Four Stage AND operation

EXP NO: 40

AIM:

To simulate a 4-stage pipeline for arithmetic operations (addition, subtraction, multiplication, division) and calculate its performance measure.

ALGORITHM:

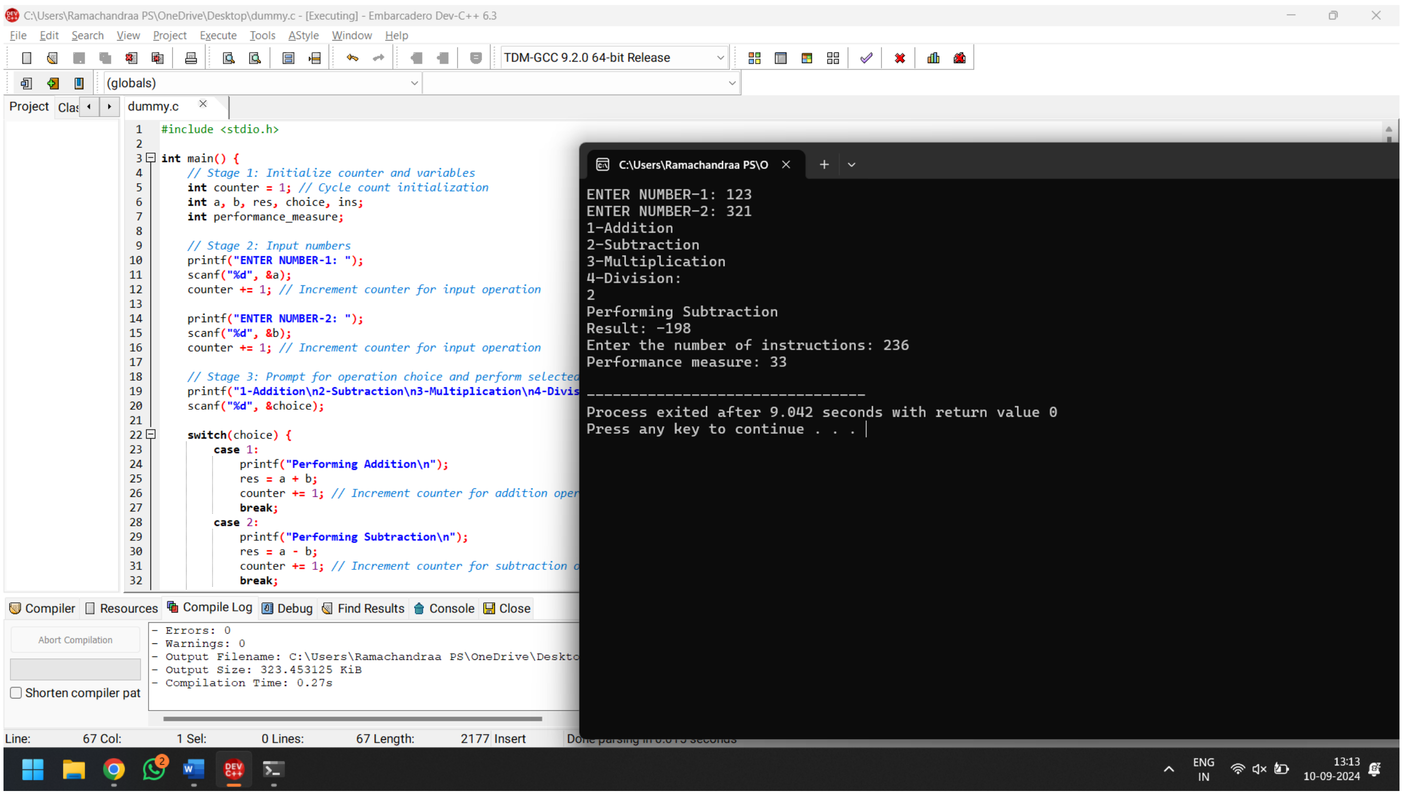
4-Stage Pipeline (Arithmetic Operations)

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 (including checking for invalid operations).
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, choice, ins;
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: Prompt for operation choice and perform selected operation
printf("1-Addition\n2-Subtraction\n3-Multiplication\n4-Division:\n");
scanf("%d", &choice);

switch(choice) {
  case 1:
    printf("Performing Addition\n");
    res = a + b;
    counter += 1; // Increment counter for addition operation
    break;
  case 2:
    printf("Performing Subtraction\n");
    res = a - b;
    counter += 1; // Increment counter for subtraction operation
    break;
  case 3:
    printf("Performing Multiplication\n");
    res = a * b;
    counter += 1; // Increment counter for multiplication operation
    break;
  case 4:
    if (b == 0) {
            printf("Denominator can't be Zero\n");
            // Handle division by zero case
            counter += 0; // No increment for invalid case
        } else {
            printf("Performing Division\n");
            res = a / b;
            counter += 1; // Increment counter for division operation
        }
        break;
    default:
        printf("Invalid choice\n");
        counter += 0; // No increment for invalid choice
        break;
    }

    // Stage 4: Display result and calculate performance measure
    printf("Result: %d\n", res);
    counter += 3; // Increment counter for additional processing

    printf("Enter the number of instructions: ");
    scanf("%d", &ins);

    performance_measure = ins / counter;
    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
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Project Clas
dummy.c
C:\Users\Ramachandraa PS\O
Compiler Resources Compile Log Debug Find Results Console Close
Abort Compilation
Shorten compiler pat
Line: 67 Col: 1 Sel: 0 Lines: 67 Length: 2177 Insert
Done parsing in 0.013 seconds
ENG IN
13:13
10-09-2024

```c
#include <stdio.h>

int main() {
    // Stage 1: Initialize counter and variables
    int counter = 1; // Cycle count initialization
    int a, b, res, choice, ins;
    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: Prompt for operation choice and perform selected
    printf("1-Addition\n2-Subtraction\n3-Multiplication\n4-Divis
    scanf("%d", &choice);

    switch(choice) {
        case 1:
            printf("Performing Addition\n");
            res = a + b;
            counter += 1; // Increment counter for addition oper
            break;
        case 2:
            printf("Performing Subtraction\n");
            res = a - b;
            counter += 1; // Increment counter for subtraction c
            break;
```

```
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Ramachandraa PS\OneDrive\Deskto
- Output Size: 323.453125 KiB
- Compilation Time: 0.27s
```

```
ENTER NUMBER-1: 123
ENTER NUMBER-2: 321
1-Addition
2-Subtraction
3-Multiplication
4-Division:
2
Performing Subtraction
Result: -198
Enter the number of instructions: 236
Performance measure: 33

--------------------------------
Process exited after 9.042 seconds with return value 0
Press any key to continue . . .
```

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