Version 1.3 For EDK 6.3i 02/21/2005

Acknowledgement

This document was created to accompany the example MP3 design we received and adopted. Many thanks to the anonymous donor. We do our best to keep it working, but no promises. The design has however been successfully tested under EDK 6.3i.

Goals

- Get a better understanding of MicroBlaze systems by using this example design, which has instructions stored in on-chip BRAMs and data stored in the ZBT external memory.
- Get an idea of how audio works on the Multimedia development board.
- Get an idea of how to offload processor-intensive computations to dedicated hardware using the FSL links.

Requirements

Access to EDK 6.3i, the Xilinx Multimedia development board, and speakers or headphones plugged into the Multimedia board headphones jack. Note that you may want to have your own (legally acquired) MP3 file under 2MB for testing playback functionality. It is unclear whether VBR coding is supported or not.

Preparation

- Download the m13.zip file to your working directory and unzip it. MAKE SURE THIS PATH HAS NO SPACES!
- Read the README file in the unzipped directory.
- Read the section on user-specified Makefiles in the EST Guide → Xilinx Platform Studio (XPS) → Flow Tool Settings and Required Files.
- Make sure your speakers/headphones are plugged into the Multimedia board and have power, if needed. Please ensure the speaker volume is not so high that it will make you go deaf or give a heart attack to your TAs. (If you can hear an audible "click" when you are downloading the bitstream to the FPGA, then the volume is too loud!)

Notes

This design has been modified several times to work with EDK 6.3i. There are a number of non-standard modifications (i.e. hacks) that this design relies upon in order to function correctly. For this reason, the design does not port easily to newer versions of EDK. The following items should be kept in mind:

- The opb_zbt_controller officially has Xilinx IP OBSOLETE status with EDK 6.2i Service Pack 2. To get around this, the /pcores directory for this core was copied and added to the /pcores directory of the example design. The MPD file was modified to describe it as having DEPRECATED status instead of OBSOLETE. Otherwise, XPS would claim that the project has errors.
- The design uses a hand-crafted version of the FSL link which provides a much deeper FIFO (2048 x 32-bit words) than originally available. For this reason, the "fsl_v20_v1_00_b" IP core was copied into the /pcores directory and the modifications were made locally. Xilinx now provides a new version of the FSL IP core ("fsl_v20_v2_00_a") which supports FIFOs up to 8192 words in depth.

- The DCM configuration used by this design is not consistent with the Xilinx recommended configuration (refer to m08 - Using ZBT Memory). This design directly instantiates the DCM primitive component along with the BUFG global clock buffers ("dcm_block_v2_00_a" and "bufg_v2_00_a," respectively). An IP core with this functionality ("dcm_module_v1_00_a") is available in the Xilinx IP library and should be used for your designs instead.
- This design uses a custom makefile (system_mp3.make) instead of the automatically-generated makefile (system.make). Why do you suppose this is required?

Step-by-step

You will use XPS to play your own MP3 song. This can be done via the GUI, by a slightly different procedure than what is in the README file.

- Compile software libraries and hardware bitstream. Make sure that you have plenty of free disk space, otherwise compilation may fail unnecessarily. This would be unfortunate because generating the netlist and bitstream takes quite a while.
- Note that with EDK 6.3i, Timing fails. What was the constraint that failed, and where is the critical path in the design?
- Make sure user input switches are in the positions 1: off, 2: on. Download the bitstream to the FPGA.
- Open the Xygwin Shell (Tools → Xygwin Shell). This is a Xilinx tool based on a Cygwin Shell (www.cygwin.com), and allows you to manually access the Xilinx Tools.
- Run XMD:

  xmd
- Connect to the MicroBlaze debug module:

  mbconnect mdm

  You will get a warning message indicating that this command is now deprecated, but you can ignore the warning safely. Notice that we cannot use the regular method of starting XMD (Tools → XMD) because it loads the memory map from the system.mhs/mss files, and refuses to download the executable as it thinks the memory addresses are out of range.
- Provide the size of the MP3 file at location 0xfa800000:

  mwr 0xfa800000 ⟨**size**⟩

  Note: You can find the size of your MP3 file by right-clicking on the file in Windows Explorer and clicking "Properties." Make sure you use the file size in bytes, and not the file size on disk.
- Provide the MP3 data to download at location 0xfa800004 (The length of the song should be less than 0xFFFFF, or 2MB):

  dow -data ../MP3Streams/classic2.mp3 0xfa800004
- Download the Microblaze executable file.

  dow my_blaze/code/executable.elf

  This step downloads code to local memory (BRAMs) and read-only data to external memory (ZBT RAM). Examine the linker script file and see if you can find where the stack is located.
- Execute the program from location 0x0:

  con 0x0
- Your song should now play on the headphone jack of the Multimedia board. Once the song has completed, the program will report "Done" via XMD. You can repeat the song by stopping the processor and re-starting it from location 0x0 (no reset is required).