2.1

Operations are performed via the CPU, central processing unit. It consists of two parts: the arithmetic/logic unit or ALU (performs data manipulation) and the control unit or CU (coordinates the machine’s activities).

Registers: general-purpose and special-purpose. General purpose serve as temporary holding places for data being manipulated by the CPU.

Cache memory is a section of high-speed memory with response times similar to that of the CPU’s registers, often located within the CPU itself.

Bus: collection of wires connecting a machine’s CPU and main memory.

Machine instructions: a short list of well-chosen tasks. Three categories:

*Data Transfer:* movement of data from one location to another.
- LOAD memory -> gpr
- STORE gpr -> memory

*Arithmetic/Logic:* instructions that tell the CU to request an activity within the ALU.
Examples: AND, OR, XOR, SHIFT & ROTATE

*Control:* instructions that direct the execution of the program rather than the manipulation of data.
Examples: JUMP, BRANCH, IF-THEN

2.2

Stored-program concept: the CU is designed to extract the program from memory, decode the instructions, and execute them.

Machine-language: collection of instructions along with the coding system.

Machine instructions typically consist of 2 parts: the op-code field and the operand field.
- Op-code field: operation (STORE, SHIFT, XOR, JUMP)
- Operand field: information about what the operation is being performed on.

2.3

A computer follows a program stored in its memory by copying the instructions from memory into the control unit as needed.

PC or program counter contains the address of the next instruction to be executed.
IR or instruction register is used to hold the instruction being executed.
The CU performs its job by continually repeating an algorithm, the machine cycle. 3 steps: **fetch, decode, and execute**.

**clock** is used to synchronize various circuits and coordinate the activities in the machine cycle.

2.4

<table>
<thead>
<tr>
<th>AND</th>
<th>OR</th>
<th>XOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>10011010</td>
<td>10011010</td>
<td>10011010</td>
</tr>
<tr>
<td>11001001</td>
<td>11001001</td>
<td>11001001</td>
</tr>
<tr>
<td>10001000</td>
<td>11011011</td>
<td>01010011</td>
</tr>
</tbody>
</table>

**masking:** example subnet mask

**Shifts** can be to the left or right.  
**circular shift** is also called a **rotation**.  
**logical shift** is a technique that discards the bit that falls of the edge and always fills the hole with a 0.  
**arithmetic shifts** leave the sign bit unchanged.

2.5

Communication between a machine and other devices is normally handled through an intermediary device known as a **controller**. It converts messages and data back and forth between forms compatible with the internal characteristics of the machine and those of the peripheral device to which it is attached. Often small computers within themselves. Connects to the same bus that connects the machine’s CPU and main memory.

**DMA, direct memory access**, the ability of a controller to access main memory. Increases performance.

**buffer** is any location where one system leaves data to be picked up later by another

The central bus can become an impediment, known as the **von Neumann bottleneck**, as the CPU and the controllers compete for bus access.

Can use the same op-codes. The main memory circuitry is designed to ignore references to particular memory locations while the controller is designed to respond to references to those locations. This is called **memory-mapped I/O** because the machine’s input/output devices appear to be in various memory locations, and the “memory” addresses assigned to a controller in this manner are collectively called a **port** in that they represent a “location” through which information enters and leaves the machine.

**status word:** a bit pattern that is generated by the peripheral device and sent to the controller. The bits in the status word reflect the conditions of the device.
The rate in which bits are transferred from one computing component to another is measured in **bits per second (bps)**. **Kbps, Mbps, Gbps.**

**parallel communication:** several bits are transferred at the same time, each on a separate line. Mbps and higher

**serial communication:** is based on transferring only one bit at a time. simpler, slower. All bits are transferred over the same line, on after the other.

**modem** short for modulator-demodulator. To achieve higher transfer rates, modems combine changes in a tone’s frequency, amplitude (volume), and phase (the degree to which the transmission of the tone is delayed). Also, data compression techniques are often applied to produce apparent transfer rates of up to 57.6 kbps.
The standard computer system is composed of two (2) fundamental subsystems: the processor and the input/output (I/O) systems.

- **Processor System**: Dicated to the performance of the machine's built-in arithmetic and logical functions. It manages fetching, decoding, and executing of the encoded instructions.

- **I/O subsystem**: Manages communication with peripherals such as mouse, keyboard, monitor, printer, scanners, etc.

- **Data/Instruction Set**: Composed of the machine language, which includes the machine instruction set.

Non-Neumann machine:
1. Performs a series of instructions at a time.
2. Clearly separates the logical design from the engineering details (the components are based on their function and not merely the mechanism that achieves that function).
3. Utilized the stored program feature of the EDVAC.

**Instruction-execution cycle**:
- Fetch the next instruction
- Decode the instruction
- Execute the instruction
- Repeat
I. Fetch
1. PC has address of next instruction
2. CU copies address to MAR
3. PC is incremented (address)
4. The instruction is retrieved from the address in memory which is currently in the MAR. The instruction is stored in the MDR.

II. Decoding
5. The instruction is copied from MDR to the IR for decoding

III. Execution
6. Once the instruction is deciphered in the IR, the CU sends appropriate signals to commence its execution. Usually carried out by ALU or more general GR.

7. Return to step 1

in MHz = one million cycles per second
400 MHz = 400 million cycles per second. A single instruction may take several cycles to complete.