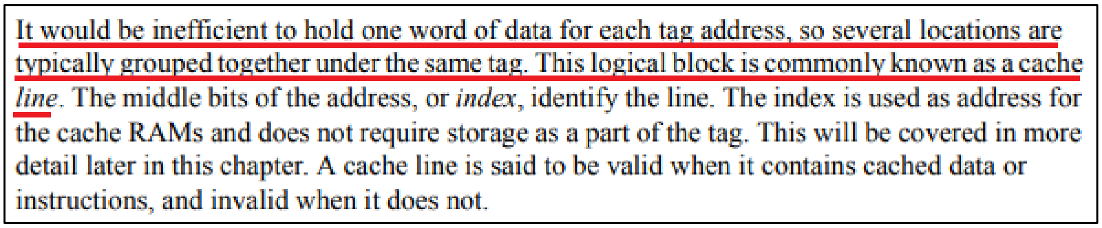


**Instruction fetch**

The instruction fetch unit fetches instructions from the L1 instruction cache and delivers up to three instructions per cycle to the instruction decode unit. It supports dynamic and static branch prediction. The instruction fetch unit includes:

- L1 instruction cache that is a 32KB 2-way set-associative cache with 64 bytes cache line and optional parity protection per 16-bits.
- 2-level dynamic predictor with BTB for fast target generation.

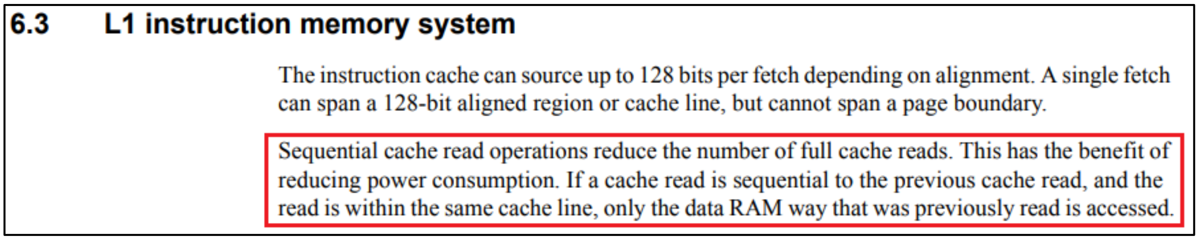
**Source:** https://static.docs.arm.com/ddi0438/i/DDI0438.pdf (Page 28)



It would be inefficient to hold one word of data for each tag address, so several locations are typically grouped together under the same tag. This logical block is commonly known as a cache *line*. The middle bits of the address, or *index*, identify the line. The index is used as address for the cache RAMs and does not require storage as a part of the tag. This will be covered in more detail later in this chapter. A cache line is said to be valid when it contains cached data or instructions, and invalid when it does not.

**Source:** https://documentation-service.arm.com/static/5e909fb6c8052b16087625aa?token=

(Page 112)

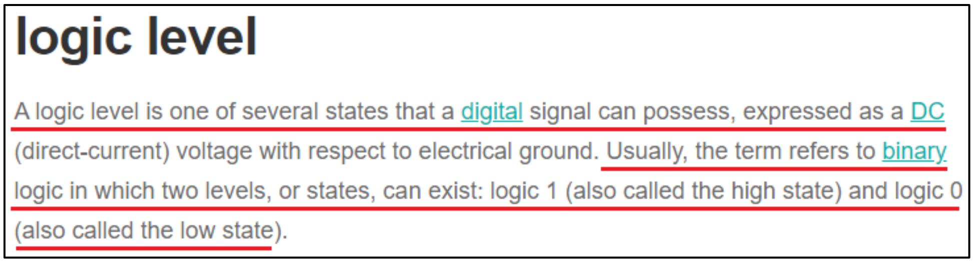


**6.3 L1 instruction memory system**

The instruction cache can source up to 128 bits per fetch depending on alignment. A single fetch can span a 128-bit aligned region or cache line, but cannot span a page boundary.

Sequential cache read operations reduce the number of full cache reads. This has the benefit of reducing power consumption. If a cache read is sequential to the previous cache read, and the read is within the same cache line, only the data RAM way that was previously read is accessed.

**Source:** https://static.docs.arm.com/ddi0438/i/DDI0438.pdf (Page 198)



logic level

A logic level is one of several states that a digital signal can possess, expressed as a DC (direct-current) voltage with respect to electrical ground. Usually, the term refers to binary logic in which two levels, or states, can exist: logic 1 (also called the high state) and logic 0 (also called the low state).

**Source:** https://whatis.techtarget.com/definition/logic-level

112. TI has had knowledge of the ’959 Patent at least as of the date when it was notified of the filing of this action.

113. Liberty Patents has been damaged as a result of the infringing conduct by TI alleged above. Thus, TI is liable to Liberty Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

114. Liberty Patents and/or its predecessors-in-interest have satisfied all statutory obligations required to collect pre-filing damages for the full period allowed by law for infringement of the ’959 Patent.

## ADDITIONAL ALLEGATIONS REGARDING INFRINGEMENT AND PERSONAL JURISDICTION

115. TI has also indirectly infringed the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent by inducing others to directly infringe the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent.

116. TI has induced the end users and/or TI’s customers to directly infringe (literally and/or under the doctrine of equivalents) the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent by using the accused products.

117. TI took active steps, directly and/or through contractual relationships with others, with the specific intent to cause them to use the accused products in a manner that infringes one

or more claims of the patents-in-suit, including, for example, claims 1 and 9 of the ’504 Patent, claims 1 and 9 of the ’156 Patent, and claims 1 and 9 of the ’496 Patent, and claims 1 and 9 of the ’959 Patent.

118. Such steps by TI included, among other things, advising or directing customers and end users to use the accused products in an infringing manner; advertising and promoting the use of the accused products in an infringing manner; and/or distributing instructions that guide users to use the accused products in an infringing manner.

119. TI performed these steps, which constitute induced infringement, with the knowledge of the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent and with the knowledge that the induced acts constitute infringement.

120. TI was and is aware that the normal and customary use of the accused products by TI’s customers would infringe the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent. TI’s inducement is ongoing.

121. TI has also induced its affiliates, or third-party manufacturers, shippers, distributors, retailers, or other persons acting on its or its affiliates’ behalf, to directly infringe (literally and/or under the doctrine of equivalents) the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent by importing, selling or offering to sell the accused products.

122. TI has a significant role in placing the accused products in the stream of commerce with the expectation and knowledge that they will be purchased by consumers in Texas and elsewhere in the United States.

123. TI purposefully directs or controls the making of accused products and their shipment to the United States, using established distribution channels, for sale in Texas and elsewhere within the United States.

124. TI purposefully directs or controls the sale of the accused products into established United States distribution channels, including sales to nationwide retailers. TI's established United States distribution channels include one or more United States based affiliates.

125. TI purposefully directs or controls the sale of the accused products online and in nationwide retailers, including for sale in Texas and elsewhere in the United States, and expects and intends that the accused products will be so sold.

126. TI purposefully places the accused products—whether by itself or through subsidiaries, affiliates, or third parties—into an international supply chain, knowing that the accused products will be sold in the United States, including Texas. Therefore, TI also facilitates the sale of the accused products in Texas.

127. TI took active steps, directly and/or through contractual relationships with others, with the specific intent to cause such persons to import, sell, or offer to sell the accused products in a manner that infringes one or more claims of the '504 Patent, the '156 Patent, the '496 Patent, and the '959 Patent including, for example, claims 1 and 9 of the '504 Patent, claims 1 and 9 of the '156 Patent, claims 1 and 9 of the '496 Patent, and claims 1 and 9 of the '959 Patent.

128. Such steps by TI included, among other things, making or selling the accused products outside of the United States for importation into or sale in the United States, or knowing that such importation or sale would occur; and directing, facilitating, or influencing its affiliates, or third-party manufacturers, shippers, distributors, retailers, or other persons acting on its or its affiliates' behalf, to import, sell, or offer to sell the accused products in an infringing manner.

129. TI performed these steps, which constitute induced infringement, with the knowledge of the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent, and with the knowledge that the induced acts would constitute infringement.

130. TI performed such steps in order to profit from the eventual sale of the accused products in the United States.

131. TI’s inducement is ongoing.

132. TI has also indirectly infringed by contributing to the infringement of the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent. TI has contributed to the direct infringement of the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent by the end user of the accused products.

133. The accused products have special features that are specially designed to be used in an infringing way and that have no substantial uses other than ones that infringe the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent, including, for example, claims 1 and 9 of the ’504 Patent, claims 1 and 9 of the ’156 Patent, claims 1 and 9 of the ’496 Patent, and claims 1 and 9 of the ’959 Patent.

134. The special features include, for example, components and/or features for applying body biasing voltages during a power mode of a processor or other integrated circuit in a manner that infringes the ’504 Patent, the ’156 Patent, and the ’496 Patent, and for executing computer instructions in an instruction cache in a manner that infringes the ’959 Patent.

135. These special features constitute a material part of the invention of one or more of the claims of the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent, and are not staple articles of commerce suitable for substantial non-infringing use.

136. TI’s contributory infringement is ongoing.

137. TI has had actual knowledge of the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent at least as of the date when it was notified of the filing of this action. Since at least that time, TI has known the scope of the claims of the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent; the products that practice the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent; and that Liberty Patents is the owner of the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent.

138. By the time of trial, TI will have known and intended (since receiving such notice) that its continued actions would infringe and actively induce and contribute to the infringement of one or more claims of the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent.

139. Furthermore, TI has a policy or practice of not reviewing the patents of others (including instructing its employees to not review the patents of others), and thus has been willfully blind of Liberty Patents’ patent rights. *See, e.g.*, M. Lemley, “Ignoring Patents,” 2008 Mich. St. L. Rev. 19 (2008).

140. TI’s actions are at least objectively reckless as to the risk of infringing valid patents, and this objective risk was either known or should have been known by TI. TI has knowledge of the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent.

141. TI’s customers have infringed the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent. TI has encouraged its customers’ infringement.

142. TI’s direct and indirect infringement of the ’504 Patent, the ’156 Patent, the ’496 Patent, and the ’959 Patent has been, and/or continues to be willful, intentional, deliberate, and/or in conscious disregard of Liberty Patents’ rights under the patents-in-suit.

143. Liberty Patents has been damaged as a result of TI's infringing conduct alleged above. Thus, TI is liable to Liberty Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

## **JURY DEMAND**

Liberty Patents hereby requests a trial by jury on all issues so triable by right.

## **PRAYER FOR RELIEF**

Liberty Patents requests that the Court find in its favor and against TI, and that the Court grant Liberty Patents the following relief:

a. Judgment that one or more claims of the '504 Patent, the '156 Patent, the '496 Patent, and the '959 Patent have been infringed, either literally and/or under the doctrine of equivalents, by TI and/or all others acting in concert therewith;

b. A permanent injunction enjoining TI and its officers, directors, agents, servants, affiliates, employees, divisions, branches, subsidiaries, parents, and all others acting in concert therewith from infringement of the '504 Patent, the '156 Patent, the '496 Patent, and the '959 Patent; or, in the alternative, an award of a reasonable ongoing royalty for future infringement of the '504 Patent, the '156 Patent, the '496 Patent, and the '959 Patent by such entities;

c. Judgment that TI account for and pay to Liberty Patents all damages to and costs incurred by Liberty Patents because of TI's infringing activities and other conduct complained of herein, including an award of all increased damages to which Liberty Patents is entitled under 35 U.S.C. § 284;

d. That Liberty Patents be granted pre-judgment and post-judgment interest on the damages caused by TI's infringing activities and other conduct complained of herein;

e. That this Court declare this an exceptional case and award Liberty Patents its reasonable attorney's fees and costs in accordance with 35 U.S.C. § 285; and

f. That Liberty Patents be granted such other and further relief as the Court may deem just and proper under the circumstances.

Dated: February 12, 2021

Respectfully submitted,

*/s/ Zachariah S. Harrington*
Matthew J. Antonelli
Texas Bar No. 24068432
matt@ahtlawfirm.com
Zachariah S. Harrington
Texas Bar No. 24057886
zac@ahtlawfirm.com
Larry D. Thompson, Jr.
Texas Bar No. 24051428
larry@ahtlawfirm.com
Christopher Ryan Pinckney
Texas Bar No. 24067819
ryan@ahtlawfirm.com
Rehan M. Safiullah
Texas Bar No. 24066017
rehan@ahtlawfirm.com

ANTONELLI, HARRINGTON
& THOMPSON LLP
4306 Yoakum Blvd., Ste. 450
Houston, TX 77006
(713) 581-3000

Stafford Davis
State Bar No. 24054605
sdavis@stafforddavisfirm.com
Catherine Bartles
Texas Bar No. 24104849
cbartles@stafforddavisfirm.com
THE STAFFORD DAVIS FIRM
815 South Broadway Avenue
Tyler, Texas 75701
(903) 593-7000
(903) 705-7369 fax

*Attorneys for Liberty Patents LLC*