Visualization and Computer Aided Design Techniques for Teaching Computer Hardware Design Course

J. Pang[1]
[1]Department of Electrical and Electronic Engineering and Program of Computer Engineering, California State University, Sacramento, Sacramento, CA, U.S.A.

Abstract - *It is essential for undergraduate computer engineering students to have thorough understanding of the computer internal architecture. However, the study of computer hardware usually involves with a lot of abstract concepts, complex hardware structural interconnections and also dynamic hardware behavior. In order to invigorate students in efficient and active learning, effective technology integration with classroom instruction is required. This paper proposes three different visualization techniques for active learning including circuit diagram visualization, waveform visualization, and signal flow visualization. In addition, this paper presents how to use computer aided design method such as using a free open source logic design tool Logisim to assign projects to students so that they are able to integrate course contents, enhance problem solving skills, and also engage in active learning.*

Keywords: Active learning, computer engineering education, visualization, computer aided design

1 Introduction

Understanding of the internal architecture of microcomputers is the basis for undergraduate computer engineering students to obtain in-depth knowledge of the design and application of computers. Traditionally, real digital hardware was used inside classroom for teaching microcomputers [1][2][3] and it was helpful for students to understand usage of microcomputers [4][5][6]. However, usually real digital hardware and associated software tools are expensive [7]. Moreover, it is time consuming for students to conduct experiments which require wiring components together. Especially, it is frustrating for students to trace problems when there are hardware related errors, such as broken wires, wrong connections, stray inductance and capacitance, power bus noise, and so on. This may also distract students from efficient understanding of the internal architecture of microcomputers. A separate and dedicated electrical engineering circuit class is necessary for students to focus on electronic circuit issues.

Active learning strategies using visualization and logic design tools have been used by author of this paper to teach computer hardware design course to undergraduate computer engineering students at CSUS. The author has observed that when students study course materials, it is hard for them to associate a lot of static texts and abstract concepts with complex internal connections of microcomputers. It is also hard for them to imagine how digital signals propagate inside microcomputers to operate different functional units. Representing information into graphic format through visualization techniques, and also using computer aided logic design tools for projects can concisely demonstrate microcomputer architecture related ideas. It is beneficial for students to enhance their involvement and comprehension of course materials, to reduce stress, to get motivated and also to have better insight into the internal architecture of microcomputers

This paper looks at the roles and implementation details of using visualization techniques and computer aided logic design tool for teaching computer hardware design course. Part 2 describes how to use multiple visualization techniques to represent information inside classroom. Part 3 presents examples of using Logisim tool [8] for student projects which help them study the internal architecture of the microcomputer. Part 4 draws conclusions of this work.

2 Use of visualization techniques in class

Visualization techniques help students understand microcomputer architecture details at various levels of complexity and they provide important supporting roles to instructors inside classroom. Three different visualization techniques are proposed in this paper.

2.1 Circuit diagram visualization

A circuit diagram representation of computer concepts enables students to approach course material in more concrete way, and to visualize abstract behavior of computer hardware architecture more clearly and effectively.

Take memory hardware which is used to store data as one example. In general, the *A1* and *A0* signals represent address of the memory, the *CS* signal enables the memory, the *RD* signal allows reading memory data, and the *OE* signal allows the memory to output data onto the data bus. Such descriptions can be visualized through the circuit diagram shown in Figure 1 which is a 4x2 memory circuit. The analysis of circuit diagram allows students to explore the functionality of the circuit, and to enhance their understanding of the memory behavior.

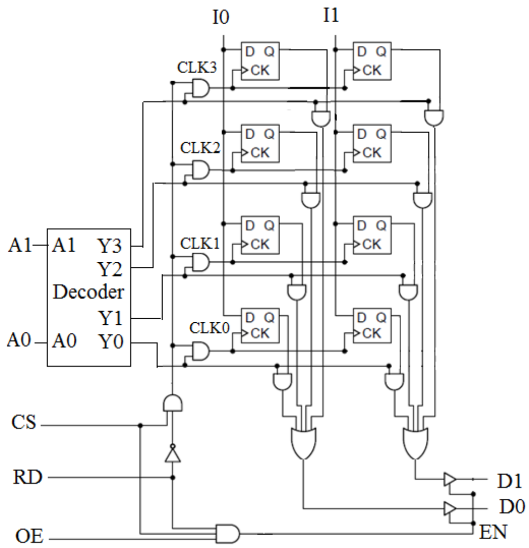

Figure 1. 4x2 memory circuit

For example, in Figure 1, when the *CS* signal is logic 0, it results in all logic 0s for the *CLK0*, *CLK1*, *CLK2*, *CLK3*, and *EN* signals, which are used to generate the clock pulses and also to enable the output tri-state buffers. Because these signals are disabled, new data inputs won't be written into D Flip-flops, and also the output data bus will be in high-impedance mode. Through such analysis, students will be able to understand why the *CS* signal can disable the memory.

Instructors can use circuit diagrams to stimulate in-class discussion, to engage students in active learning and to initiate a collaborative effort among students for finding answers and solutions to the functionality of circuits.

2.2 Waveform visualization

Waveforms can help students find relationship among multiple signals, and visualize signal patterns. Figure 2 shows the simulation waveform for the memory circuit in Figure 1.

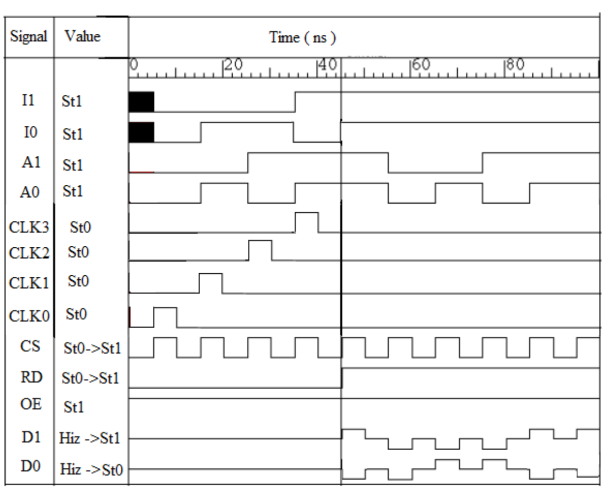

Figure 2. Memory simulation waveform

In Figure 2, it is obviously to see that the *CLK0*, *CLK1*, *CLK2*, and *CLK3* pulses are generated at different time. For example, the *CLK3* pulse is generated due to the input of *CS* pulse from 35 ns to 45 ns. At this time, the *A1A0* signals point to memory address 3, and the memory data input signals *I1* and *I0* are binary "10". Because *RD* is logic 0, it allows binary data "10" to be written into the memory address location 3. Here, multiple signals show joint actions in order to write data into memory. In another word, to write data into memory location 3, the *CS* signal needs to be logic 1 to enable the memory hardware. The address signals need to be set at value 3, and the *RD* signal needs to be logic 0 to allow writing to occur.

From 45 ns to 50 ns in Figure 2, because the *RD* and *CS* signal are activated with logic 1s, the memory cells at address 3 can output data "10" onto the data bus *D1D0*. This shows how memory reading works. It also proves the previous writing was successful.

In general, waveforms can be used to promote hardware functionality analysis and extend students' understanding of key concepts of computer hardware.

2.3 Signal flow visualization

To analyze the dynamic behavior of interconnected computer system and also to provide a more complete view of how computer hardware works, a signal flow diagram is beneficial for students to visualize the hardware complexity in a more comprehensible way. Signal values can be marked on the hardware diagram for different case studies as displayed in Figure 3. Figure 3 shows the Mic-1 microprocessor [9] consisting of data path and control parts interweaved together.

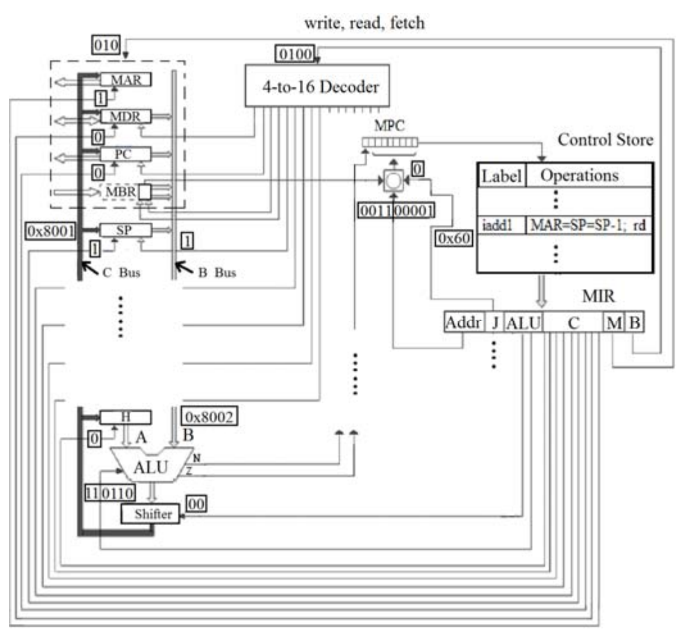

Figure 3. Signal flow on Mic-1 microprocessor

Mic-1 architecture implements Integer Java Virtual Machine (IJVM) instruction set. Figure 3 shows signal flow values marked inside black rectangles for iadd1 micro-operation. IADD instruction is one of IVJM instructions, which pops two words from the stack memory, and then push their sum into the stack. To implement IADD, three micro-operations of iadd1, iadd2, and iadd3 are needed. iadd1 is used to read next-to-top stack data. The opcode 0x60 for IADD points to the control store memory address where iadd1 micro-operation is stored. The iadd1 micro-operation is fetched into the micro-instruction register *MIR*, which controls hardware in the data path.

In Figure 3, the four bits of "B field" inside the *MIR* register are equal to "0100" which are connected with a 4-to-16 decoder. This enables the 5th output of the decoder so that the SP register can output content "0x8002" onto the B bus.

The "ALU field" of the *MIR* register has totally 8 bits. Two bits are "00" which disable the shifter circuit from shifting. And the other 6 bits are "110110" which choose ALU function of B - 1. As a result, the B bus data "0x8002" is subtracted by 1 through ALU and then generates 0x8001 on the C bus, which is the next-to-top stack address value.

The "C field" of the *MIR* register outputs binary numbers which are used as write enable signals for different registers connected with the C bus. Since only the write enable signals of *MAR* and *SP* registers are logic 1s, the *MAR* and *SP* registers are activated. As a result, both *MAR* and *SP* registers will be updated by the next-to-top stack address.

Since the "M field" of the *MIR* register outputs binary number "010" for write, read and fetch control of the main memory, the main memory will be set to read mode. In another word, the Mic-1 processor is going to read next-to-top stack data from the main memory and store it into the memory data register *MDR*.

After iadd1 micro-operation is processed by data path hardware, Mic-1 is going to process next micro-operation of iadd2. The logic 0 values of the "J filed" in the *MIR* register allow the *MPC* register to choose the next micro-instruction address from the "Addr filed" of the *MIR* register, and it is "001100001" (0x61). The *MPC* register stores the control store memory address for the next micro-operation. At the address 0x61 location of the control store, the micro-operation iadd2 machine code is stored. Then the whole processing steps will be repeated to implement the new micro-operation.

Signal flow helps explain cause and effect of different dynamic actions inside the computer hardware with the support of data. It can engage students in the active learning process, and also increase their attention and focus. Moreover, it helps students understand context in a tangible way.

Signal flow data can be fully provided by instructors, or partially by instructors and partially by students through questions and problem solving inside the classroom. Such practice increase students' curiosity for course contents, and also promotes meaningful learning experiences among students.

3 Computer aided design projects

Project assignments using computer aided design tools help students integrate topics covered in the course, and engage them in active learning instead of just memorizing knowledge transferred from instructors in class. As a free and user friendly open source educational tool, Logisim [8] allows students to design and simulate digital logic circuits. This tool has been used by author of this paper to teach computer hardware design course to undergraduate computer engineering students at CSUS.

Figure 4 shows one project idea of using Logisim. It requires students to create the register logic sub-circuit to be connected with the data bus, the ALU, the control store, and the main memory of the Mic-1 processor.

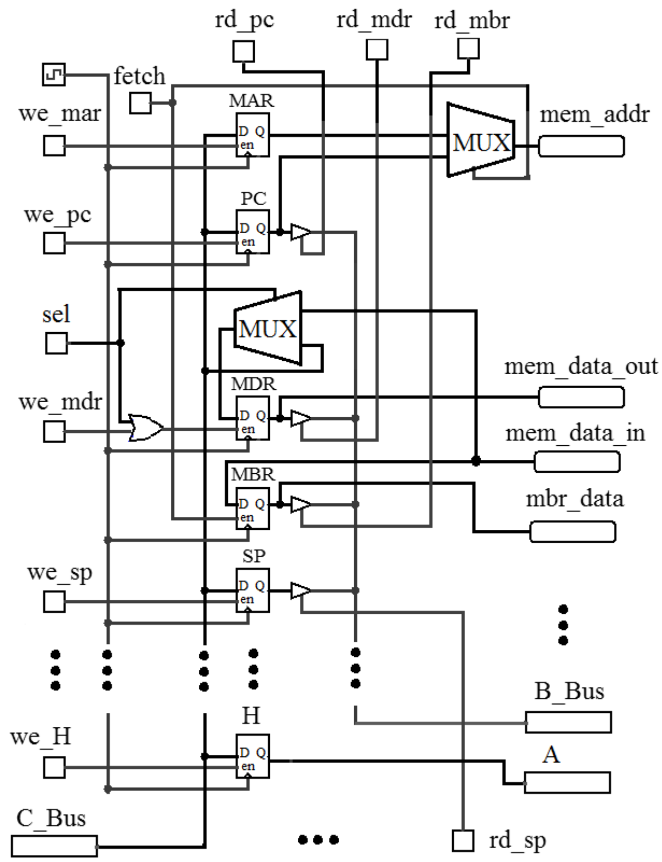

Figure 4. Registers and data bus

The ALU needs two operands A and B. The operand A comes from the register *H*. The operand B comes from one of those registers with read enable controls and also connected with the data bus *B_Bus*. Every register connected with the data bus *C_Bus* has write enable control for storing the ALU result. For example, the *rd_pc* is the read enable control for the *PC* register, and the *we_pc* is the write enable control for the *PC* register. In addition, registers *MAR*, *PC*, *MDR*, and *MBR* are used to interface with another sub-circuit of the memory block.

In Figure 4, when the *fetch* signal is logic 0, it selects the *MAR* register output to be the memory address *mem_addr*. When the *fetch* signal is logic 1, it selects the *PC* register output to be the memory address, and also it allows reading instruction *mem_data_in* from the main memory into the *MBR* register. The *MDR* register has write enable controls from *we_mdr* and *sel*. The signal *we_mdr* allows the *C_Bus* data to be written into the *MDR* register. The signal *sel* allows the memory data *mem_data_in* to be written into the *MDR* register. The signal *mem_data_out* comes from the *MDR* register output which can be written into the main memory. The output *mbr_data* can be used as one of inputs to the CONTROL_STORE subcircuit to determine the next microinstruction address.

Moreover, Logisim has built-in RAM which can be configured as the Mic-1 memory space to hold the main IJVM instructions, local variable, stack, and constant pool data. Logisim also has built-in ROM which can be configured as control store to hold micro-instructions. Students are required to design appropriate micro-instructions to be loaded into the control store to interact with the data path. In conclusion, the whole Mic-1 processor can be created, and simulated by using the Logisim tool. Modification of the Mic-1 is also possible which may stimulate students for more creative design ideas.

4 Conclusions

Three different visualization techniques are discussed in this paper including circuit diagram visualization, waveform visualization, and signal flow visualization. These techniques allow students to study computer hardware in different perspectives, to visualize computer hardware concepts in more tangible ways, and to improve their learning experience. In addition, computer aided design projects stimulate students to think creatively, to integrate course contents, and to enhance their problem solving skills. Combining different visualization techniques for instruction inside classroom, and also using computer aided design project assignments for students outside classroom, instructors are able to teach computer hardware course more effectively and efficiently, and also engage computer engineering students in active learning.

5 References

[1] S. Vijay Iyengar, Larry L. Kinney. “A Design-Oriented Microprocessor Laboratory”; IEEE Transactions on Education, Vol. 24, Issue 1, Pages 43 – 46, Feb. 1981.

[2] James E. Cross, Raphael A. Soetan. “Teaching Microprocessor Design Using the 8086 Microprocessor”; IEEE Conference Proceedings on Southeastcon, Pages 175 – 180, Apr. 1988.

[3] A. J. D. Casaca, J. V. Woods. “Teaching Digital-Systems Design with a Microprocessor-Controlled Patchboard”; IEE Proceedings on Computers and Digital Techniques, Vol. 130, Issue 3, Pages 75 – 82, May 1983.

[4] Maosheng Wu, Ximeng Wu, “Design of the Single-chip Microcomputer Experiment Teaching System for Application”; 2011 International Conference on Electrical and Control Engineering, Pages 649 – 652, Sept. 2011.

[5] S. Arunaganesan, J. Adhavan, G. Shivakanth Reddy, M. Venkatesan. “Data Acquisition System Based on 8051 Microcontroller for Cutting Tool Temperature Measurement”; 2013 IEEE International Conference on Computational Intelligence and Computing Research, Pages 1 – 4, Dec. 2013.

[6] Hussam M. Elbehiery, M. Ghada Abdelmouez. “Implementation of New Symmetric Ciphering on ATMEGA 32”; 2010 International Computer Engineering Conference, Pages 1 – 8, Dec. 2010.

[7] Tom Stoltz, Mark Paulik, Nizar Al-Holou. “A Microcontroller Laboratory Hardware Platform for the Academic Environment: The UDM-EVB”; Proceedings of 35th ASEE/IEEE Annual Conference on Frontiers in Education, Session S2G-1 – S2G-6, Oct. 2005.

[8] Carl Burch. “Logisim: A Graphical System for Logic Circuit Design and Simulation”; Journal on Educational Resources in Computing, Vol. 2, Issue 1, Pages 5 – 16, March 2002.

[9] Andrew S. Tanenbaum, Todd Austin. “Structured Computer Organization”. Pearson publisher, 2012.