TVP5158 - How to Load RAM Code

Digital Video Department

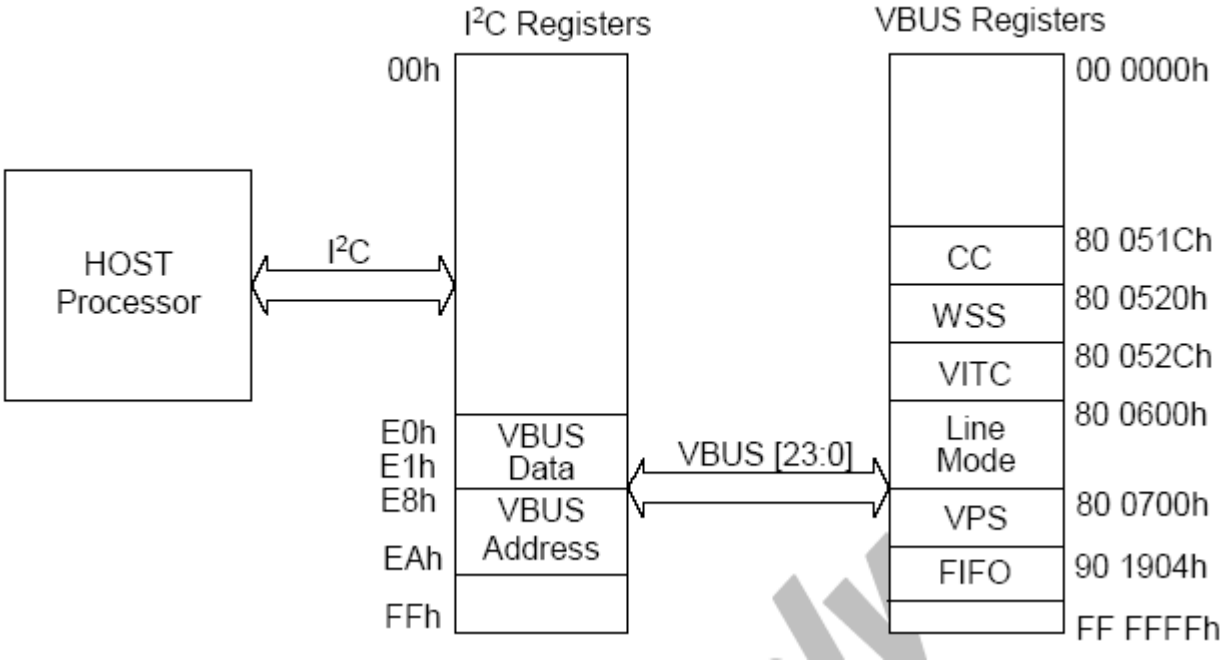
IMPORTANT NOTICE

1 Overview

This application note explains how to load RAM code into the TVP5158 video decoder. The TVP5158 video decoder by default executes firmware from internal ROM on power up. Special functions or optimizations are available by utilizing the internal RAM of the TVP5158. The following describes the processed required to access and load code into the TVP5158 RAM.

2 Understanding the VBUS

It is important to understand that loading RAM code is not a feature provided by the standard I2C register map. This procedure requires I2C writes to the physical hardware of the TVP5158 CPU. These internal registers of the TVP5158 video decoder are known as VBUS registers. The figures below show a typical VBUS register access.



VBUS Write

Single Byte

S	B8	ACK	E8	ACK	VA0	ACK	VA1	ACK	VA2	ACK	P

S	B8	ACK	E0	ACK	send data	ACK	P

Multiple Bytes

S	B8	ACK	E8	ACK	VA0	ACK	VA1	ACK	VA2	ACK	P

S	B8	ACK	E1	ACK	send data	ACK	···	send data	ACK	P

VBUS Read

Single Byte

S	B8	ACK	E8	ACK	VA0	ACK	VA1	ACK	VA2	ACK	P

S	B8	ACK	E0	ACK	S	B9	ACK	read data	NAK	P

Multiple Bytes

S	B8	ACK	E8	ACK	VA0	ACK	VA1	ACK	VA2	ACK	P

S	B8	ACK	E1	ACK	S	B9	ACK	read data	MACK	·	read data	NAK	P

The examples above use default the I2C address, 0xB8. The following acronyms are detailed below.

ACK - Acknowledge generated by the slave

MACK - Acknowledge generated by the master

NAK - No Acknowledge generated by the master

3 The Process

There are 6 steps required in order to properly load RAM code into the TVP5158.

Place the CPU into Reset

By writing a 1 to bit 0 of the first byte in the 0xB00060 VBUS address, the internal processor is placed into a reset state. This is necessary in order to load RAM code. To do this the VBUS address must first be set. Set VBUS address to 0xB00060 by making the following I2C writes.

0xE8, 0x60

0xE9, 0x00

0xEA, 0xB0

Where:

0xE8, 0xE9, 0xEA indicates the bytes of the address being setup (byte1, etc)

0x60, 0x00, 0xB0 indicates the bytes of the physical VBUS address

Once these writes have been performed, the current VBUS address is set to 0xB00060. Use the non-incrementing data register, 0xE0, to set the Reset bit by setting bit 0 to 1.

0xE0, 0x01

1. Set the I2C to write to 4 decoders

   0xFE, 0x0F

2. Set the VBUS to the Beginning of Program RAM

   Now that the internal processor is in a reset state, the following I2C writes will set the VBUS to the beginning of Program RAM. This is the location in which the RAM code will be stored during the loading process. Set the VBUS address to the beginning of Program RAM, 0x400000.

   0xE8, 0x00

   0xE9, 0x00

   0xEA, 0x40

3. Load the RAM Code

   With the VBUS now set to the beginning of Program RAM, start loading the provided RAM code *.bin file using the following writes. Since the firmware code data is loaded at once, the incrementing VBUS data register, 0xE1 must be used, where:

   0xE1, (RAM Code Data)

Using the above technique, all of the bytes of the firmware should be written using a single I2C transaction. See below for details.

ST B8 E1 D0 D1 ... DN-1 SP

Where:

- ST = I2C start condition
- B8 = TVP5158 device I2C address for writes (could also be BA depending on the GLCO/I2CA pin at the end of RESET)
- E1 = I2C sub-address of the incrementing VBUS data register
- D0 D1 ... DN-1D = Data from the binary firmware file. N is the number of bytes in the firmware file.
- SP = I2C stop condition

4. Set the RAM Loaded Bit

In order for the default ROM code to understand RAM load has been used, the RAM Loaded bit must be used. This is used by the internal CPU to execute out of RAM instead of ROM. To set the RAM Loaded Bit set VBUS address to 0xB00060.

0xE8, 0x60

0xE9, 0x00

0xEA, 0xB0

Write 0x03 to the non-incrementing VBUS data register, 0xE0 sets the RAM Loaded Bit and keeps the CPU RESET bit set.

0xE0, 0x03

5. Release the CPU Reset

To restart the CPU and release it from its reset state, a write of 0x02 to the same VBUS address as above (0xB00060) is necessary.

0xE8, 0x60

0xE9, 0x00

0xEA, 0xB0

0xE0, 0x02

.