South China University of Technology

Experimental Report of Computer Organization and Architecture II

Name: Huang Zejun Student ID: 202130670058

Instructor: Xie Xiaomeng

Experimental Topic 1： Bus and register experiments

Experiment Overview

【Experimental objectives and requirements】

Purpose of the experiment:

1) Master the operation of Proteus software

2) Master the use of devices such as 74LS244, 74LS374, 74LS273, 74LS194, etc.

3) Understand the concept of bus and bus operation characteristics

Experimental requirements:

Design the schematic diagram in Proteus software and simulate it according to the PPT requirements. Answer the following questions based on the simulation results (attach simulation screenshots)

1) What is the function of the 74LS244? How is data transmission controlled?

2) What is the difference between the 74LS273 and 74LS374? How do I store the data 0xF0 into the 74LS273 and 74LS374?

How to send the data stored in 74LS273 and 74LS374?

3) How do you use two 74LS194s to form two 8-bit shift registers? How do you use left shift, right shift, and parallel transfer to make the 74LS194 registers equal to 0xF0?

4) What are the characteristics of a bus? What functions should the multiple chips connected to the data bus have?

【Experimental environment】

Operating system: Windows 10

Experimental content

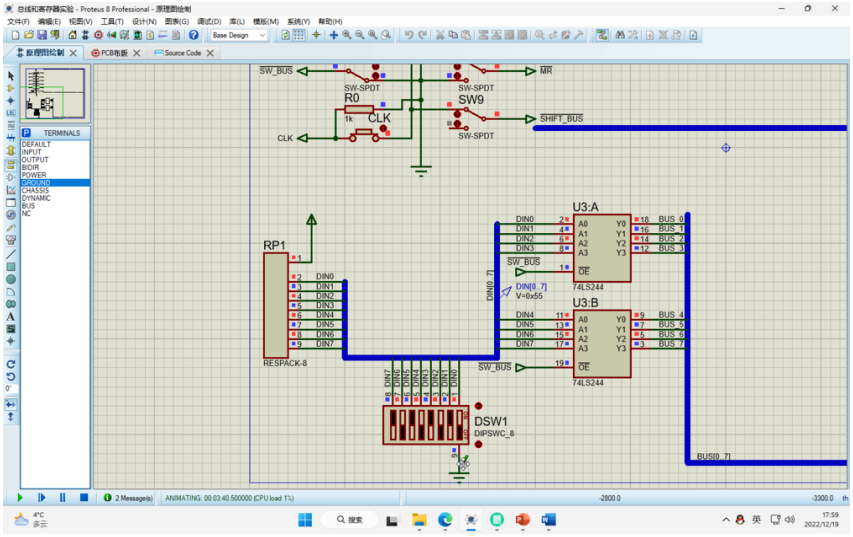
【Experimental process】

1. Experimental steps:

Experiment (1) Bus and buffer experiment

a) SW_BUS = 1; manually operate the DIP switch on the DIN bus to set data 0x55 on the DIN bus, blocking buffer 244. Compare the similarities and differences between the DIN bus and the BUS bus status.

b) SW_BUS=0, compare the similarities and differences between the bus DIN and BUS states, and record the BUS data.



Experiment (2) General register experiment

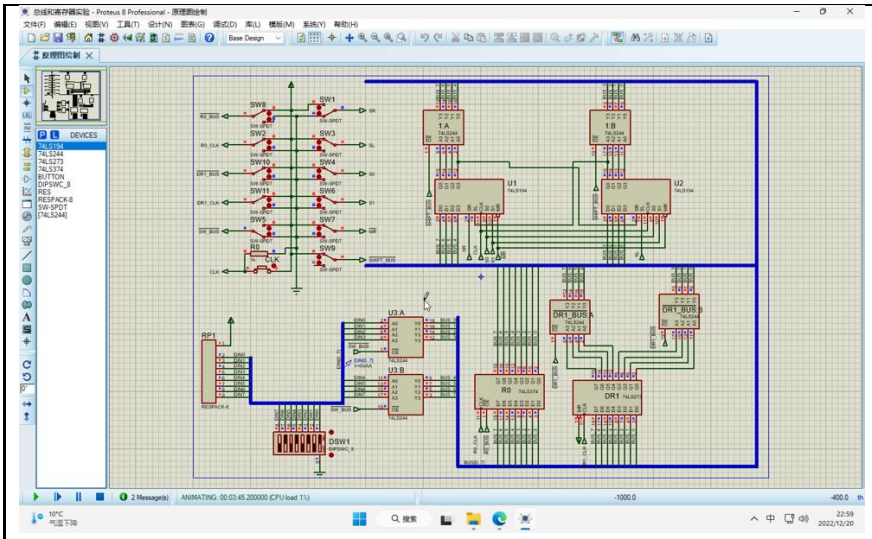
a) SW_BUS=0, R0_BUS=DR1_BUS=SHIFT_BUS=1; operate the DIP switch to input data 0xAA to the bus, and observe the respective states of the output 6 terminals QX of 74LS374 and 74LS273 at this time.

b) The rising edge of the CLK terminal R0_CLK of 74LS374 stores the 0xAA data on the bus into the R0 register (74LS374).

c) The three-state gate 244 of the DIP switch is blocked (SW_BUS=1), and the status of the bus BUS is observed at this time.

d) The output of 74LS374 is enabled (R0_BUS=0) and the status of the bus BUS is observed.

e) The rising edge of DR_CLK on the 74LS273 stores the 0xAA data on the bus into the DR register (74LS273). Observe the states of the QX output terminals of the 74LS374 and 74LS273.

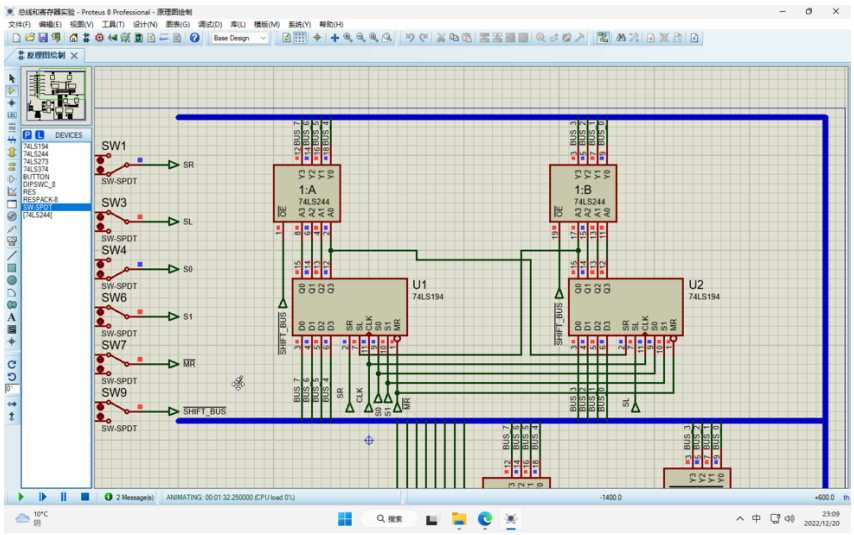


Experiment (3) Shift register experiment

a) SW_BUS=0, R0_BUS=DR1_BUS=SHIFT_BUS=1; any eight-bit binary number is sent to the bus BUS through the DIP switch and assigned to the input terminals D0D1D2D3 of 74LS194.

b) Set the MR, S1, S0, SL, and SR terminals of 74LS194 respectively and observe the rising edge of CLK. , the status of the output terminals Q0Q1Q2Q3 at the time of falling edge transition, verifying the function of 74LS194.

c) Observe the method of combining two 74LS194s in the circuit to form an 8-bit shifter, and the shifting effect.



2. Experimental data:

Experiment (1) Bus and buffer experiment

SW_B	BUS_	BUS_	BUS_	BUS_	BUS_	BUS_	BUS_	BUS_

US	0	1	2	3	4	5	6	7
0	1	0	1	0	1	0	1	0
1	Z	Z	Z	Z	Z	Z	Z	Z

Experiment (2) General register experiment

a) state:

74LS374;10101010 74LS273:00000000

c) The bus is blocked and there is no signal.

d) Data changed from none to oxAA

e) state:

74LS374: 10101010 74LS273: 10101010

3. Main process of the experiment:

The following are answers to the questions based on simulation results:

1) 74LS244 is a 3-state 8-bit buffer,It is mainly used for tri-state output. When OE is low,The output immediately assumes the state of the input pin; When OE is high, the outputHigh resistance state.

2) right74LS273

a. When MR is low, all output pins output 0, that is, all are reset.

b. When MR is high, when CLK has a rising edge, the level state of the input pin is immediately latched and immediately presented on the output pin.

right74LS374

a. When OE is high, regardless of the input or LE, all outputs are in high impedance state.

b. When OE is low, as long as a falling edge appears on LE, the output immediately assumes the state of the input pin.

3) Connect Q3 of the low-order chip to the right-shift serial input terminal SR of the high-order chip, and connect Q0 of the high-order chip to the left-shift serial input terminal SL of the low-order chip. At the same time, connect S1, S0, CLK, and MR of the two chips together as corresponding signal input terminals.S1 = 1, S0 = 0, MR = 1. SL is set to low level, CLK clicks 4 times; SL is set to high level, CLK clicks 4 times, making the value of 74LS194 0xF0.

4) The characteristics of the bus include physical characteristics, functional characteristics, electrical characteristics, and time characteristics. The output port connected to the data bus should have a latch function, and the input port

connected to the data bus should have a tri-state buffer function.

summary

The biggest problem encountered during the experiment was that I didn't know much about these chips. During the experiment, I had to look at the chip function table frequently to understand the purpose of each experimental step. But after the whole experiment, I understood that if I didn't74LS244, the output data may be affected by the data bus and produce incorrect results, clarifying74LS273 andThe difference and use of 74LS374Two 74LS194s are combined to form an 8-bit shifter.

Experimental Question 2： Serial and parallel adder experiments

Experiment Overview

【Experimental objectives and requirements】

Purpose of the experiment:

1) Understand the implementation ideas of ripple carry adder and carry lookahead adder

2) Design a 4-bit ripple carry adder and a carry-lookahead adder

3) Implementation of subtraction function

Experimental requirements:

designSimulate a 4-bit ripple carry adder/subtractor and a carry-lookahead subtract adder/subtractor. Answer the following questions based on the simulation results (attach simulation screenshots).

1) Compare the similarities and differences between the ripple-carry adder and the carry-lookahead adder, as well as their respective advantages.

2) How to modify the adder to implement the subtraction function?

3) Use the designed circuit to implement 5+2 and 5-2. What are the simulation results?

4) Use the designed circuit to implement 4+5 and 4-5. What are the simulation results?

5) How do you implement overflow detection? Please add the relevant circuit to the schematic diagram.

【Experimental environment】

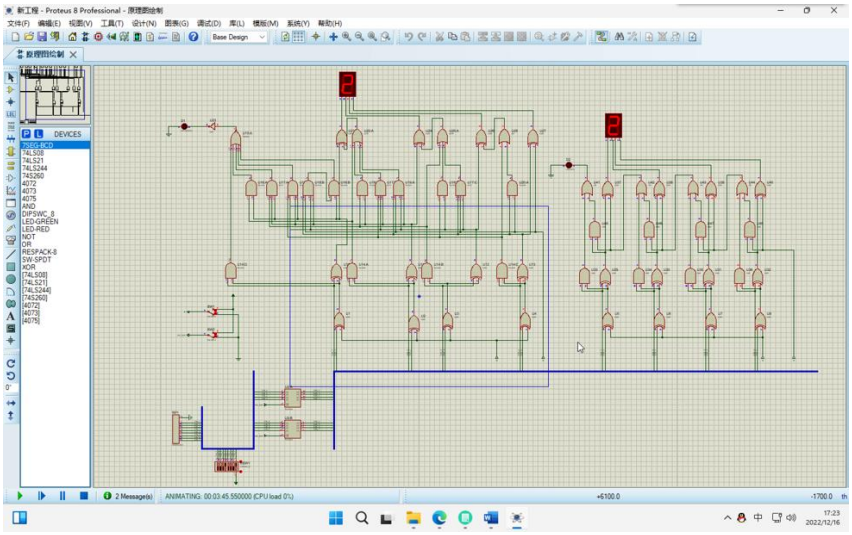
Operating system: Windows 10

Experimental content

【Experimental process】

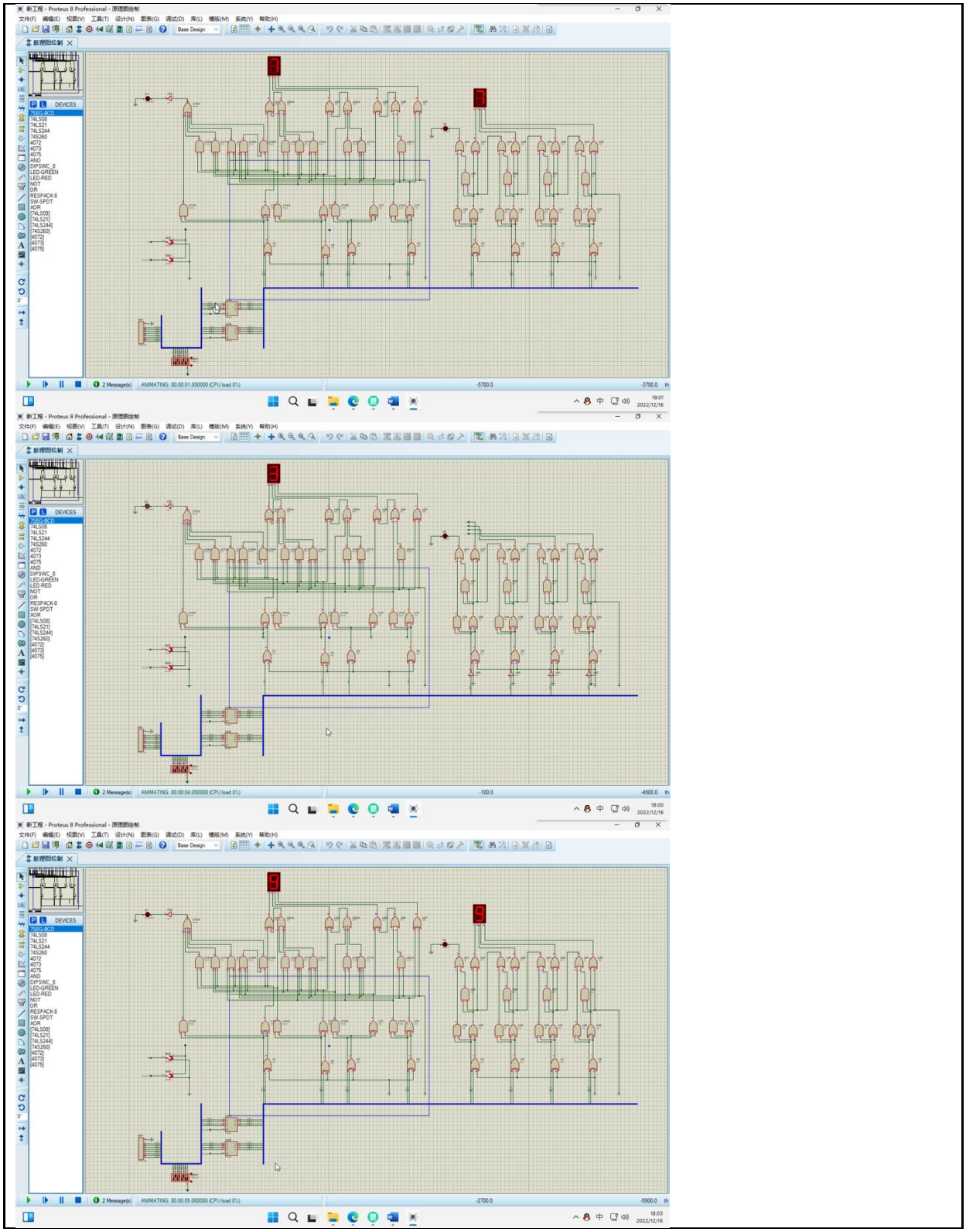
1. Experimental steps:

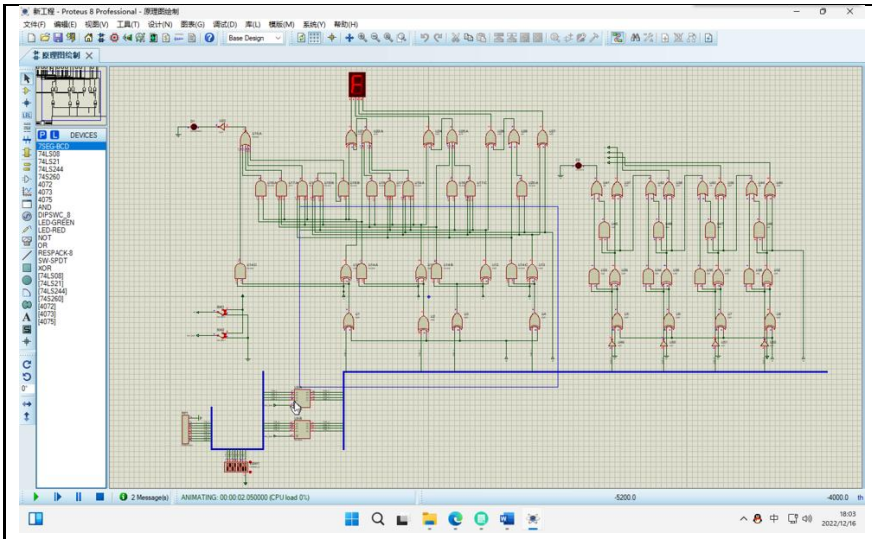
1) First, draw the serial adder and the parallel adder according to the experimental principle diagram.
2) Serial adder: Use logic gates to implement a 4-bit serial adder. The lower 4 bits and upper 4 bits of the DIP switches are used as addend and augend respectively. Observe the result and the highest carry.
3) Parallel adder: Use logic gates to implement a 4-bit parallel adder. The lower 4 bits and upper 4 bits of the DIP switches are used as addends and augends respectively. Observe the result and the highest carry.



4) Using the principle of two's complement, the adder is modified to implement subtraction. The minuend plus the two's complement form of the subtrahend (all bits are inverted and 1 is added) is the result.

2. Experimental data:





3. Main process of the experiment:

The following are answers to the questions based on simulation results:

1）In a carry-lookahead adder, each carry is generated independently of the others and depends only on the input data A, B, and C_in. This eliminates the carry cascade propagation between stages. While the circuit implementation is relatively complex, it can reduce the delay caused by the carry. In contrast, a ripple carry adder is relatively simple to implement, but has a longer delay than a carry-lookahead adder.

2）Add a component to the adder that can convert the subtrahend (which is stored in two's complement) into two's complement when encountering a subtraction.

3）7,3

4）9, F

5）Double sign method (modified two's complement) X and Y use double sign two's complement to participate in the operation. The double sign bit of positive numbers is 00, and the double sign bit of negative numbers is 11. When the two signs of the operation result are different (01 or 10), overflow occurs.

summary

After using the 7SEG-BCD digital tube, it is more intuitive and convenient to observe the results of the experiment.I still encountered many difficulties with the carry-lookahead adder, and I also consulted some information. From Ci+1=Ai*Bi+(Ai+Bi)*Ci, I deduced step by step that the carry of each bit C1=G0+P0C0, C2=G1+P1G0+P1P0C0, C3=G2+P2G1+P2P1G0+P2P1P0C0...

, where Gi = AiBi and Pi = Ai + Bi. This makes the carry only related to the input data A and

B, reducing the delay time. However, due to the more complex structure, wiring errors are often made during the wiring process, so careful inspection is required after the connection is completed.

Experimental Question 3：Arithmetic unit, memory and data path experiments

Experiment Overview

【Experimental objectives and requirements】

Purpose of the experiment:

1) Master the design ideas of 8-bit ALU using 74181

2) Use ALU to implement +, -, +1, -1, AND, OR, NOT, XOR, and pass-through functions

3) Understand the operation timing of the memory and analyze the memory address space

4) Master the concept of data path and analyze the data path of conventional instructions based on the CPU structure

Experimental requirements:

According to the given schematic diagram and experimental PPT, first master the working principle of the circuit, thenPerform experimental simulation operations and answer the following questions based on the simulation results (attach simulation screenshots)

1) How to design an 8-bit ALU using 74181?

2) When performing +, -, AND, OR, and XOR operations on DR1 and DR2, what values do M, CN, and S3-S0 take?

3) What values do M, CN, and S3-S0 take when performing +1, -1, inversion, and straight-through operations on DR1?

4) What are the address spaces of the 6116 and 2764? How do I read and write to the 6116?

5) Perform corresponding operations according to the data given in the data path part and store the results in 6116.

【Experimental environment】

Operating system: Windows 10

Experimental content

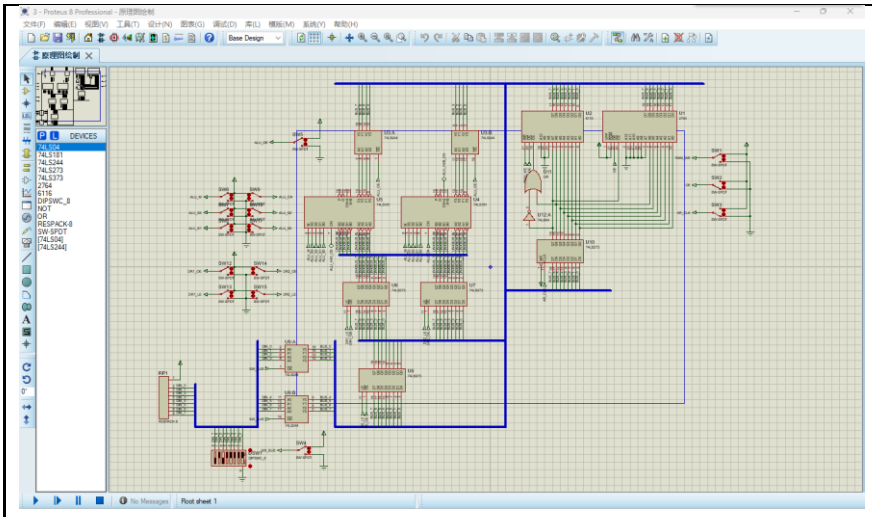
【Experimental process】

1. Experimental steps:

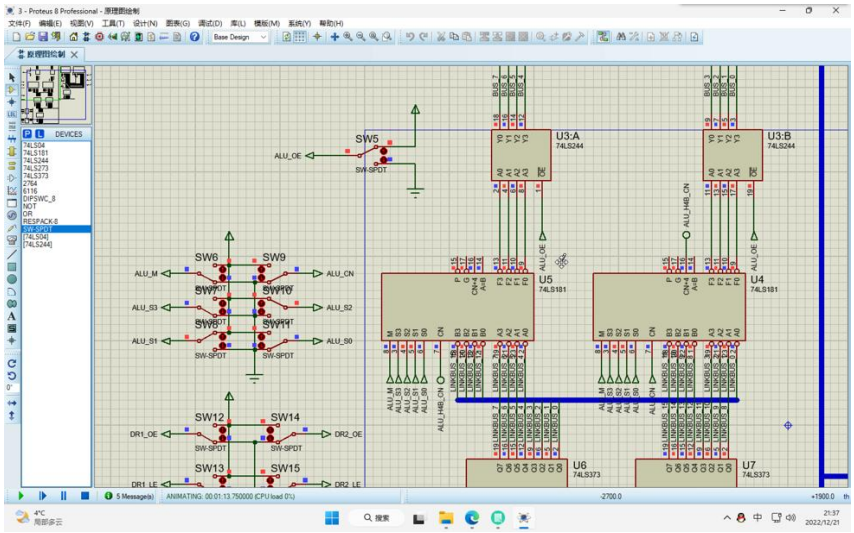
1) Put the data 0xAA into general register R0.
2) Latch the data in R0 to the temporary register DR1.
3) Place the data 0x55 into general register R0.
4) Latch the data in R0 to the temporary register DR2.
5) Perform logical "AND, OR, XOR" on DR1 and DR2 and observe the operation results.
6) DR1 and DR2 perform arithmetic "+, -" operations , and observe the operation results.
7) Perform inversion, +1, -1, and pass-through operations on DR1 and observe the results.

Memory experiment steps:

1) Write the data AAH, FFH, 55H, 01H into ROM 2764, address [00..03] in sequence (see the appendix for detailed operation methods).
2) The address 00H is latched into the AR register.
3) The read function of ROM 2764 is enabled, and the data at address [00] is sent to the bus and latched into R0.
4) Pause the simulation and observe the storage contents of RAM 6116 and ROM 2764 (see the appendix for detailed operation methods).
5) Latch address 80H into the AR register.
6) Enable the RAM 6116 write function and write the data of R0 to the address of RAM [80] .
7) Pause the simulation and repeatedly observe the RAM and ROM storage contents.



2. Experimental data:



	Operation results
+	1111 1111
-	0101 0101
and	0000 0000
or	1111 1111
XOR	1111 1111
+1	1010 1011
-1	1010 1001
Negation	0101 0101
Direct	1010 1010

3. Main process of the experiment:

The following are answers to the questions based on simulation results:

1) Combine two 74181s, connecting the output CN+4 of the first 74181 to the input CN of the second. SW_SPDT sets the current ALU function, controlling arithmetic or logical operations. The result is output to the bus via the 74LS244.

2) 74181 pin values for each operation performed on DR1 and DR2

	M	CN	S3	S2	S1	S0
+	0	1	1	0	0	1
-	0	0	0	1	1	0
and	1	1/0	1	0	1	1
or	1	1/0	1	1	1	0
XOR	1	1/0	0	1	1	0

3) 74181 pin values for each operation performed on DR1 and DR2

	M	CN	S3	S2	S1	S0
+1	0	0	0	0	0	0
-1	0	1	1	1	1	1
Negation	1	1/0	0	0	0	0
Direct	1	1/0	1	1	1	1

4) The address range of 6116 is 00H~7FH

The address range of 2764 is 80H~FFH

Input the address you want to store in RAM to AR through the bus, refresh AR through AR_CLK, input the data of D7-D0 terminal to Q7-Q0 terminal, input the address of A7-A0 address terminal of RAM, and then input the data you want to store to the bus, such as F0H in the figure, and then give it a low level with RAW_WE to put it in write state, and then you can view the data at address 80H.

summary

This experiment taught meHow to design an 8-bit ALU using two 74181s, simulate the internal principles of the CPU, and complete the memory and data path experiments. The course experiments made me understand the importance of combining theory with practice.Only by drawing conclusions from theory and applying them in practice can we improve our practical skills and independent thinking ability.It deepened my understanding of the knowledge in the textbook.

Instructor's comments and grades

Comments:

score: Instructor's signature:

Review date: