Engineer To Engineer Note EE-203

Technical Notes on using Analog Devices' DSP components and development tools
Contact our technical support by phone: (800) ANALOG-D or e-mail: dsp.support@analog.com
Or visit our on-line resources http://www.analog.com/dsp and http://www.analog.com/dsp/EZAnswers

Interfacing the ADSP-BF535/ADSP-BF533 Blackfin® Processor to NTSC/PAL video decoder over the asynchronous port

Contributed by Thorsten Lorenzen

September 1, 2003

Introduction

The purpose of this note is to describe how to hook up video devices such as the ADV7183 NTSC/PAL Video Decoder to the external bus of the ADSP-BF535 Blackfin® Processor. Because of its architecture and video processing capabilities, Blackfin will interface with video devices. The ADSP-BF535 as the first part of the Blackfin Processor family is not equipped with a standard interface that glueless interacts with video devices, but as this EE-Note shows, the Asynchronous Interface can be used to receive video data in ITU-656 format.

Future Blackfin derivatives will be equipped with interfaces designed to support video devices (PPI interface).

ADV7183 NTSC/PAL Video Decoder

The ADV7183 is an integrated video decoder that automatically detects and converts a standard analog base band television signal compatible with world wide standards NTSC or PAL into 4:2:2 or 4:1:1 component video data compatible with 16-bit/8-bit CCIR601/CCIR656 or 10/20-bit extended standards.

The advanced and highly flexible digital output interface enables performance video decoding and conversion in both frame buffer based and line locked based systems “LLC Mode”. This makes the device ideally suited for a broad rage of applications with diverse analog video characteristics including tape based sources, broadcast sources, security/surveillance cameras and professional systems.

Basics of analog video composite signals

To fully understand the decoders digital output it is helpful to review the background of the NTSC standard and how it came along. If you are familiar with it already please go to the next chapter.

The first color television system was developed in the United States and began broadcasting in 1954. For economic reasons, a requirement was made that monochrome receivers must be able to display the black and white portion of a color broadcast and that color receivers must be able to display monochrome broadcast. For broadcasting purposes all required signals must be fitted into a single line. However, the CVBS signal looks still as years ago for compatibility purposes.

The analog composite video signal was designed a way it is shown as follows.

Imagine, a CRT (Cathode Ray Tube) of a TV such as shown in Figure 1 has been designed to display a video image based on N lines in the visual area “active video” of the monitors surface. In general, the ray of the CRT starts at the top left side and goes to the right. After reaching the right end it turns off and goes back to the left, a line below. Reaching the left side it turns on again and draws the next line on the screen. By repeating this the monitor will always be drawn with the help of the CRTs ray.

What kind of signals are required to fully drive the CRT?

In Figure 1 a NTSC Video Composite Signal can be seen. It drives the CRT in a sense that it runs as discussed above.

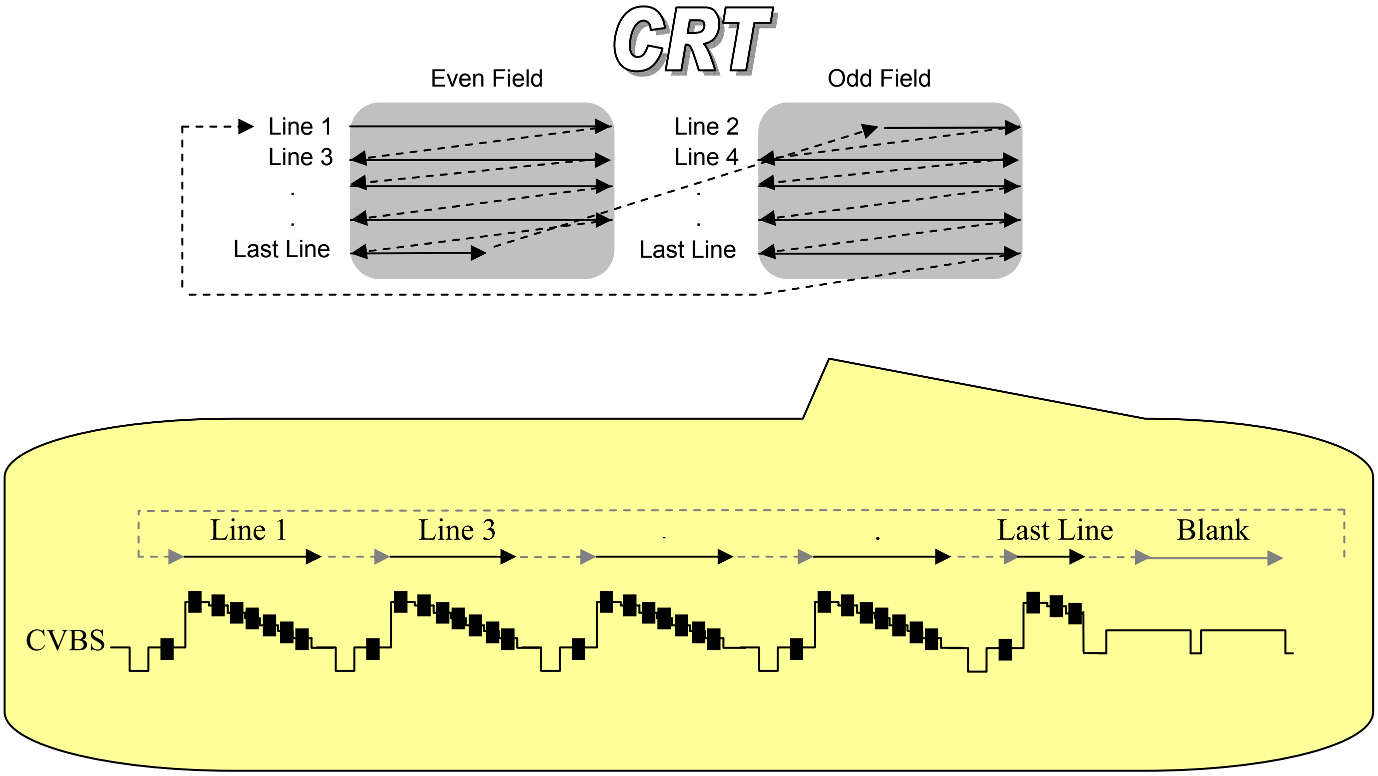


Figure 1: Multiple lines of an analog video composite signal (CVBS)

In general the video composite signal has two major tasks.

First, the CRT (Figure 1) runs from the left to the right, from the top to the bottom controlled by its own oscillators and sidetrack units. But it must be controlled (triggered) by signals that comes along (is aligned to) with the signal that holds the image data. Otherwise the image can never be placed on the screen in a correct order. Such signals are called:

Horizontal Sync (Line Blanking Signal), to force the CRT to turn off the ray and start with the next line. The Line Blanking area in Figure 2 shows such a signal.

Vertical Sync, to force the CRT to turn off the ray at the end of a frame/field and start with the next frame/field. The right end of the CVBS signal on Figure 1 "Blank" shows such a signal.

Second, the image data is split into parts of luminance and chrominance information's "(Y)

(C)". "Figure 2" shows the two information's combined with the syncs in a single line.

Luminance: As it can be seen in the area of "Active Video Line" the steps show different levels of luminance displaying a bar of different values in brightness.

Chrominance: The gray squares in Figure 2 represents 9 cycles of a 3.58MHz frequency as shown underneath the signal. The cycles are carrying the color information. The first gray square in the area of Color Burst drives an extra oscillator build into the video receiver and holds the cycles as a reference. The following squares "cycles" will be compared with the reference (Color Burst). The phase shift between these determines the hue. Color saturation is implemented by the cycles amplitude.

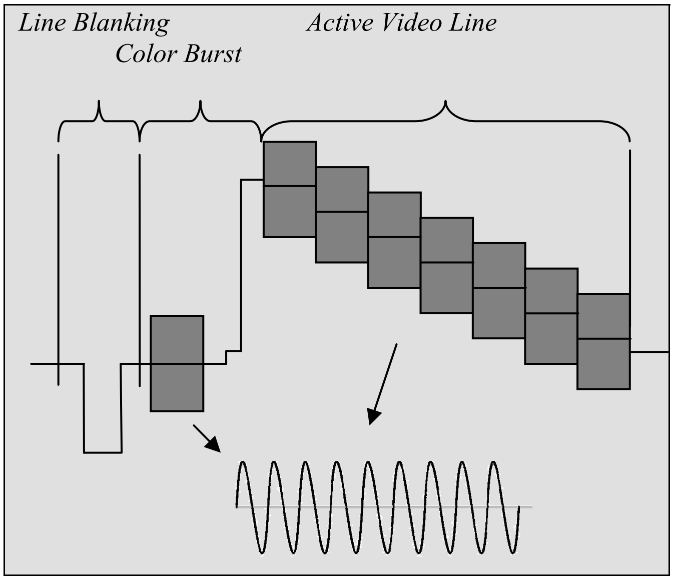


Figure 2: One line of a Video Composite Video Signal

As many as lines the video resolution includes as many line signals must be implemented in the video composite signal.

Decoding the analog composite video signal by the ADV7183 in LLC mode

It has been discussed how the analog composite video signal looks like. This section is dedicated to understand the signal path of the decoder when in CVBS mode. Just one of the two ADCs are used for conversion

After the analog signal has passed the ADC it goes into a luminance path to separate the luminance information. Also, it goes into a chrominance path to separate the chrominance information.

In the *luminance* path the 10-bit data from the ADC is applied to an antialiasing low pass filter that is designed to band-limit the input video signal such that antialiasing does not occur. In CVBS mode a notch filter must be used to remove the unwanted chrominance data that lies around the subcarrier. The next step is a peaking filter. This filter offers a sharpness function on the luminance path. The data is then passed through a resampler to correct for linelength variations in the input video. The resampler is designed to always output 720 pixels per line.

In the *chrominance* path the 10-bit data from the ADC is first demodulated as it is achieved by multiplying the locally generated quadrature subcarrier, where the sign of the cos subcarrier is inverted from line to line according to the PAL switch. The low pass filter is applied to remove components at twice the subcarrier frequency. The chrominance data is then passed through an band-pass filter to remove unwanted luminance data. After a shaping filter is applied the data is passed through a resampler to correct for line length variations in the input video. It always outputs 720 pixels per line.

The output formatter block does take the luminance and chrominance data and put it in the order as it is shown in Figure 3.

As is can be seen the actual video data has been embedded in the data block (Active Video). Active video data is separated by blanking and synchronization. Due to the ITU-656 recommendation it always begins and ends with the preamble (“FF 00 00 DA” 10 80 10 80 … 10 80 10 80 “FF 00 00 C7”).

Figure 13 shows a video stream blanking and the preamble included.

Alternatively, the synchronization values embedded in the data stream can also be provided with the help of external lines “pins”. Figure 3 show such lines “DV, VREF, HREF”.

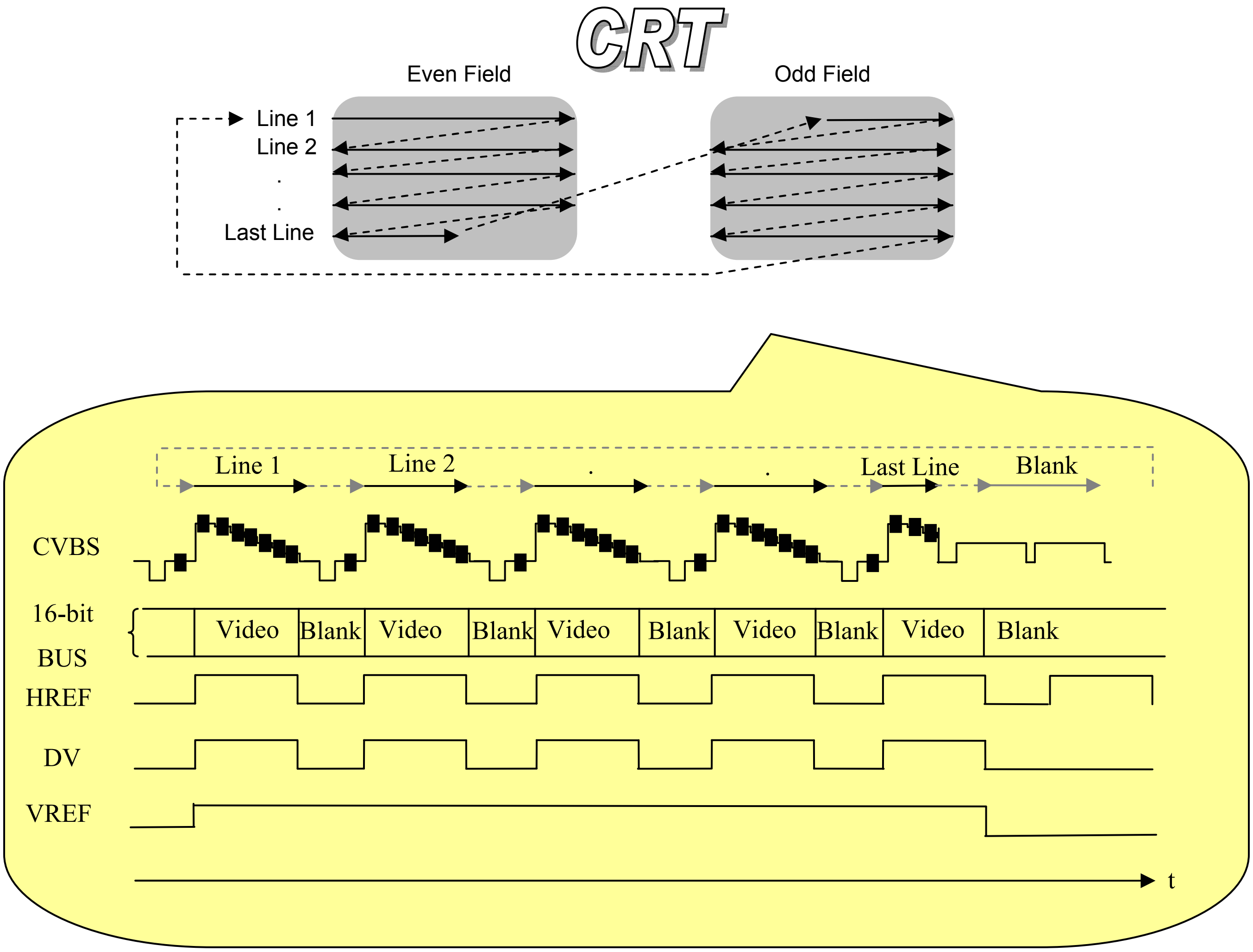


Figure 3: Analog input and digital output of an video decoder

Asynchronous Interface of the ADSP-BF535

The processors asynchronous interface is used to receive the video data. 32-bit data can be fetched in a manner it is shown in Figure 4.

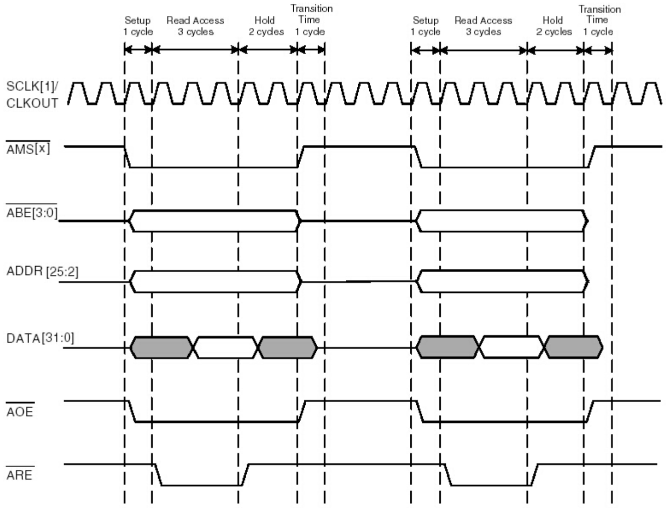


Figure 4: ADSP-BF535 Asynchronous Timing

As mentioned in the ADSP-BF535 Blackfin Hardware Reference Manual after a read cycle is initiated the Async Memory Select line (/AMSx), Async Output Enable line (/AOE) and the Async Read Enable line (/ARE) become asserted. After a multicycle "Read Access" delay (Configured by the Async Interface Bank Control Register), the /ARE pin normally de-assert to complete the read operation. As the processors external memory map is split into four banks of asynchronous memory regions there are four separate select lines "AMS[3:0]" dedicated to each memory bank. The timing is equivalent to the read enable line "/ARE" unless additional hold cycles are inserted. The /AMSx lines can be taken to act as a chip select "CS" to different devices by accessing different memory banks.

Due to the architecture of the ADSP-BF535, a DMA-controlled data download is somewhat non-intuitive. Each DMA data transfer is split into bursts of eight read accesses. After the burst, a gap appears because of internal bus activity. Figure 5 illustrates this. As shown below the first DMA is set up to read 32x32-bit data words (shown as Channel 2, the /ARE signal). However, the DMA channel can be programmed to a maximum number of 64k words. The gap between these 32 words is required for loading the next DMA descriptor.

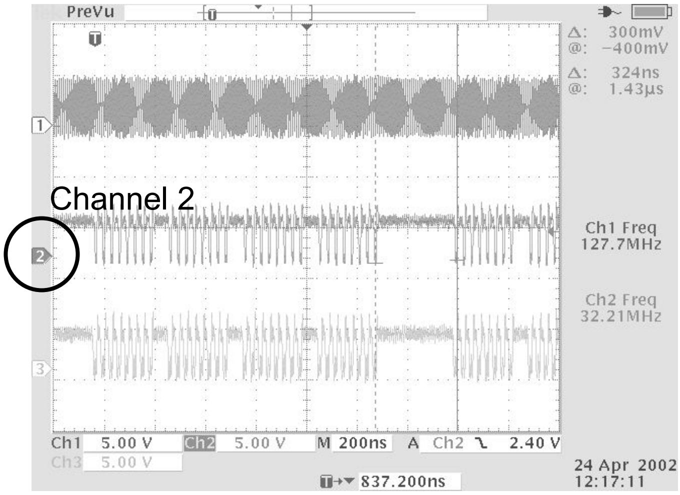


Figure 5: ADSP-BF535 DMA Read Bursts

Because of each DMA transfer is split into bursts of eight accesses (in this configuration, four bursts per DMA execution) the understanding of this behavior is crucial for developing a proper DMA interface. Figure 6 zooms into one of these burst patterns to analyze how many cycles are taken for each access.

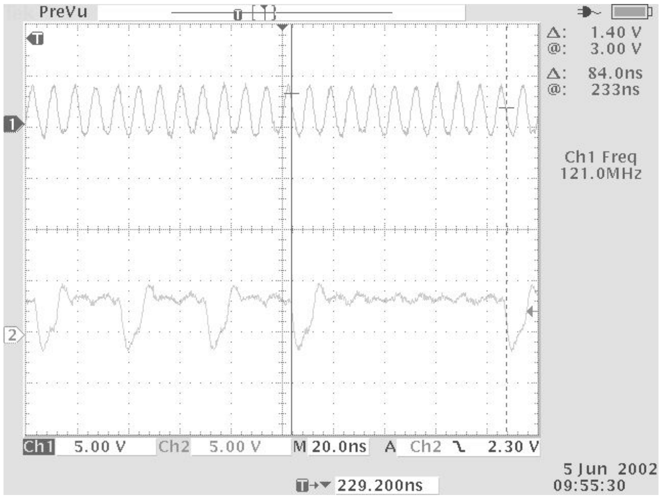


Figure 6: ADSP-BF535 DMA Read Accesses

The peripheral clock "SCLK" is displayed in channel 1 and channel 2 shows the /ARE pin. After eight read strobes are done nine extra SCLK cycles are taken to place the data into internal memory.

Interface the ADSP-BF535 into the ADV7183

The ADSP-BF535 asynchronous interface is configured to make full use of its 32-bit external memory interface, in order to gain maximum throughput. As the ADV7183 is configured to run in 16-bit LLC synchronous mode its output does not meet the timing requirements of the processors asynchronous interface in more than one topic.

1. The 16-bit bus must be extended to gain maximum throughput over the 32-bit bus.
2. As discussed in the chapter before, due to the descriptor reload the processor is not able to receive data while it is in the reloading process. Data of the ADV7183 would get lost without the use of any hold mechanism.
3. SDRAM could never be accessed without the use of external buffers as long as the video decoder transfers data sequentially (Data/Address busses are multiplexed).
4. Without any logic active video data as well as control information's "Blanking" embedded in the data would be received. Although, the control information's are not needed it would be transferred and waste memory space.

All these topics has been taken to design an interface that serves these. In this application example the decision was made to take FIFOs of IDT. The IDT72V06 is a dual port FIFO memory that operates at a power supply voltage between 3.0V and 3.6V. It loads and empties data on a first-in/first-out basis. It uses Full, Half Full and Empty flags to prevent data overflow and underflow. The reads and writes are internally sequential through the use of ring pointers, with no address information required to load and unload data. Data is toggled in and out of the devices through the use of the Write (/W) and Read (/R) pins. The IDT72V06 FIFO incorporates a 9-bit data path. In order to take it four of them must be connected in parallel to make it compatible to the ADSP-BF535 32-bit interface.

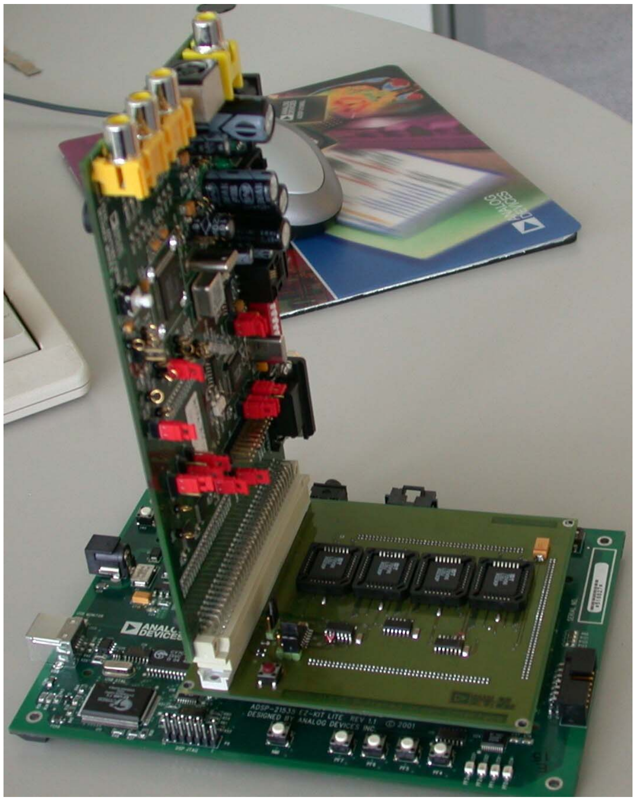


Figure 7: Picture of the actual application

Figure 7 shows the actual application. Its designed as a daughter card that fits on the ADSP-BF535 EZ-KIT Board. As it can be seen on top is the ADV7183 evaluation board connected. Both boards are part of Analog Devices evaluation environments.

IDT today offers more enhanced FIFOs as it has been taken in this EE-Note. The chapter "Design Improvements" shows an alternative part that fits better with less glue logic

As it can be seen in Figure 8 the FIFOs are designed in between the processor and the video decoder. Lets start with the connections to the Blackfin “ADSP-BF535”. As discussed earlier the ADSP-BF535 is fitted with the 32-bit asynchronous interface. The 32-bit data bus is linked to the FIFOs output “Q[7:0]”. The Asynchronous Memory Select Line 1 “/AMS1” is linked to the Read Line “/R” of the FIFOs. Taking /AMS1 indicates that the processors Memory Bank 1 must be accessed to activate the FIFOs. The FIFOs Half Full Flag “/HF” indicates a half full and the ADSP-BF535 should start a download to prevent a data overflow. That’s why its connected to programmable flag GPIO1. A processor interrupt driven by the programmable flag GPIO1 starts the download. In general, there is no connections of the Blackfin to the video device. It just waits until it indicates that the FIFO must be emptied. The key point can be found between the Video Decoder and the FIFOs. Two main functions must be implemented.

First: Automatically, it should be detected the start of the first frame otherwise the frames can not be reconstructed.

Second: The FIFOs should not be filled with blank data. Blank data just wastes memory.

The D-Flip Flop “74LV74D” at the bottom of Figure 8 is designed in to detect the start of the first frame after power up or manually reset activated by a programmable flag. With the help of the signal VREF and FIELD supplied by the video decoder the Data transfer will be enabled at the first stage. The Data Valid signal “DV” indicates fractions of active video. With the help of a logical AND the DV signal and the Flip Flop output of VREF and FIELD defines a signal that enables transfer at the second stage.

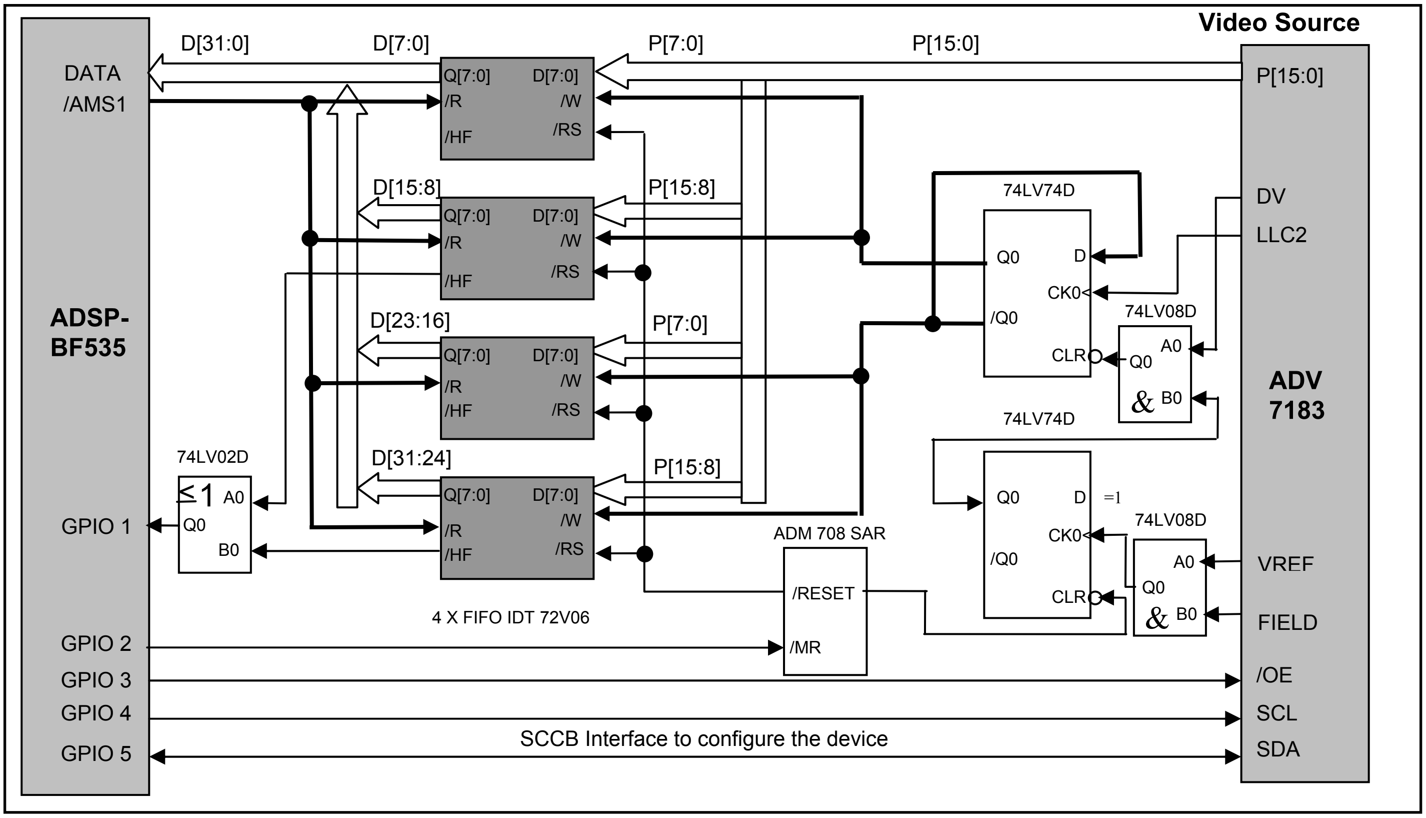


Figure 8: ADSP-BF535 video connection block diagram

To extend the 16-bit bus of the Video Decoder to become a 32-bit bus compatible with the ADSP-BF535 a second Flip Flop is used at the final stage. As it can be seen on top of Figure 8 the second Flip Flop half's the frequency. It links the output line Q0 to the top FIFOs write line and links the output line /Q0 to the bottom FIFOs write line. Alternately, the 16-bit bus will be linked to both FIFO pairs.

Due to this logic the FIFOs will just be filled with video data.

Data Download Procedure

With regards to the discussion above the normal procedure to receive video data in this system is as follows. After power up the FIFOs will be filled with data when a start of a field occurred. All this will be ignored because the processor is not ready to run. The first action of the Blackfin software is to assert GPIO2 in order to reset the system manually. The programmable flag drives a supervisory circuit effecting the Flip Flops and FIFOs to reset. The first video field after reset fills the FIFOs. After the FIFOs are half full the /HF pin will be asserted. The assertion of /HF generates an processors interrupt via GPIO1. The interrupt service routine includes instructions to start a descriptor based DMA. The IDT72V06 includes a 16k address space so that the /HF pin asserts after it has been filled with 8k of data. To prevent memory underflow the maximum number of DMA transfers must be limited to 8k. When the DMA transfer is finished the next descriptor will be loaded automatically but the transfer will not continue. The data already received can be processed. As soon as the FIFOs indicate been half full again the next DMA transfer is activated.

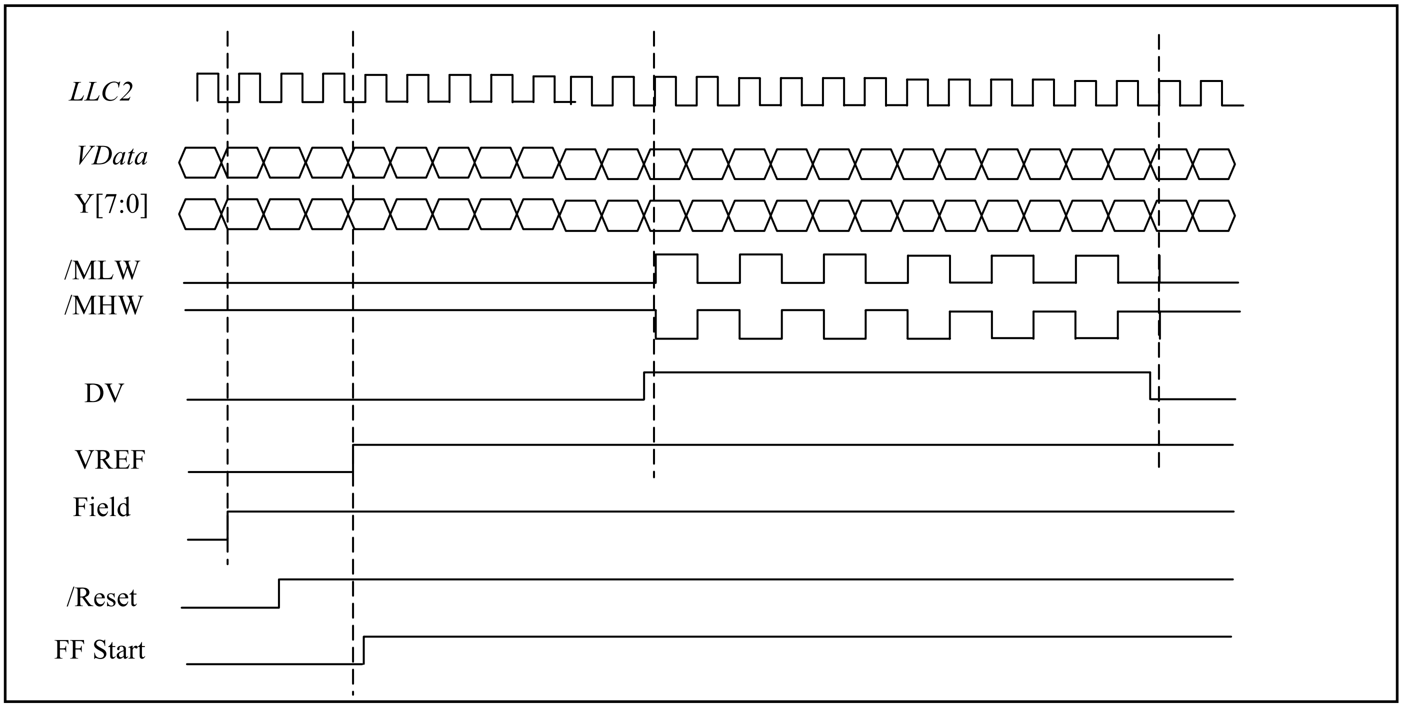


Figure 9: ADV7183 output and video system timing

This process repeats as long as it is not turned off by software.

In general a DMA engine is designed to transmit data independently of core activities. While the first portion of data will be processed the next portion can be received without interrupting the core.

Figure 9 shows the timing from ADV7183 to the FIFOs. After RESET has been de-asserted VREF will assert at the beginning of the next field. The pin DV will remain de-asserted as long as just blanking goes over the bus. At the beginning of the first visual line (active video) DV will assert and remain asserted while active video is transmitted in the actual line. At the end of the line DV will de-assert and remain until the next visual line. While DV remains asserted alternatively the FIFOs gets filled triggered by the top Flip Flop (/MLW and /MHW). As its output and inverted output are taken (Q0 and /Q0) the top FF expends the 16-bit bus to an 32-bit bus.

Design Performance.

As in the chapter "Interface the ADSP-BF535 into the ADV7183" covered the SDRAM could never be accessed if the bus would be taken for the video download already. Due to the fact that the Blackfin Processor empties the FIFO much faster than it is filled by the video decoder the bus remain idle in between. Figure 10 (channel 1) shows blocks of memory strobes (/AMS1). If the processors peripheral clock (SCLK) runs at 120MHz and the external port is set to run at maximum speed it takes 400µs to download 8k words (32-bit). After each download the bus remains in idle. Channel 2 of Figure 10 shows a signal peak triggered by the half full flag of the FIFOs. That results in a Blackfin processor interrupt which starts the next DMA transmission. From the end of transmission one to transmission two takes 1.04ms.This is left for SDRAM or any other access via the external port.

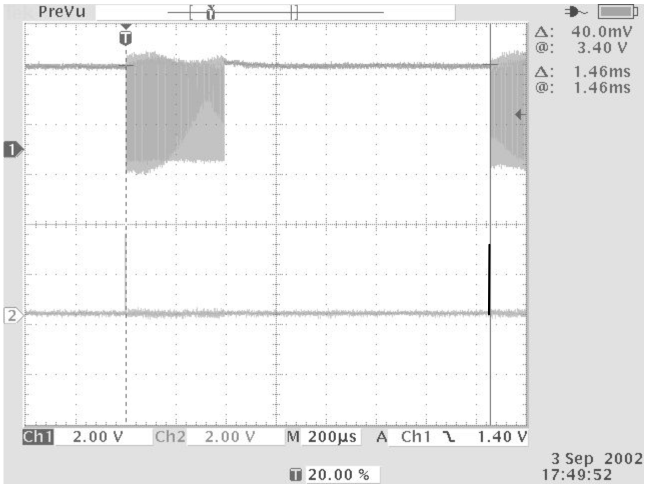


Figure 10: ADSP-BF535 performance test result

Design Improvements

Today's FIFOs are much more enhanced than the one taken for this project. The main reasons for taken the IDT72V06 was the availability as well as the package size. The IDT72V06 is available in a PLCC package instead of the IDT72T36105 discussed below. However, the next IDT generation of FIFOs like the IDT72T36105 provides bus widths up to 36 bits in a single chip. It supports features like asynchronous and synchronous timing, retransmit, programmable almost full and empty flags, bus matching… These chips provide a lot of functionalities that decrease the number of external logic required.

Asynchronous vs. Synchronous Timing:

The device can be programmed to meet the asynchronous timing as it is required for Blackfin as well as synchronous timing required for the video device.

Retransmit:

Allows to transmit the data stored many times.

Programmable almost full/empty flags:

A programmable threshold of a certain level when the pin asserts. As mentioned before taking the half full flag wastes a lot of memory. A cheaper chip with less memory could be chosen if the almost full fag is programmable.

Bus Matching:

The input bus of the FIFO can be limited to 18 bits while the output keeps the width of 36. This way the device collects 18 bit words and packs it to be 36 bit words. No glue logic is required for alteration. Figure 11 shows a schematic that includes such a device. This schematic is just a recommendation but has never been tested on hardware.

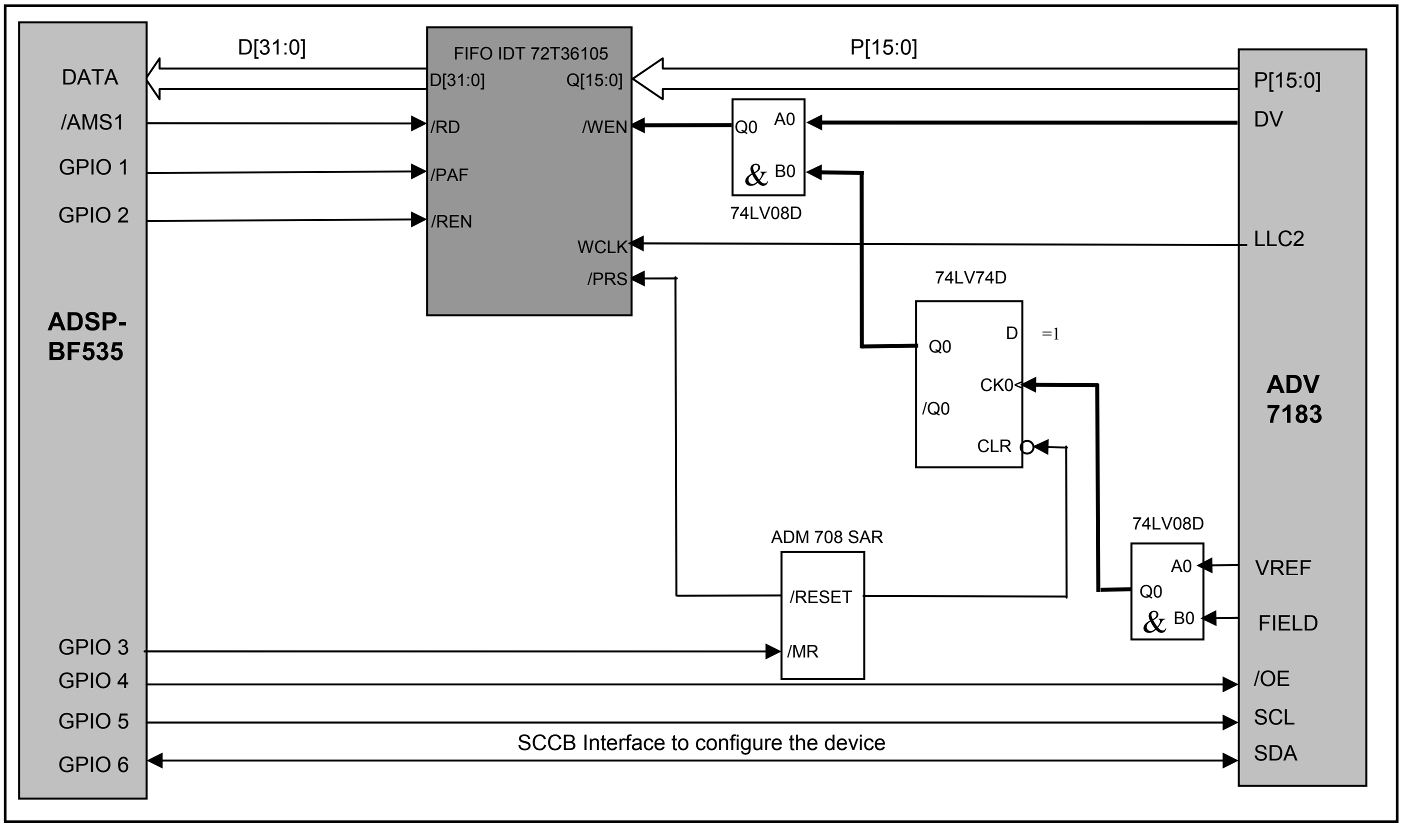


Figure 11: Enhanced ADSP-BF535 video connection block diagram

Conclusion

This note should have given an idea on how to connect devices like the ADV7183 video decoder to the ADSP-BF535. It has been discussed the processors behavior of the external port (EBIU) especially in combination with DMA transfers. A standard video source (ADV7183), the digital signal obtained from and the way it can be implemented in the video signal chain has been covered. Furthermore the glue logic required and the connections to such devices.

Derivates as such as the ADSP-BF533 will be equipped with interfaces (PPI) that fit glueless to this sort of video devices. Additionally, the PPI interface will include features like ITU-656/601 support. The data stream will be investigated by the PPI and optionally skip blanking, skip even or odd samples to select bw/color samples or skip field 1 or 2. Also, external sync signals are supported (HSYNC/VSYNC).

Future processors will have even more support as such as on-fly separation of luminance and chrominance values, windowing of an area of interest, re-sampling 4:4:4>4:2:2, RGB/YUV conversion, down sampling or up sampling.

For lower speed devices it is referred to the note EE-181. EE-181 is dedicate to show how single chip CIF cameras can be connected to the ADSP-BF535. Such cameras supply an output clock of about 8MHz.

ADSP-BF531/532/533 Support

The ADSP-BF531/532/533 is already equipped with a PPI Interface. The use of any glue logic is not necessary. Due to the fact that the PPI is a half duplex bi-directional port it is not capable for multiple video devices. But if for any reason two PPI interfaces are required the 16-bit external port of the ADSP-BF531/532/533 can also be taken as it is done in this note to make a second PPI. Figure 12 shows this. With the help of just two FIFOs and nearly the same logic the ADV7183 can also be connected to the ADSP-BF531/532/533 external port.

The ADSP-BF531/532/533 does not show the same behavior as the ADSP-BF535 on the external port due to design changes. The gaps are not existing anymore. Please see the Hardware Reference Manual

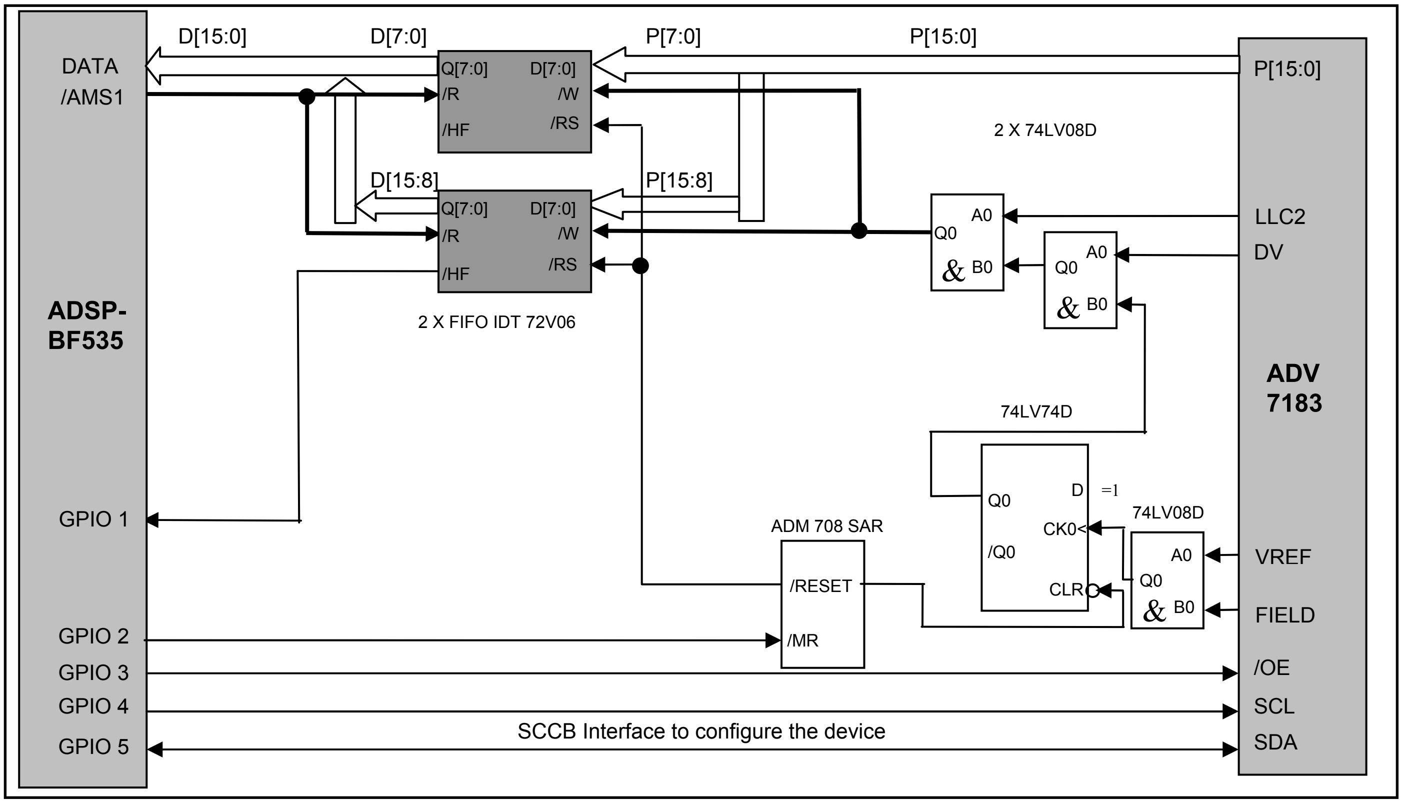
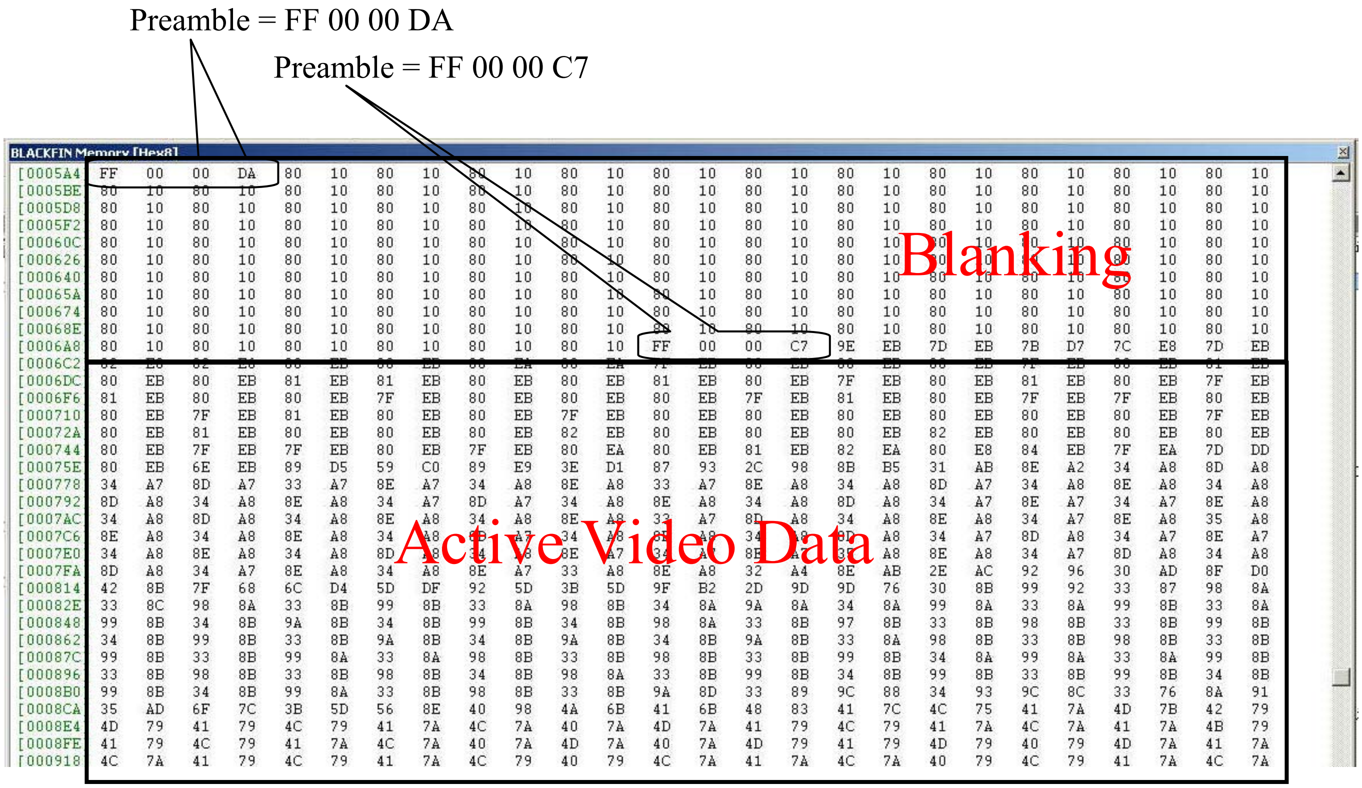


Figure 12: ADSP-BF533 video connection block diagram



```
[0005A4] FF 00 00 DA 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10
[0005BE] 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10
[0005D8] 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10
[0005F2] 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10
[00060C] 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10
[000626] 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10
[000640] 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10
[00065A] 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10
[000674] 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10
[00068E] 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10 80 10
[0006A8] 80 10 80 10 80 10 80 10 80 10 80 10 FF 00 00 C7 9E EB 7D EB 7B D7 7C E8 7D EB
[0006C2] 82 E8 82 E6 80 EB 80 EB 80 EA 80 EA 7F EB 80 EB 80 EB 80 EB 7F EB 80 EB 81 EB
[0006DC] 80 EB 80 EB 81 EB 81 EB 80 EB 80 EB 81 EB 80 EB 7F EB 80 EB 81 EB 80 EB 7F EB
[0006F6] 81 EB 80 EB 80 EB 7F EB 80 EB 80 EB 80 EB 7F EB 81 EB 80 EB 7F EB 7F EB 80 EB
[000710] 80 EB 7F EB 81 EB 80 EB 80 EB 7F EB 80 EB 80 EB 80 EB 80 EB 80 EB 80 EB 7F EB
[00072A] 80 EB 81 EB 80 EB 80 EB 80 EB 82 EB 80 EB 80 EB 80 EB 82 EB 80 EB 80 EB 80 EB
[000744] 80 EB 7F EB 7F EB 80 EB 7F EB 80 EA 80 EB 81 EB 82 EA 80 E8 84 EB 7F EA 7D DD
[00075E] 80 EB 6E EB 89 D5 59 C0 89 E9 3E D1 87 93 2C 98 8B B5 31 AB 8E A2 34 A8 8D A8
[000778] 34 A7 8D A7 33 A7 8E A7 34 A8 8E A8 33 A7 8E A8 34 A8 8D A7 34 A8 8E A8 34 A8
[000792] 8D A8 34 A8 8E A8 34 A7 8D A7 34 A8 8E A8 34 A8 8D A8 34 A7 8E A7 34 A7 8E A8
[0007AC] 34 A8 8D A8 34 A8 8E A8 34 A8 8E A8 33 A7 8D A8 34 A8 8E A8 34 A7 8E A8 35 A8
[0007C6] 8E A8 34 A8 8E A8 34 A8 8D A7 34 A8 8E A8 34 A8 8D A8 34 A7 8D A8 34 A7 8E A7
[0007E0] 34 A8 8E A8 34 A8 8D A8 34 A8 8E A7 34 A7 8E A7 8E A8 8E A8 34 A7 8D A8 34 A8
[0007FA] 8D A8 34 A7 8E A8 34 A8 8E A7 33 A8 8E A8 32 A4 8E AB 2E AC 92 96 30 AD 8F D0
[000814] 42 8B 7F 68 6C D4 5D DF 92 5D 3B 5D 9F B2 2D 9D 9D 76 30 8B 99 92 33 87 98 8A
[00082E] 33 8C 98 8A 33 8B 99 8B 33 8A 98 8B 34 8A 9A 8A 34 8A 99 8A 33 8A 99 8B 33 8A
[000848] 99 8B 34 8B 9A 8B 34 8B 99 8B 34 8B 98 8A 33 8B 97 8B 33 8B 98 8B 33 8B 99 8B
[000862] 34 8B 99 8B 33 8B 9A 8B 34 8B 9A 8B 34 8B 9A 8B 33 8A 98 8B 33 8B 98 8B 33 8B
[00087C] 99 8B 33 8B 99 8A 33 8A 98 8B 33 8B 98 8B 33 8B 99 8B 34 8A 99 8A 33 8A 99 8B
[000896] 33 8B 98 8B 33 8B 98 8B 34 8B 98 8A 33 8B 99 8B 34 8B 99 8B 33 8B 99 8B 34 8B
[0008B0] 99 8B 34 8B 99 8A 33 8B 98 8B 33 8B 9A 8D 33 89 9C 88 34 93 9C 8C 33 76 8A 91
[0008CA] 35 AD 6F 7C 3B 5D 56 8E 40 98 4A 6B 41 6B 48 83 41 7C 4C 75 41 7A 4D 7B 42 79
[0008E4] 4D 79 41 79 4C 79 41 7A 4C 7A 40 7A 4D 7A 41 79 4C 79 41 7A 4C 7A 41 7A 4B 79
[0008FE] 41 79 4C 79 41 7A 4C 7A 40 7A 4D 7A 40 7A 4D 79 41 79 4D 79 40 79 4D 7A 41 7A
[000918] 4C 7A 41 79 4C 79 41 7A 4C 79 40 79 4C 7A 41 7A 4C 7A 40 79 4C 79 41 7A 4C 7A
```

Figure 13: ITU-656 Digital Video Data Stream

References

[1] IDT 72V06 Datasheet, August 2001 Integrated Device Technology Inc

[2] IDT72T36105 Datasheet, June 2002 Integrated Device Technology Inc

[3] ADV7183 Datasheet, Rev 0, 2002. Analog Devices Inc

[4] ADSP-BF535 Datasheet, Rev PrC, 2002. Analog Devices Inc

[5] ADSP-BF535 Blackfin® Processor Hardware Reference Manual, Second Edition, March 2003

[6] Video Demystified by Keith Jack, Third Edition, 2001

Document History

Version	Description
September 01, 2003 by T. Lorenzen.	Initial Release