



ADV202 Errata for ADV202-ES3 samples

This document presents all known bugs/anomalies, work arounds and product changes for the ADV202 ES3 silicon.

ADV202 parts to which this document refers to are branded as follows:

ADV202xxx

xxxxxxx

xxxx 0.2

1) Line noise in decode slave mode.

Issues: This applies only to applications that use the ADV202 in decode slave mode. The error is visible on the slave device due to random lines that are delayed by one sample.

Workaround: Use the same clock source for VCLK and MCLK.

Fix: This issue has been fixed on present ES5 samples.

2) JDATA mode

Issues: When used in JDATA mode and an RD/ signal are shared with other devices in a system. When in JDATA mode and HOLD [DACK0/] is driven low, the ADV202 will incorrectly drive the HDATA [15:0] pins when RD/ is asserted. Accesses to the ADV202 itself function properly, but accesses to other devices which share the RD/ signal with the ADV202 will cause contention on the HDATA[15:0] pins.

Workaround: It is required to gate pins RD/ and CS/, so that the RD/ connected to the ADV202 only goes low when CS/ is low as well. This will ensure that RD/ only goes low when the ADV202 is being accessed and no bus contention will occur.

Fix: This issue has been fixed on ES5 samples.

Issues: The initial assertion of VALID by the ADV202 required that HOLD is de-asserted first, regardless of the state of the CODE FIFO.

Workaround: It is required to de-assert HOLD to allow VALID to be asserted.

Fix: Implemented on ES5.

3) Fly-By DMA mode - single DMA transfer

Issues: When in Fly-By Single Transfer mode the ADV202 will assert DREQ0/, DREQ1/ one additional time, although the FIFO is either full or empty. When reading the FIFO, this will cause the host to read invalid data. When writing to the FIFO, no data loss occurs, as the 2-deep write buffer will hold the data. When a FIFO overflow/underflow condition occurs, bits PFERR, DFERR, AFERR, NFERR in the EIRQFLG register will be set. This indicates that the last access to a FIFO may not have been successful.

Workaround: It is recommended to use DREQ/DACK DMA mode in single or burst transfer mode, DMA Fly-By mode in burst mode or Dedicated Chip Select mode.

Fix: This issue has been fixed on ES5 samples.

4) DACKx/

Issues: DACKx/ will drive the HDATA bus when not tied high, even if the DMA channels are disabled.

Workaround: It is required to tie DACKx/ high at all times if the DMA interface is not used, even if the DMA channels are disabled.

Fix: This issue has been fixed on ES5 samples.

5) DACK0/ and DACK1/ polarity

Issues: If DACK0/ or DACK1/ polarity is set to active high [EDMOD0, DA0POL and EDMOD1, DA1POL], DREQ/DACK DMA mode and Fly-By DMA mode in single or burst configuration will not function properly. HDATA will not be driven with valid data. JDATA mode is not affected, VALID and HOLD polarities function normally.

Workaround: If used in DREQ/DACK DMA mode and Fly-By mode in single or burst configuration, the polarity of DACK0/ and DACK1/ should be set to active low. Active low polarity is enabled by default. DACK0/ and DACK1/ polarity are set in indirect registers:

at address: 0xFFFF1408,
register: EDMOD0,
control bit: DA0POL and

at address: 0xFFFF140C,
register: EDMOD1,
control bit: DA1POL.

Fix: This issue has been fixed on ES5 samples.

6) Raw mode

Issues: When used in Video Raw mode, in either encode or decode mode the ADV202 cannot robustly handle conditions where the interface is starved for data by the user. In encode mode, the error may cause previously transferred data, no longer valid but still in the pixel interface sync FIFO to be re-qualified as valid. In decode mode, the error might cause valid data awaiting transfer to be overwritten.

Workaround: It is recommended to use MP-656 mode.
[Indirect register 0xFFFF044C, VMODE register, control bit: MP_656].

Fix: This issue has been fixed on ES5 samples.