Assignment 2 "Adding an ALU and a Stack "

Group - SS05

The CPU is based on MIPS architecture.

- **Memory:** 2048 locations of 1 bytes each (mem[2048]) = 2KB of memory

Memory	Memory Address	Address in program	Locations (bytes)
Data memory	0x0481 to 0x07FF	mem[1053] to mem[2047]	896
Stack	0x0400 to 0x0480	mem[1024] to mem[1152]	128
Instruction memory	0x0200 to 0x03FF	mem[512] to mem[1023]	512
OS memory	0x0000 to 0x01FF	mem[0] to mem[511]	512

- **Instruction size:** 4 bytes = 32 bits

Opcode (8 bits)	Operand 1 (8 bits)	Operand 2 (8 bits)	Operand 3 (8 bits)

- **Opcodes:**

Register	Opcode	Instruction	Opcode
r0	0x00	lw	0x00
r1	0x01	sw	0x01
r2	0x02	add	0x02
r3	0x03	sub	0x03
r4	0x04	mul	0x04
r5	0x05	div	0x05
r6	0x06	mod	0x06
r7	0x07	push	0x07
r8	0x08	pop	0x08

- **Status register:** 8bits

-	-	OF	SF	ZF	AC	PF	CF
(Bit 7)	(Bit 6)	(Bit 5)	(Bit 4)	(Bit 3)	(Bit 2)	(Bit 1)	(Bit 0)

Instructions representation:

1. **add r0,r1,r2**

   Big Endian machine

add (0x02)	r0 (0x00)	r1 (0x01)	r2 (0x02)

   Little Endian machine

r2 (0x02)	r1 (0x01)	r0 (0x00)	add (0x02)

2. **sub r0,r1,r2**

   Big Endian machine

sub (0x03)	r0 (0x00)	r1 (0x01)	r2 (0x02)

   Little Endian machine

r2 (0x02)	r1 (0x01)	r0 (0x00)	sub (0x03)

3. **mul r0,r1,r2**

Big Endian machine

mul (0x04)	r0 (0x00)	r1 (0x01)	r2 (0x02)

Little Endian machine

r2 (0x02)	r1 (0x01)	r0 (0x00)	mul (0x04)

4. **div r0,r1,r2**

Big Endian machine

div (0x05)	r0 (0x00)	r1 (0x01)	r2 (0x02)

Little Endian machine

r2 (0x02)	r1 (0x01)	r0 (0x00)	div (0x05)

5. **mod r0,r1,r2**

Big Endian machine

mod (0x06)	r0 (0x00)	r1 (0x01)	r2 (0x02)

Little Endian machine

r2 (0x02)	r1 (0x01)	r0 (0x00)	mod (0x06)

6. **push r0,0,0**

Big Endian machine

push (0x07)	r0 (0x00)	0 (0x00)	0 (0x00)

Little Endian machine

0 (0x00)	0 (0x00)	r0 (0x00)	push (0x07)

7. **pop r3,0,0**

Big Endian machine

pop (0x08)	r3 (0x03)	0 (0x00)	0 (0x00)

Little Endian machine

0 (0x00)	0 (0x00)	r3 (0x03)	pop (0x08)

- **CPU State initially**

```
Machine selection 1.Little Endian 2. Big Endian : 1

************** Cpu state initially **************

Content of instruction Memory:

Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value
200     ff    201    ff    202    ff    203    ff    204    ff    205    ff
206     ff    207    ff    208    ff    209    ff    20a    ff    20b    ff
20c     ff    20d    ff    20e    ff    20f    ff    210    ff    211    ff
212     ff    213    ff    214    ff    215    ff    216    ff    217    ff
218     ff    219    ff    21a    ff    21b    ff    21c    ff    21d    ff
21e     ff    21f    ff

Content of Stack Memory:

Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value
400     ff    401    ff    402    ff    403    ff    404    ff    405    ff
406     ff    407    ff    408    ff    409    ff    40A    ff    40B    ff
40C     ff    40D    ff    40E    ff    40F    ff    410    ff    411    ff
412     ff    413    ff    414    ff    415    ff    416    ff    417    ff
418     ff    419    ff    41A    ff    41B    ff    41C    ff    41D    ff
41E     ff    41F    ff

Content of Data Memory:

Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value
481     ff    482    ff    483    ff    484    ff    485    ff    486    ff
487     ff    488    ff    489    ff    48A    ff    48B    ff    48C    ff
48D     ff    48E    ff    48F    ff    490    ff    491    ff    492    ff
493     ff    494    ff    495    ff    496    ff    497    ff    498    ff
499     ff    49A    ff    49B    ff    49C    ff    49D    ff    49E    ff
49F     ff    4A0    ff

Content of General Purpose Register:

R0 = 00000000 R1 = 00000000  R2 = 00000000  R3 = 00000000  R4 = 00000000  R5 = 00000000  R6 = 00000000  R7 = 00000000
R8 = 00000000 R9 = 00000000 R10 = 00000000 R11 = 00000000 R12 = 00000000 R13 = 00000000 R14 = 00000000 R15 = 00000000

Contents of Status register:    0 0 0 0 0 0 0 0

Content of Program Counter (PC): 00000000

Content of Stack Pointer (SP):  00000400

Content of Base Pointer (BP):   00000400
```

- **Storing predefined values in registers to demonstrate assignment no. 2**

  R1 = 0x00000FFF R2 = 0x00000002

```
Content of General Purpose Register:

R0 = 00000000 R1 = 00000fff  R2 = 00000002  R3 = 00000000
R8 = 00000000 R9 = 00000000 R10 = 00000000 R11 = 00000000
```

- **CPU State after loading instructions in memory (Little Endian Machine)**

```
Machine selection 1.Little Endian 2. Big Endian : 1

************** Cpu state after loading instructions into memory **************

Content of instruction Memory:

Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value
200     02    201    01    202    00    203    02    204    02    205    01
206     00    207    03    208    02    209    01    20a    00    20b    04
20c     02    20d    01    20e    00    20f    05    210    02    211    01
212     00    213    06    214    00    215    00    216    00    217    07
218     00    219    00    21a    03    21b    08    21c    ff    21d    ff
21e     ff    21f    ff
```

CPU State after loading instructions in memory (Big Endian Machine)

```
Machine selection 1.Little Endian 2. Big Endian : 2

************** Cpu state after loading instructions into memory **************

Content of instruction Memory:

Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value
200     02    201    00    202    01    203    02    204    03    205    00
206     01    207    02    208    04    209    00    20a    01    20b    02
20c     05    20d    00    20e    01    20f    02    210    06    211    00
212     01    213    02    214    07    215    00    216    00    217    00
218     08    219    01    21a    00    21b    00    21c    ff    21d    ff
21e     ff    21f    ff
```

- **After add r0,r1,r2**

```
Content of General Purpose Register:

R0 = 00001001 R1 = 00000fff  R2 = 00000002  R3 = 00000000
R8 = 00000000 R9 = 00000000 R10 = 00000000 R11 = 00000000

Contents of Status register:    0 0 0 0 0 0 1 0

Content of Program Counter (PC): 00000204

Content of Stack Pointer (SP):  00000400

Content of Base Pointer (BP):   00000400
```

- **After sub r0,r1,r2**

```
Content of General Purpose Register:

R0 = 00000ffd R1 = 00000fff  R2 = 00000002  R3 = 00000000
R8 = 00000000 R9 = 00000000 R10 = 00000000 R11 = 00000000

Contents of Status register:    0 0 0 0 0 0 1 0

Content of Program Counter (PC): 00000208

Content of Stack Pointer (SP):  00000400

Content of Base Pointer (BP):   00000400
```

- **After mul r0,r1,r2**

```
Content of General Purpose Register:

R0 = 00001ffe R1 = 00000fff  R2 = 00000002  R3 = 00000000
R8 = 00000000 R9 = 00000000 R10 = 00000000 R11 = 00000000

Contents of Status register:    0 0 0 0 0 0 1 0

Content of Program Counter (PC): 0000020c

Content of Stack Pointer (SP):  00000400

Content of Base Pointer (BP):   00000400
```

- **After div r0,r1,r2**

```
Content of General Purpose Register:

R0 = 000007ff R1 = 00000fff  R2 = 00000002  R3 = 00000000
R8 = 00000000 R9 = 00000000 R10 = 00000000 R11 = 00000000

Contents of Status register:    0 0 0 0 0 0 0 0

Content of Program Counter (PC): 00000210

Content of Stack Pointer (SP):  00000400

Content of Base Pointer (BP):   00000400
```

- **After mod r0,r1,r2**

```
Content of General Purpose Register:

R0 = 00000001 R1 = 00000fff  R2 = 00000002  R3 = 00000000
R8 = 00000000 R9 = 00000000 R10 = 00000000 R11 = 00000000

Contents of Status register:    0 0 0 0 0 0 1 0

Content of Program Counter (PC): 00000214

Content of Stack Pointer (SP):  00000400

Content of Base Pointer (BP):   00000400
```

- **After push r0,0,0**

```
Content of Stack Memory:

Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value
400     01    401     00    402     00    403     00    404     ff    405     ff
406     ff    407     ff    408     ff    409     ff    40A     ff    40B     ff
40C     ff    40D     ff    40E     ff    40F     ff    410     ff    411     ff
412     ff    413     ff    414     ff    415     ff    416     ff    417     ff
418     ff    419     ff    41A     ff    41B     ff    41C     ff    41D     ff
41E     ff    41F     ff

Content of Data Memory:

Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value
481     ff    482     ff    483     ff    484     ff    485     ff    486     ff
487     ff    488     ff    489     ff    48A     ff    48B     ff    48C     ff
48D     ff    48E     ff    48F     ff    490     ff    491     ff    492     ff
493     ff    494     ff    495     ff    496     ff    497     ff    498     ff
499     ff    49A     ff    49B     ff    49C     ff    49D     ff    49E     ff
49F     ff    4A0     ff

Content of General Purpose Register:

R0 = 00000001 R1 = 00000fff  R2 = 00000002  R3 = 00000000  R4 = 00000000  R5 = 00000000
R8 = 00000000 R9 = 00000000 R10 = 00000000 R11 = 00000000 R12 = 00000000 R13 = 00000000

Contents of Status register:    0 0 0 0 0 0 1 0

Content of Program Counter (PC): 00000218

Content of Stack Pointer (SP):  00000404

Content of Base Pointer (BP):   00000400
```

- **After pop r3,0,0**

```
Content of Stack Memory:

Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value
400    ff    401    ff    402    ff    403    ff    404    ff    405    ff
406    ff    407    ff    408    ff    409    ff    40A    ff    40B    ff
40C    ff    40D    ff    40E    ff    40F    ff    410    ff    411    ff
412    ff    413    ff    414    ff    415    ff    416    ff    417    ff
418    ff    419    ff    41A    ff    41B    ff    41C    ff    41D    ff
41E    ff    41F    ff

Content of Data Memory:

Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value  Addr  Value
481    ff    482    ff    483    ff    484    ff    485    ff    486    ff
487    ff    488    ff    489    ff    48A    ff    48B    ff    48C    ff
48D    ff    48E    ff    48F    ff    490    ff    491    ff    492    ff
493    ff    494    ff    495    ff    496    ff    497    ff    498    ff
499    ff    49A    ff    49B    ff    49C    ff    49D    ff    49E    ff
49F    ff    4A0    ff

Content of General Purpose Register:

R0 = 00000001 R1 = 00000fff  R2 = 00000002  R3 = 00000001  R4 = 00000000  R5 = 00000000
R8 = 00000000 R9 = 00000000 R10 = 00000000 R11 = 00000000 R12 = 00000000 R13 = 00000000

Contents of Status register:    0 0 0 0 0 0 1 0

Content of Program Counter (PC): 0000021c

Content of Stack Pointer (SP):  00000400

Content of Base Pointer (BP):   00000400
```

- **For demonstration of setting overflow flag we set R1=0x0FFFFFFF and R2=0x00000020 and executed instruction mul r0,r1,r2 Following is the output. (Status register's description is given above)**

```
Content of General Purpose Register:

R0 = fffffffe0 R1 = 0fffffff  R2 = 00000020  R3 = 00000000
R8 = 00000000 R9 = 00000000 R10 = 00000000 R11 = 00000000

Contents of Status register:    0 0 1 0 0 0 1 1

Content of Program Counter (PC): 0000020f

Content of Stack Pointer (SP):  00000400

Content of Base Pointer (BP):   00000400
```