

ALUNOS: Luciano Bonzatto Junior

Hadryan Salles

DATA: 21/12/2022

DESCRIÇÃO:

https://github.com/lucianobonzatto/vhdl_fpu

Construção de uma fpu de 32 bits que pode ser acessada via socket utilizando FPGA que possui as quatro operações básicas soma, subtração, multiplicação e divisão. Para a fpu utilizamos uma biblioteca presente na opencores.org.

utiliza-se floats segundo a IEEE:

- 1bit de sinal
- 8bits de expoente
- 23bits de mantissa

Como interação, o server recebe uma string e envia para o NIOS que utiliza para formular os valores e a operação com isso, o NIOS envia para user hardware esses valores, salvando eles nos seus respectivos registradores e então le o resultado obtido e envia ele ao server.

ARQUIVOS:

- tcpserver_fpu -> server em python que a FPGA se conecta
- DE2_NET -> projeto do Quartus II versão 13
- DE2_NET/vhdl -> arquivos VHDL que fazem a conexão com o NIOS
- DE2_NET/fpu -> biblioteca utilizada para a execução da fpu de 32bits
- DE2_NET/simulation -> arquivos de simulação do multisim utilizados para testar a biblioteca
- DE2_NET/software -> workspace do eclipse onde esta o projeto do NIOS

COMO RODAR:

- 1 Utilize o Quartus II versão 13.0sp1
- 2 Vá na aba Files do Project Navigator e delete o arquivo system_0/synthesis/system_0.qip
- 3 Abre o Qsys em Tools->Qsys
- 4 Selecione o arquivo system_0.qsys e clique em Abrir
- 5 De OK ou Close no que aparecer
- 6 Va para aba Generation e clique em Generate
- 7 Feche o Qsys
- 8 No Project Navigator, aba Files, clique com o botão direito em Files->Add/Remove Files in Project
- 9 Clique no [...], entre em system_0/synthesis, mude o tipo de arquivo para todos, selecione o arquivo system_0.qip

10 Clique em Add-> Apply-> OK

11 Abra o Programmer, aperte Auto Detect e aperte Start, caso o Progress esteja verde, continue caso contrário verifique conexão da JTAG

12 Deixe aberto o Programmer e a janela "OpenCore Plus Status" e inicie o Eclipse em Tools -> Nios II Software Build Tools for Eclipse

13 No Workspace selecione SeuCaminhoAtéaPasta\DE2_NET\software

14 Agora existem 2 opções:

14.1 Se os projetos server1 e server1_bsp estão aparecendo e são expandíveis (pastinha cor amarela):

14.1.1 Clique com o botão direito em cada projeto e clique em Clean Project

14.1.2 Abra o arquivo iniche_init.c, dentro da função SSSInitialTask irá encontrar 2 linhas de código e fazer alterações:

- `sa.sin_port = htons(7777);`

- `res = inet_pton(AF_INET, "192.168.1.110", &sa.sin_addr);`

Também atualize o IP no servidor (tcpserver_fpu)

14.1.3 Vá em Project -> Build All

14.1.4 Se tudo compilar sem erros, é possível fazer Botão direito em simple_socket -> Run As...->Nios II Hardware

14.1.5 Se der algum erro na hora de rodar apenas, tente de novo

14.1.6 Se rodar como deve, algo assim deve aparecer no Nios II Console:

InterNiche Portable TCP/IP, v3.1

Copyright 1996-2008 by InterNiche Technologies. All rights reserved.

Your Ethernet MAC address is 00:07:ed:ff:15

prepped 1 interface, initializing...

dm9ka_init

Created "Inet main" task (Prio: 2)

Created "clock tick" task (Prio: 3)

mctest init called

IP address of et1 : 10.3.2.175

Acquired IP address via DHCP client for interface: et1

IP address : 192.168.1.101

Subnet Mask: 255.255.255.0

Gateway : 192.168.1.1

Simple Socket Server starting up

estou aqui

Socket criado

14.1.7 Se deu certo, precisa testar com um server agora. Primeiro pare o programa do Eclipse.

14.1.8 Pegue o tcpserver.py (fazendo alterações de ip e porta dentro) fornecido e rode ele fazendo o comando python .\tcpserver.py na linha de comando no local que estiver o arquivo

14.1.9 Rode novamente com Run As->Nios II Hardware

14.1.10 Se o roteador não bloqueou nada, está pronto!

14.2 Se os projetos server1 e server1_bsp estão aparecendo e NÃO são expandíveis (pastinha cor azul):

14.2.1 Delete ambos com botão direito -> Delete (sem deletar do computador)

14.2.2 Botão direito no Project Explorer->Import...->Nios II Software Build Tools Project-> Import Custom Makefile for Nios II Software Build Tools Project-> Next-> Browse...->

Vá para a pasta do projeto DE2_NET/software/server1_bsp->OK/Abrir

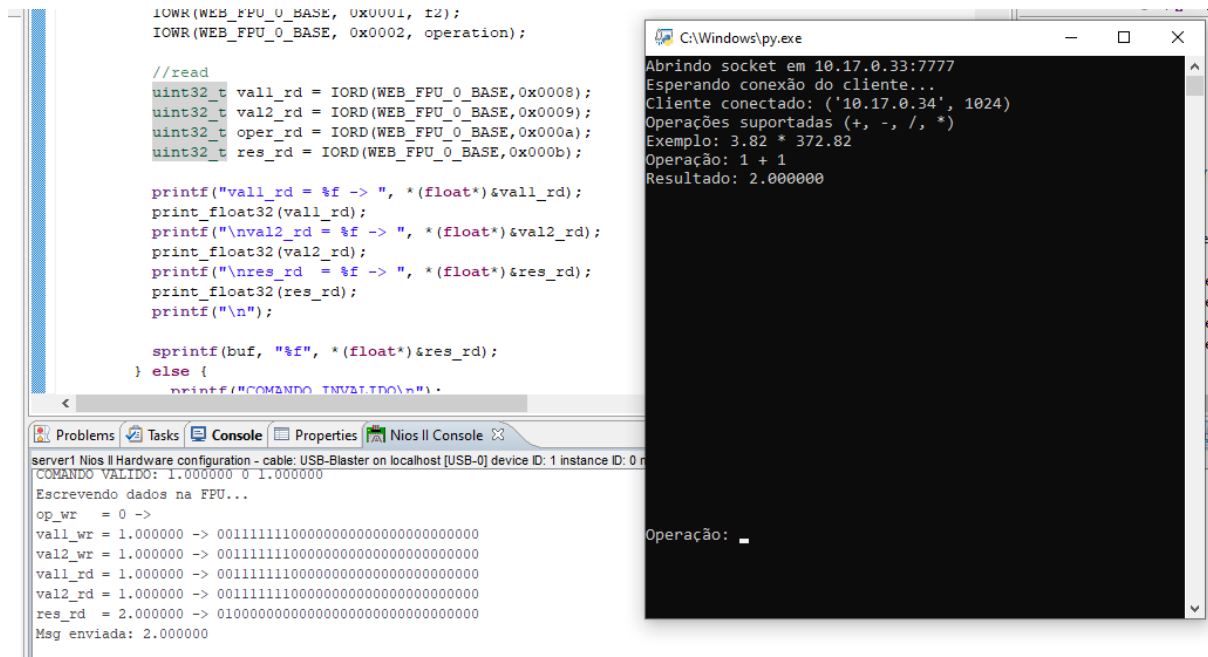
Project name: server1_bsp

Faça a mesma coisa para o server1

14.2.3 Retorne a partir de 14.1.1

EXEMPLOS DE EXECUÇÃO:

1 - soma:



The image shows a screenshot of the Nios II IDE. The left pane displays the source code for a program that reads two floating-point values from memory, performs an addition, and prints the result. The right pane shows the console output, which includes the program's execution steps and the final result of the addition.

```
IOWR(WEB_FPU_0_BASE, 0x0001, 12);
IOWR(WEB_FPU_0_BASE, 0x0002, operation);

//read
uint32_t val1_rd = IORD(WEB_FPU_0_BASE, 0x0008);
uint32_t val2_rd = IORD(WEB_FPU_0_BASE, 0x0009);
uint32_t oper_rd = IORD(WEB_FPU_0_BASE, 0x000a);
uint32_t res_rd = IORD(WEB_FPU_0_BASE, 0x000b);

printf("val1_rd = %f -> ", *(float*)&val1_rd);
printf("val2_rd = %f -> ", *(float*)&val2_rd);
printf("val2_rd = %f -> ", *(float*)&val2_rd);
printf("nres_rd = %f -> ", *(float*)&res_rd);
printf("n");

sprintf(buf, "%f", *(float*)&res_rd);
} else {
    printf("COMANDO INVÁLIDO\n");
}
```

server1 Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0
COMANDO VALIDO: 1.000000 0 1.000000
Escrevendo dados na FPU...
op_wr = 0 ->
val1_wr = 1.000000 -> 00111111100000000000000000000000
val2_wr = 1.000000 -> 00111111100000000000000000000000
val1_rd = 1.000000 -> 00111111100000000000000000000000
val2_rd = 1.000000 -> 00111111100000000000000000000000
res_rd = 2.000000 -> 01000000000000000000000000000000
Msg enviada: 2.000000

CA\Windows\py.exe
Abrindo socket em 10.17.0.33:7777
Esperando conexão do cliente...
Cliente conectado: ('10.17.0.34', 1024)
Operações suportadas (+, -, /, *)
Exemplo: 3.82 * 372.82
Operação: 1 + 1
Resultado: 2.000000
Operação: _

```
IOWR(WEB_FPU_0_BASE, 0x0001, 12);
IOWR(WEB_FPU_0_BASE, 0x0002, operation);

//read
uint32_t val1_rd = IORD(WEB_FPU_0_BASE, 0x0008);
uint32_t val2_rd = IORD(WEB_FPU_0_BASE, 0x0009);
uint32_t oper_rd = IORD(WEB_FPU_0_BASE, 0x000a);
uint32_t res_rd = IORD(WEB_FPU_0_BASE, 0x000b);

printf("val1_rd = %f -> ", *(float*)&val1_rd);
print_float32(val1_rd);
printf("\nval2_rd = %f -> ", *(float*)&val2_rd);
print_float32(val2_rd);
printf("\nres_rd = %f -> ", *(float*)&res_rd);
print_float32(res_rd);
printf("\n");

sprintf(buf, "%f", *(float*)&res_rd);
} else {
    printf("COMANDO INVÁLIDO\n");
}

server1 Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0
COMANDO VALIDO: 10000.000000 0 123456.000000
Escrevendo dados na FPU...
op_wr = 0 ->
val1_wr = 10000.000000 -> 01000110000111000100000000000000
val2_wr = 123456.000000 -> 01000111111100010010000000000000
val1_rd = 10000.000000 -> 01000110000111000100000000000000
val2_rd = 123456.000000 -> 01000111111100010010000000000000
res_rd = 133456.000000 -> 01001000000001001010100000000000
Msg enviada: 133456.000000
```

```
C:\Windows\py.exe

Operação: 10000 + 123456
Resultado: 133456.000000

Operação: _
```

```
IOWR(WEB_FPU_0_BASE, 0x0001, 12);
IOWR(WEB_FPU_0_BASE, 0x0002, operation);

//read
uint32_t val1_rd = IORD(WEB_FPU_0_BASE, 0x0008);
uint32_t val2_rd = IORD(WEB_FPU_0_BASE, 0x0009);
uint32_t oper_rd = IORD(WEB_FPU_0_BASE, 0x000a);
uint32_t res_rd = IORD(WEB_FPU_0_BASE, 0x000b);

printf("val1_rd = %f -> ", *(float*)&val1_rd);
print_float32(val1_rd);
printf("\nval2_rd = %f -> ", *(float*)&val2_rd);
print_float32(val2_rd);
printf("\nres_rd = %f -> ", *(float*)&res_rd);
print_float32(res_rd);
printf("\n");

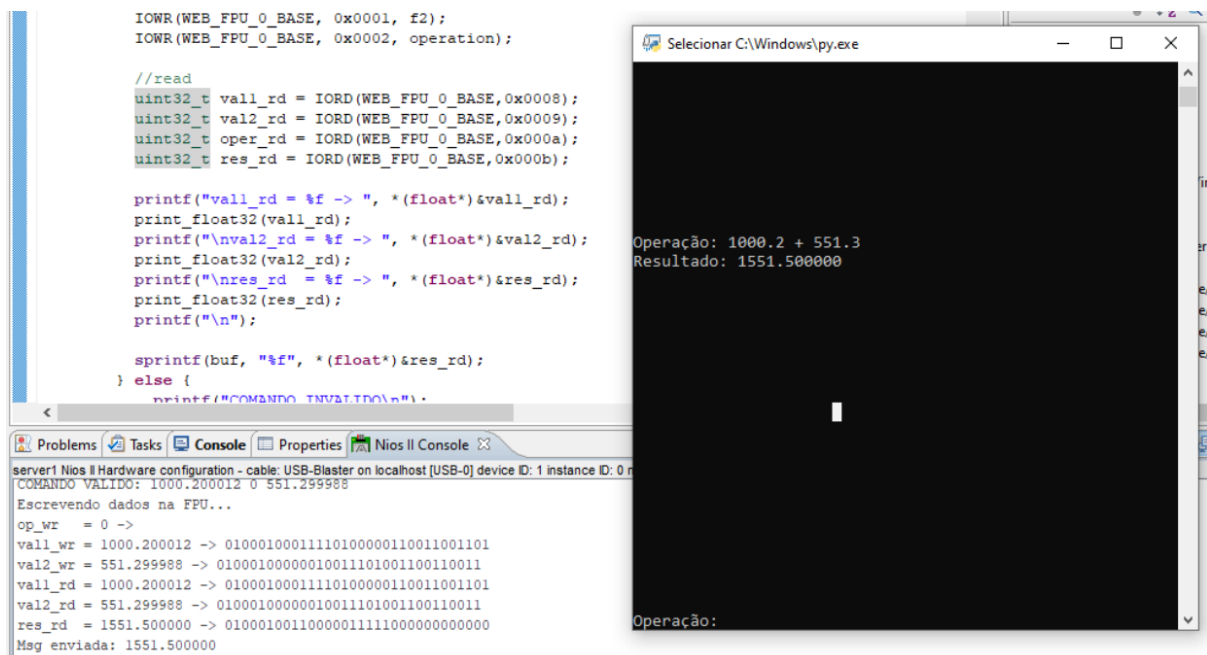
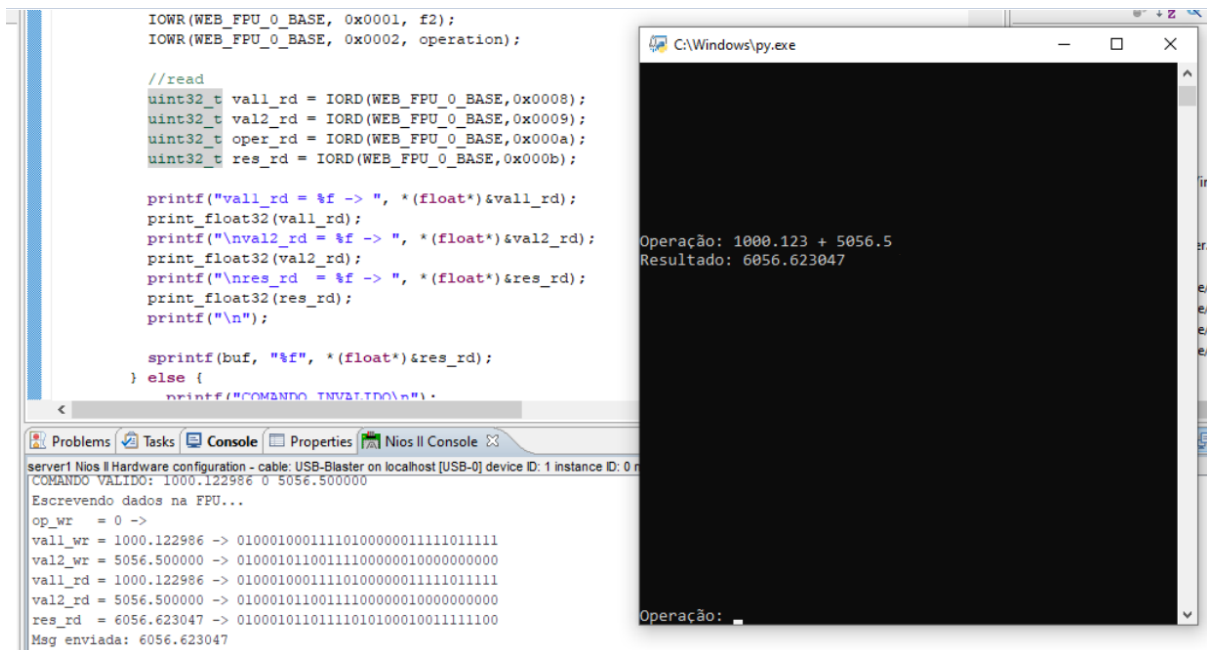
sprintf(buf, "%f", *(float*)&res_rd);
} else {
    printf("COMANDO INVÁLIDO\n");
}

server1 Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0
COMANDO VALIDO: 1.200000 0 2.500000
Escrevendo dados na FPU...
op_wr = 0 ->
val1_wr = 1.200000 -> 00111111100110011001100110011010
val2_wr = 2.500000 -> 01000000001000000000000000000000
val1_rd = 1.200000 -> 00111111100110011001100110011010
val2_rd = 2.500000 -> 01000000001000000000000000000000
res_rd = 3.700000 -> 01000000011011001100110011001101
Msg enviada: 3.700000
```

```
C:\Windows\py.exe

Operação: 1.2 + 2.5
Resultado: 3.700000

Operação: _
```



```
IOWR(WEB_FPU_0_BASE, 0x0001, f2);
IOWR(WEB_FPU_0_BASE, 0x0002, operation);

//read
uint32_t val1_rd = IORD(WEB_FPU_0_BASE, 0x0008);
uint32_t val2_rd = IORD(WEB_FPU_0_BASE, 0x0009);
uint32_t oper_rd = IORD(WEB_FPU_0_BASE, 0x000a);
uint32_t res_rd = IORD(WEB_FPU_0_BASE, 0x000b);

printf("val1_rd = %f -> ", *(float*)&val1_rd);
print_float32(val1_rd);
printf("\nval2_rd = %f -> ", *(float*)&val2_rd);
print_float32(val2_rd);
printf("\nres_rd = %f -> ", *(float*)&res_rd);
print_float32(res_rd);
printf("\n");

sprintf(buf, "%f", *(float*)&res_rd);
} else {
    printf("COMANDO INVALIDO\n");
}

server1 Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0
COMANDO VALIDO: 5.000000 0 -1.000000
Escrevendo dados na FPU...
op_wr = 0 ->
val1_wr = 5.000000 -> 01000000101000000000000000000000
val2_wr = -1.000000 -> 10111111100000000000000000000000
val1_rd = 5.000000 -> 01000000101000000000000000000000
val2_rd = -1.000000 -> 10111111100000000000000000000000
res_rd = 4.000000 -> 01000000100000000000000000000000
Msg enviada: 4.000000
```

C:\Windows\py.exe

Operação: 5 + -1
Resultado: 4.000000

Operação:

2 - subtração:

```
IOWR(WEB_FPU_0_BASE, 0x0001, f2);
IOWR(WEB_FPU_0_BASE, 0x0002, operation);

//read
uint32_t val1_rd = IORD(WEB_FPU_0_BASE, 0x0008);
uint32_t val2_rd = IORD(WEB_FPU_0_BASE, 0x0009);
uint32_t oper_rd = IORD(WEB_FPU_0_BASE, 0x000a);
uint32_t res_rd = IORD(WEB_FPU_0_BASE, 0x000b);

printf("val1_rd = %f -> ", *(float*)&val1_rd);
print_float32(val1_rd);
printf("\nval2_rd = %f -> ", *(float*)&val2_rd);
print_float32(val2_rd);
printf("\nres_rd = %f -> ", *(float*)&res_rd);
print_float32(res_rd);
printf("\n");

sprintf(buf, "%f", *(float*)&res_rd);
} else {
    printf("COMANDO INVALIDO\n");
}

server1 Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0
COMANDO VALIDO: 5.000000 1 8.000000
Escrevendo dados na FPU...
op_wr = 1 ->
val1_wr = 5.000000 -> 01000000101000000000000000000000
val2_wr = 8.000000 -> 01000000100000000000000000000000
val1_rd = 5.000000 -> 01000000101000000000000000000000
val2_rd = 8.000000 -> 01000000100000000000000000000000
res_rd = -3.000000 -> 11000000010000000000000000000000
Msg enviada: -3.000000
```

C:\Windows\py.exe

Operação: 5 - 8
Resultado: -3.000000

Operação:

```
IOWR(WEB_FPU_0_BASE, 0x0001, f2);
IOWR(WEB_FPU_0_BASE, 0x0002, operation);

//read
uint32_t val1_rd = IORD(WEB_FPU_0_BASE, 0x0008);
uint32_t val2_rd = IORD(WEB_FPU_0_BASE, 0x0009);
uint32_t oper_rd = IORD(WEB_FPU_0_BASE, 0x000a);
uint32_t res_rd = IORD(WEB_FPU_0_BASE, 0x000b);

printf("val1_rd = %f -> ", *(float*)&val1_rd);
print_float32(val1_rd);
printf("val2_rd = %f -> ", *(float*)&val2_rd);
print_float32(val2_rd);
printf("nres_rd = %f -> ", *(float*)&res_rd);
print_float32(res_rd);
printf("\n");

sprintf(buf, "%f", *(float*)&res_rd);
} else {
    printf("COMANDO INVALIDO\n");
}

server1 Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0
COMANDO VALIDO: 8.000000 1 5.000000
Escrevendo dados na FPU...
op_wr = 1 ->
val1_wr = 8.000000 -> 01000001000000000000000000000000
val2_wr = 5.000000 -> 01000000101000000000000000000000
val1_rd = 8.000000 -> 01000001000000000000000000000000
val2_rd = 5.000000 -> 01000000101000000000000000000000
res_rd = 3.000000 -> 01000000010000000000000000000000
Msg enviada: 3.000000
```

```
C:\Windows\py.exe

Operação: 8 - 5
Resultado: 3.000000 0 .3 .5

Operação:
```

```
IOWR(WEB_FPU_0_BASE, 0x0001, f2);
IOWR(WEB_FPU_0_BASE, 0x0002, operation);

//read
uint32_t val1_rd = IORD(WEB_FPU_0_BASE, 0x0008);
uint32_t val2_rd = IORD(WEB_FPU_0_BASE, 0x0009);
uint32_t oper_rd = IORD(WEB_FPU_0_BASE, 0x000a);
uint32_t res_rd = IORD(WEB_FPU_0_BASE, 0x000b);

printf("val1_rd = %f -> ", *(float*)&val1_rd);
print_float32(val1_rd);
printf("val2_rd = %f -> ", *(float*)&val2_rd);
print_float32(val2_rd);
printf("nres_rd = %f -> ", *(float*)&res_rd);
print_float32(res_rd);
printf("\n");

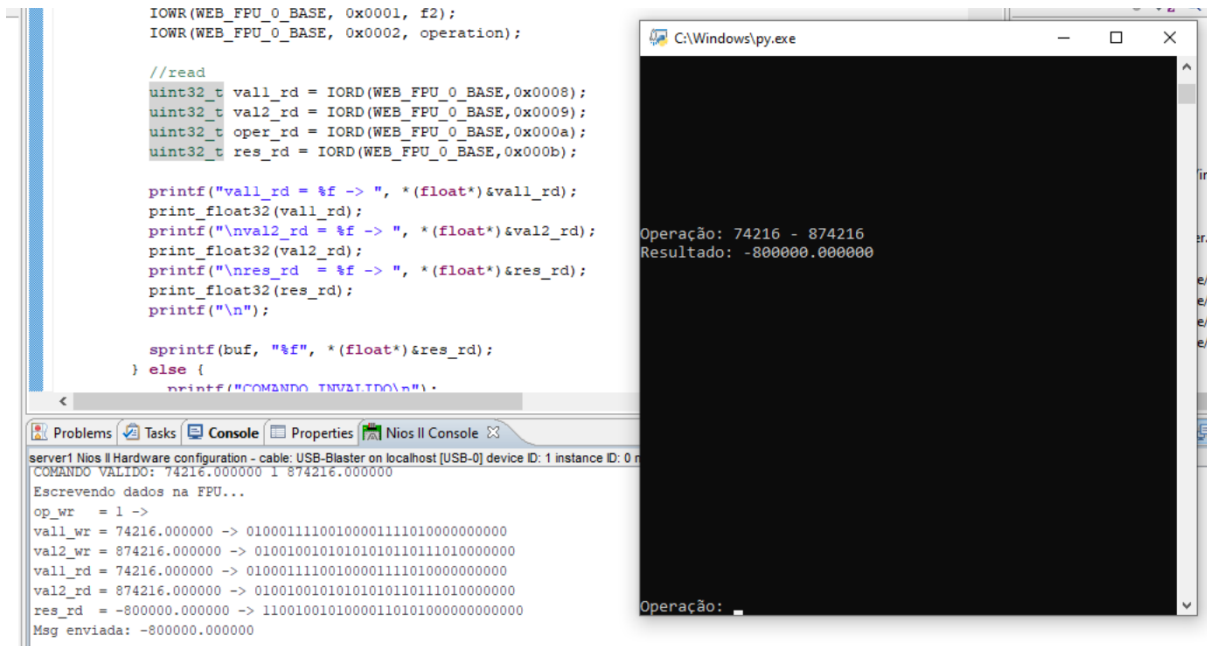
sprintf(buf, "%f", *(float*)&res_rd);
} else {
    printf("COMANDO INVALIDO\n");
}

server1 Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0
COMANDO VALIDO: 8.000000 1 5.000000
Escrevendo dados na FPU...
op_wr = 1 ->
val1_wr = 8.000000 -> 01000001000000000000000000000000
val2_wr = 5.000000 -> 01000000101000000000000000000000
val1_rd = 8.000000 -> 01000001000000000000000000000000
val2_rd = 5.000000 -> 01000000101000000000000000000000
res_rd = 3.000000 -> 01000000010000000000000000000000
Msg enviada: 3.000000
```

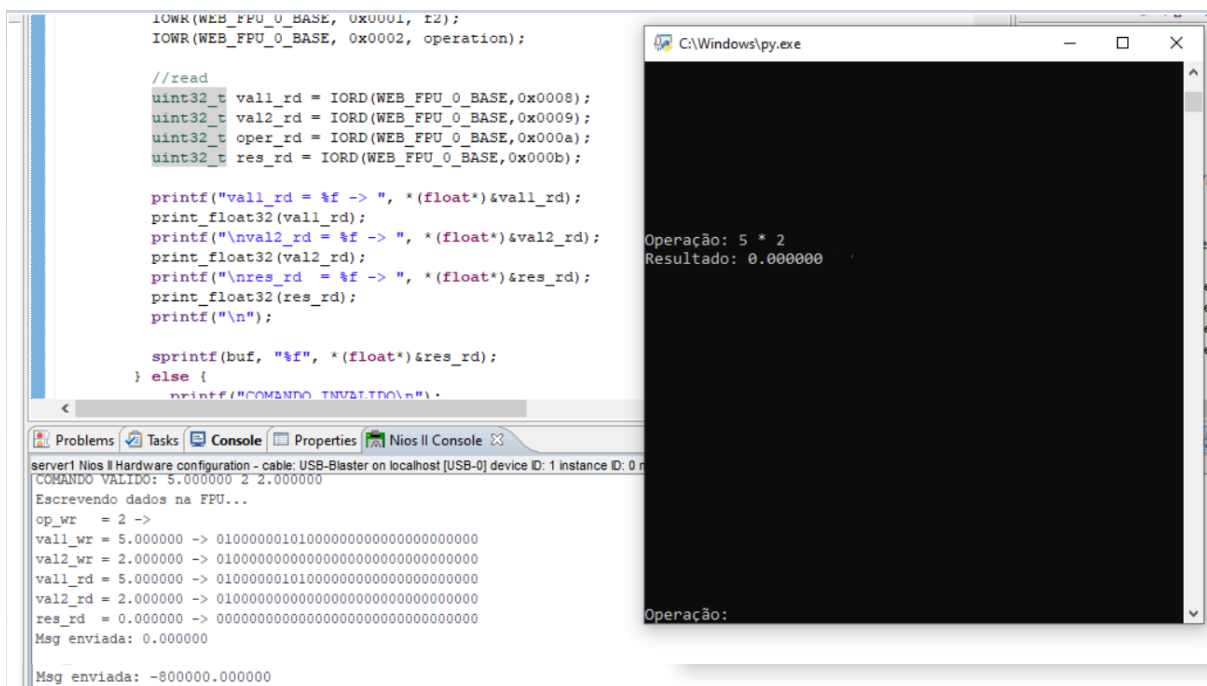
```
C:\Windows\py.exe

Operação: 8 - 5
Resultado: 3.000000

Operação:
```



3 - multiplicação:




```
IOWR(WEB_FPU_0_BASE, 0x0001, r2);
IOWR(WEB_FPU_0_BASE, 0x0002, operation);

//read
uint32_t val1_rd = IORD(WEB_FPU_0_BASE, 0x0008);
uint32_t val2_rd = IORD(WEB_FPU_0_BASE, 0x0009);
uint32_t oper_rd = IORD(WEB_FPU_0_BASE, 0x000a);
uint32_t res_rd = IORD(WEB_FPU_0_BASE, 0x000b);

printf("val1_rd = %f -> ", *(float*)&val1_rd);
print_float32(val1_rd);
printf("\nval2_rd = %f -> ", *(float*)&val2_rd);
print_float32(val2_rd);
printf("\nres_rd = %f -> ", *(float*)&res_rd);
print_float32(res_rd);
printf("\n");

sprintf(buf, "%f", *(float*)&res_rd);
} else {
    printf("COMANDO INVALIDO\n");
}
```

server1 Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0

COMANDO VALIDO: 1000.500000 2 1.500000

Escrevendo dados na FPU...

op_wr = 2 ->

val1_wr = 1000.500000 -> 0100010001111010001000000000000000

val2_wr = 1.500000 -> 0011111111000000000000000000000000

val1_rd = 1000.500000 -> 0100010001111010001000000000000000

val2_rd = 1.500000 -> 0011111111000000000000000000000000

res_rd = 0.000000 -> 0000000000000000000000000000000000

Msg enviada: 0.000000

C:\Windows\py.exe

Operação: 1000.5 * 1.5

Resultado: 0.000000

Operação:

4 - divisão:

```
IOWR(WEB_FPU_0_BASE, 0x0001, r2);
IOWR(WEB_FPU_0_BASE, 0x0002, operation);

//read
uint32_t val1_rd = IORD(WEB_FPU_0_BASE, 0x0008);
uint32_t val2_rd = IORD(WEB_FPU_0_BASE, 0x0009);
uint32_t oper_rd = IORD(WEB_FPU_0_BASE, 0x000a);
uint32_t res_rd = IORD(WEB_FPU_0_BASE, 0x000b);

printf("val1_rd = %f -> ", *(float*)&val1_rd);
print_float32(val1_rd);
printf("\nval2_rd = %f -> ", *(float*)&val2_rd);
print_float32(val2_rd);
printf("\nres_rd = %f -> ", *(float*)&res_rd);
print_float32(res_rd);
printf("\n");

sprintf(buf, "%f", *(float*)&res_rd);
} else {
    printf("COMANDO INVALIDO\n");
}
```

server1 Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0

COMANDO VALIDO: 5.000000 3 2.000000

Escrevendo dados na FPU...

op_wr = 3 ->

val1_wr = 5.000000 -> 0100000010100000000000000000000000

val2_wr = 2.000000 -> 0100000000000000000000000000000000

val1_rd = 5.000000 -> 0100000010100000000000000000000000

val2_rd = 2.000000 -> 0100000000000000000000000000000000

res_rd = 0.500000 -> 0011111110000000000000000000000000

Msg enviada: 0.500000

C:\Windows\py.exe

Operação: 5 / 2

Resultado: 0.500000

Operação:

