DAMP - Delft Altera-based Multimedia Platform

Willem Zwart, Jos Eilers, Georgi Gaydadjiev and Sorin Cotofana
Computer Engineering Laboratory Electrical Engineering Department,
Delft University of Technology,
Delft, The Netherlands
{wzwart,jeilers,georgi,sorin}@CE.ET.TUDelft.NL

Abstract—**This paper describes the DAMP (Delft Altera-based Multimedia Platform) architecture. DAMP's main goal is to provide a low cost platform for the embedded systems specification and hardware-software co design, with the main focus on (mobile) multimedia applications. This platform is based on the Altera Excalibur device incorporating a 400k gates FPGA and a state of the art ARM922T processor. In addition, 32MB flash and up to 512MB SDRAM memories are supported to provide sufficient application design space. Wide range of peripheral interfaces, such as ethernet, PS/2, IDE, 24-bit audio in/out and video out are also supported by DAMP. Additionally DAMP provides a methodology for application mapping.**

Keywords— **hardware-software co design, FPGA, ARM CPU core, reconfigurable hardware, multimedia development platform**

I. Introduction

In the Industry, rapidly increasing design complexity and very tight time-to-market schedules are typical for many years now. In the field of Embedded systems design a paradigm shift occurs from system on chip towards reconfigurable systems on chip designs. This shift requires new tools, skills, and methodologies. One way to help developers deal with these new challenges is by providing them with Development Kits. This practice is widely used even for systems based on simple micro-controllers, e.g., 8051-based systems, and provides two advantages. First, by utilizing development kits better design space exploration can be performed. This makes selection of the most appropriate solution for the end product feasible. Second, the utilization of working hardware together with example designs can significantly speed up the development process and shorten the tools learning time.

All University engineering students need to be prepared for the challenges in their engineers life after graduation. Therefore, it is important to include up-to-date software tools and state of the art hardware into the University educational programs. Many of today's Electrical Engineering and Computer Science students are the Industries tomorrow's embedded systems designers and need to have practical experience very close to the real situation in the Industry.

As viewed from the academic perspective, the existing development kits, such as Altera EPXA10 Development Kit[2] and Xilinx HW-AFX-FG for complex FPGA based devices, suffer from two main disadvantages, namely very high cost and insufficient flexibility. Those two factors make them practically unusable for academic and small research/industry groups.

In order to satisfy the above needs (with a very limited budget in mind), we decided to initiate the DAMP project. Its main goal is to provide a low cost platform for the embedded systems specification and hardware-software co design, with the main focus on (mobile) multimedia applications. The Altera ARM based Excalibur device[1] with 400k gates around an ARM922T CPU[3] was chosen as the main chip. Our selection was based on the popularity of the ARM processor in many embedded devices, some of them targeting the 3G and 4G mobile devices with high expectations on multimedia applications. DAMP incorporates a rich set of possible interfaces usually present in multimedia mobile devices. An audio CODEC from Texas Instruments was implemented, which supports DVD quality 24-bit 96 kHz audio. DAMP also contains a VGA interface, which is capable of producing 256 colors. Furthermore a 10/100 Mbit Ethernet controller from Cirrus Logic was implemented as well as an IDE interface. DAMP supports up to 32MB of flash memory, which can be used for booting the ARM and programming the FPGA of the Excalibur device. In addition to the flash memory, DAMP also incorporates an SDRAM interface, capable of data rates of 133MHz and supports up to 512 MB. An extension interface was designed to support other future interfaces, e.g., Bluetooth or Home RF, making DAMP an open environment. A set of basic drivers, interfaces and FPGA configurations were designed to support application development on DAMP. In addition to the above DAMP provides a methodology for application mapping. Considering all of the above, DAMP offers a powerful, flexible yet low-cost development platform.

This paper is organized as follows. Section II discusses the initial design requirements and states the design restrictions. In Section III the different DAMP features and pros and cons about their potential implementations are discussed. The DAMP architecture is described in Section IV. The application mapping procedure is presented in Section V. Finally, conclusions and future work are described in Section VI.

II. Requirements and Restrictions

Before proceeding with the investigation of the DAMP organization, we would like first to highlight the main requirements and restrictions we took into consideration in our quest for the DAMP architecture.

DAMP is meant to be used as a platform for embedded systems development, mainly targeted on multimedia and mobile applications.

DAMP is supposed to provide a solid base needed to study modern chip design and testing techniques, and to build experimental multi-media applications. The DAMP board is not targeted for customer-ready end products with optimal cost and size.

The development board is intended to be used as a platform for embedded systems development, mainly targeted on multimedia and mobile applications. The following features were considered:

- High Quality Audio In/Out (stereo)
- Video Out
- Ethernet Controller
- USB v.2.0 Controller
- User I/O
- SDRAM Interface
- Daughter Card Interface
- Optional Altera Apex FPGA (socket)
- Connector to access Excalibur I/O Pins
- Flash Memory
- JTAG Interface
- UART Interface
- IDE Interface
- PS/2 Interface
- Power Circuit (external ATX power supply is to be used)
- Clock Circuit
- Reset Circuit

In order to keep the cost of the development board reasonable, the number of layers on the PCB (Printed Circuit Board) should be minimal and the use of buried vias is not allowed.

DAMP should work glue-lessly with QUARTUS II development software and Altera download cables.

DAMP is supposed to provide a natural Nios-to-Excalibur migration path for any customer developed 3.3 V extension boards. Customer Nios boards using 5 V are not covered by DAMP in its current definition.

III. Design Space Exploration

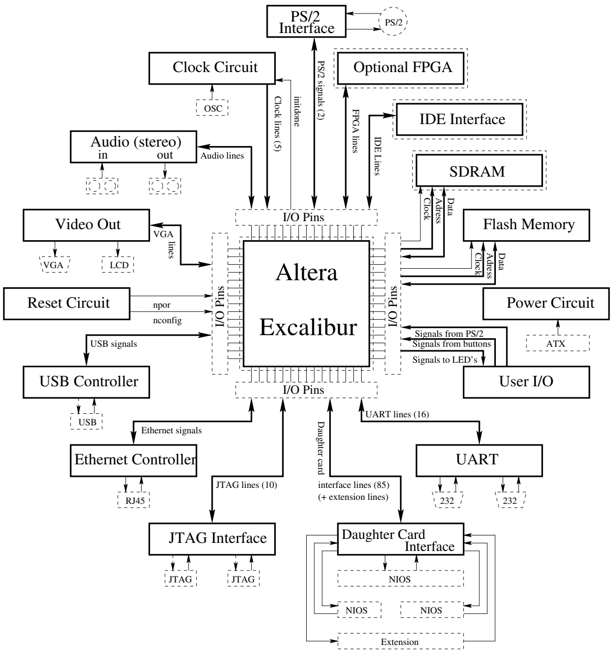

Fig. 1. Proposed features, requirements and signalflow of the development board.

In the next subsections, different implementations of the proposed features will be described. A short analysis is presented for each possibility in respect to cost and utilization of Excalibur resources.

The proposed features and the signalflow are shown schematically in Figure 1.

A. High Quality Audio In/Out

High quality audio input and output is needed in order to develop multimedia applications.
This can be supported via one of the following solutions:
1. Separate Analog to Digital Converters (ADCs) and Digital to Analog Converters (DACs) implemented on the development board.
2. ADCs and DACs implemented on a daughter card.
3. Audio CODEC implemented on the development board.
4. Audio CODEC implemented on a daughter card.

The advantage of the first two schemes is their simplicity. The audio can be processed by the ARM922T processor, if a simple driver is implemented in the FPGA, or it can directly be processed by the FPGA. This is, the FPGA can be used as a simple DSP. A disadvantage is the number of Excalibur device pins that will be used and the occupation of some FPGA space. The second option decreases the cost of the development board, as the audio daughter card becomes an optional extension to the DAMP mainboard, but uses a lot of daughter card interface pins.

An advantage of the last two options is that audio CODECs often feature some DSP functionality, thus more performance can be provided. Another advantage is that the number of required interface signals is very small, because the audio is encoded into a bitstream. A disadvantage of this is that the Excalibur device has to contain a driver which takes care of the communication protocol. This driver has to be implemented in the FPGA of the Excalibur device, which leaves less room for other functionality to the end user.

The fourth option decreases the cost of the development board, but will also use some of the daughter card interface pins.

B. Video Out

Regarding the video out feature, we considered the following options.
There are several options:
1. VGA connector on the development board, with a simple VGA controller implemented in the FPGA of the Excalibur device.
2. VGA connector on the development board, with an advanced VGA controller implemented in the FPGA of the Excalibur device. The outputs of the controller should be directed to external DACs.
3. Use a third party LCD controller on the development board.

An advantage of the first option is that a simple VGA controller has to be implemented in the FPGA of the Excalibur device. Therefore only eight data signals (Red, Green and Blue) and two synchronization signals are required. This solution gives the possibility to have up to 256 different colors. Another advantage is that a standard PC monitor can be used. A disadvantage here is that the VGA controller requires some FPGA space, which leaves less room for other functionality to the end user.

The main advantage of the second option is that it produces 24-bit colors on a standard PC monitor. However, it requires a lot of signals and occupies more space on the FPGA of the Excalibur device than the first option, which leaves even less room for other functionality. In addition external DACs will increase the cost significantly.

The last option is intended for LCDs. This means that a standard PC monitor cannot be used. The use of an LCD controller also increases the cost of the development board and it requires a lot of signals, but only a very simple driver has to be implemented on the FPGA of the Excalibur device, so little FPGA space is lost there, which is beneficial to the end user.

C. Ethernet Controller

The ethernet controller consists of a physical layer (PHY) and a protocol layer. There are different ways to implement the ethernet controller:
1. Use a third party PHY chip on the development board and implement the protocol layer on the FPGA of the Excalibur device.
2. Use a third party chip that contains both layers on the development board.
3. Use a third party chip that contains both layers on a daughter card.
4. Combination of the first and third option: Use a third party PHY chip on the development board and use a third party chip that contains both layers on a daughter card.

The first option is a low cost solution and offers a lot of flexibility. The user can create its own network interface or network controller for example. A disadvantage is that the controller implementation will occupy FPGA space on the Excalibur device, which leaves less room for other functionality to the end user.

The second and third option doesn't offer the flexibility of the first option, but occupies only little space for a driver on the FPGA of the Excalibur device, which is beneficial to the end user. An extra disadvantage is that a lot of the Excalibur device pins are used.

However, if one decides to add a controller with both layers to the address space of the Excalibur device, a driver is no longer needed. This gives the second option an advantage.

An advantage of the third option is that it decreases the cost of the development board. But it uses a lot of daughter card interface pins.

The fourth option is the most flexible, but has the disadvantages of the first and third option: it occupies a lot of FPGA space and uses a lot of daughter card interface pins, both leaving less space for other functionality to the end user.

D. USB v.2.0 Controller

USB v.2.0, like ethernet, can be split up into two distinct layers: a physical layer and a protocol layer. Furthermore, the USB controllers can be split in two different types, host controllers and device controllers. There are several options for the development board:
1. Use a third party USB v.2.0 host/device controller chip on the development board.
2. Use a third party USB v.2.0 transceiver chip on the development board and implement the protocol layer of the

USB controller in the FPGA of the Excalibur device.
3. Use a third party USB v.2.0 host/device controller device on a daughter card.

Both the first two options occupy a lot of Excalibur device pins. The first option occupies FPGA space for a driver, which leaves less space for other functionality, the second option contains an entire protocol layer, which decreases the free space of the FPGA even more. The second option is however, the most flexible solution of the two. Custom USB controller implementations are feasible.

The third option leaves it to the designer to implement a controller on the daughterboard (with or without the use of an FPGA). The advantage here is that the cost of the development board decreases. The disadvantage is that it occupies the daughter card interface.

E. User I/O

The board should contain buttons, jumpers, dipswitches and LEDs to feed information to and to receive information from the Excalibur device. There is also the possibility of using a touchscreen controller, which is integrated in some audio CODECs.

F. SDRAM Interface

Random access memory is needed to run programs on the ARM processor and to store data. The EPXA4 Excalibur device has 128 KB SRAM and 64 KB DPRAM onboard, however, in many practical situations this amount of memory will not suffice. Therefore external memory is required. The Excalibur device features an SDRAM controller that is supporting single data rate (SDR, 133 MHz) memory and double data rate (DDR, 266 MHz) memory.
The considered options are:
1. SDR memory mounted directly on the development board.
2. DDR memory mounted directly on the development board.
3. DIMM sockets mounted directly on the development board.
4. Combination of all options: the use of memory and sockets on the development board.

The first two options have the advantage that there is always memory present, but increases the cost of the development board. A major weakness for both is that the memory size cannot be varied. The use of sockets will solve this problem, which gives the third option a significant advantage.

The last option has all the advantages of the other options, but is also the most expensive one.

G. Daughter Card Interface

As stated in Section II, one of the requirements is that the daughter card interface should be pin compatible with the daughter card interface of the Altera Nios development kit. However, only the 3.3 V interface will be supported. This offers an easy design migration path from the Nios development board to the Excalibur development board. The interface consists of three headers: 14, 20 and 40 pins wide. In addition to the Nios-interface, the user can define his own external interface, which can be any arbitrary size.

H. Optional Altera Apex FPGA

A very flexible way of adding extra functionality to the DAMP is to use an extra FPGA. This FPGA can contain any required functionality. There are several options:
1. Altera Apex FPGA used on the development board.
2. Altera Apex FPGA used on a daughter card.
3. FPGA socket for an Altera Apex FPGA on the development board.

The first option is going to make the development board very flexible, but increases the cost of the development board. The cost of the development board can be reduced by the second option, but this option increases the utilization of daughter card interface pins. The last option lets the user decide whether or not to use an extra FPGA, so the cost of the development board increases only by a small amount.

I. Connector to access Excalibur I/O Pins

To be as flexible as possible, we would like to read, write and monitor the pin values of the Excalibur device. Therefore we would like to implement some pin headers around the device. There are two options:
1. Connect every device pin to a corresponding test pin.
2. Use a selection of the most interesting device pins and connect those device pins to pin headers.

The first option offers the most flexibility, but requires a lot of pin headers and will make the size of the headers very big, compared to the size of the Excalibur device. This is solved by the second option, but this option only gives a subset of the Excalibur device pins.

J. Flash Memory

The Altera Excalibur device is capable of booting from external program memory. Altera offers tools to program flash memory by means of the JTAG interface. Therefore the development board should incorporate flash memory. The Altera Excalibur device is capable of handling four external flash memory devices, each up to 32 MB. Therefore the development board has to support four flash memory devices. The end user can choose whether he/she will implement one, two or four devices on his board.

K. JTAG Interface

The JTAG interface is used for testing, programming and debugging the Excalibur device or the peripheral devices. There are two JTAG channels available on the Excalibur device. One of them is connected directly to the ARM922T processor. The other one is connected to the FPGA of the Excalibur device. Therefore two JTAG connectors will have to be used.

L. UART Interface

The Excalibur device has support for one UART interface, which can be used to interact with a terminal program running on a PC. Such a program can display program information on the monitor of the PC. It is also possible to implement more than 1 UART interface, but support for the additional interface must be programmed in the Excalibur FPGA.

M. IDE Interface

Support for CD-ROM players/recorders and harddisks will become available when an IDE interface is implemented. This interface can be made possible by adding a 40-pins header on the development board and implementing a protocol in the FPGA of the Excalibur device.

N. PS/2 Interface

Most computers use the PS/2 interface to connect to a mouse or keyboard. Support for this interface can be given by placing a PS/2 connector on the development board and by implementing a driver in the FPGA of the Excalibur device.

O. Power, Clock and Reset Circuit

The Excalibur device cannot operate properly without proper power, clock and reset signals. Thus circuits have to be implemented that provide these signals. The power circuit will be based on an external ATX power supply. Therefore an ATX connector is implemented on the development board.

The Excalibur requires a 1.8 V level for its core and a 3.3 V level for its I/O. When the power supply is turned on and the 3.3 V output reaches the required level, the Excalibur reset signal will be generated.

The Excalibur device features four clock inputs and one reference clock input. The clock circuit, which is implemented on the development board, generates these clock signals and offers the possibility to use an external clock. One switch of the User I/O-part will be used for the selection of an onboard reference clock signal or an external clock signal, which can come from an external source.

IV. DAMP Architecture

Based on the discussion in Section III we decided to incorporate the following features:

- High Quality Audio In/Out (stereo)
- Video Out
- Ethernet Controller
- User I/O
- SDRAM Interface
- Daughter Card Interface
- Connector to access Excalibur I/O Pins
- Flash Memory
- JTAG Interface
- UART Interface
- IDE Interface
- PS/2 Interface
- Power Circuit (external ATX power supply is to be used)
- Clock Circuit
- Reset Circuit

We decided to implement a Texas Instruments TLC320AD77C CODEC[5] on the development board, because of the low cost and high-quality. Furthermore it doesn't occupy much Excalibur I/O pins. An FPGA driver was developed to support the use of audio.

For video support, we decided to incorporate a VGA connector on the development board and have implemented a simple VGA controller on the FPGA of the Excalibur device. We have chosen this solution, because of the low cost and because a simple VGA driver occupies only a small amount of FPGA space, which leaves more space available for other functionality to the end user.

A Cirrus Logic CS8900A[4] controller, incorporating both ethernet layers, was implemented on the development board, because of its low cost. We decided to connect the ethernet controller through the address space of the Excalibur device, because then a driver in the FPGA space is not needed and a remote (via network) FPGA reconfiguration is possible.

We decided to add one 8-dipswitch header and four low-active press buttons for input, and two 7-segments LED displays for output.

Because of the low cost and its high flexibility, we decided to mount a DIMM socket on the development board in order to support SDRAM.

One of the requirements is that a Nios development board compatible daughter card interface is present. DAMP only supports the 3.3 V interface. We decided to add an extension to provide more support to the developer of daughter cards. This extension is a 2x20 pins header.

To offer high flexibility and design testability, we decided to implement a one-to-one connection from each Excalibur I/O pin to a test pin. To support this, 12 2x20 pins headers and 4 2x24 pins headers were used on the development board.

Support for up to four 4M x 16 bit AMD AM29LV640 flash memory devices is provided on the development board in order to test the functionality of the development board.

Because one of the requirements is that the board should be compatible with Altera download cables, we decided to mount a MasterBlaster compatible connector on the development board, which is connected to the Excalibur device. In order to support the debug the ARM CPU, we decided to mount a JTAG connector, which is directly connected to the JTAG interface of the ARM core.

We decided to mount one RS-232 connector on the development board, which is connected through a MAX232 compatible level convertor to the Excalibur device.

In order to provide support to IDE compatible devices, we mounted a 2x20 pins header on the development board and we have written a driver in the FPGA space.

We decided to mount a PS/2 connector on the development board in order to support a mouse or PC keyboard. A PS/2 driver was written in the FPGA space.

The Power Circuit implements the ATX power connector and the circuit which generates the different voltages needed on the board. The Clock Circuit consists of the oscillator(s) and components for providing the clock frequencies. The Reset Circuit consists of a power-on reset circuit, which generates the proper reset to the Excalibur device after power is up.

V. Application Mapping Methodology

The designer who wants to port a specific application to the DAMP has first to check if the development board meets all the requirements for the application, for example to check if it meets all the in/output requirements. After the check is performed and the actual application porting to the DAMP is done, the designer then can try to optimize his design specifically for DAMP. For example the designer can use Apex FPGAs to accelerate specific functions via a hardware-software co design strategy, partially supplied by the Quartus II design software. We will discuss what the designer has to do for each feature on the DAMP to be able to use it.

The Audio CODEC hardware driver decodes the bitstream from the CODEC and also provides an interface to the ARM Processor. This driver is provided by DAMP, hence the designer only has to develop a software driver which reads/writes data to the DAMP interface.

The designer has to develop his own Video Out Controller in the FPGA, written in HDL. A general driver which implements an interface between the ARM processor and the FPGA is provided by DAMP.

The designer doesn't have to develop a driver for the ethernet controller because the ethernet controller is connected to the Expansion Bus Interface, which is mapped into memory space.

The User I/O can be accessed by the ARM processor through a hardware driver, provided by DAMP. The designer only has to develop a software driver which reads/writes data to the interface.

The designer doesn't have to develop a driver for the SDRAM, because the SDRAM controller is directly mapped into memory space.

The designer doesn't have to develop a driver for the flash memory, because it is connected to the Expansion Bus Interface, which is mapped into memory space. However, if the flash memory is to be used as a program storage place, the designer should develop his own driver in order to write to flash.

A hardware driver which implements an interface between the ARM processor and the Daughter Card Interface is provided by DAMP. The designer only has to write a software driver which reads/writes data to the interface.

There are two JTAG Interfaces available, both of them already have a driver, provided by Altera.

The designer doesn't have to develop a driver for the UART interface because the UART is connected to the UART Interface which is implemented on the Altera Excalibur device. The interface is mapped into memory space.

The hardware driver that delivers IDE-support provides an interface to the ARM processor. The designer only has to develop a software driver which reads/writes data to the interface.

A hardware driver which implements an interface between the ARM processor and the PS/2 connector is provided by DAMP. The designer only has to develop a software driver which reads/writes data to this interface.

VI. Conclusions and Future Work

In this paper we discussed the DAMP (Delft Altera-based Multimedia Platform) architecture. DAMP's main goal is to provide a low cost platform for the embedded systems specification and hardware-software co design, with the main focus on (mobile) multimedia applications. This platform is based on the Altera Excalibur device incorporating a 400k gates FPGA and a state of the art ARM922T processor. In addition, 32MB flash and up to 512MB SDRAM memories are supported to provide sufficient application design space. Wide range of peripheral interfaces, such as ethernet, PS/2, IDE, 24-bit audio in/out and video out are also supported by DAMP. Additionally DAMP provides a methodology for application mapping. Future work can incorporate porting Linux, developing daughter cards featuring Bluetooth, Home RF, WLAN, etc.

REFERENCES

[1] Altera Corporation. *ARM-Based Embedded Processor PLDs Hardware Reference Manual 2.0*, 2002.

[2] Altera Corporation. *EPXA10 Development Kit Getting Started - User Guide*, 2002.

[3] ARM Limited. *ARM922T Technical Reference Manual rev 0*, 2000.

[4] Cirrus Logic. *CS8900A - Product Data Sheet*, 2001.

[5] Texas Instruments. *TLC320AD77C 24-bit 96 KHz Stereo Audio Codec - Data Manual*, 1999.