Week 2 – CPU Operation

Suggested Answer:

Problem 1

Step-by-Step Solution

1. **Identify the coefficients**:
   - $a = 5$
   - $b = 6$
   - $c = 1$
2. **Calculate the discriminant**:

$$\Delta = b^2 - 4ac$$

   Substitute the values of a, b, and c into the formula:

$$\Delta = 6^2 - 4 \cdot 5 \cdot 1 = 36 - 20 = 16$$

3. **Determine the nature of the roots**:
   - Since $\Delta > 0$, there are two distinct real roots.
4. **Apply the quadratic formula**:

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

   Substitute the values of a, b, and Δ into the formula:

$$x = \frac{-6 \pm \sqrt{16}}{2 \cdot 5} = \frac{-6 \pm 4}{10}$$

5. **Calculate the two roots**:
   - For the positive root:

$$x_1 = \frac{-6 + 4}{10} = \frac{-2}{10} = -0.2$$

   - For the negative root:

$$x_2 = \frac{-6 - 4}{10} = \frac{-10}{10} = -1$$

Solutions

The solutions to the quadratic equation $5x^2 + 6x + 1 = 0$ are:

$$x_1 = -0.2$$

$$x_2 = -1$$

Week 2 – CPU Operation

Problem 2

```
START
    // Input the coefficients a, b, and c
    INPUT a, b, c

    // Calculate the discriminant
    discriminant <- (b * b) - (4 * a * c)

    // Check if the discriminant is non-negative (since we are taking the square root)
    IF discriminant < 0 THEN
        PRINT "No real solutions"
    ELSE
        // Calculate the square root of the discriminant
        sqrt_discriminant <- sqrt(discriminant)

        // Calculate the solutions using the quadratic formula
        x1 <- (-b + sqrt_discriminant) / (2 * a)
        x2 <- (-b - sqrt_discriminant) / (2 * a)

        // Print the solutions
        PRINT "The solutions are: ", x1, " and ", x2
    END IF
END
```

Week 2 – CPU Operation

Problem 3

a) - Memory – to store the data

- Register – temporary storage of data during ALU operations
- PC – Program counter that indicate the address where the machine instructions of the program are stored.
- IR – Instruction register that store the instruction to be decoded
- CU – control unit that coordinator the operation of the CPU, including the decoding of the instruction.

□

CPU Instruction Execution Cycle

The CPU Instruction Execution Cycle, also known as the fetch-decode-execute cycle, is the process by which a CPU retrieves, interprets, and executes instructions. This cycle can be broken down into several steps:

1. **Fetch**:
   - **Description**: The CPU fetches the instruction from the memory location pointed to by the Program Counter (PC).
   - **Example**: Consider the instruction to compute $x1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$. The CPU fetches this instruction from memory.
2. **Decode**:
   - **Description**: The CPU decodes the fetched instruction to understand what action is required. The instruction is interpreted by the Control Unit (CU), which determines the operation to be performed and the operands involved.
   - **Example**: The fetched instruction is decoded to identify that it involves arithmetic operations such as negation, addition, multiplication, subtraction, and square-root calculation.
3. **Execute**:
   - **Description**: The CPU executes the instruction by performing the specified operations. This involves using the Arithmetic Logic Unit (ALU) for arithmetic and logical operations.

4. **Store**:
    - **Description**: The result of the executed instruction is stored back into a specified register or memory location.
    - **Example**: The result $x1$ is stored in register $R5$ or in memory for later use.
5. **Update Program Counter**:
    - **Description**: The Program Counter (PC) is updated to point to the next instruction to be fetched.
    - **Example**: The PC is incremented to the address of the next instruction in the sequence.

Problem 4

Example:

Step-by-Step Instructions Using 4 Registers

1. **First Step: Compute b^2:**
   - **R0** = Copy b into Register R0.
   - **R1** = Copy b into Register R1.
   - **R2** = Multiply $R0$ by $R1$ (using ALU), store the result in $R2$.

   Now, $R2$ holds b^2.

2. **Second Step: Compute $4ac$:**
   - **R0** = Copy a into Register R0.
   - **R1** = Copy 4 into Register R1.
   - **R3** = Multiply $R0$ by $R1$ (using ALU), store the result in $R3$ (i.e., $4a$).
   - **R0** = Copy c into Register R0.
   - **R1** = Multiply $R3$ by $R0$ (using ALU), store the result in $R1$ (i.e., $4ac$).

   Now, $R1$ holds $4ac$.

3. **Third Step: Compute $b^2 - 4ac$:**
   - **R0** = Subtract $R1$ from $R2$ (using ALU), store the result in $R0$.

   Now, $R0$ holds $b^2 - 4ac$.
4. **Fourth Step: Compute $\sqrt{b^2 - 4ac}$:**
   - **R1** = Compute the square root of $R0$ (using ALU), store the result in $R1$.

   Now, $R1$ holds $\sqrt{b^2 - 4ac}$.
5. **Fifth Step: Compute $-b$:**
   - **R2** = Copy b into Register $R2$.
   - **R3** = Negate $R2$ (using ALU), store the result in $R3$ (i.e., $-b$).

   Now, $R3$ holds $-b$.
6. **Sixth Step: Compute $-b + \sqrt{b^2 - 4ac}$:**
   - **R2** = Add $R3$ (which holds $-b$) to $R1$ (which holds $\sqrt{b^2 - 4ac}$) (using ALU), store the result in $R2$.

   Now, $R2$ holds $-b + \sqrt{b^2 - 4ac}$.
7. **Seventh Step: Compute $\frac{-b+\sqrt{b^2-4ac}}{2a}$:**
   - **R0** = Copy a into Register $R0$ (before the multiplication).
   - **R3** = Copy 2 into Register $R3$.
   - **R1** = Multiply $R3$ by $R0$ (using ALU), store the result in $R1$ (i.e., $2a$).
   - **R3** = Divide $R2$ by $R1$ (using ALU), store the result in $R3$.

   Now, $R3$ holds $\frac{-b+\sqrt{b^2-4ac}}{2a}$.