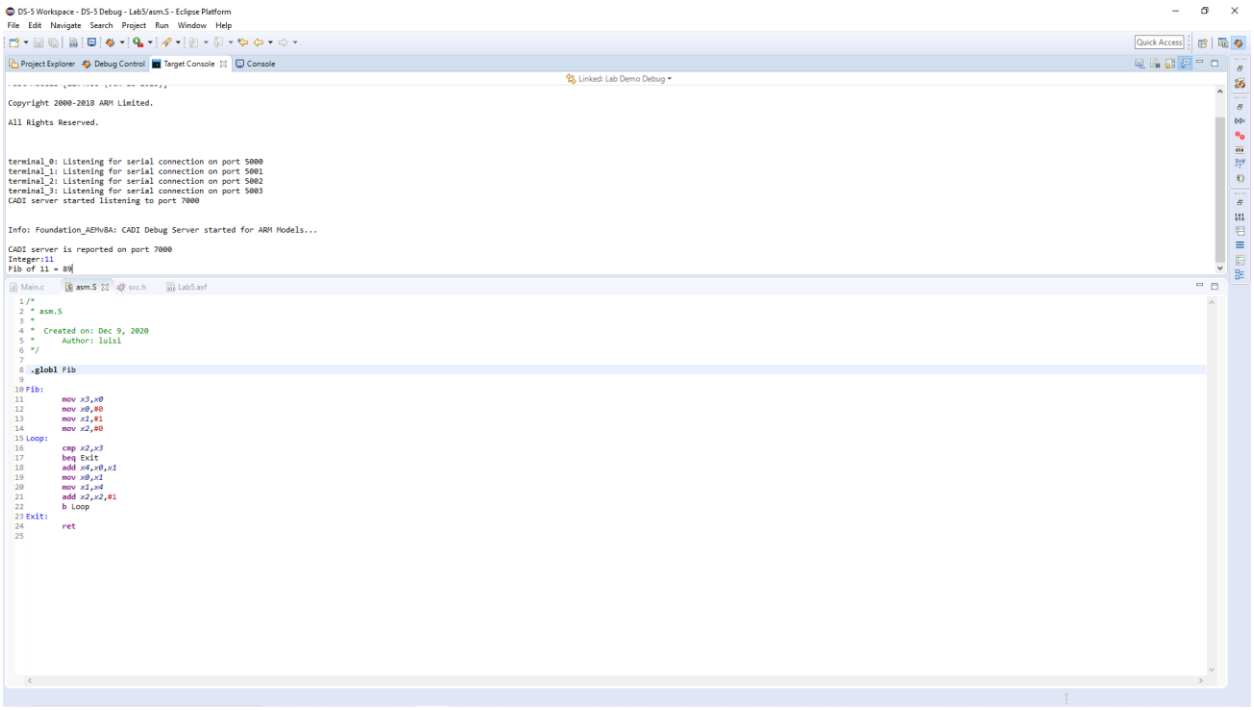
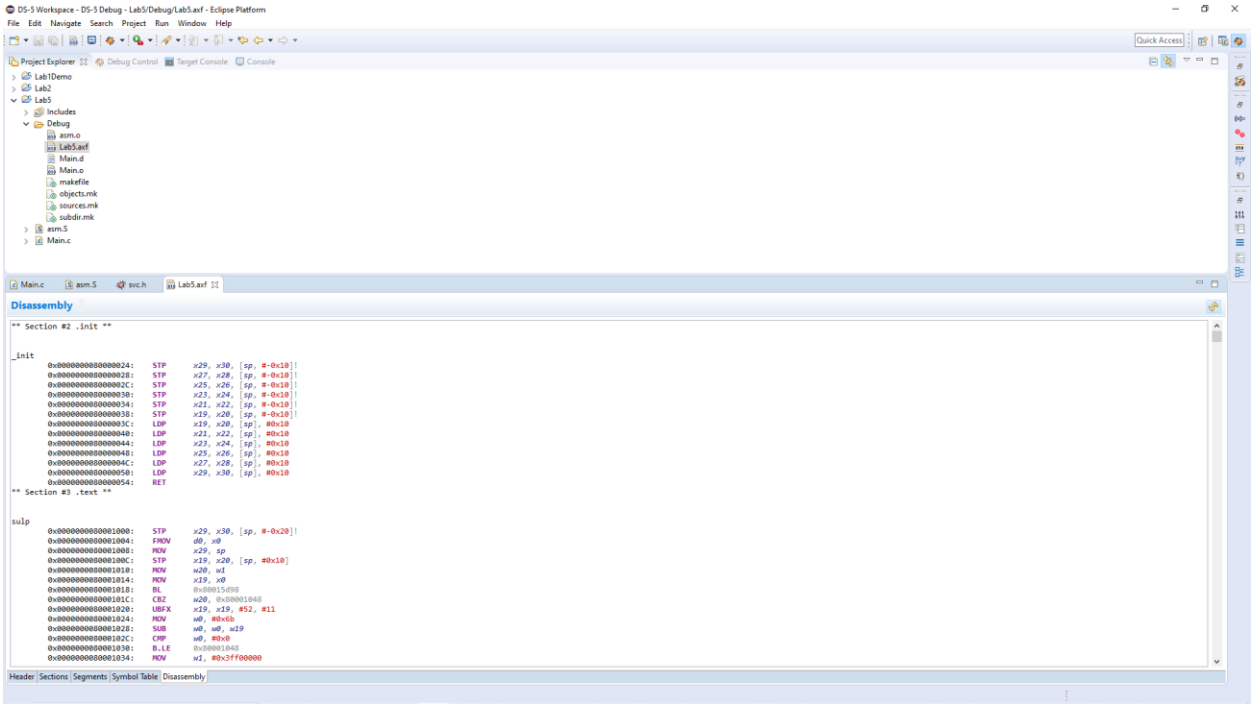
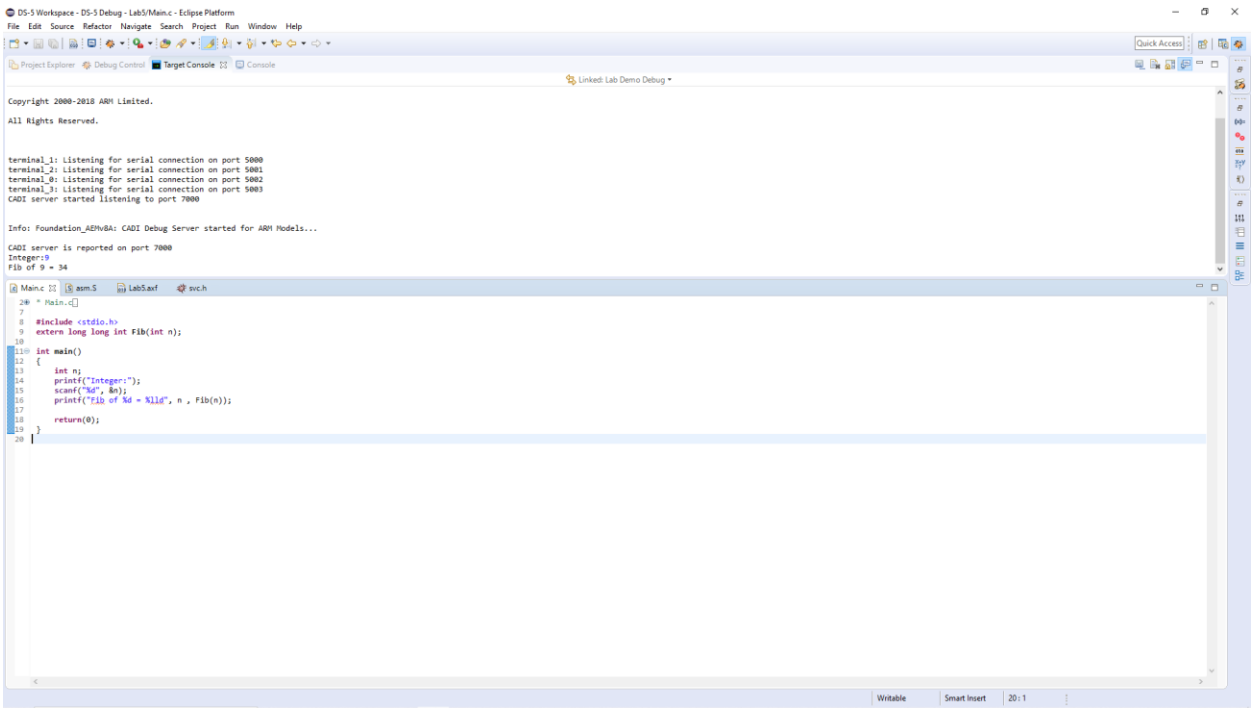
Lab 5:

C program and assembly program along with screenshots (building and executing in debug window displaying the output along with project explorer tab).




DS-5 Workspace - DS-5 Debug - Lab5/Debug/Lab5.axf - Eclipse Platform
File Edit Navigate Search Project Run Window Help
Project Explorer
Lab1Demo
Lab2
Lab5
Includes
Debug
asm.o
Lab5.axf
Main.d
Main.o
makefile
objects.mk
sources.mk
subdir.mk
asm.S
Main.c
Main.c
asm.S
svc.h
Lab5.axf
Disassembly
** Section #2 .init **
_init
0x0000000080000024: STP x29, x30, [sp, #-0x10]!
0x0000000080000028: STP x27, x28, [sp, #-0x10]!
0x000000008000002C: STP x25, x26, [sp, #-0x10]!
0x0000000080000030: STP x23, x24, [sp, #-0x10]!
0x0000000080000034: STP x21, x22, [sp, #-0x10]!
0x0000000080000038: STP x19, x20, [sp, #-0x10]!
0x000000008000003C: LDP x19, x20, [sp], #0x10
0x0000000080000040: LDP x21, x22, [sp], #0x10
0x0000000080000044: LDP x23, x24, [sp], #0x10
0x0000000080000048: LDP x25, x26, [sp], #0x10
0x000000008000004C: LDP x27, x28, [sp], #0x10
0x0000000080000050: LDP x29, x30, [sp], #0x10
0x0000000080000054: RET
** Section #3 .text **
sulp
0x0000000080001000: STP x29, x30, [sp, #-0x20]!
0x0000000080001004: FMOV d0, x0
0x0000000080001008: MOV x29, sp
0x000000008000100C: STP x19, x20, [sp, #0x10]
0x0000000080001010: MOV w20, w1
0x0000000080001014: MOV x19, x0
0x0000000080001018: BL 0x80015d98
0x000000008000101C: CBZ w20, 0x80001048
0x0000000080001020: UBFX x19, x19, #52, #11
0x0000000080001024: MOV w0, #0x6b
0x0000000080001028: SUB w0, w0, w19
0x000000008000102C: CMP w0, #0x0
0x0000000080001030: B.LE 0x80001048
0x0000000080001034: MOV w1, #0x3ff00000
Header Sections Segments Symbol Table Disassembly


DS-5 Workspace - DS-5 Debug - Lab5/Main.c - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer
Debug Control
Target Console
Console
Linked: Lab Demo Debug
Copyright 2000-2018 ARM Limited.
All Rights Reserved.
terminal_1: Listening for serial connection on port 5000
terminal_2: Listening for serial connection on port 5001
terminal_0: Listening for serial connection on port 5002
terminal_3: Listening for serial connection on port 5003
CADI server started listening to port 7000
Info: Foundation_AEMv8A: CADI Debug Server started for ARM Models...
CADI server is reported on port 7000
Integer:9
Fib of 9 = 34
Main.c
asm.S
Lab5.axf
svc.h
* Main.c
#include <stdio.h>
extern long long int Fib(int n);
int main()
{
int n;
printf("Integer:");
scanf("%d", &n);
printf("Fib of %d = %lld", n , Fib(n));
return(0);
}
Writable
Smart Insert
20 : 1

In brief mention the different variables and registers that are used by the program by making use of the debug control window and the .axf file. Also provide the start address and end address of each function (init, start, main, and the address of all the implemented functions)

The value n in the c program represents X0 in the program. In the first part of the assembly code, I move X0 to X3, so that the value entered by the user would be stored. I decided to go with this method after initially having trouble having X1 giving the N value, when X0 had the value given by the user, and having a better explanation from Geraldine on what exactly the initial variables should be.

I decided to give X0-X2 a value, X0 would be 0, X1 would be 1 and X2 would be 0. X0 and X1 represents the first two numbers in the Fibonacci sequence. The loop begins with comparing X2, 0, and X3, the integer the user inputted. If the value of X2 and X3 are equal, it will stop the loop and give the Fibonacci value. If they are not equal than the program will begin by having X4 equal to the value of X0 + X1. Afterwards, the value of X1 would move to the X0 register, and the value of X4 would move to the X1 register. By moving these values, it duplicates how the value of any Fibonacci integer would be the addition of the previous two values before the selected N. Finally, 1 would be added to X2 and the loop would repeat until X2 and X3 are the same value.

/*Main.c*/

```c
#include <stdio.h>
 extern long long int Fib(int n);

 int main()
 {
        int n;
        printf("Integer:");
        scanf("%d", &n);
        printf("Fib of %d = %lld", n , Fib(n));

        return(0);
 }
```

/*asm.S*/

```
.globl Fib

Fib:
        mov x3,x0
        mov x0,#0
        mov x1,#1
        mov x2,#0
Loop:
        cmp x2,x3
        beq Exit
        add x4,x0,x1
        mov x0,x1
        mov x1,x4
        add x2,x2,#1
        b Loop
Exit:
        ret
```