**CIS 350 – INFRASTRUCTURE TECHNOLOGIES**

**HOMEWORK #4**

Student Name(s):   Louis Ries

(You may do this homework in groups of 2 students maximum.)

**Topics**: The CPU and Memory (Chapters 7 and 8)

Ex. Show an instruction format that could be used to move data or perform arithmetic between two registers (the source register and the destination register). Assume that the instruction is 32 bits wide and that the computer has 8 general-purpose registers. If the op code uses 7 bits, how many bits are spares, available for other purposes, such as special addressing techniques? Draw a detailed diagram with the instruction format. Figure 7.21 on p. 228 in your textbook depicting a "register to register" format may be very helpful.

25 spare bits

|  |  |  |
| --- | --- | --- |
| Op code | Rsrc | R­dst |

0 6 7 18 19 31

Ex. 7.3, p. 232

One large modern computer has a 48-bit memory address register. How much memory can this computer address?

2^48 = 281,474,976,710,656 bits

Ex. 7.7 a (only), p. 232

What is the effect of shifting an unsigned number in a register two bits to the left? One bit to the right? Assume that 0s are inserted to replace bit locations at the end of the register that have become empty due to the shift.

Two bits to the left => number is multiplied by 4.

One bit to the right => number is divided by 2 and floored (no remainder, no decimal)

Ex. Suppose that the following instructions are found at memory locations 25 and 26. Suppose that the following data are found at memory 55 and 56.

Address Instruction

--------------------------------

25 LDA 55

26 ADD 56 Addresses 25-26 represent the program area

--------------------------------

--------------------------------

Data

55 125 Addresses 55-56 represent the data area

56 10

Show the contents of the PC, the MAR, the MDR, the IR, and the A as each step of the fetch-execute cycle is performed for instructions at addresses 25 and 26. (The machine cycle for instruction LDA I worked in class on the white board would be helpful. See page 5 in the lecture notes for chapter 7. I asked students to take notes. Also, look at Assignment One in in-class small group activity #4.)

Instruction: 25 LDA 55

PC MAR MDR IR A

1. PC→ MAR 25 25 LDA 55 ? ?
2. MDR→ IR 25 25 LDA 55 LDA 55 ?
3. IR[addr] → MAR 25 55 125 LDA 55 ?
4. MDR→ A 25 55 125 LDA 55 125
5. PC+1→ PC 26 55 125 LDA 55 125

Instruction: 26 ADD 56

PC MAR MDR IR A

1. PC→ MAR 26 26 ADD 56 ? 125
2. MDR→ IR 26 26 ADD 56 ADD 56 125
3. IR[addr] → MAR 26 56 10 ADD 56 125
4. MDR+A→ A 26 56 10 ADD 56 135
5. PC+1→ PC 27 56 10 ADD 56 135

**Short essay questions. Your answers should capture the essence of the questions. There is no credit for 1- or 2-sentence answers.**

Ex. 8.4 and 8.5, p. 263.

8.4 Suppose that a CPU always executes the two instructions following a branch instruction, regardless of whether the branch is taken or not. Explain how this can eliminate most of the delay resulting from branch dependency in a pipelined CPU. What penalties or restrictions does this impose on the programs that are executed on this machine?

Without knowing what will happen at a branch, the CPU already has the information processed for if the branch is ignored. If the branch isn’t ignored, the CPU can dispose of the data stored in the registers for that pipeline. This style of branching takes up more registers than otherwise. This can cause issues when the CPU needs additional registers to execute a new pipeline.

8.5 Some systems use a branch prediction method known as static branch prediction, so called because the prediction is made on the basis of the instruction, without regard to history. One possible scenario would have the system predict that all conditional backward branches are taken and all forward conditional branches are not taken. Recall your experience with programming in the Little Man Computer language. Would this algorithm be effective? Why or why not? What aspects of normal programming, in any programming language, support your conclusion?

This algorithm would be very effective, but less efficient. It would use more resources, but in the case of a correct prediction, would speed up the flow of processing.

Ex. 8.16, p. 263. Also briefly explain how cache memory works.

Cache memory allows for resources to be accessed in the future more quickly. It is considered more volatile than other types of memory as cache memory is cleared whenever the space is needed for more-active memory.

Ex. 8.17, p. 263

Modern computers are usually described as multicore. What does this mean? Under ideal conditions, what performance gain would be achieved using a four-core processor over a single-core processor?

A four-core processor could process four times as many processes or execute processes faster by taking advantage of multi-threaded/parallel processing. This would mean a huge performance boost over single-core processors.

Ex. 8.18, p. 263

Identify and briefly explain two different ways of configuring a multiprocessing system. Which configuration is more effective for general-purpose computing? Which configuration is more effective for handling specialized processing tasks, such as those used in game applications?

One configuration is Symmetrical Multiprocessing. This allows for each CPU to have equal access to resources. This configuration is reliable, makes fault tolerant support simple, and produces a balanced workload across CPUs. An alternate configuration would be Master-Slave Multiprocessing. This configuration involves a CPU that manages all other CPUs by assigning them tasks and controlling resources and process scheduling. It’s simple and helps protect the system from conflicting processes, but is unreliable. If the master CPU fails, the entire system will fail. Symmetrical processing would be better suited for general-purpose computing. Master-slave processing would be best suited to perform specialized processing tasks.