FPGA Design - Creating an Intel 8086 Microprocessor with Modern Technology

RUTGERS
THE STATE UNIVERSITY OF NEW JERSEY

Independent Study by: Elie Rosen

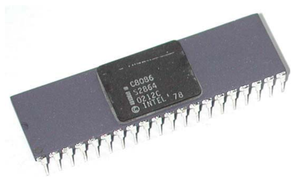
Advisor: Dr. Michael Caggiano

Objective

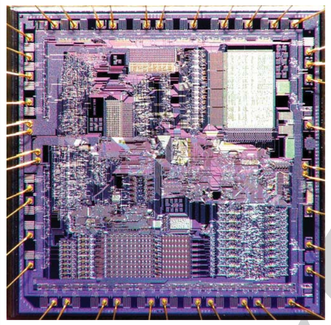
The primary goal of this project is to re-create an Intel 8086 Microprocessor based on an open source implementation of the processor's hardware description language. In order to accomplish this, it was necessary to: 1) create and embed a BIOS into static ROM and 2) load the compiled hardware onto the FPGA device. Once completed, an x86-based operating system, such as MS-DOS 6.22, was installed as a proof-of-concept demonstrating that the implemented processor worked.

Intel 8086

- First produced in 1978.
- The first official 16-bit processor capable of supporting up to a revolutionary 1 megabyte of memory and 64 kilobytes of Input / Output data (I/0).



- Backwards compatable with 8-bit applications to encourage corporations to switch to the new technology.
- Paved the way to future computing with the introduction of x86 architecture.

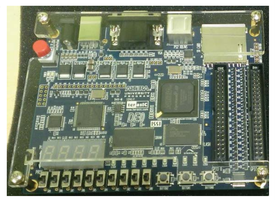


Intel 8086 silicon die - the hardware for this design features over 20,000 transistors

References:
[1] Altera Corporation. (2012, Aug.) Cyclone III device handbook. [Online]. Available: http://www.altera.com/literature/hb/cyc3/cyclone3handbook.pdf
[2] Z. G. Marmolejo. (2011, Feb.) Zet Processor - Source Code. [Online]. Available: https://github.com/marmolejo/zet
[3] Open Watcom. (2010, Jun.) Welcome to Open Watcom. [Online]. Available: http://www.openwatcom.org/index.php/Main Page
[4] M. Toy and G. Wichman, "Rogue (video game)," 1980.
[5] Altera Corporation. (2012, Aug.) Altera University Program - Learning Through Innovation. [Online]. Available: http://www.altera.com/education/univ/unv-index.html

FPGA Hardware

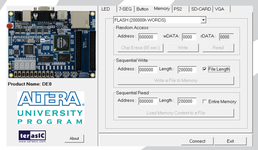
- The field-programmable gate array consists of over 10,000 individual logic slices that can be programmed to act as individual low level logic devices and gates.
- By configuring the logic elements to work together, large and complex devices can be created (for example, an Intel 8086 microprocessor).



Altera DE0 FPGA Development Board

Procedure

- Compile Basic Input/Output System (BIOS)
  - Performed with OpenWatcom compiler, the BIOS tells the processor where to look for further instruction when turning on.
  - Provides necessary drivers for running hardware on the board such as the SD slot which acts like a floppy disk drive.
- Mount operating system to SD Flash Memory
  - MS-DOS 6.22 was placed byte-by-byte onto a 128 megabyte SD flash memory card to allow a visual means to determine that the processor works.
- Compile hardware description for the Intel 8086
  - The x86 16-bit based processor was compiled using the Quartus II IDE for Verilog; this included all of the necessary components to create the processor such as: the ALU, Register Memory, VGA Drivers, and PS2 Device Drivers.
- Load hardware to FPGA
  - The compiled hardware description was then loaded onto the FPGA by using Active Serial programming which saves the description to the on-board flash memory.
- Mount BIOS to FPGA in static ROM
  - Once all of these steps were completed, the BIOS was inserted into a static ROM which was created on the device at memory location 0x00000.



DE0 memory flashing software

MS-DOS

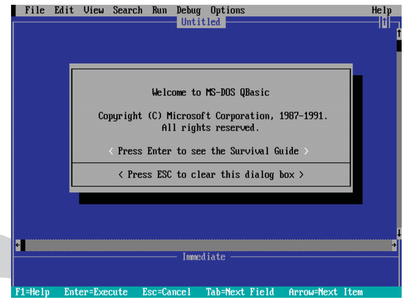
- MS-DOS initially released in 1981.
- It was designed specifically to work with the x86 instruction set used by the Intel 8086 microprocessor.
- The operating system came prepackaged with software which allowed users to develop their own programs in the BASIC language.

FPGA configured to behave like an Intel 8086 microprocessor demonstrating MS-DOS 6.22

- The release of MS-DOS brought foward a new generation of PC gaming. With the addition of better graphics drivers and support for additional memory, popular games such as Doom and Rogue changed the PC into a respected platform for doing more than just work and computations.

Rogue, a popular dungeon game created by Michael Toy and Glenn Wichman

QBASIC, a prepackaged IDE and interpreter for the BASIC programming language.



Acknowledgements

This project would not have been possible if it were not for the Altera University Program which provided both an Altera DE0 and DE0-Nano FPGA Development boards.