FIELD PROGRAMMABLE GATE ARRAY HARD DISK SYSTEM

FIELD OF THE INVENTION

The present invention relates generally to the field of memory systems incorporating a hard disk drive and more specifically to a system which can provide a variety of fully configurable interfaces to or processes for a hard disk drive.

BACKGROUND OF THE INVENTION

The current state of the art allows in-circuit programmability for a field programmable gate array (FPGA). Typically, this data resides in memory or is passed on to the FPGA via a host computer. Ultimately, this data resides in some storage device (RAM, ROM, or a hard disk all accessed via host intervention). This data is used to program the FPGA to perform its specified function. Practical restrictions on the number of interfaces exist, due to the limitation of memory size and the load on the CPU to steer this data to the FPGA.

Such a restraint would have special importance in a device such as the data shuttle disclosed in the related application which is incorporated herein by reference. In this application, a single data shuttle is disclosed which is especially useful in portably storing

input data stream from a number of sources including television signals, SPDIF formatted data, and information received over buses such as a USB bus or ATA bus or 1394 bus. Each of these requires its own interface, multiplying the number of chips which must be incorporated, increasing the amount of functional silicon and therefore the cost of such a multi-interface product.

SUMMARY OF THE INVENTION

It is an objective of this invention to create a singular assembly, compatible with a multitude of interfaces.

More specifically, in this invention a number of interface chips are replaced by one or more FPGA chips.

More specifically, in the present invention an FPGA is characterized by programming information stored on an associated hard disk.

More specifically, the present invention is characterized by an FPGA integrated with a hard disk assembly which stores the associated in-circuit programming data.

Yet another characteristic is that the FPGA is integrated with a hard disk assembly and the programming is modified by an embedded controller in the hard disk assembly.

In summary, the FPGA/HD assembly is self-contained by integrating with the FPGA with the HDD assembly, with the HDD storing the in-circuit programming for the FPGA, and modifying the FPGA to cooperate with any selected interface under the control of the embedded controller in the HDD. This presents the advantage in this invention that the in-circuit programming data is stored directly on a selective partition of the HDD, leaving as much space as needed for any other data traditionally stored on the HDD. The controller would locate the specific in-circuit data and locate the appropriate soft core into the FPGA upon receiving a command identifying the bus which is to be interfaced within any selected operation.

Other features and advantages of the invention will become apparent to a person of skill in the art who studies this disclosure given in association with the following drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram schematic of the basic elements of the invention; and

Fig. 2 is a block diagram of a board level multi-interface product in which the present invention is useful.

DETAILED DESCRIPTION OF AN EMBODIMENT

The following description describes a system which combines a field programmable gate array (FPGA) with a hard disk drive assembly (HDD) in order to provide a variety of fully configurable interfaces to or processors for the hard disk drive. However, it should be recognized that the features and advantages of this invention are not to be limited to the specific block diagram described herein. The present features may be used with any number of interfaces or processors; further, the FPGA data could be stored in a partition segment of any size disk drive.

Referring to Fig. 1, the basic elements implementing the present invention include the disk drive assembly 100 which includes an embedded controller 102 and preferably an embedded or closely associated field programmable gate array FPGA 104. As is well known in the field of FPGA technology, the functions of this device 104 can be modified from time to time based on data which is downloaded to the FPGA to specify specific functions to its "soft core". For example, in the field of a device which needs to utilize multiple interfaces, for example an ATA interface, a 1394 interface, or a USB interface, the FPGA 104 could adopt the necessary signal processing structures and functions at any given time based on the data loaded into it by the controller 102. According to the present invention, this data can be stored on a separate partitioned region 110 of the HDD 100 after the desired interfaces have been defined. At any time during the use of the FPGA, based on some external control signal, a time function or the like, the FPGA is to provide a particular interface to the overall system 10, the controller 102 which is also incorporated on board the disk drive can download the data from the FPGA data partition 110 on the disk drive into the FPGA 104. As soon as the FPGA data is downloaded, the FPGA serves as that particular programmable device.

Thus, according to the present invention, any programmable interface from the group identified above or others not specifically identified is available for any user to

assign as needed. The soft core data can be time multiplex loaded or otherwise under control of a single external signal received from the external host computer be assigned to be unloaded without further host computer intervention and loading. This will significantly reduce any manufacturing costs by homogenizing the HDD assembly and eliminates previously required functional silicon which would be required to implement each desired interface.

An example of a board level system utilizing multiple interfaces which could well be implemented by a person of skill in the art in this field and which could very profitably incorporate this invention is the data shuttle utilizing a disk storage device shown in Fig. 2.

The shuttle accepts continuous streams of digital information from a variety of sources and conveys them through various interfaces incorporated into the data shuttle and conveys them across a bus into a hard disk drive. In this figure, the inputs from the various devices or sources of data are shown on the left as are the outputs to potential destinations. If the received data is in analog form, it is digitized as shown for example at the upper left where the composite TV video signals 700 and the associated audio 702 are applied to appropriate A to D converters 704, and 706 and then conveyed over buses to an MPEG-2 encoder 710. The outputs of this MPEG-2 encoder 710 are transferred through a data packetizer 712 to the disk processor 714 which does the appropriate file management, bus arbitration, content management and stream management functions so that the data can be stored on a local hard disk drive 720. In this way, any desired video input stream can be converted, digitized, processed and stored for selective access on the data shuttle. The MPEG encoders and decoders could be embodied as an FPGA that was reprogrammed under control of the on-board microprocessor 270 utilizing data stored on the local HDD 220. In this way, the number of actual encoder/decoder chips could be substantially reduced.

The shuttle can also be connected across an interface to a larger hard disk drive which is incorporated in a nesting or docking device 760 for the shuttle. The disk processor 714 can then further transmit the stored digital data from the local disk drive 720 onto a nesting disk drive 740 across an ATA bus which would have a larger capacity. In this way, the shuttle can be moved from one apparatus to another and store input data

from one or several sources through the various interfaces shown.

The shuttle operates under control of its own local processor 770 and includes a power supply and monitor 772 and controls 780-784.

Among other interfaces, the shuttle also includes an input/output bus 720 operative to handle SPDIF format. This input/output bus 720 runs directly to the data packetizer 712 and then across a bus to the disk processor 714. Another SPDIF input 722 for receipt of digital audio is an input to the MPEG-2 encoder 710; the outputs of this MPEG-2 encoder are also conveyed to the disk processor 714 for storage on the local hard disk 720 or the nesting hard disk 740. This digital audio source 722 can also be applied to the MP3 encoder 724 whose outputs are connected directly to the data packetizer 712 and then to the disk processor 714, so that any data in SPDIF format can be stored and selectively accessed.

A plurality of bidirectional buses including a USB bus 730, a 1394 bus 732 and an ATA bus 734 are also provided. The USB bus 730 may provide a bidirectional connection for example to an MP3 player, a digital camera or a PC. Through a USB PHY 740, and a packetizer 742, any of these devices is coupled directly to the data packetizer 712 with their inputs and outputs then conveyed through the processor 714 to the hard disk drive 720. In similar fashion, the 1394 bus 732 could be connected to a digital video camera or a PC or a digital VCR through an appropriate PHY 744 and packetizer 746 to the data packetizer 712 and disk processor 714. Finally, the ATA bus 734 could connect a flash memory or other data storage device directly to the disk processor 714 and then to the disk drive 720.

On the output side, even as the SPDIF input 722 can be conveyed through an MP3 encoder 724 for storage, an MP3 decoder 750 is provided whose output may be coupled to an SPDIF output bus 752 or alternatively through an audio processor 754 to a modulator amp 756. This provides several alternative output lines including an RF modulated AV 758, a stereo phone output 760 and audio output 762. The audio output would more typically be used with the television output 764 which comes through the modulator amp via a digital video encoder 766 and an MPEG-2 decoder 768. The MPEG-2 decoder receives its video information from the depacketizer 712 and the disk processor 714 which as noted above can selectively access any file on the local disk 720.

All of these functions are conducted under the control of the CPU 770 which in this example is Motorola 823E which is supported by a power supply 772 and monitor.

The functions are selected and the input and output sources and destinations are recognized through an IR control 780 and the selected function displayed on an LCD display 782 on the face of the shuttle. Both of these are supported through a control I/O 784 incorporated into the shuttle and controlling the functions of the CPU 770 over the bus 786.

Similarly to the above, where each bus includes a packetizer/depacketizer, a FPGA could be utilized. When a bus is selected, the microprocessor could download the necessary data from the disk drive to program the FPGA to serve the necessary packetizer/depacketizer.

In this way the processor board could be substantially simplified without extra burdens being placed on the host computer, since the on-board processor would have the time available to download the core data from the on-board disk drive without any conflict with its data storage control functions.

Other uses, features and advantages of the present invention will become apparent to a person of skill in the art who studies the above invention disclosure. Therefore, the scope of the present invention is to be limited only by the following claims.

WHAT IS CLAIMED:

1. A system for configuring a field programmable gate array (FPGA) to perform multiple different functions, the system comprising a FPGA integrated with a hard disk drive assembly, the hard disk drive assembly further comprising an integrated microprocessor, the hard disk drive storing, in a separate partitioned region, data to configure the field programmable gate array to perform multiple different functions under control of the microprocessor without intervention by the host of the overall system.

2. A system as claimed in claim 1 including a plurality of interfaces, each connected to the hard disk drive through an encoder or decoder, one or more of the encoders and decoders being implemented by a single FPGA which is reprogrammed to work with a different interface by the data stored on the hard disk drive.

3. A system as claimed in claim 1 including a plurality of interfaces, each connected to the hard disk drive through a packetizer or depacketizer, one or more of the packetizer and depacketizer being implemented by a single FPGA adapted to be programmed to work with a selected one of the plurality of interfaces by data stored on the hard disk drive.

4. A system as claimed in claim 3 including a disk controller adapted to store a plurality of data sets for programming the FPGA in identifiable sections of said hard disk drive, and to select one of said data sets for programming the FPGA in response to a command from the microprocessor dependent on the selected interface which is to send or receive data from the system.

5. A method of configuring a field programmable gate array (FPGA) to perform multiple different functions, the FPGA being integrated with a hard disk drive assembly, the hard disk drive being connected to a plurality of interfaces, the hard disk drive assembly further comprising an integrated microprocessor, the hard disk drive storing, in a separate partitioned region, data to configure the field programmable gate array to perform multiple different functions under control of the microprocessor without intervention by the host of the overall system, the method comprising storing a plurality of data sets for programming the field programmable

gate array on a partitioned region of the hard disk drive, identifying a selected one of the interfaces which is to send or receive data from the system, and programming the FPGA with one of the data sets from the hard disk drive under the control of the microprocessor in response to the identification of the selected interface.

6. A system as claimed in claim 1 including a plurality of interfaces each connected to the hard disc drive, the hard disk drive assembly further comprising means for storing a plurality of data sets for programming the field programmable gate array on said partitioned region of the hard disk drive, and programming the FPGA with one of the data sets from the hard disk drive under the control of the microprocessor in response to the identification of the selected interface.