Automotive Audio Bus (A²B) Transceiver

Silicon Anomaly List — AD2426/AD2427/AD2428

ABOUT AD2426/AD2427/AD2428 SILICON ANOMALIES

These anomalies represent the currently known differences between revisions of the A²B® AD2426/AD2427/AD2428 product(s) and the functionality specified in the AD2426/AD2427/AD2428 data sheet(s) and the Technical Reference manual.

SILICON REVISIONS

A silicon revision number with the form "x.x" is branded on all parts. The silicon revision can be electronically determined by reading the **A2B_VERSION** register.

Silicon Revision	A2B_VERSION
0.3	0x03
0.2	0x02
0.1	0x01
0.0	0x00

APPLICABILITY

Some anomalies apply only to specific A2B transceiver parts. In the table below an "x" indicates that the feature only applies to the part indicated, and the specific anomalies for that feature appear in the rightmost column. In addition, each anomaly applies to specific silicon revisions as listed in the SUMMARY OF SILICON ANOMALIES table.

Feature	AD2426	AD2427	AD2428	Anomalies
Bus Monitor Mode			x	18000028
Main node Capable			x	18000051

ANOMALY LIST REVISION HISTORY

The following revision history lists the anomaly list revisions and major changes for each anomaly list revision.

Date	Anomaly List Revision	Data Sheet Revision	Additions and Changes
3/19/2026	H	Rev.F	Modified Anomaly:18000052 Added Anomalies:18000075, 18000076
7/24/2023	G	Rev.E	Removed Obsolete AD2420 and AD2429 models Added Silicon Revision 0.3 Modified Anomalies:18000060,18000052
11/23/2022	F	Rev.D	Modified Anomaly:18000027, 18000052 Added Anomalies:18000068, 18000069, 18000073
11/16/2021	E	Rev.C	Added Silicon Revision 0.2

NOTE

1. For mixed node systems with different A²B part numbers, refer to all applicable A²B anomaly sheets.
2. Unless specifically mentioned, AD242x refers to AD2421/AD2422/AD2425/AD2426/AD2427/AD2428; and AD243x refers to AD2430/AD2431/AD2432/AD2433/AD2435/AD438.

NR004699H

Document Feedback

One Analog Way, Wilmington, MA 01887 U.S.A.

Technical Support www.analog.com

SUMMARY OF SILICON ANOMALIES

The following table provides a summary of AD2426/AD2427/AD2428 anomalies and the applicable silicon revision(s) for each anomaly.

No.	ID	Description	Rev 0.0	Rev 0.1	Rev 0.2	Rev 0.3
1	18000027	PRBS Test Mode May Erroneously Report Bit Errors	x	x	x	x
2	18000028	BMMCFG.BMMRXEN Bit Does Not Behave as Expected	x	x	x	x
3	18000030	I^2C Signals Clamp When I^2C Bus Voltage Is Higher Than IOVDD	x	.	.	.
4	18000044	I^2C t_{LOW} Spec Violation In Fast Mode I^2C Operations on Subordinate Node	x	x	.	.
5	18000050	Upslot Data Corruption During Partial Rediscovery	x	x	x	x
6	18000051	Transmit Data Channels On A^2B Main Node Shifts When First Subordinate Node Disconnects	x	x	x	x
7	18000052	Concurrent Use Of GPIO-Over-Distance And Remote Register/Peripheral Access May Result In Error	x	x	x	x
8	18000059	Bus Read Access To Subordinate Node register May Be Corrupted By Simultaneous Local I^2C Read	x	x	.	.
9	18000060	A^2B Subordinate Node May Drive Free-running I^2C Signals When Exposed To Invalid I^2C Protocol	x	x	x	x
10	18000068	Host Access To Sub-node Register Or Remote Peripheral May Fail When Downstream Node Discovery Process Completes	x	x	x	x
11	18000069	PDM Interface Performance Specifications Deviations In The Datasheet	x	x	x	x
12	18000073	Bus Write Access To Subordinate Node Register May Be Corrupted By Simultaneous Local I^2C Write	x	x	x	.
13	18000075	I2CERR And NACK May Not Be Reported Under Certain Scenarios For A Remote I^2C Peripheral Access	x	x	.	.
14	18000076	Bus Access To Subordinate Node Register May Fail When Simultaneous Local Access During Discovery	x	x	x	x

Key: x = anomaly exists in revision
. = Not applicable

DETAILED LIST OF SILICON ANOMALIES

The following list details all known silicon anomalies for the AD2426/AD2427/AD2428 including a description, workaround, and identification of applicable silicon revisions.

1. 18000027 - PRBS Test Mode May Erroneously Report Bit Errors:

DESCRIPTION:
The PRBS (pseudorandom binary sequence) test mode may erroneously report bit errors for following conditions:

1. When I2S/TDM transmit block is enabled.
2. Data slots with ECC/compression enabled.
3. When a subordinate node contributes downstream slots.

Conditions 1 and 2 can occur on both A2B main node and subordinate nodes. Condition 3 occurs only at the source subordinate node.

WORKAROUND:
Do not run PRBS mode with I²S/TDM transmit block enabled. In order to make sure I²S/TDM transmit block is disabled, the **A2B_I2SCFG.TX0EN** and **A2B_I2SCFG.TX1EN** bits must be cleared on each node before enabling PRBS mode.
Do not run PRBS mode with data slot formatting enabled. Make sure that **A2B_SLOTFMT.DNFMT** and **A2B_SLOTFMT.UPFMT** bits are cleared before enabling PRBS mode.
Also, make sure not to run PRBS mode with any subordinate node contributing downstream slots.

APPLIES TO REVISION(S):
0.0, 0.1, 0.2, 0.3

2. 18000028 - BMMCFG.BMMRXEN Bit Does Not Behave as Expected:

DESCRIPTION:
In bus monitor mode, the **A2B_BMMCFG.BMMRXEN** bit allows software to control enable of A-side LVDS receiver and when the bus monitor node locks to the bus. This bit is used to keep the A-side LVDS receiver static while bus monitor node is being attached, and it is also used to re-initiate the bus monitor lock sequence without physically detaching the bus monitor node.

The A-side LVDS receiver is expected to be disabled when **BMMRXEN** = 0. However, due to this anomaly, the **BMMRXEN** bit is not functional when PLL is not locked and the A-side LVDS receiver can only be disabled/re-enabled after PLL is locked i.e. when PLL is not locked, the **BMMRXEN** bit is ignored and the A-side LVDS receiver is always enabled. Therefore, if bus monitor node is enabled (**BMMEN** = 1) with **BMMREXN** = 0, then it may cause the transceiver to lock and unlock from the bus.

WORKAROUND:
The **BMMEN** and **BMMRXEN** bits must be set in a single write when entering bus monitor mode operation.
Software control of bus monitor lock time must be implemented external to the transceiver. For example, use an external switch to control the LVDS traffic going to the A-side of a transceiver in bus monitor mode. Refer app note **AN-1391** for details.

APPLIES TO REVISION(S):
0.0, 0.1, 0.2, 0.3

3. 18000030 - I²C Signals Clamp When I²C Bus Voltage Is Higher Than IOVDD:

DESCRIPTION:

If the I²C Bus Voltage (I2C_VBUS) is greater than IOVDD, the I²C signals may clamp to a voltage level that is lower than I2C_VBUS. This is due to activation of internal ESD diodes on the SCL and SDA pins of transceiver. The resulting clamping may result in violation of **VIH** spec of other devices on the I²C bus. The following table summarizes the effect of this anomaly for different cases:

I2C_VBUS	IOVDD	Remarks
1.8V	1.8V	No Issue
3.3V	3.3V	No Issue
3.3V	1.8V	I²C signals clamp to an intermediary voltage level. Other devices connected to the I²C bus must have maximum **VIH** specs that are below the clamped voltage. Low reliability risk. A worst-case circuit analysis must be performed at the system level to verify operation at these voltage levels
1.8V	3.3V	Not supported
5V	1.8V or 3.3V	Not supported

WORKAROUND:

Operate both I2C_VBUS and IOVDD at same 1.8V or 3.3V domain only.

1. Preferably connect I²C pull-up resistors to IOVDD i.e. IOVDD -> I2C_VBUS.
2. If I2C_VBUS and IOVDD are supplied from different power domains, then pull-up resistors on the SCL/SDA pins ensures that the internal ESD diodes are not damaged when I2C_VBUS is powered while IOVDD is not:
   a. When I2C_VBUS = 1.8V, use at least 1KOhm I²C pull-up resistors.
   b. When I2C_VBUS = 3.3V, use at least 2KOhm I²C pull-up resistors.

APPLIES TO REVISION(S):

0.0

4. 18000044 - I²C t_{LOW} Spec Violation In Fast Mode I²C Operations on Subordinate Node:

DESCRIPTION:

This anomaly is applicable only for A²B subordinate node if acting as I²C controller in Fast mode of I²C operations (400KHz). In this mode, the Low period of SCL clock (t_{LOW}) spec doesn't meet the minimum duration as per I²C 2.1 specification. The transceiver can assert a minimum of 1.21us instead of 1.3us as per I²C 2.1 specification.

If there is sufficient margin for the data setup(t_{DS}) and data hold time(t_{DH}), then t_{LOW} violation will not lead to functional failures. Also, if the target peripheral on I²C bus supports Fast Mode Plus operating mode adhering to I²C 2.1 specification, then there will not be functional failures due to this anomaly, as the Fast Mode Plus devices have margin for data setup and data hold time with I²C clock low period of 1.21us.

WORKAROUND:

Use one of the following workarounds:

1. Operate subordinate node I²C in Standard Mode (100KHz)
2. Choose I²C target peripherals that supports Fast Mode Plus

APPLIES TO REVISION(S):

0.0, 0.1

5. 18000050 - Upslot Data Corruption During Partial Rediscovery:

DESCRIPTION:

When a subordinate node disconnects from the A²B bus during run-time, the host processor can attempt partial rediscovery of disconnected nodes without affecting data exchange between active upstream nodes. In the case where host processor initiates partial rediscovery of disconnected node and it results in a successful discovery, there would be data corruption in upslots contributed by other upstream nodes in the A²B network. This issue only occurs when disconnected nodes were contributing upslots to active nodes. The data corruption starts after successful rediscovery of the node and continues until the node is initialized.

For example, consider an A²B network *Main_node - Sub_node0 - Sub_node1*, with each subordinate node sending some upslots to *Main_node*. When *Sub_node1* disconnects from the network during run-time, the upslots of *Sub_node1* are replaced with previous known good samples and the communication with *Sub_node0* continues as-is without any data corruption. Now if the host processor attempts a partial rediscovery of *Sub_node1*, then upon successful rediscovery, there will be data corruption in the upslots contributed by *Sub_node0*, until the *Sub_node1* is completely initialized with its upslots settings.

WORKAROUND:

Use one of the workarounds listed below:

1. Once the subordinate node rediscovery is successful, the host processor must configure the **A2B_LUPSLOTS, A2B_UPSLOTS, A2B_SLOTFMT, A2B_DATCTL, A2B_CONTROL.NEWSTRUCT** registers of discovered node before configuring the remaining registers. This reduces the duration of upslots data corruption.
2. Before initiating the partial rediscovery, program the **A2B_UPSLOT** register of the last active upstream node to zero and remap the TDM channels of receiver nodes accordingly. In the example considered, the workaround involves clearing the **A2B_UPSLOT** register of *Sub_node0* and remapping the TDM channels of *Main_node*.
   Once there is a successful rediscovery and initialization of the dropped nodes, reconfigure the registers to their original value and revert back the TDM map of receiver node.

APPLIES TO REVISION(S):

0.0, 0.1, 0.2, 0.3

6. 18000051 - Transmit Data Channels On A²B Main Node Shifts When First Subordinate Node Disconnects:

DESCRIPTION:

If the first subordinate node disconnects from the A²B bus during run-time, then the TDM channels on data transmit (DTX) pins of A²B main node shift right by one channel. If both the DTX pins are enabled on A²B main node, then the channel shift happens on both the lines. This issue is applicable only when I²S transmit data offset is zero (**A2B_I2STXOFFSET.TXOFFSET** = 0) in A²B main node.

WORKAROUND:

Use one of the workaround listed below:

1. Use a non-zero I²S/TDM transmit data offset (**A2B_I2STXOFFSET.TXOFFSET** > 0)
2. In the host processor software, ignore the TDM channels from A²B main node when the first subordinate node is disconnected.

APPLIES TO REVISION(S):

0.0, 0.1, 0.2, 0.3

7. 18000052 - Concurrent Use Of GPIO-Over-Distance And Remote Register/Peripheral Access May Result In Error:

DESCRIPTION:
When host processor performs a remote I2C peripheral access (read/write) or subordinate register access(read/write) and a GPIO-over-Distance(GPIOD) communication happens concurrently, then there will be an arbitration conflict between the subordinate node register access/remote peripheral access and GPIOD signaling, as both would compete for the SCF and SRF fields of A2B superframes. Due to this arbitration conflict, it may lead to either one of the following erroneous behaviors:

1. GPIOD request may be dropped.
2. Remote I^2C access timeout due to subordinate node failing to acknowledge the transaction.
3. Remote I^2C peripheral access or subordinate register access corruption.

WORKAROUND:
Use one of the workarounds listed below:

1. Do not use GPIOD feature and remote register/ I^2C peripheral access concurrently. Disable GPIOD before initiating a remote register/ I^2C peripheral access and then re-enable it upon access completion.
2. If the GPIOD is configured from subnode to main node
   a. Use GPIO interrupt instead of GPIOD
   b. Configure the subnode GPIO pin as input pin, and poll A2B_GPIOIN register on sub node for GPIO status.

APPLIES TO REVISION(S):
0.0, 0.1, 0.2, 0.3

8. 18000059 - Bus Read Access To Subordinate Node register May Be Corrupted By Simultaneous Local I^2C Read:

DESCRIPTION:
The registers of A^2B subordinate node can be accessed by the host processor via A^2B bus and by the local processor via I^2C port. If the subordinate node receives read requests via A^2B bus and local I^2C port in the same system clock cycle (SYSBCLK), then the register value of local access is returned to both requests. The local I^2C requests have higher priority than Bus requests.
For example, if a subordinate node receives a bus read request to Mailbox Data register (**A2B_MBOX0Bn**) and a local read request to **A2B_NODE** register in the same SYSBCLK cycle, then it erroneously returns **A2B_NODE** register value to the bus request. In this case, the subordinate node returns an incorrect register value to bus request, but there is no corruption to actual register value.

The bus read request can also be internally initiated by the A^2B main node as part of bus protocol. For example, when a subordinate node generates any interrupt, the A^2B main node internally sends a bus read request to the node for INTTYPE register. If the subordinate node receives a read request from the local I^2C port at the same time, then the register value of local read request is erroneously returned as INTTYPE to the A^2B main node. In this case, host processor would get an incorrect interrupt (INTTYPE) from the A^2B network.

There is no error generated or reported to the host processor when this bus read access is corrupted. This issue occurs at all I^2C frequencies and bus superframe rates. The base probability of this issue occurring is 1/1024 per superframe, but the actual probability at the system level depends on bus and local access rates.

WORKAROUND:

1. Avoid parallel read accesses to the subordinate node registers from host processor and local processor using a handshake mechanism. For example:
   a. Register based handshake (using Mailbox)
   b. GPIO based handshake

APPLIES TO REVISION(S):
0.0, 0.1

9. 18000060 - A²B Subordinate Node May Drive Free-running I²C Signals When Exposed To Invalid I²C Protocol:

DESCRIPTION:

The A²B subordinate node (when acting as an I²C controller accessing a remote peripheral) may toggle the I²C lines indefinitely when exposed to an invalid I²C protocol. This behavior has been observed in the following two scenarios:

1. This scenario is applicable to silicon revisions 0.0 and 0.1 only.If the A²B subordinate node loses access arbitration due to an errant I²C target device (For example, I²C device provides ACK one bit early):
   According to the I²C-bus specification ver2.1, when an I²C controller loses arbitration during an access, it aborts the access by immediately turning off the SDA output driver, but it may drive the clock until the end of byte in which it lost the arbitration. However, when A²B subordinate node is acting as an I²C controller and loses arbitration, it continues to drive the clock on the SCL line for the whole access (until it sees a STOP condition on the bus).
   In true multi-I²C controller systems, if an A²B subordinate node loses access arbitration, then it drives the clock for the whole ongoing access of winning I²C controller and stops driving the clock in response to the STOP bit at the end of access. The access of the winning I²C controller continues through the collision without any problem, as there would be clock synchronization between both the I²C controllers. The I²C bus becomes free after the completion of access.
   However, if the I²C access of an A²B subordinate node (acting as an I²C controller) is corrupted by an errant I²C target device or due to glitch on SDA line, then the subordinate node aborts its access but would drive the clock indefinitely, as there would be no STOP bit for this access.
2. If the SCL line of an A²B subordinate node is spuriously pulled low, either by an external device or due to some event before the start of a remote I²C transaction from the subordinate node:
   When a subordinate node receives a remote peripheral access request over the A²B bus, it checks the status of the local I²C bus. If the I²C bus is free, the A²B subordinate node starts the access; and if the I²C bus is busy, the subordinate node aborts the access. If the I²C bus is detected as busy (for example, due to glitch on the SCL line), but there is no valid I²C access on-going on the bus, then the A²B subordinate node will start driving the I²C signals continuously when the SCL line is released.

   Invalid activity on the I²C bus causes this scenario. There is no issue, when the I²C bus is detected as busy due to proper I²C access ongoing on the bus.

There is no error generated or reported when this issue occurs, however one can observe that the remote peripheral accesses will always fail while the subordinate node register accesses will pass.

WORKAROUND:

Use one of the steps listed below to exit the error condition:

1. Generate a STOP bit on the I²C bus of the subordinate node.
2. Power cycle the A²B subordinate node.
3. For Scenario 2, apply Soft Reset to subordinate node from host processor. This is applicable only to silicon revisions 0.2 and 0.3

APPLIES TO REVISION(S):

0.0, 0.1, 0.2, 0.3

10. 18000068 - Host Access To Sub-node Register Or Remote Peripheral May Fail When Downstream Node Discovery Process Completes:

DESCRIPTION:

During a node discovery, if host processor performs an access to upstream sub-node or it's peripheral, then the access may fail if the node discovery process completes in the same superframe. This happens when Main node receives the BUS access on I^2C port and waiting for next superframe SCF to send the access to sub-node over A^2B bus, but meanwhile receives the Discovery Done response from newly discovered sub-node via SRF in the same superframe. In this case, the acknowledgement from the discovered node is seen by the A^2B Main node as an ACK to the Host access and decides that access is already complete before it has been sent to sub-node over the A^2B bus.
If the access is a write access, it is simply aborted by Main node and ACK is provided to I^2C access of Host. If the access is a read access, it is also aborted by Main node and provides the read value as Response Cycle of newly discovered node. There is no I^2C Error interrupt (**`A2B_INTTYPE = 25`**) raised or status bit (**`A2B_INTPND2.I2CERR`**) not changed to indicate this I^2C access fail.
This issue can happen in optimized discovery flow and advanced discovery flow, wherein a node is initialized during discovery of next downstream node. It can also happen during partial rediscovery of dropped node.

WORKAROUND:

For optimized or advanced discovery flow, use below steps to avoid getting Sub-node Discovery Done interrupt while initializing upstream sub-node. For example, to perform discovery of node-n:

a. Turn-ON the switch in upstream node n-1 :
   M NODEADDR = n-1
   Sn-1 SWCTL = 0x01
   With this step, node n-1 starts diving the SCFs on port-B and node-n starts locking its PLL.
b. Initialize earlier discovered nodes .....(WR Sn-1/Sn-2/Sn-3.....)
c. Start the discovery of node n :
   M NODEADDR = n-1
   M DISCVRY = Sn_RESPCYC
   If the PLL of node-n is locked during step-b, then discovery process completes and DISC_DONE interrupt raised immediately with this step. So, discovery time is not impacted.

APPLIES TO REVISION(S):

0.0, 0.1, 0.2, 0.3

11. 18000069 - PDM Interface Performance Specifications Deviations In The Datasheet:

DESCRIPTION:

Following PDM specs are incorrect in the datasheet :

1. Fig-18 refers to "Total Harmonic Distortion (THD) vs. Normalized Frequency" instead of "Total Harmonic Distortion + Noise (THD+N) vs. Normalized Frequency".
2. The SNR spec (A-weighted filter) of PDM block in the Table-11 of datasheet is incorrectly specified as 120dB instead of 108dB.

WORKAROUND:

None

APPLIES TO REVISION(S):

0.0, 0.1, 0.2, 0.3

12. 18000073 - Bus Write Access To Subordinate Node Register May Be Corrupted By Simultaneous Local I²C Write:

DESCRIPTION:

The registers of A²B subordinate node can be accessed by the host processor via A²B bus and by the local processor via I²C port. If the subordinate node receives register write requests via both A²B bus and local I²C port in the same system clock cycle (SYSBCLK), then the local access write value is written to both registers.

For example, if a sub node receives bus access to write REG_B with a value of 'X' and also receives a local access to write REG_L with a value of 'Y' in the same SYSBCLK cycle, then both registers are written with the value of 'Y' i.e. REG_B is written with value 'Y' instead of 'X'. The local I²C requests have higher priority than Bus requests.

A²B main node also initiates internal bus write accesses to sub-node interrupt registers during its interrupt handling. For example, when a sub-node generates any interrupt, the A²B main node internally reads INTTYPE register first and then clears the interrupt latch in sub-node by writing to INTPNDn or MBOXn_STAT register depending on the interrupt type. If this internal write is corrupted by simultaneous local write access in same SYSBCLK cycle, then the interrupt may not get cleared and will be reported again. This can lead to duplicate interrupts, or it could clear another pending interrupt in the same INTPNDn register before servicing it.

There is no error generated or reported to the host processor when this bus write access is corrupted. This issue can occur at any I²C frequencies (100K/400KHz) and bus superframe rates (48KHz/44.1KHz). The base probability of the occurrence is 1/1024 per superframe. However, actual probability of occurrence will be low as both write accesses have to be received by the sub-node in the same SYSBCLK cycle.

WORKAROUND:

Use one of the workarounds listed below:

1. Avoid parallel write accesses to the subordinate node registers from host processor and local processor using GPIO or register (e.g. Mailbox) based handshake mechanism.
2. Initialize the sub-node completely from Host processor. However, if it is required to partially initialize the node locally, perform a handshake mechanism between processors at system software level before configuring the node registers.
3. After writing to a sub-node configuration register, the Host processor should readback the register value to confirm if the write was successful.
4. When using Mailbox, keep CRC checksum on data payload to detect the data corruption. Also, when Host processor gets Mailbox interrupt, confirm the MBOX FIFO status before proceeding. For example, when Host processor gets MBOX_Full interrupt, check the bit#0 (MBnFULL) of MBOX_STAT register to confirm MBOX Data FIFO status before reading it.

APPLIES TO REVISION(S):

0.0, 0.1, 0.2

13. 18000075 - I2CERR And NACK May Not Be Reported Under Certain Scenarios For A Remote I²C Peripheral Access:

DESCRIPTION:

In a system with AD2428 Rev 0.0 or 0.1 as main node and AD243x or AD2428/AD2427/AD2426 Rev 0.2 or 0.3 as subordinate node, NACK provided by remote I²C peripheral to a subordinate node may be incorrectly reported as ACK at the main node during a remote peripheral access. I2CERR may also not be reported during this scenario.

WORKAROUND:

Use AD2428 Rev 0.2 or 0.3 as the main node.

APPLIES TO REVISION(S):

0.0, 0.1

14. 18000076 - Bus Access To Subordinate Node Register May Fail When Simultaneous Local Access During Discovery:

DESCRIPTION:

During discovery if there is a local access(read/write) at sub node around its PLL lock time, any bus access including internal accesses may fail till the ongoing local access is completed. Errors may not be reported to the Main-node.

For example, if a local access starts at time **t0** and ends at **t2**, with the PLL locking at **t1** (where **t0** < **t1** < **t2**), any internal or bus access to the sub-node during **t1** to **t2** may exhibit the following issues:

1. Internal writes to response cycle register which gets updated during the PLL lock time might be ignored causing intermittent discovery failures.
2. Bus writes might be ignored.
3. Bus reads might return incorrect data.

After time **t2**, bus accesses function normally, even with simultaneous local accesses.

WORKAROUND:

1. Avoid Bus accesses till local access is completed:
   a. Delay Bus Access After DSCDONE Interrupt: After receiving the DSCDONE interrupt at the main node, introduce a delay equivalent to the local access duration (**t2**- **t0**) before initiating any bus access to the sub-node.
   b. Initialize Response Cycle Register as part of the sub-node initialization during the discovery process.
2. Avoid Local accesses till Discovery is completed, use GPIO output as discovery indication at sub node to initiate local access.

APPLIES TO REVISION(S):

0.0, 0.1, 0.2, 0.3