A Senior-level Computer Hardware Organization Course: Designing a Single Board Computer

James M. Conrad and Luke M. Hassell
University of Arkansas, Computer Systems Engineering
(Current contact info: jmconrad@uncc.edu)

Abstract

This paper describes a course at the University of Arkansas with the following catalog description: CSEG/ELEG 4983: Computer Hardware Organization. Design of a complete single board computer including basic hardware organization, memory subsystem design, peripheral interfacing, DMA control, interrupt control, and bus organization.

Introduction

A Senior-level course at the University of Arkansas provides a current yet inexpensive method to teach computer hardware design. The course, Computer Hardware Organization, is cross-listed between the Electrical (ELEG) and Computer Systems (CSEG) Engineering Departments. The major objective of the course is to acquaint the student with basic computer hardware organization and to provide hands-on experience in designing a single board computer. The major vehicle for learning is based on ten laboratory experiments. Each lab requires initial specification of the problem and a proposed design of the solution (prelab). Students design and simulate their solution using accurate timing models of components (Smart Models). Students write a report for each lab justifying their design decisions and describing their experiences with the design and simulation process.

Prerequisite Knowledge

Students entering the class will have taken an ELEG or CSEG digital systems design and an ELEG microprocessor organization course.

Textbook and Lab Manuals

Since this course uses the Motorola 68000 as a base of the single board computer, students are restricted to a book with a detailed hardware design built around this chip. Two books in particular have some good sections, but each books does not provide all the material needed. These books are:

- *Microprocessor Systems Design: 68000 Hardware, Software, and Interfacing*, Alan Clements, PWS-Kent, Boston, MA, 1992.
- *The 68000 Microprocessor, Hardware and Software Applications*, James L. Antonakos, second edition, Merril, 1990.

The books are optional and available in the campus bookstore. Students often pair up and purchase one book between them.

Two references are required: the Motorola 68000 User's Manual, and the CSEG/ELEG 4983 Lab Procedures and Exercises Manual. The Lab Manual contains the specifications for each of the lab exercises, CAD tools information, and helpful hints.

Four optional references include:

- *CSEG/ELEG 4983 Data Sheets (74 series)*. This contains copies of selected data sheets of 74ALS and 74LS components.
- *CSEG/ELEG 4983 Data Sheets (Non-74 series).* This contains copies of selected data sheets for ROM, RAM, Direct Memory Access (DMA), and interface components.
- *Motorola 68000 Programmers Reference Manual.*
- *An Introduction to UNIX and the vi Editor,* an internally-developed guide for those not familiar with UNIX.

Design Tools and Lab Equipment

The hardware platforms used are SUN SPARCstations. The front-end software is Viewlogic's PowerView (ViewDraw and ViewSim), with device models provided by LogicModeling's Smart Models.

One lab requires building a small computer board. The equipment required for this lab is typical of a hardware lab (soldering irons, tools, etc.).

Lab Exercises

Students design a single board Motorola 68000-based system throughout the semester using PowerView. Ten exercises require students to understand topics discussed in class and presented in the book. Students work in two-person teams to complete the assignments.

The lab exercises are listed below.

Lab 1 - Oscillator Circuit

The first lab is purposely simple, because most time is spend ensuring that the design tools can be executed. Some students will have run PowerView in previous classes, but have never accessed the new parts libraries necessary for this class. The objective is to build a simple oscillator circuit consisting of an oscillator component and two inverters. The oscillator requires that the students define the speed as specified in a Unix ``.mem" file. Students must ensure that the inverters exhibit a small delay consistent with the actual delay of the physical device. An example of a lab schematic is shown in Figure 1. An example of the wave trace for the schematic is shown in Figure 2.

Lab 2 - Memory Decode

The purpose of the second lab is to design and simulate the decode logic needed to access memory of the single board computer. Students determine whether they want to use full decoding, partial decoding, logic gates exclusively, or logic gates and decoders (74ALS138). Students are expected to minimize the number of logic levels needed to implement their decode, and again ensure the components exhibit proper propagation delays. An example of the wave trace for the decode schematic is shown in Figure 3.

A simple introduction to VHDL with combinatorial logic is planned for future semesters, with the results to be programmed into a PAL, FPGA, or other programmable component.

Lab 3 - Memory Subsystem

In this lab students design and simulate the entire memory subsystem, including all data, address, and control lines. Students simulate the bus cycle of the 68000 processor to ensure the decode, memory access, and data acknowledge functions operate properly. Students ``load" ROM and RAM chips with a Unix ``.mem" file. The decode logic from Lab 2 is used for this lab, and students are urged to make the Lab 2 circuit into an individual ``component," or symbol, to make their designs more readable.

Lab 4 - 68000 Processor Initialization

Lab 4 requires the students to combine the designs from Labs 1 through 3 and add some additional logic to make a working Motorola 68000-based design. In past labs, students provided specific signals to test their circuits. In this lab, students need only worry about the HALT and RESET lines or the 68000 processor. They are required to write a short assembly language program, assemble it, and create the Unix ``.mem" files to load the ROM and RAM chips. The entire design then runs based on the code ``burned" into the ROM and RAM components. An example of a lab schematic is shown in Figure 4.

Lab 5 - Building a Simple Circuit Board

One comment from students is that throughout the CSEG (and sometimes the ELEG) curriculum they have little hands-on activity with real hardware. Lab 5 gives students a chance to build a simple printed circuit board. The board only has one resistor pack, four components, and two connectors, so it is not too complex. Students are required to drill the PCB, solder the chips and connectors on the board, and test their work using a PC and a bank of LEDs. They are required to write code on the PC to exercise the board and verify its function. Lab 5 also gives students a brief break from using the CAD tools.

Lab 6 - I/O and Interfacing

Lab 6 introduces students to the concepts of interrupt procession. The circuit designed for Lab 4 is modified to add a priority encoder for the 68000 interrupt lines IPL2-0. More effort is spent writing and simulating interrupt service routines than on hardware design. Students are required to ensure the correct operation of their circuit when separate level 3 and 5 interrupt signals are received, and when the signals are received concurrently.

Lab 7 - I/O Device (6821 Parallel Port)

This lab builds on Lab 6 progress. Students add a Motorola 6821 Peripheral Interface Adapter device to their design, as well as additional decode circuitry for the 6821. Students then simulate inputs from the two parallel ports of the 6821 separately and then concurrently.

Lab 8 - DMA (68440)

Lab 8 is the final 68000-based exercise. Students add a Motorola 68440 Direct Memory Access (DMA) chip to their existing circuit. They need to create the error-checking circuitry, and write a new interrupt service routine. They demonstrate the correctness of their design by simulating a memory-to-memory DMA transfer of 16 bytes, including an interrupt to the processor when the transfer is complete. Again, much of the effort of this assignment is devoted to programming the I/O device. An example of a lab schematic is shown in Figure 5.

Lab 9 - 68020 Processor Initialization

Lab 9 provides students with the last chance to use the design tool, though many students indicate they will not miss their CAD experiences! They demonstrate their grasp of the material learned in the class by implementing a processor and memory subsystem design using the Motorola 68020 processor. Students must design the decode circuitry and allow the access of one, two, three, and four bytes of data. Alternate decode methods, like a ROM, are examined.

Lab 10 - System Cost

Since engineering entails design based on a reasonable cost, the last lab examines the price for a 68000 system board. Students are required to generate a parts list and look up prices for these parts. They also make an estimate of the printed circuit board size based on the size of the components. The printed circuit board, assembly, and testing costs are based on specifications from actual companies. Students see how certain components that were over-specified greatly increase the cost of the entire system.

Assessment/Grading

There are several ways of assessing student performance and the classroom environment. First, each student group submits a report for each lab exercise. Students are specifically asked to address the question ``What did you learn" in their report, and are asked to write more than just ``I learned how to use the tool." The level of learning is judged by assessing if the student was able to formulate a solution to the problem and report how and why they choose their particular solution. This demonstrates their critical thinking skills.

The results of labs, weekly quizzes, and midterm and final exams demonstrate if students have a good mastery of the material. Weekly evaluation essays and the midterm evaluation help convey the mood of the class.

Evaluation

This course is well received by the students, though there still are some problems. The most frequent negative comment from students is the frustration with the PowerView design package. This problem has been addressed by compiling a ``List of Hints," as written by students from previous semesters. This includes suggestions related to the PowerView tool, as well as general suggestions like "start early."

Another negative comment from students is related to poor Engineering Computer Network performance. The current network carries too much traffic and small hardware problems cripple performance. These problems are compounded the night before a lab is due, when most students complete the assignments. This problem has been addressed by requiring a prelab report be turned in one week before the final report. The prelab report requires some design be performed, and helps distribute the assignment design effort across several days. Network hardware solutions are currently being implemented.

The Course from a Student's Point-of-View

This course is deemed "an excellent class" by the student co-author. Creating a single board computer from scratch for the first time is no easy experience. It requires large amounts of time and is a problem cited by many students who have taken the class. However, whether students realize it or not, the amount of time placed in conceptualization, design, and simulation greatly benefits their overall engineering abilities and performance. With enough effort, the student realizes a great sense of accomplishment and pride once the course is completed.

There are limitations in the simulation package, PowerView. Students must spend a lot of time discovering these limitations for themselves. This most often results in the loss of time. Since this is a software usage problem and not related to hardware design, these limitations are documented and distributed to the next semester's class.

The course material is well planned. It has a proper level of difficulty for a first time computer hardware design class. Students are enlightened as to how a computer is built and how instructions affect the system, from a physical point of view. This allows the student to link information from previous classes together so that the entire undergraduate program comes to focus.

Conclusions

This course gives students a good, first-hand opportunity to design hardware and work in a team environment. The design software and lab environment is very inexpensive and easy to use. The course is well received by the students and is an enjoyable to teach.

Acknowledgement

The author would like to thank Professor David Andrews at the University of Arkansas, Electrical Engineering Department, for use of the materials he developed when this class was first offered.

Schematics and waveforms were developed by Luke Hassell during the Fall Semester, 1994.

This paper was reset into MS Word and pdf in May, 2003. The content is unchanged, except for contact information. The primary author can now be reached at jmconrad@uncc.edu.

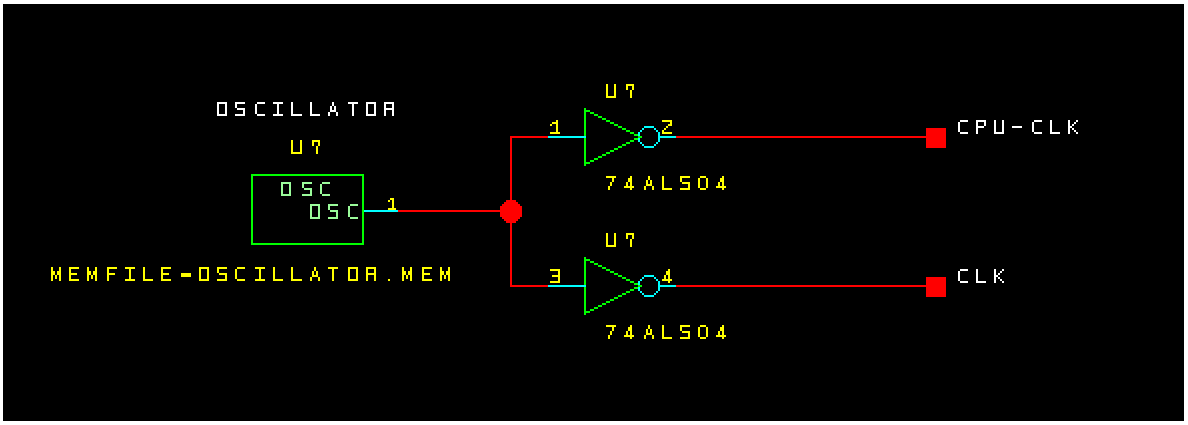


Figure 1: Lab 1 - Oscillator Circuit

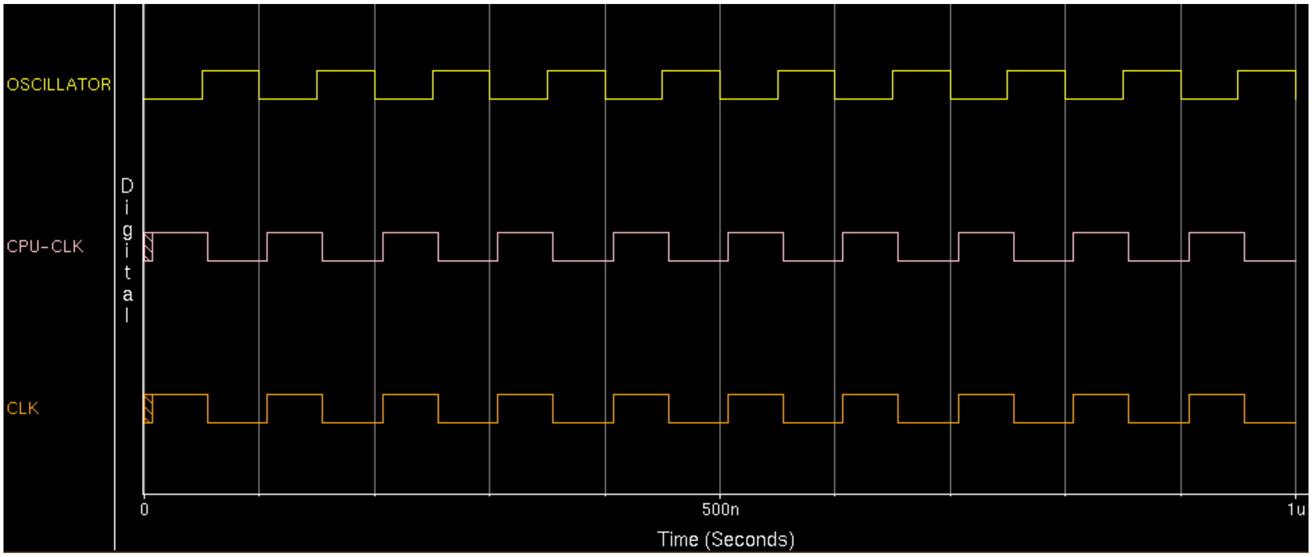


Figure 2: Lab 1 - Oscillator Circuit Wave Trace

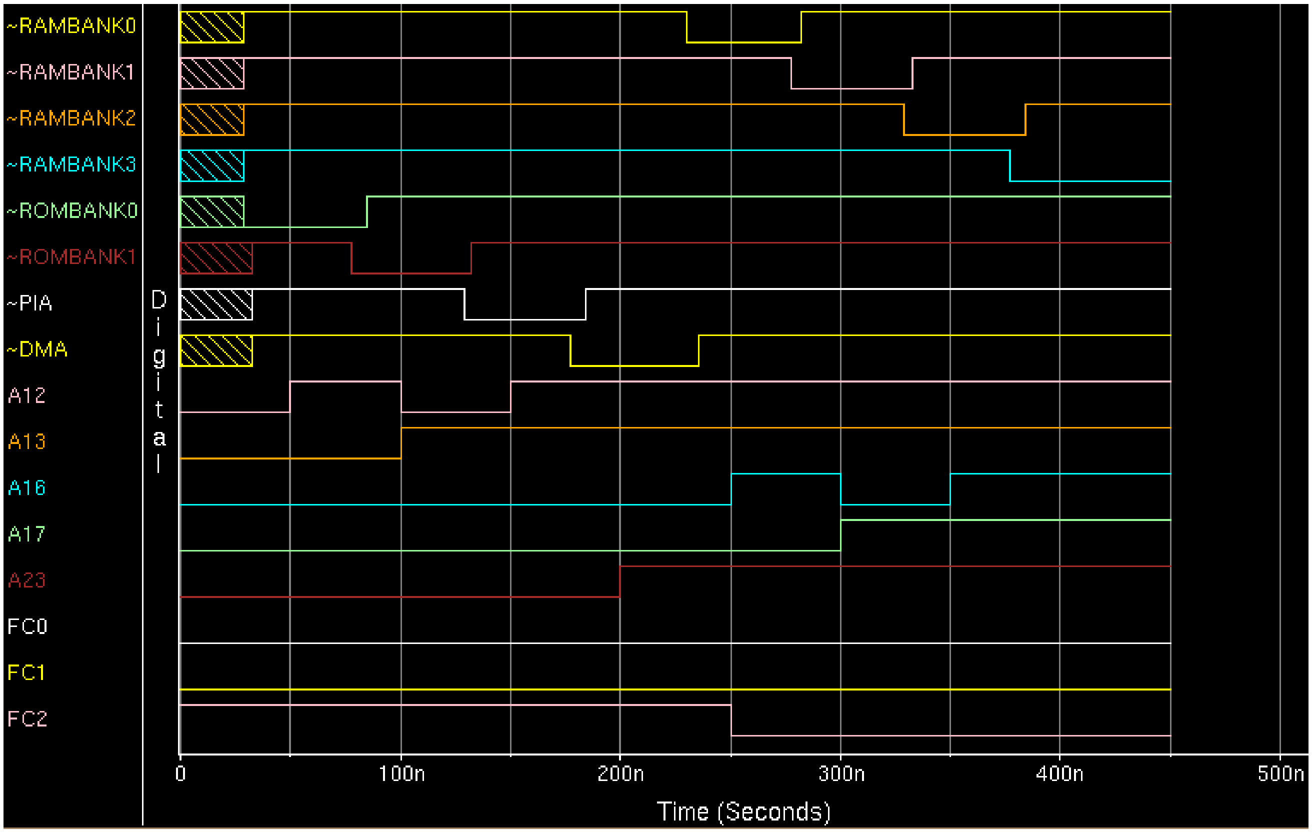
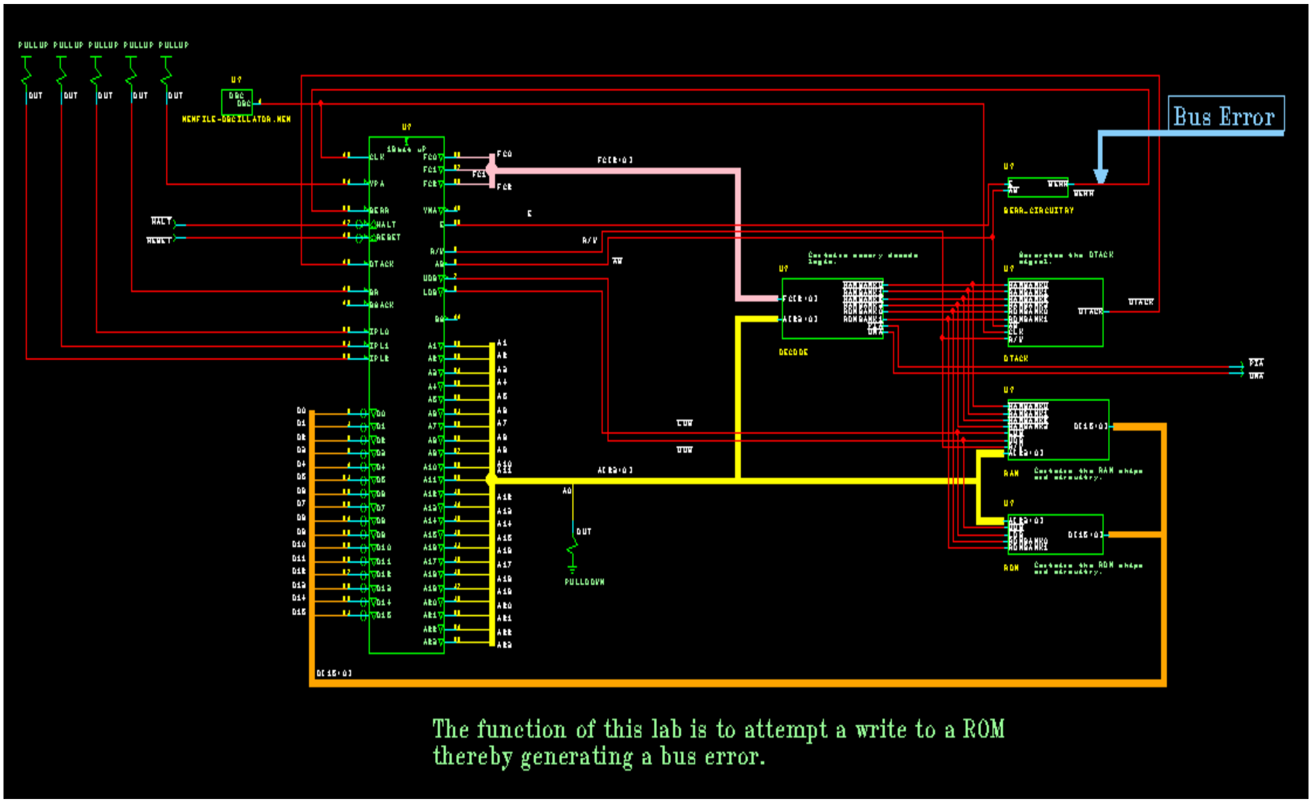


Figure 3: Lab 2 - Memory Decode Circuit Wave Trace



Bus Error

The function of this lab is to attempt a write to a ROM thereby generating a bus error.

Figure 4: Lab 4 - 68000 Processor Initialization

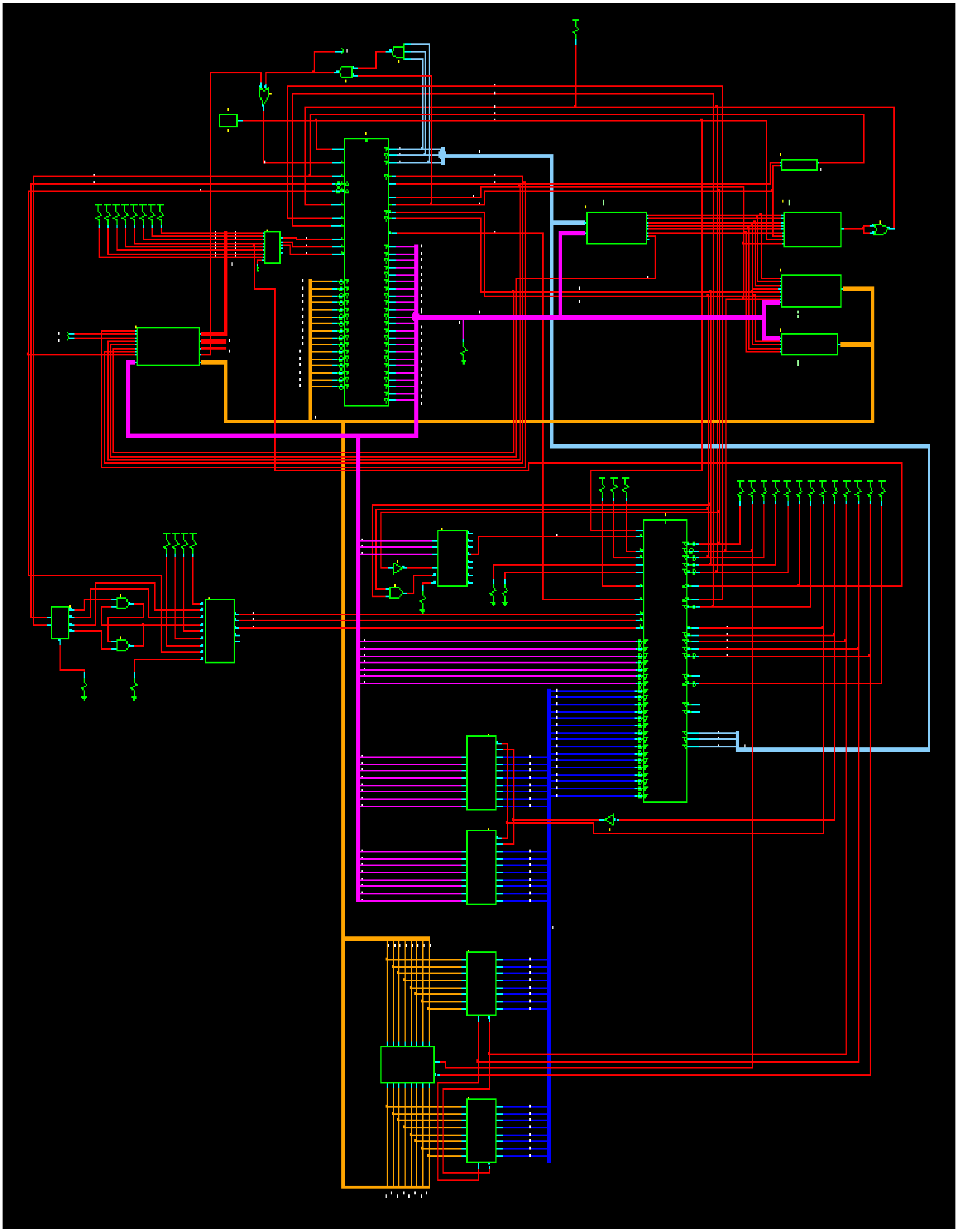


Figure 5: Lab 8 - DMA using the 68440 - Final Full System