Student name: LAN CHANG
HOMEWORK 1 The due date to submit – September 13, 15

	Answer:		References:
1. Why **the Von Neumann model** is essential in understanding computers?	The CPU was to include ALU, memory, and CU components. The control unit read instructions from memory and executed them. A method of handling I/O through the control unit was also established. The instruction set contained instructions representing all the essential features of a modern computer. In other words, von Neumann's machine contained every major feature considered essential to modern computer architecture. Modern computer architecture is still referred to as von Neumann architecture.		Textbook p.27
2. **Numbers:** Please write TWO examples representing the numerical data in any possible base, including binary, hexadecimal and octal, as well as floating point number notations.	$(53)_{10}$ $= (110101)_2$ $= (65)_8$ $= (35)_{16}$ = 01000010 01010100 00000000 00000000 $(7030)_{10}$ $= (1101101110110)_2$ $= (15566)_8$ $= (1B76)_{16}$ = 01000101 11011011 10110000 00000000		
3. **Data** - Please describe any TWO examples representing different formats of data used for still images (bitmap versus object images), video, audio and alphanumerical data.	**PNG** format is the best-known losslessly compressed alternative to GIF. PNG can store up to 48 bits of color per pixel, and additionally can store a transparency percentage value and a correction factor for the color in a monitor or printer. Its compression algorithm is often more efficient than that used with GIF. Unlike GIF, PNG stores only a single image in a file. **MP3** is the predominant digital audio data format for the storage and transmission of music. It is characterized by reasonable audio quality and small file size. MP3 uses a number of different tactics and options to achieve its small file sizes. These include options for different audio sampling rates, fixed or variable bit rates, and a wide range of bit rates that represent different levels of compression.		Textbook p.116 Textbook p.123
4. **LMC** - Explain the inner workings of the Little Man Computer and its relation with real life computers, including the	There is a series of one hundred mailboxes, a calculator, a two-digit hand counter and the Little Man in the Little Man Computer. There is a three-digit number used as a small group of instructions. We can combine some instructions into a program to have the Little Man do some useful work. It is a strength of the original model that it operates so similarly to a real computer that it is still an accurate		Textbook p.179-181

basics of assembly instructions.	representation of the way that computers work more than 45 years after its introduction.		
5. **CPU-memory** – Explain how the CPU and memory communicate. Concept of a register (including MAR/MDR).	There are two registers including the MAR and the MDR. There are some cells, each of which can hold a single value, and each of which has a single address. The MAR holds the address in the memory and the MDR connects to every cell in the memory unit. The CPU copies an address from some register in the CPU to the MAR. Then, the CPU turns on the switch that connects the MDR with the register by using the activation line, and data will be transferred from the memory to the MDR. Using the MAR, a single row of cells is activate, the MDR has access to the values in that single row.		Textbook p.200-204
6. **Fetch-execute** – What is the fetch-execution?	The fetch–execution instruction cycle is the basis for every capability of the computer. The operation of every instruction is defined by its fetch–execute instruction cycle. Ultimately, the operation of a computer as a whole is defined by the primary operations that can be performed with registers, such as to move data between registers, to add or subtract data to a register, to shift data within a register, and to test the value in a register for certain conditions, such as negative, positive, or zero.		Textbook p.207
7. **Stack** - How the stack is permanently used through any subroutine call to better write code?	Stack is an excellent method for storing the return addresses and arguments from subroutine calls. Program routines that are recursive must "call themselves". Suppose the return address were stored in a fixed location. If the routine is called a second time, from within itself, the original returning address is lost and replaced by the new return address. The program is stuck in an infinite loop. The return address is stored on a stack. This time when the routine is again called, the original address is simply pushed down the stack, below the most recent address. We always return from the last called subroutine to the one just previous.		Textbook p.220-222
8. **I/O** – Please list different types of Input/Output: Programmed I/O vs Interrupts and explain how they each work, as well as their advantages and disadvantages.	**Programmed I/O** Programmed I/O is a method of transferring data between the CPU and a peripheral. In general, programmed I/O happens when software running on the CPU uses instructions that access I/O address space to perform data transfers to or from an I/O device. **Advantage** Easy to program and understand. **Disadvantages** Slow and inefficient. **Interrupts** In systems programming, an interrupt is a signal to the processor emitted by hardware or software indicating an event that needs immediate attention. An interrupt alerts		http://www.quora.com/What-is-the-difference-between-programmed-driven-I-O-and-interrupt-driven-I-O-What-is-one-advantage-

	the processor to a high-priority condition requiring the interruption of the current code the processor is executing. The processor responds by suspending its current activities, saving its state, and executing a function called an interrupt handler (or an interrupt service routine, ISR) to deal with the event. This interruption is temporary, and, after the interrupt handler finishes, the processor resumes normal activities. **Advantage** Fast and efficient. **Disadvantage** Can be tricky to write if you are using a low level language. Can be tough to get the various pieces to work well together. Usually done by the hardware manufacturer or the OS maker e.g. Microsoft.		and-one-disadvantage-of-each https://en.wikipedia.org/wiki/Interrupt https://en.wikipedia.org/wiki/Programmed_input/output
9. **DMA** - How Direct Memory Access works and when it is useful to use it?	A DMA contains several processor registers that can be written and read by the CPU. To carry out an input, output or memory-to-memory operation, the host processor initializes the DMA controller with a count of the number of words to transfer, and the memory address to use. The CPU then sends commands to a peripheral device to initiate transfer of data. The DMA controller then provides addresses and read/write control lines to the system memory. It is useful to use it when the CPU cannot keep up with the rate of data transfer, or when the CPU needs to perform useful work while waiting for a relatively slow I/O data transfer.		Textbook p.286-288 https://en.wikipedia.org/wiki/Direct_memory_access
10. **Buses** – Please list the advantages and limitations of different types of buses (serial vs parallel with many examples).	**Parallel Bus** **Advantages** The parallel bus is characterized by high-throughput capability. Most internal operations and registers are parallel. **Limitations** Parallel buses are expensive and consume a considerable amount of space. The higher the data rate, the worse the interference, which ultimately limits the speed at which the parallel bus can operate. There is a slight difference in time delay on different lines, known as skew, as signals traverse the bus. The cost of fiber optic technology makes a parallel optical cable impractical. **Serial Bus** **Advantages** Serial buses are often set up for point-to-point connection; no addressing is required in this case. **Limitations**		Textbook p.211-213

	The throughput of a serial bus would be lower than that of a parallel bus theoretically capable of the same per line transfer rate.		
11. **Peripherals** - How computer peripherals work, including magnetic disk drives (floppy disks, hard drives), optical disk drives (CD-R, CD-RW, DVDROM, DVD+R, DVD-R, DVD+RW, DVD-RW), displays (CRT and LCD monitors) and laser printers and realize why it is important to limit the number of disk-read phases when writing programs.	**Floppy disk** is a disk storage medium composed of a disk of thin and flexible magnetic storage medium, sealed in a rectangular plastic carrier lined with fabric that removes dust particles. Floppy disks are read and written by a floppy disk drive. **Hard disk drive** is a data storage device used for storing and retrieving digital information using one or more rigid rapidly rotating disks coated with magnetic material. The platters are paired with magnetic heads arranged on a moving actuator arm, which read and write data to the platter surfaces. **Optical disc drive** is a disk drive that uses laser light or electromagnetic waves within or near the visible light spectrum as part of the process of reading or writing data to or from optical discs. **LCD** is an electronically modulated optical device made up of any number of segments controlling a layer of liquid crystals and arrayed in front of a light source (backlight) or reflector to produce images in color or monochrome. **CRT** is a vacuum tube containing one or more electron guns, and a phosphorescent screen used to view images. It has a means to accelerate and deflect the electron beam(s) onto the screen to create the images. **Laser printing** is an electrostatic digital printing process. It produces high-quality text and graphics by repeatedly passing a laser beam back and forth over a negatively charged cylindrical drum to define a differentially-charged image. The drum then selectively collects electrically charged powdered ink (toner), and transfers the image to paper, which is then heated in order to permanently fuse the text and/or imagery. It is important to limit the number of disk-read phases when writing programs, because most of the access time specified for secondary storage devices consists of seek time, so even the fastest disks are a million times slower than the slowest memory. It should be apparent that a lot of CPU instructions can be performed while waiting for a disk transfer to take place. The more disk-read phases, the longer waiting time.		https://en.wikipedia.org/wiki/Floppy_disk https://en.wikipedia.org/wiki/Hard_disk_drive https://en.wikipedia.org/wiki/Optical_disc_drive https://en.wikipedia.org/wiki/Liquid-crystal_display https://en.wikipedia.org/wiki/Cathode_ray_tube https://en.wikipedia.org/wiki/Laser_printing Textbook p.297