Digital Talking Book Player for the Visually Impaired Using FPGAs

Azadeh Nazemi, Cesar Ortega-Sanchez, Iain Murray
Electrical and Computer Engineering Department
Curtin University
Perth, Western Australia
azadeh.nazemi@postgrad.curtin.edu.au, c.ortega@curtin.edu.au, i.murray@curtin.edu.au

Abstract— The Digital Talking Book (DTB) player is a device for the visually impaired to read, search, navigate and bookmark written material using DAISY and EPUB standards. This paper presents the design and implementation of a DTB player in an **FPGA**-based embedded system to play audio books containing MP3 (Daisy) files and utilize Text to Speech Synthesis (TTS) for text only books or EPUB books.

Keywords- Assistive Technology; FPGA; Daisy Player; EPUB; uClinux; Text-to-Speech Synthesis

I. INTRODUCTION

Visually impaired people traditionally use one of several reading methods such as Braille and audio tapes. Braille is bulky, has a limited range, is expensive to produce and has a complex coding system (hard to learn) that requires high tactile sensitivity. Analogue books, such as tape and record, or digital books in MP3 format, have sequential access without bookmarking and navigation features [1, 2]. To enable navigation there are two standards: Digital Accessible Information System (DAISY) and Electronic Publication (EPUB). DAISY is an international open standard that provides full navigation and search capabilities, allows bookmarking and can handle human-read and synthetic TTS [3]. EPUB is a text-only standard compatible with the Daisy standard and has some navigation abilities [4].

Existing DAISY players are either hardware or software based, both of which have significant drawbacks. Hardware players tend to be expensive due to their relatively limited market (compared to mainstream consumer devices). Software players require the user to be familiar with computers and their often complex assistive technology.

This paper describes an embedded system-on-chip DTB player under development at Curtin University to address the special reading needs of visually impaired people.

The system is microprocessor-based, containing custom IP, specialized hardware and a customized Linux kernel. uClinux -an embedded Operating System (OS)- has been installed on MicroBlaze as a suitable platform for hardware /software co-design [5, 6]. The system has been implemented in a Field-Programmable Gate Array (FPGA) to allow reconfiguration of hardware after manufacturing to meet future DTB standards [7, 8].

The final goal is to be able to offer a portable, low-cost, low-power, fully-accessible digital talking book player for the visually impaired.

II. HIGH-LEVEL SYSTEM DESCRIPTION

Figure 1 shows the high-level block diagram of Curtin's DTB player. The current prototype was implemented in the ***Spartan3E*** Starter kit board by Digilent [10].

The player uses two RS-232 channels to communicate with a Personal Computer and a TTS module. The computer supplies the media files to the DTB. Alternatively, a Media Access Controller (MAC) coupled to an on-board physical layer network chip supplies a standard Ethernet connection to get DAISY or EPUB books over an FTP connection. Digilent's PMODs DA2 and AMP1 [12] are used by an MP3 decoder to play audio files. A user interface based on push buttons communicates user requirements and commands to the DTB player.

An RC8660 DOUBLETALK module is connected to the board via a secondary RS232 serial port. This module receives DAISY text files and "reads" them out loud by means of sophisticated text-to-speech algorithms. [13].

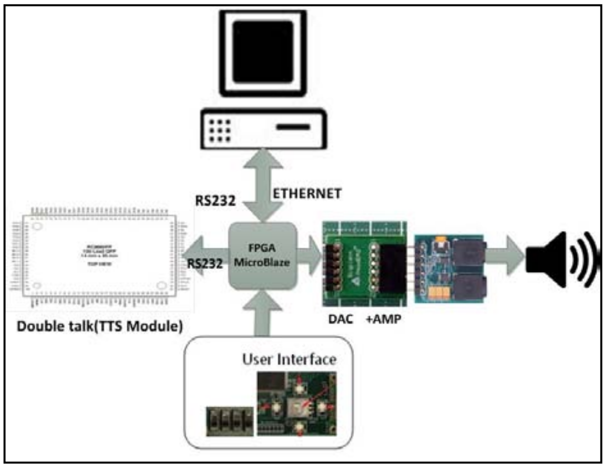

Figure 1. FPGA-based embedded system for DTB player

A Linux host computer enables FTP and trivial file transfer protocol (TFTP) services. A serial communication program or terminal emulator (Kermit in this instance) is required to communicate with the MicroBlaze Linux machine. Petalinux is used as a tool chain to port uClinux OS to Microblaze [11].

III. BUILDING THE DTB PLAYER

Building the DTB player requires the following steps:

1. Build software application in C++ and port the DTB application into the kernel image as user-application.
2. Build the hardware platform using Xilinx's EDK environment.
3. Prepare Petalinux requirements (copy-autoconfig).
4. Build uClinux kernel.

A. Software application

The DTB Player software application is written in C++ so as to take advantage of existing Linux libraries and portability across various possible platforms. The flowchart in Figure 2 shows the functionality of the DTB player.

The DTB player imports book files over the ftp channel and recognizes the relevant standard or format. If the input file is in EPUB format, then the extension is changed to .zip and then files associated to navigation levels are extracted according to the map file. Once extracted, these file can be treated as a DAISY standard book. The folder to store books is 16 Mbytes in size.

EPUB is a text-only format that does not include pre-recorded MP3 files. The map file contains references to XML files that form the different navigation levels of the book (chapters, sections, paragraphs). Audio is generated by sending these files to an external TTS hardware module [13].

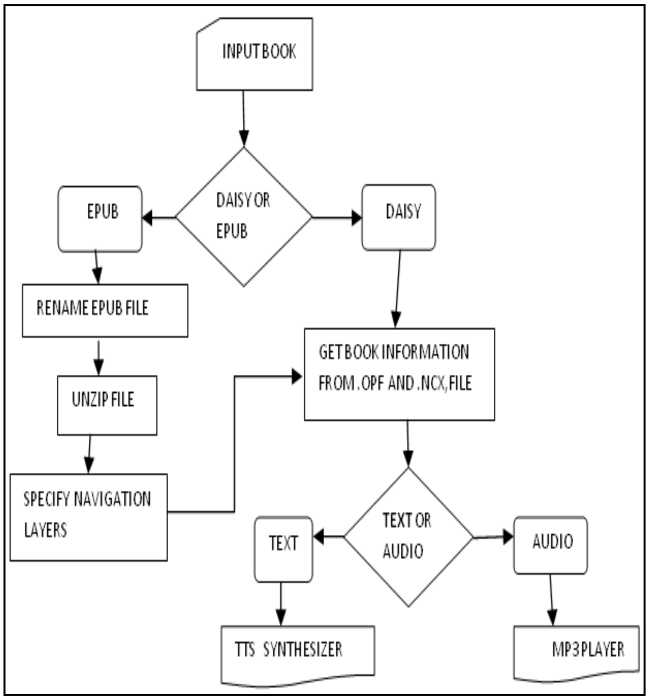

Figure 2. Flowchart of application

The Daisy standard supports three types of books [1]:

- Full-text /full-audio.
- Audio only.
- Text only.

In the first two types, the file contains a collection of MP3 files organized according to the different navigation levels of the book. These files are then played using a conventional MP3 player program. For text only Daisy books and EPUB books, text to speech (TTS) is needed. In this project, hardware TTS has been used to simplify processing and reduce the computational load on the Microblaze controller.

Curtin's DTB player has the following playing functions: *Pause, Stop and Navigate by chapter, section, subsection or paragraph*. Bookmarking is also supported in the current version of the DTB player [14]. A special bookmark saves the location where the user last stopped reading so that the player may continue from this location in the next reading session.

The completed program was compiled in uClinux with the *microblaze-uclinux-g++* compiler. The DAISY software program executed in the uClinux operating system environment includes its own set of build rules that are architecture specific and are built independently of the uClinux kernel.

The Most critical issue encountered in creating a software application under uClinux with MicroBlaze is the lack of a Memory Management Unit (MMU). Without a MMU, MicroBlaze is limited to operating systems with simplified protection and a virtual memory-model, so segmentation faults or bus errors occur when the CPU attempts to access a location in memory that does not exist [9].

B. Microblaze Hardware Platform

The minimum requirements to build a uClinux hardware platform are:

- MicroBlaze soft core processor
- Timer Interrupt controller.
- Two Universal Asynchronous Receiver Transmitter (UART) channels:
    - DCE: DTB Player for personal computers
    - DTE: connection between TTS module and MicroBlaze.
- 32 Mb of external DDRAM for loading and running DTB Player application
- 64 Mb of flash memory for storing kernel image of OS, DTB Player and device drivers.
- Ethernet MAC - A Media Access Controller coupled to an on-board physical layer network chip supplies a standard Ethernet connection to get DAISY or EPUB book as input data over FTP protocol.
- Push buttons and DIP Switches provide the user interface. These devices are accessed using the *XIo-In32(Button-Base-Address)* function. Table I shows the return values of this function and how they are used by the DTB software application. [10]
- PmodDA2 (DAC) and PmodAMP1 (audio amplifier).

TABLE I. PUSH BUTTON VALUES FOR DIFFERENT FUNCTIONS

***XIo-In32 (Button-Base-Address)* return value**	**Button**	**Navigation function**
0X8	LEFT	Last same level
0X4	RIGHT	Next same level
0x2	UP	Upper level
0X6	DOWN	Lower level
0XC	RIGHT+LEFT	Play
0XA	LEFT+UP	Exit

Audio generation is provided by two modules external to the FPGA: The PmodDA2 and PmodAMP1 by Digilent [12]. The DAC was incorporated into the hardware platform using EDK's *Create or Import Peripheral Wizard*. The PLB bus was used to connect DAC to MicroBlaze. The wizard generates a generic VHDL skeleton that is used to incorporate external peripherals to MicroBlaze-based embedded systems. VHDL code to control the DAC was added to this file. The DAC IP core output signals were declared as *External Ports* and the *User's Constraints File* (UCF) was modified to establish the connection between FPGA pins and DAC ports [10, 12].

The PmodAMP1 is a speaker/headphone amplifier that amplifies low power audio signals to drive either stereo headphones or a monophonic speaker. The PmodAMP1 directly interfaces with the PmodDA2 module.

In the process of creating the embedded system using EDK's wizard, the *Instruction Cache* and *Data Cache* have to be set to at least 2KB [11]. Figure 3 shows the block diagram of the hardware platform of the DTB.

C. *Hardware Platform Requirements for Petalinux*

Once the hardware platform has been created using EDK's wizard, the *system.mss* file must be modified to set the parameter *OS_NAME* to Petalinux. Then the option *Add Software Application Project* must be set to *create FS-Boot application*. In this application the compiler's *optimization level* was set to *Size Optimized*. The FS-Boot code, including source files and header files, is part of the PETALINUX package. FS-Boot is a simple boot loader developed by PetaLogix, intended to serve as the primary bootstrap mechanism when the MicroBlaze CPU first boots [15].

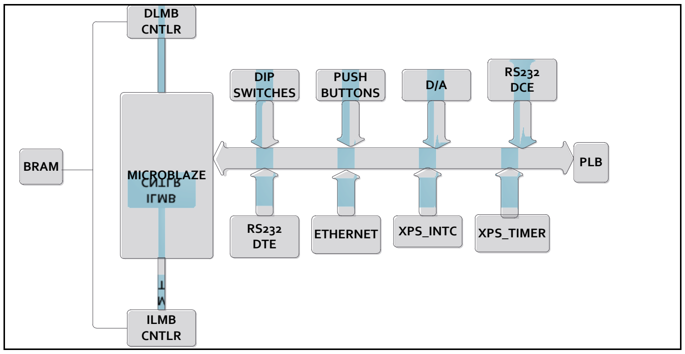

Figure 3. MicroBlaze-based embedded system for the DTB.

FS-Boot's primary purpose is to bootstrap the main system boot loader from flash memory. FS-Boot has features such as small code footprint to minimize on-chip BRAM resource usage, serial download for main boot loader (U-Boot) SREC images and auto-boot capability [11]. Once the *create FS-Boot application* option has been selected, the bitstream and libraries can be generated. The resulting configuration file (.bit) has to be downloaded to the FPGA to test the sytem.

When EDK builds a hardware project, the PetaLinux BSP *kconfig.in* file for 2.6 kernel will be also created and must be copied across to the selected PetaLinux platform. The kernel will automatically detect the software project installation and will install the corresponding hardware *AutoConfig* files. PetaLinux *AutoConfig* framework allows the hardware configurations to be propagated to the boot loader and Linux kernel configurations.

D. *Adding the DAC to the Hardware Platform*

The DAC was added to the embedded system using EDK's wizard. The wizard created hardware to interface the DAC with MicroBlaze through the PLB bus. Also, a couple of VHDL files were automatically created: *user_logic.vhd* and *dac.vhd*. These files had to be modified to complete the interface between MicroBlaze and the DAC. Digilent provides VHDL code to control and interface the PmodDA2 [12].

MicroBlaze controls the DAC with two signals *START* and *DONE*. *START* initiates a digital-to-analogue conversion. At the end of conversion the DAC asserts *DONE* to signal MicroBlaze that a new conversion may be started. This signal exchange is coordinated by the finite state machine shown in figure 4. The description of this controller was added to the *dac.vhd* file. Details of the connection between the DAC and the PLB bus were added to the *user_logic.vhd* file.

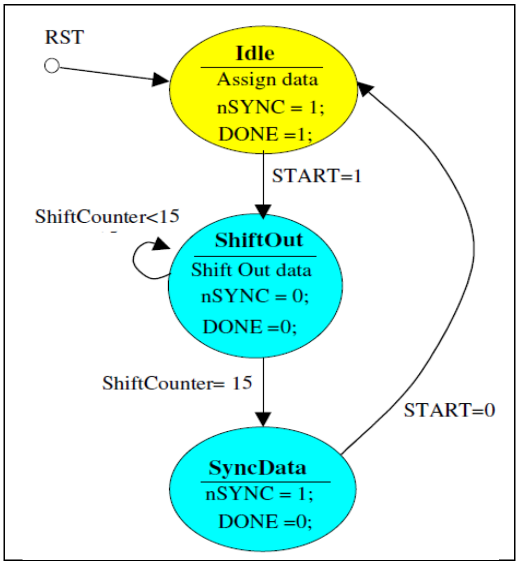

Figure 4. FSM of PmodDA2

E. *Building the Linux Kernel Embedded OS*

The uClinux OS was installed on flash memory and configured to the following settings:

- Kernel: 2.6
- File system: CRAMFS
- Vendor: Xilinx
- Platform: Spartan3e
- Network Protocols: FTP,TFTP.

Table II shows flash partitioning for CRAMFS, the file management program. After downloading the *download.bit* file from the EDK via the JTAG cable, the SPARTAN3E board was connected to the Linux machine using one of the RS232 channels. Before the connection is established, a Kermit terminal emulator must be running on the Linux machine. This step is necessary for monitoring OS loading on MicroBlaze. When the *FS-Boot* application runs, the FS-Boot prompt must be interrupted with the character '*s*' to select the SREC image download option. This option loads the *u-boot.srec* file to RAM via the RS232 cable. After this, a second stage of bootloader will start running.

When the loading process is completed, the prompt *u-boot>* will appear on the Kermit terminal. Table II shows the partitioning table for the system's flash memory. The file *u-boot-s.bin* must be saved in the flash base address and *image.bin* in the base address + 0x8000.

IV. IMPLEMENTATION AND TEST

The design was implemented and tested in a Xilinx Spartan-3E FPGA with a 50MHz clock. According to the device utilisation report generated by EDK, the complete design occupies less than 53% of the FPGA programmable logic. Implementation and test were carried out using Xilinx's ISE suite, version 10.1.

To test the audio sub-system, a PCM file was transferred to uClinux over FTP and then read into an array. Samples were sent to the DAC at the sample rate stored in the file. The sample rate was extracted from the PCM file using the function *XIo_Out32(XPAR_DAC_BASEADDR, samples)*. Several music MP3 files were reproduced using this method.

To test the text-to-speech synthesis (RC8660 module), DAISY and EPUB files were transferred from the Linux machine to the DTB player using FTP over Ethernet. The player stored the file's text navigation block into an array and then sent it character by character to the TTS synthesiser. The transfer took place over the second RS232 channel. The function used to transmit characters is *XUartLite_SendByte(DCE_UART_BASE_ADDRESS, buffer[i])*. Text files were correctly reproduced in spoken English.

TABLE II. FLASH PARTITIOING TABLE

Name	***Size***	***Purpose***
BOOT	0X20000	Store the U-Boot. Mandatory.
BOOTENV	0X20000	Store the U-Boot configurations. Protected read only. Mandatory.
CONFIG	0X40000	Used by the **flatfsd** daemon
IMAGE	0X50000	Store the Linux kernel/ramdisk .Used by the **auto-boot framework**. Mandatory

Once all the DTB player's hardware sub-systems were tested, the FPGA's configuration file was converted to the *mcs* format and permanently stored in the SPI flash memory so that the DTB player gets loaded into the FPGA after power-up. By doing so, the personal computer is no longer necessary to download the design after power-up, and it is only used as an FTP server for book transfers.

V. CONCLUSION AND FUTURE WORK

This paper presented the design and test of an embedded system implementing a digital talking book player. The device can reproduce full text/full audio or audio only DAISY books. EPUB and text-only DAISY books which do not include MP3 or audio files in them, were successfully reproduced using an external text-to-speech synthesis chip set. However, the TTS sub-system significantly increases the cost of the player. Future work is required to integrate hardware-based TTS synthesis and MP3 playing into the design. Also, work is under way to provide additional features like: playback speed control, volume control, rewind and fast forward. Once a commercial version of the DTB goes into production, the full project documentation will be made available.

REFERENCES

[1] Kerscher, G. (August 20 - August 28, 1999). "Braille Production the DAISY Way." 65th IFLA Council and General Conference.

[2] Deltour, R. (2010-02-19). "DAISY Open Source Projects: A greater collaboration for better production tools." DAISY Symposium in Tokyo.

[3] Daisy Standard, Available at: http://www.daisy.org , [Accessed 15 Oct 2010].

[4] International Digital Publishing Forum, Available at: http://idpf.org/ epub, [Accessed 29 Mar 2011].

[5] Ernest, R. 1998. Hardware/Software Co-Design. Co- design of Embedded System, Volume 15 Issue 2, 45-54.

[6] De Michelie, G. 2001. Reading in Hardware/Software Codesign. 1st ed. USA: Academic Press.

[7] Hauck, S. 2002.A Survey of Systems and Software. Reconfigurable Computing, Vol. 34, No. 2, 1-40.

[8] Santamborgio, M. 2009. Reconfigurable System Design and Verification . 1st ed. USA: CRC Press, Inc. Boca Raton.

[9] Hollabaugh, C. 2005. Embedded Linux, 1st ed. USA: Pearson Education.

[10] Xilinx, "Spartan-3E Starter Kit Board User Guide", Mar. 2006. http://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf

[11] Load OS in Microblaze, Available at: http://www.petalogix.com/ resources/documentation/petalinux/petalinux-installation-guide, [Accessed 20 Feb 2011].

[12] Diligent, "PmodDA2 Reference Component", Dec 2008. http://digilentinc.com/Data/Products/PMOD-DA2/PmodDA2 RefComp.zip.

[13] RC SYSTEMS, "RC8660 Voice Synthesizer", http://www.rcsys.com/ chips.htm. [Accessed 20 June 2011].

[14] Custom survey. Results to be published. http://www.surveymonkey.com/s/JHTDRP6

[15] Petalogix, First Stage Bootloader, in Petalogix website http://www.petalogix.com/univ/documentation/userguide/Bootloaders /FSboot. [Accessed 20 July, 2011].