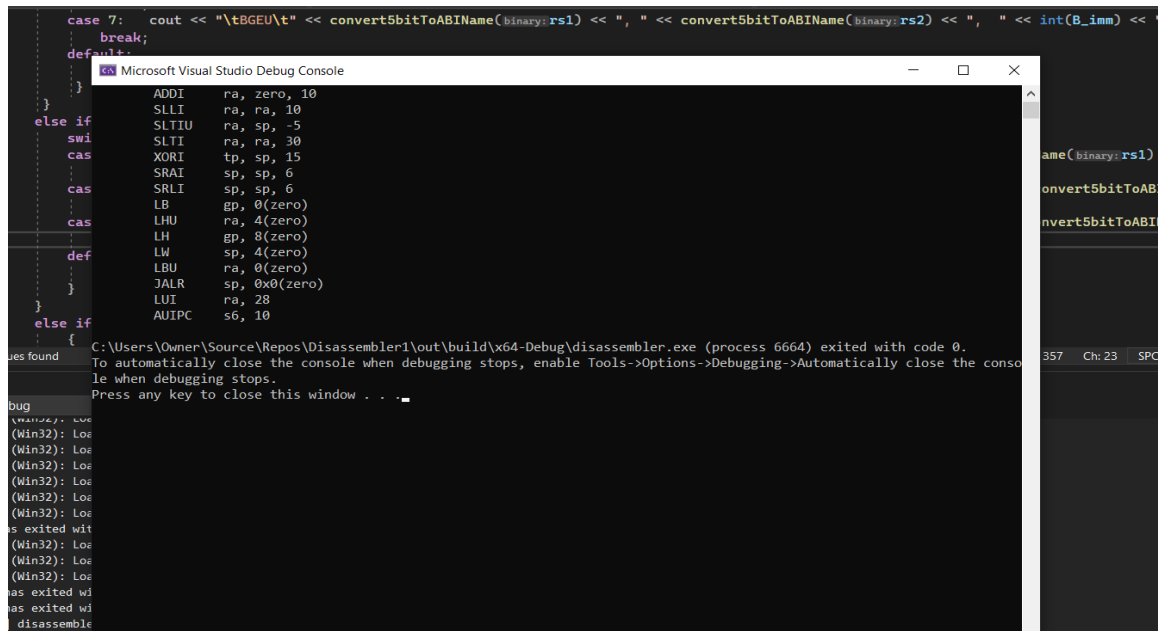


I instructions (32 bits)

Instruction name	Hex representation
addi x1, x0, 10	0x00a00093
slli x1, x1, 1	0x00a09093
sltiu x1, x2,	0xffb13093
slti x1, x1, 30	0x01e0a093
xori x4, x2, 15	0x00f14213
srai x2, x2, 6	0x40615113
srli x2, x2, 6	0x00615113
lb x3, 0(x0)	0x00000183
lhu x1, 4(x0)	0x00405083
lh x3, 8(x0)	0x00801183
lw x2, 4(x0)	0x00402103
lbu x1,0(x0)	0x00004083
jalr x2, 0(x0)	0x00000167
lui x1, 40	0x000280b7
auipc s6, 16	0x00010b17

Screenshot:



## R instructions (32 bits)

Instruction	Hex representation
sub x2, x1, x2	0x40208133
add x2, x1, x2	0x00208133
sll x4,x3,x2	0x00219233
slt x1,x2,x1	0x001120b3
sltu x2,x2,x3	0x00313133
xor x3,x2,x3	0x003141b3
sra x3,x2,x1	0x01151b3
srl x2,x2,x4	0x00415133
or x1, x2, x3	0x003160b3
and x3,x2,x3	0x003171b3

screenshot:

The screenshot shows a Visual Studio IDE with a C++ file named `main.cpp` open. The code is as follows:

```

96  inFile.open("C:/Users/Owner/Desktop/riscv1.txt");
97  string inst;
98  unsigned int t;
99
100 while (!inFile.eof()) {
101     inFile >> inst;
102     stringstream ss;
103     ss << inst;
104     ss >> hex >> t;
105     instDecExec(t);
106 }
107
108 return 0;
109

```

Below the code editor, the `Microsoft Visual Studio Debug Console` is open, displaying the assembly output for the program. The assembly instructions are:

```

SUB    sp, ra, sp
ADD    sp, ra, sp
SLL    tp, gp, sp
SLT    ra, sp, ra
SLTU   sp, sp, gp
XOR    gp, sp, gp
SRA    gp, sp, ra
SRL    sp, sp, tp
OR     ra, sp, gp
AND    gp, sp, gp

```

Below the assembly instructions, the debug console shows the following message:

```

C:\Users\Owner\source\repos\Disassembler\out\build\x64-Debug\disassembler.exe (process 2392) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

```

At the bottom of the debug console, the output from the debug session is shown:

```

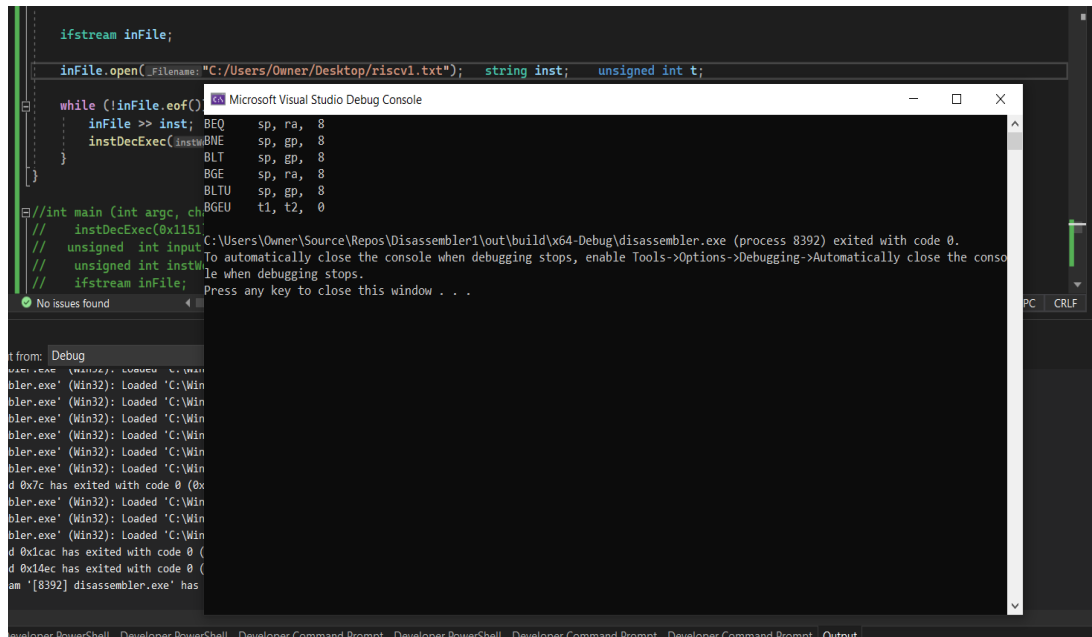
output from: Debug
disassembler.exe (Win32): Loaded 'C:\Windows\System32\
disassembler.exe (Win32): Loaded 'C:\Windows\System32\
disassembler.exe (Win32): Loaded 'C:\Windows\System32\
disassembler.exe (Win32): Loaded 'C:\Windows\System32\
disassembler.exe (Win32): Loaded 'C:\Windows\System32\
disassembler.exe (Win32): Loaded 'C:\Windows\System32\
Thread 0x12e0 has exited with code 0 (0x0).
disassembler.exe (Win32): Loaded 'C:\Windows\System32\
disassembler.exe (Win32): Loaded 'C:\Windows\System32\

```

## B instructions and J instructions

Instruction	Hex representation
beq x2,x1,L3	0x00110463
bne x2,x3,L4	0x00311463
blt x2,x3,L2	0x00314463
bge x2,x1,L2	0x00115463
bltu x2,x3,L3	0x00316463
L1: bgeu t1,t2,l1	0x00737063
jal x1,L5	0x008000ef

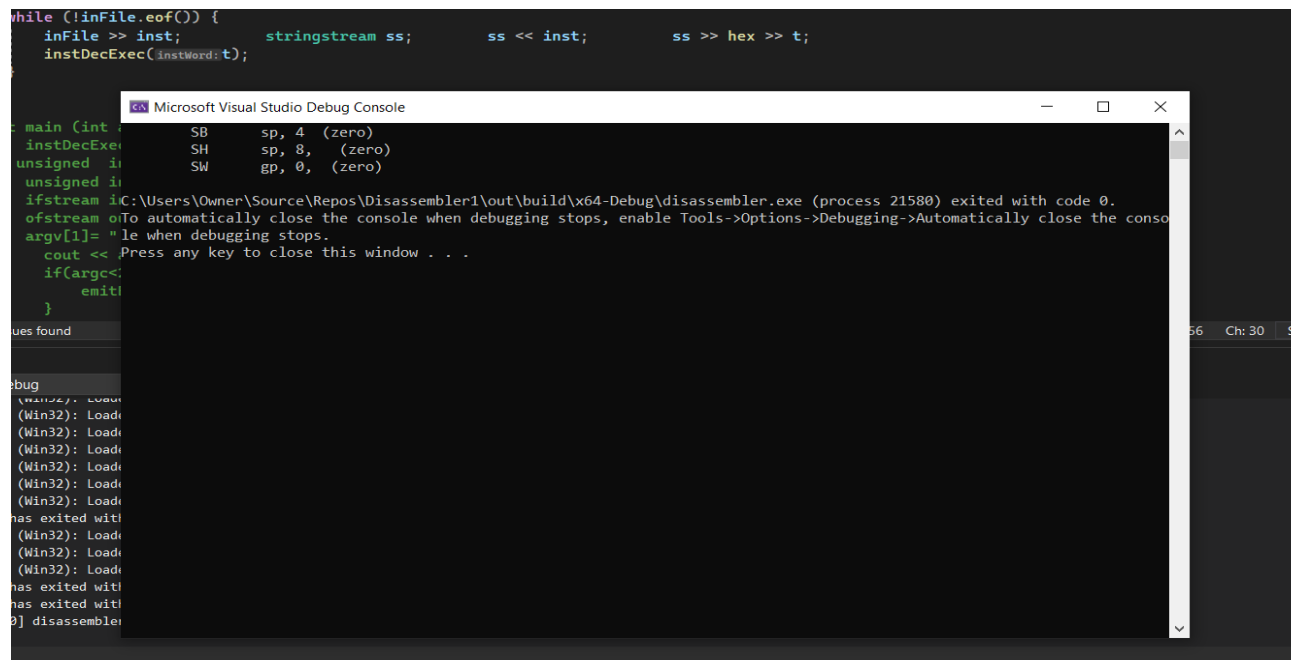
screenshot:



S instructions:

instruction	Hex representation
sb x2, 4(x0)	0x00200223
sh x2, 8(x0)	0x00201423
sw x3, 0(x0)	0x00302023

Screenshot:



The screenshot shows a Visual Studio IDE with a C++ source file and a Microsoft Visual Studio Debug Console window. The source file contains the following code:

```
while (!inFile.eof()) {  
    inFile >> inst;    stringstream ss;    ss << inst;    ss >> hex >> t;  
    instDecExec(instWord:t);  
}  
  
int main (int argc, char* argv[]) {  
    instDecExec(SB, sp, 4, (zero));  
    instDecExec(SH, sp, 8, (zero));  
    unsigned int gp = 0;    instDecExec(SW, gp, 0, (zero));  
    ifstream iC:\\Users\\Owner\\Source\\Repos\\Disassembler1\\out\\build\\x64-Debug\\disassembler.exe (process 21580) exited with code 0.  
    ofstream oC:\\Users\\Owner\\Source\\Repos\\Disassembler1\\out\\build\\x64-Debug\\disassembler.exe (process 21580) exited with code 0.  
    argv[1] = "le when debugging stops."  
    cout << "Press any key to close this window . . ."  
    if (argc < 2) {  
        emitl  
    }  
}
```

The Debug Console window displays the following output:

```
(Win32): Load  
(Win32): Load  
(Win32): Load  
(Win32): Load  
(Win32): Load  
(Win32): Load  
(Win32): Load  
has exited with  
(Win32): Load  
(Win32): Load  
(Win32): Load  
has exited with  
has exited with  
0] disassembler
```

**Code used for test run:**

(file contains one instruction per line in hexadecimal)

```
int main(){

    ifstream inFile;

    inFile.open("riscv1.txt");
    string inst;
    unsigned int t;

    while(!inFile.eof()){
        inFile>>inst;
        stringstream ss;
        ss<<inst;
        ss>>hex >> t;
        instDecExec(t);
    }
}
```