MicroBlitz Overview

Harry H. Porter III

HHPorter3@gmail.com

14 December 2023

MicroBlitz is a hardware implementation of the Blitz-64 processor core, written in SystemVerilog.

MicroBlitz Features

SystemVerilog

MicroBlitz consists of several files written in **SystemVerilog**. Together, these files implement a **Blitz-64 processor core**.

Open Source

MicroBlitz is **free and and open**. It can be modified and used by anyone.

Blitz-64 Architecture

MicroBlitz implements the **full Instruction Set Architecture** of the Blitz-64 processor core. In Blitz-64, all registers are 64 bits wide and the design embraces 64 bit operations as the default.

Intel Quartus Prime

MicroBlitz was developed with the **Intel Quartus Prime software (Lite version)**, which includes **ModelSim**. Presumably it will compile and function in other environments.

FPGA

MicroBlitz was developed and tested on an FPGA. Specifically, it was developed on the **"Cyclone V GX Starter Kit" development board**. In addition to the **"Altera Cyclone V GX" FPGA**, this board also contains a number of communications ports, buttons, switches, LEDs, memory devices, etc.

Lines of Code

The primary SystemVerilog files are **approximately 6,000 lines of SystemVerilog code**, including comments. In addition, there are several support files, such as **MBBooter.s** (the bootstrap program which is loaded into ROM) and files related to testing and verification.

Pipelined Design

The MicroBlitz core is **a pipelined architecture** in which each instruction requires several pipeline stages to complete. Generally speaking, one instruction is completed per cycle. However, instructions that access memory and instructions that branch require additional cycles for memory accesses. The MMU will introduce additional cycles when the page table must be walked.

Memory Management Unit

The MicroBlitz MMU implements page tables as described by the Blitz-64 architecture spec. **Page table walks occur in hardware**. The MMU implements **Translation Lookaside Buffers** (TLBs) in hardware. Instructions are loaded into an **instruction prefetch buffer**.

Execution Modes

At any time, the MicroBlitz core is either in **Kernel Mode** or **User Mode**, to facilitate the implementation of Unix-like OS kernels. The instructions **syscall** and **sysret** are used to invoke the kernel and return to userland code.

Exceptions and Interrupts

When error conditions arise during instruction execution or interrupts from devices occur, **trap processing** will occur, per the Blitz-64 architecture spec.

Timer Interrupts

Time-slicing is necessary for any **multitasking OS kernel**. MicroBlitz supports this.

Serial Communication

MicroBlitz accommodates communication using the **UART protocol** (Universal Asynchronous Receiver-Transmitter).

Digital I/O

MicroBlitz accommodates **input and output on digital pins**. Digital output is also shown in hex on the 4 digit, seven-segment LED display on the board.

Emulated Instruction Capability

In the Blitz-64 spec, the implementation of some instructions is optional. The ISA specifies exactly how **emulated instructions are trapped and handled**. For example, **floating point instructions are not implemented** in hardware. Instead, their use causes an “Emulated Instruction Exception”, which allows the operation to be performed in software.

Startup Sequence

After pressing the RESET button, the MicroBlitz core will begin executing a **boot loader program in ROM**, as mandated in the spec. The boot loader program is assembled using the Blitz assembler. From this, there is a Blitz tool that automatically produces SystemVerilog code, which is included directly into MicroBlitz SV code. Thus, the boot loader ROM is downloaded with the rest of the core during FPGA initialization. **The target program** to be executed will be **compiled, assembled, and linked separately on a laptop**. When RESET is pressed, the boot loader will execute and will download a target program (such as an OS kernel) from the laptop via a serial UART connection to the laptop. Finally, the boot loader will jump to the target program and execution begins.

KPL Capability

The MicroBlitz core can execute **programs written in the KPL** (Kernel Program Language), which is part of the Blitz-64 project. Such a program will call on a number of library and support functions, such as the code required for printf. The **KPL support environment** is downloaded to the FPGA along with the target program and is used during program execution.

Future Development

MicroBlitz currently lacks the following features which are planned for the future:

- Multiple Cores
- Interface to DDRAM
- Interface to SPI, I^2C, USB
- Other Memory-Mapped I/O devices, such as MicroSD cards
- Compressed Instruction Set

You might be able to contribute toward these goals.

Additional Documentation

The document **"MicroBlitz: A SystemVerilog Implementation of Blitz-64"** describes the design, implementation, and use of the system in detail. Also, a video series is under consideration.