

Assignment #1 Design of an 8 bit Instruction Set

Due Monday January 31, 2000

During the quarter, you will be designing a complete 8 bit processor which is optimized to execute three particular programs. All of the instructions executed by your processor must complete in one cycle. An optimal processor will maximize speed of execution and minimize the amount of hardware used. Because all instructions are executed in one cycle, maximum speed is attained by minimizing the dynamic instruction count, and by simplifying the hardware design to reduce the time necessary to execute a single cycle. With this in mind, your goals for the design are:

1. Minimize dynamic instruction count for the three programs.
2. Design the processor with the minimum quantity of hardware components.

For assignment #1, you will design the instruction set for your processor. Your instruction set should contain only fixed length (8 bit) instructions. All data and registers should be 8 bit. Your storage space should be divided into bytes. Obviously, to fit within 8 bits, you will have to be clever about the design. Before finalizing your design, you should carefully consider the specific needs for execution of the three programs. To make the design more manageable, the instructions (code) are stored in a separate memory, which itself need not be addressable by the instruction set (other than through a Program Counter and branches).

You will turn in a lab report no longer than 5 pages. The report will contain a detailed instruction set specification, including:

1. The operations and associated opcodes
2. Detailed instruction format description (we should be able to build an assembler based on this).
3. Architecture of processor; i.e. number and type of registers, specialized functions of registers (if any), stacks, etc.
4. Main memory architecture and size
5. Addressing modes to be implemented.

Your report will also include the answers to the questions that follow. If the question is answered by your instruction set description, you may answer the question by reference to the appropriate part of that description. Remember, the person grading your report only understands your solution to the extent you describe it in the report. Be specific and complete.

The three programs:

1. Write a program to determine the parity of the lower 7 bits of the bytes of an array. Place a 1 in the 8th bit of each byte if the parity is even, place a 0 there if it is odd. The array is stored in locations 0 thru 31
2. Write a program to calculate $A**B$ without using a multiply instruction. A is stored in memory address 7 and B is in location 8. Place the double byte result in locations 14 and 15. Assume unsigned integers for both A and B. B is always less than or equal to 4.
3. Write a program which sorts an array such that all numbers greater than X are at the front (low addressed end) of the array, and all numbers less than X are at the other end. The array occupies locations 0 thru 31. X is stored in location 32. Put a pointer to the lowest addressed location that contains a value less than X into location 33.

The questions:

1. What instruction formats are supported and what does each format look like? Provide an example of each.
2. List all of the instructions in your set and give their opcodes.
3. Describe the registers and give information about their purpose if other than general purpose registers.
4. Describe your memory addressing architecture. What addressing modes are supported? How are addresses calculated? Give examples. How large is main memory and how is it structured?
5. What did you do to optimize your instruction set to minimize dynamic instruction count?
6. What did you do to optimize your instruction set to minimize the hardware used to implement it?
7. Describe any other optimizations you did other than for 5 and 6 above.
8. What do you believe will be the bottleneck in your design?
9. How would you have changed your instruction set design if you had 2 more bits in the instruction? 4 more bits?
10. Can you classify your machine in any of the classical ways, such as register, accumulator, stack, etc.? If so, which? If not, give a name for your machine type.
11. Give an example of an assembly language instruction in your machine and then give the machine code.
12. Write a program in your assembly language to do program 1.
13. What is the dynamic instruction count for your program (if it varies with the data, tell me the worst case count)?
14. Write a program in your assemble language to do program 2.
15. What is the dynamic instruction count if $A = 61$ and $B = 14$?
16. Write a program in your assembly language to do program 3.
17. What is the dynamic instruction count for your program (if it varies with the data, tell me the worst case count)?