Three Stage AND operation

EXP NO: 39

AIM:

To simulate 3-stage and 4-stage pipelines for logical and arithmetic operations, and calculate their performance measure.

ALGORITHM:

3-Stage Pipeline (Logical AND Operation)

1. **Start**
2. **Initialize** counter to track the number of cycles.
3. **Input** the first operand (a), increment counter.
4. **Input** the second operand (b), increment counter.
5. **Perform** the logical AND operation: res = a and b, increment counter.
6. **Output** the result.
7. **Increment** counter by 2 for printing results and waiting time.
8. **Input** the number of instructions (INS).
9. **Calculate** the performance measure as INS / counter.
10. **Output** the performance measure.
11. **End**

4-Stage Pipeline (Arithmetic Operations)

1. **Start**
2. **Initialize** counter to track the number of cycles.
3. **Input** the first operand (a), increment counter.
4. **Input** the second operand (b), increment counter.
5. **Prompt** for operation choice:
    - 1 for Addition
    - 2 for Subtraction
    - 3 for Multiplication
    - 4 for Division
6. **Perform** the selected operation based on choice:
    - **Addition:** res = a + b, increment counter.
    - **Subtraction:** res = a - b, increment counter.
    - **Multiplication:** res = a * b, increment counter.
    - **Division:** Check if b is zero. If not, perform division: res = a / b, increment counter.
7. **Handle** invalid inputs by skipping counter increment.

8. **Output** the result.
9. **Increment** counter by 3 for additional processing.
10. **Input** the number of instructions (ins).
11. **Calculate** the performance measure as ins / counter.
12. **Output** the performance measure.
13. **End**

PROGRAM:

```
#include <stdio.h>

int main() {
   // Stage 1: Initialize counter and variables
   int counter = 1; // Cycle count initialization
   int a, b, res, INS; // Variables for input numbers, result, and number of instructions
   int performance_measure;

   // Stage 2: Input numbers
   printf("ENTER NUMBER-1: ");
   scanf("%d", &a);
   counter += 1; // Increment counter for input operation

   printf("ENTER NUMBER-2: ");
   scanf("%d", &b);
   counter += 1; // Increment counter for input operation

   // Stage 3: Perform AND operation
   res = a & b; // Bitwise AND operation
   counter += 1; // Increment counter for operation

   // Display result
   printf("Result of AND operation: %d\n", res);
   counter += 2; // Increment counter for display operation

   // Stage 4: Input number of instructions
   printf("Enter number of instructions: ");
```

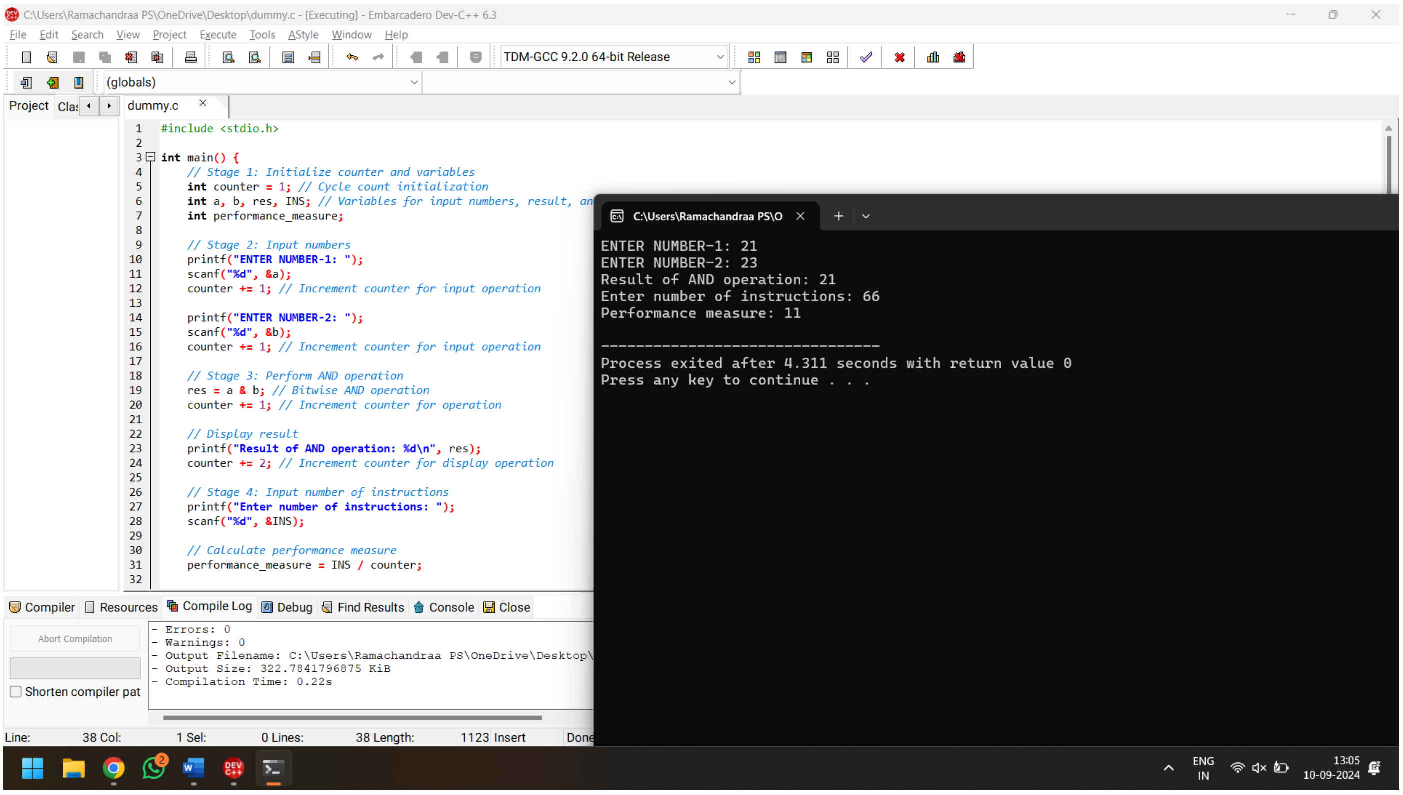
```
    scanf("%d", &INS);


    // Calculate performance measure
    performance_measure = INS / counter;


    // Display performance measure
    printf("Performance measure: %d\n", performance_measure);


    return 0;
}
```

INPUT & OUTPUT:



C:\Users\Ramachandraa PS\OneDrive\Desktop\dummy.c - [Executing] - Embarcadero Dev-C++ 6.3

```c
#include <stdio.h>

int main() {
    // Stage 1: Initialize counter and variables
    int counter = 1; // Cycle count initialization
    int a, b, res, INS; // Variables for input numbers, result, an
    int performance_measure;

    // Stage 2: Input numbers
    printf("ENTER NUMBER-1: ");
    scanf("%d", &a);
    counter += 1; // Increment counter for input operation

    printf("ENTER NUMBER-2: ");
    scanf("%d", &b);
    counter += 1; // Increment counter for input operation

    // Stage 3: Perform AND operation
    res = a & b; // Bitwise AND operation
    counter += 1; // Increment counter for operation

    // Display result
    printf("Result of AND operation: %d\n", res);
    counter += 2; // Increment counter for display operation

    // Stage 4: Input number of instructions
    printf("Enter number of instructions: ");
    scanf("%d", &INS);

    // Calculate performance measure
    performance_measure = INS / counter;

```

```
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Ramachandraa PS\OneDrive\Desktop\
- Output Size: 322.7841796875 KiB
- Compilation Time: 0.22s
```

```
ENTER NUMBER-1: 21
ENTER NUMBER-2: 23
Result of AND operation: 21
Enter number of instructions: 66
Performance measure: 11

--------------------------------
Process exited after 4.311 seconds with return value 0
Press any key to continue . . .
```

RESULT: Thus, the program was executed successfully using DevC++