TINFO 320

Chapter 7 Exercise Questions

7.1. What does *ALU* stand for? What is its corresponding component in the Little Man Computer? What does *CU* stand for? What is its corresponding LMC component?

ALU stand for arithmetic logic unit. The corresponding component in the Little Man Computer is the calculator.

CU stand for control unit. The corresponding component in the Little Man Computer is the Little Man.

7.2. What is a *register*? Be precise. Name at least two components in the LMC that meet the qualifications for a register. Name several different kinds of values that a register might hold.

Register is a hardware component which provides temporary data storage capability within the CPU. The program counter and the calculator are two components in the LMC that meet the qualifications for a register.

Register might hold data being processed, instruction being executed, I/O address to be accessed and binary status code.

7.3. What is the purpose of the *instruction register*? What takes the place of the instruction register in the LMC?

The purpose if instruction register (IT) is to holds the actual instruction being executed. It takes place in the input and output basket, where user give instruction for the Little Man.

7.4. When a value is compied from one register to another, what happens to the value in the source register? What happens to the value in the destination register?

The value in source register will not change, but the destination register will be overwritten by the copy of source register.

7.5. There are four primary operations that are normally performed on a register. Desscribe each operation.

- Hold data needed for currently executing program.

-Transfer data between memory locations.

-Store information about status data and currently executing program.

-Address of next program instruction.

7.6. Explain the relationship between the memory address register, the memory data register, and memory itself.

The memory address register holds the address of the memory location. The memory data register holds data value that is being stored or retrieved from memory location. Memory is any address location that is stored.

7.7. If the memory register for a particular computer is 32 bits wide, how much memory can this computer support?

4 GB RAM

7.8. What is the difference between volatile and non-volatile memory. Is RAM volatile or non-volatile? Is ROM?

Volatile memory loses its content when power is shut off, while non-volatile memory maintains its content when power is shut off. RAM is volatile and ROM is non-volatile.

7.9. Registers perform a very important role in fetch-execute cycle. What is the functions of registers in the fetch execute instruction cycle?

The importance of registers in the cycle is that it allows the data to be temporarily held in memory until the CPU or I/O requires that particular data to be used.

7.10. Explain each step of the fetch execute cycle. What are the similarities between this and the little man cycle?

The function includes:

- Registers can be loaded for other locations.
- Data is added or subtracted from the previous loaded data (only leaves the difference in the register).
- Data in the register can be rotated/shifted right or left.
- Value of data in register can test for 0, positive, negative, or too large to fit in register.

This is similar to the little man computer because they have the same operations available.

7.11. Once the fetch operation is complete, what is the first step of the execution phase for any instruction that accesses a memory address for data (e.g., load, store)?

In the fetch execution cycle, as discussed in the textbook, is the basic for every computer and every action that the computer completes. First the instructions are fetched from memory. After instructions are fetched, the first step of the execution phase to move from PC > MAR.

7.14. Define a bus. What are buses used for?

A bus is a physical connection that makes it possible to transfer data from one location in the computer system to another. Most common use for buses is transferring data between computer peripherals and the CPU, but can also be used to transfer data between hardware and CPU.

7.15. What 3 types of data might a bus carry?

Types of data that a bus can carry includes, data, addressing, control, and power. Data lines carry data from one location to another. The addressing line of the bus sends memory addresses over the line. Control lines provide control and timing signals for the proper synchronization and operation of the bus and of the modules. A power line is simply just the power to the hardware or peripheral that allows it to function within the other lines. Each bus line can be used in a different way except for the power line.

7.16. Explain how data travels on a bus when the bus is *simplex*, *half-duplex*, and *full-duplex*.

Simplex: 1 way only - unidirection
Half-duplex: 2-way, non-simultaneous
Full-duplex: 2-way, simultaneous communication

7.17. What is the difference between a *multipoint* bus and a *point-to-point* bus? Draw diagrams that illustrate the difference.

Point-to-point: usb ----> printer
One point to one point; a connection

Multi-point: storage -----V----------V----------V
Cpu memory video controller
One point to many points; a network

7.18. Briefly describe each of the major disadvantages of parallel buses.

Disadvantages of a parallel bus.
Expensive, big, connectors are also expensive due to large number of pins

7.19. Which Little Man Computer instructions would be classified as data movement instructions?

Load and Store

Load instruction: Load instruction move data from memory to register. Means data is moved from mailboxes to calculator.

Store instruction: Store instruction move data from register to memory. Means data is moved from calculator to mailboxes.

7.20. What operations would you expect the arithmetic class of instructions to perform?

Instruction is given to perform specific function. Instruction is represented in either zero or one because computer understand only two languages, that is zero and one. For the arithmetic class of instruction that would consider about the instructions shift and rotate instructions etc.

7.21. Explain the difference betweenn SHIFT and ROTATE instruction.

Shift : Shift instruction move the data bits left or right one more bits.

Shift instruction does not form closed loop.

In shift operation data is lost.

Rotate: Rotate instruction also move the data bits one or more left to right but the bit that is shifted out of the end is placed the vacated space at the other end.

Rotate instruction from the closed loop.

In rotate instruction data is not lost.

7.22. What do program control instructions do?

The program control instruction control the flow of the program. These instructions include jump and branch instructions, sub-routine calls and returns instructions.

The following instructions in the little man computer could be classified as program control instructions:

- Branch
- Branch on zero
- Branch on positive
- Branch on negative
- Halt

7.23. What is a stack? Explain how a stack works. Create a diagram that shows how PUSH and POP instructions are used to implement a stack.

Stacks are a data storage structure used in programming. The most recent data is “pushed” on to the top of the stack. This is called LIFO (last-in, first-out). When data is called from the stack, it is “popped” back into the program. Stacks can be useful for completing complex calculations where intermediate values can be pushed and popped from memory. Stacks are often used in functions (subroutines). When a function calls another function, the calling function is pushed and only pulled when the called function returns.

7.24. What is a privileged instruction? Which LMC instructions would normally be privileged?

Instructions that can be executed only when the computer is in a special privileged mode that is generally available to an operating or executive system, but not to user programs. Privileged instruction include storage protection setting, interrupt handling, timer control, input/output, and special processor status-setting instructions,

Privileged instructions are instructions that are used by the operation system to control memory access, provide security, and other background tasks that the user should not be able to modify directly. The HALT instruction for the LMC is one example. An individual user should not be able to halt the computer while other users are still performing tasks.

7.25. Show a 32-bit instruction format that allows 32 different op codes. How many bits are available for addressing in your format?

Op code *Addressing field*

In the 32-bit instruction format that allows 5 bits for 32 different op codes there are 27 bits available for addressing.

7.26. Show an instruction format that could be used to move data or perform arithmetic between two registers. Assume that the instruction is 32 bits wide and that the computer has sixteen general-purpose data registers. If the opcode uses 8 bits, how many bits are spares, available for other purposes, such as special addressing techniques?

10101010	1010	10101010101010101010

8-bit opcode + 4 bits for 16 registers = 12-bits; 32 bits - 12 bits = 20 bits for other purposes