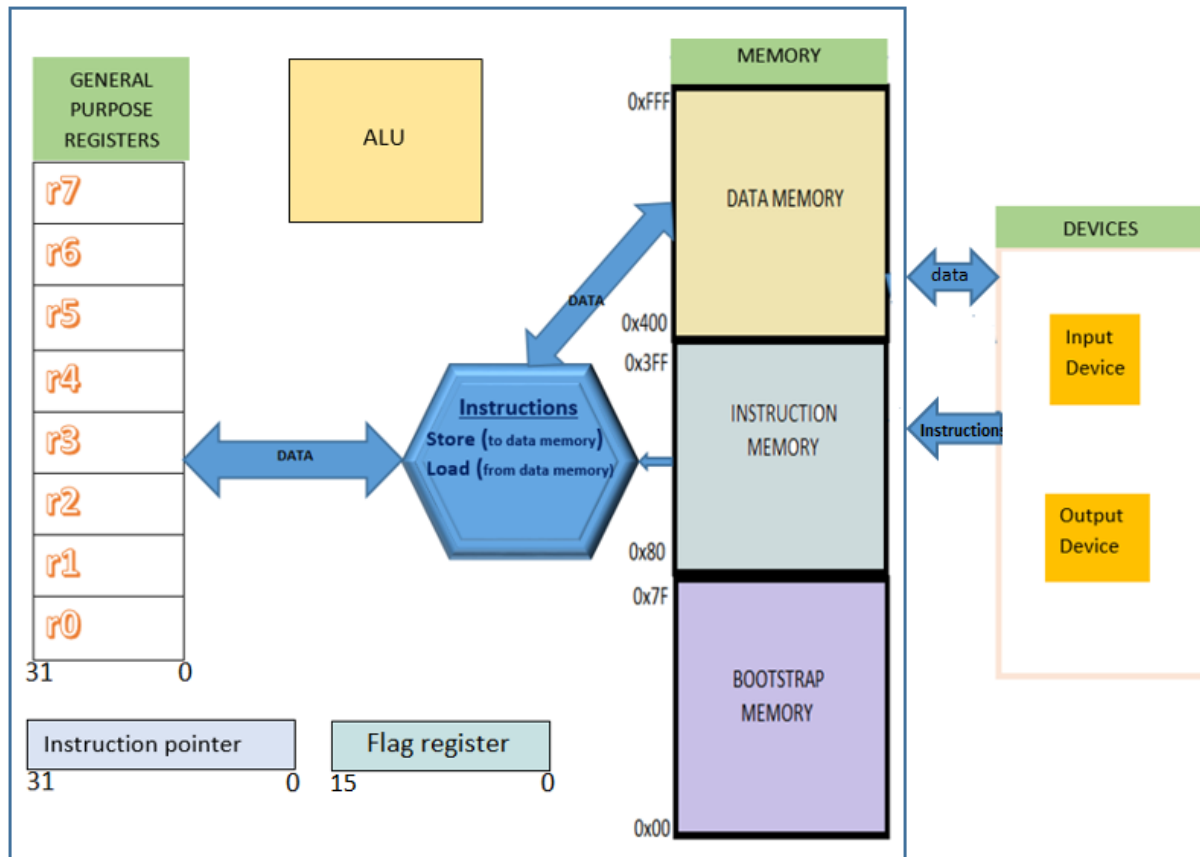


CPU ARCHITECTURE :



Specifications

Main memory:

1. *Bootstrap memory* : 0-127 : 128 bytes
2. *Instruction memory* : 128 - 1023 : 896 bytes
3. *Data memory* : 1024 - 4095 : 3.072 KB

Description: Memory is byte addressable (data type-uint8_t), and word size is 32 bits/4 bytes.

Registers:

General purpose registers: r0-r7 : all 4 word i.e. uint32_t

Number of registers : 8 registers

Special Purpose Registers:

Flag_register : 16 bit register that updates in case of Memory overflow with a value to set the 12 bit.

Instruction Pointer: 32 bit register which holds the address of the next instruction to be executed. It is incremented by 4 bytes after every instruction and thus points to the next instruction.(Since instruction is 32 bits wide)

Arithmetic Logical Unit:

Performs logical operation on the data and provides it to the registers(To be implemented in future)

Operations

1. Load Operation

Instruction: lw register, (offset)memory address

Method Implemented: int loadreg(char* reg1, uint32_t* reg_num, int memory_addr);

- Method Name: loadreg
- Return type: int
- Register name: reg1 (r0-r7)
- Pointer to the Register: reg_num (pointer to reg1)
- Memory Address: memory_addr

Method Description: After taking input from user, 32-bit register is loaded with values that are present in 4 consecutive 8 bit/byte addressable memory locations. The final memory location is the result of addition of offset address as well as memory address.

2. Store Operation

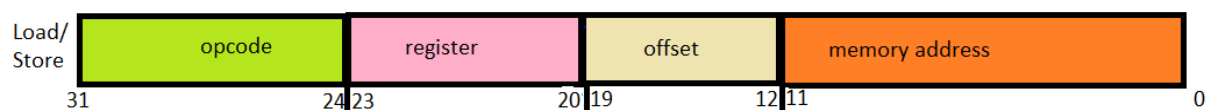
Instruction: sw register,(offset)memory address

Method Implemented: int storereg(char*reg1,uint32_t* reg_num, int memory_addr);

- Method Name: storereg
- Return type: int
- Register name: reg1 (r0-r7)
- Pointer to the Register: reg_num (pointer to reg1)
- Memory Address: memory_addr

Method Description: After taking input from user, values are loaded from the register into 4 consecutive 8 bit/byte addressable memory locations. The final memory location is the result of addition of offset address as well as memory address.

Instruction format



Instruction size : 4 bytes (opcode+register+offset+memory address)

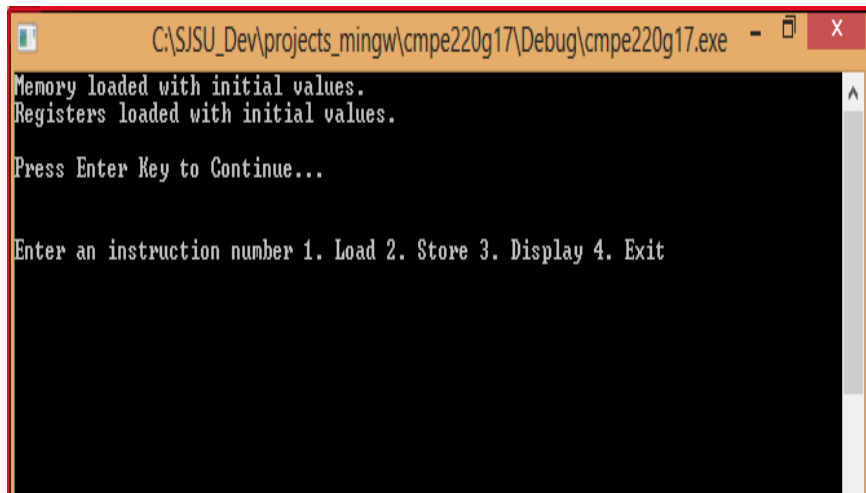
The CPU implements a 32-bit instruction set architecture with Little-Endian format. Each memory related instruction (Load/Store) is designed as below with the following fields:

- **Opcode:** 8 bits (0x00→Load, 0x01→Store) This field specifies the type of the operation being performed in the instruction. The MSB byte of the instruction represent the opcode.
- **Register:** 4 bits (0x00-0x07 → r0-r7) This field represents the number corresponding to the actual register which is being used in the Load/Store operation. This field occupies the MSB nibble next to the opcode.

- **Offset:** 8 bits (0x00-0xff) This field specifies the offset amount of the memory address to be accessed. This takes the byte next to the register field in the instruction.
- **Memory:** 12 bits (0x00-0xfff) This field represents the address of memory from/to which a memory access should be performed. The width of this LSB field is chosen such that the instruction allows to access the memory available (i.e 4096 unique memory locations).

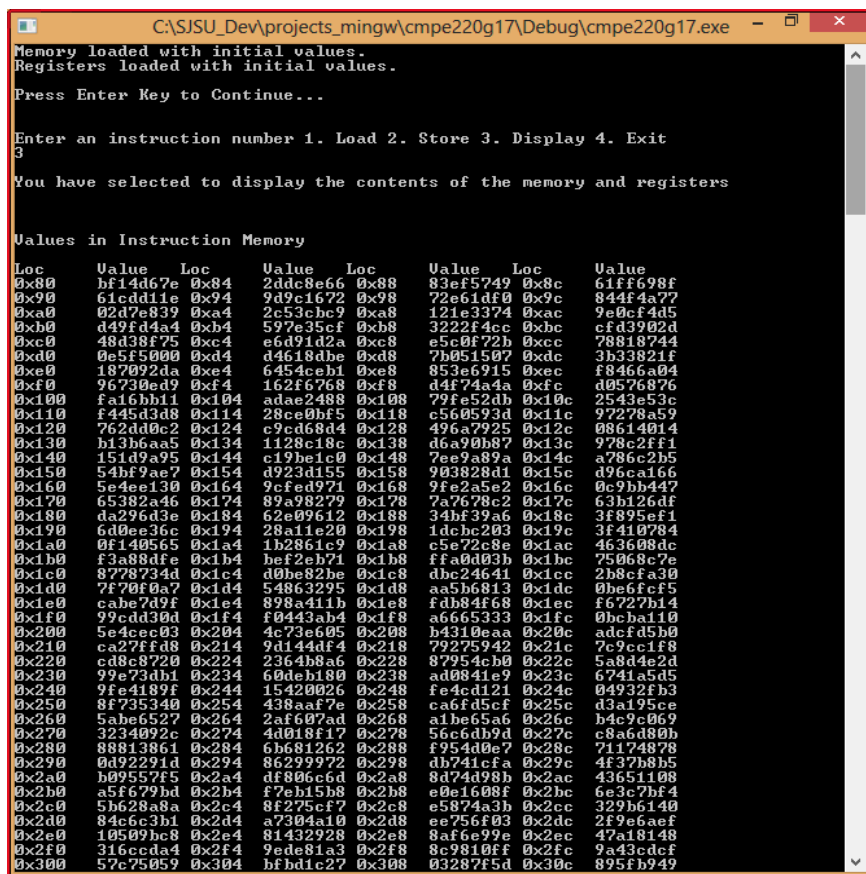
Screen Shots

1) Initial Processor Start :



```
C:\JSU_Dev\projects_mingw\cmpe220g17\Debug\cmpe220g17.exe
Memory loaded with initial values.
Registers loaded with initial values.
Press Enter Key to Continue...
Enter an instruction number 1. Load 2. Store 3. Display 4. Exit
```

2.1) Display of Memory and Registers:



```
C:\JSU_Dev\projects_mingw\cmpe220g17\Debug\cmpe220g17.exe
Memory loaded with initial values.
Registers loaded with initial values.
Press Enter Key to Continue...
Enter an instruction number 1. Load 2. Store 3. Display 4. Exit
3
You have selected to display the contents of the memory and registers

Values in Instruction Memory
Loc      Value      Loc      Value      Loc      Value      Loc      Value
0x80     bf14d67e    0x84     2ddc8e66    0x88     83ef5749    0x8c     61ff698f
0x90     61cdd11e    0x94     9d9c1672    0x98     72e61df0    0x9c     844f4a77
0xa0     02d7e839    0xa4     2c53cbc9    0xa8     121e3374    0xac     9e0cf4d5
0xb0     d49d4a4a    0xb4     597e35cf    0xb8     3222f4cc    0xbc     cfd3902d
0xc0     48d38f75    0xc4     e6d91d2a    0xc8     e5c0f72b    0xcc     78818744
0xd0     0e5f5000    0xd4     d4618dbe    0xd8     7b051507    0xdc     3b33821f
0xe0     187092da    0xe4     6454ceb1    0xe8     853e6915    0xec     f8466a04
0xf0     96730ed9    0xf4     162f6768    0xf8     d4f74a4a    0xfc     d0576876
0x100    fa16bb11    0x104    adae2488    0x108    79fe52db    0x10c    2543e53c
0x110    f445d3d8    0x114    28ce0bf5    0x118    c560953d    0x11c    97278a59
0x120    762ad0e2    0x124    c9cd68d4    0x128    496a7925    0x12c    08614014
0x130    b13b6aa5    0x134    1128e18c    0x138    d6a90b87    0x13c    978c2ff1
0x140    151d9a95    0x144    c19be1c0    0x148    7ee9a89a    0x14c    a786c2b5
0x150    54bf9ae7    0x154    d923d155    0x158    903828d1    0x15c    d96ca166
0x160    5e4ee130    0x164    9cfd9771    0x168    9fe2a5e2    0x16c    0c9bb447
0x170    65382a46    0x174    89a98279    0x178    7a7678c2    0x17c    63b126df
0x180    da296d3e    0x184    62e09612    0x188    34bf39a6    0x18c    3f895ef1
0x190    6d0ee36c    0x194    28a11e20    0x198    1dc8c203    0x19c    3f410784
0x1a0    0f140565    0x1a4    1b2861c9    0x1a8    c5e72c8e    0x1ac    463608dc
0x1b0    f3a88dfe    0x1b4    hef2eb71    0x1b8    ffa0d03b    0x1bc    75068c7e
0x1c0    8778734d    0x1c4    d0be82be    0x1c8    d0c24641    0x1cc    2b8cfa30
0x1d0    7f70f0a7    0x1d4    54863295    0x1d8    aa5b6813    0x1dc    0be6fcf5
0x1e0    cabef7d9f    0x1e4    898a411b    0x1e8    fdb84f68    0x1ec    f6727b14
0x1f0    99cdd3bd    0x1f4    f0443ab4    0x1f8    a6665333    0x1fc    0bcbca110
0x200    5e4cec03    0x204    4c73e605    0x208    b4310eaa    0x20c    adcfd5b0
0x210    ca27ffd8    0x214    9d144df4    0x218    79275942    0x21c    7c9cc1f8
0x220    cd8c8720    0x224    2364b8a6    0x228    87954cb0    0x22c    5a8d4e2d
0x230    99e73db1    0x234    60deb180    0x238    ad0841e9    0x23c    6741a5d5
0x240    9fe4189f    0x244    15420026    0x248    fe4cd121    0x24c    04932fb3
0x250    8f735340    0x254    438aaf7e    0x258    ca6fd5cf    0x25c    d3a195ce
0x260    5ab65277    0x264    2af607ad    0x268    a1be65a6    0x26c    b4c9e069
0x270    2234092c    0x274    4d010f17    0x278    56c6db9d    0x27c    c8a6d80b
0x280    88813861    0x284    6b681262    0x288    f954d0e7    0x28c    71174878
0x290    0d92291d    0x294    86299972    0x298    db741cfa    0x29c    4f37b8b5
0x2a0    b0955575    0x2a4    df806c6d    0x2a8    8d74d98b    0x2ac    43651108
0x2b0    a5f679bd    0x2b4    f7eb15b8    0x2b8    e0e1608f    0x2bc    6e3c7bf4
0x2c0    5b628a8a    0x2c4    8f275cf7    0x2c8    e5874a3b    0x2cc    329b6140
0x2d0    84c6c3b1    0x2d4    a7304a10    0x2d8    ee756f03    0x2dc    2f9e6aef
0x2e0    10509bc8    0x2e4    81432928    0x2e8    8af6e99e    0x2ec    47a18148
0x2f0    316ccda4    0x2f4    9ede81a3    0x2f8    8c9810ff    0x2fc    9a43cdcf
0x300    57c75059    0x304    bfbd1c27    0x308    03287f5d    0x30c    895fb949
```

2.2) Display Memory and Registers (Contd..)

```
C:\JSU_Development\projects_mingw\cmpe220g17\Debug\cmpe220g17.exe
0xd30 bcc9b100 0xd34 6ec0e848 0xd38 11a8fee0 0xd3c fc0e99e3
0xd40 b0fee8db 0xd44 5d763fd7 0xd48 a81b01be 0xd4c ab2bc3e2
0xd50 3db3aed8 0xd54 74022588 0xd58 695da880 0xd5c 3bf4f98e
0xd60 57157d8d 0xd64 f6a0e47f 0xd68 e7bb0ddc 0xd6c 8ec6232a
0xd70 2d920dce 0xd74 62cd0522 0xd78 f1c186c7 0xd7c c43f6c3d
0xd80 30d557b0 0xd84 7a475015 0xd88 9a3daf76 0xd8c 3e3a3b8a
0xd90 12c49489 0xd94 3f0bee3e 0xd98 313c5f5e 0xd9c 9e4d53b8
0xda0 c4a73ded 0xda4 f255c9c2 0xda8 490bb034 0xdac c46d532b
0xdb0 76ce0cb2 0xdb4 13a3c906 0xdb8 b237faec 0xdbc d1a0ae48
0xdc0 9af1f8ec 0xdc4 65b198ae 0xdc8 7d8cd7bd 0xdcc 2749b335
0xdd0 e0fc3cf0 0xdd4 e77d3ea0 0xdd8 fb18201a 0xddc 66860cf5
0xde0 3a1c5154 0xde4 db430500 0xde8 bd28eeba 0xdec 6fb5a3cf
0xdf0 d9bfeec0 0xdf4 c2817534 0xdf8 95499990 0xdfc 64719f46
0xe00 2d910dce 0xe04 5cf357e1 0xe08 9247bbd2 0xe0c a5986a21
0xe10 e87fe84e 0xe14 ac97b5eb 0xe18 06302df5 0xe1c d348ad64
0xe20 fab7f701 0xe24 f1a9d878 0xe28 835108ae 0xe2c d0dc5154
0xe30 95a4f3bd 0xe34 ada748ec 0xe38 d837ad42 0xe3c bcf37f77
0xe40 29f5d696 0xe44 a14dceee 0xe48 18ae9bf5 0xe4c f868a090
0xe50 6896d761 0xe54 ce997460 0xe58 92c5c52e 0xe5c 26595da6
0xe60 43b57134 0xe64 75c7826a 0xe68 d9c73c50 0xe6c 25249ffc
0xe70 eabe5c63 0xe74 1655826f 0xe78 bd415952 0xe7c 19648e19
0xe80 cf5e9184 0xe84 73ff3c15 0xe88 4f02b567 0xe8c 60f895c1
0xe90 a2824aeb 0xe94 8ec3ba40 0xe98 e0152805 0xe9c 9dfb5cf9
0xea0 5558fe2d 0xea4 a5de4416 0xea8 01c8cde1 0xeac b0cccc06
0xeb0 184b6820 0xeb4 3ccc63fc 0xeb8 83a8faf0 0xebc bb060d6e
0xec0 530e80d8 0xec4 13dae196 0xec8 77824b67 0xecc 1e888af4
0xed0 d5827fab 0xed4 2a56c6ca 0xed8 2f6297ba 0xedc 7a6eea9f
0xee0 70dedf2d 0xee4 c42c5cbd 0xee8 3a96f8a0 0xeec b11418b3
0xef0 608d5733 0xef4 604a2cd3 0xef8 6aabc70c 0xefc e3193bb5
0xf00 153be2d3 0xf04 c06dfdb2 0xf08 d16e9c35 0xf0c 7158be6a
0xf10 41d6b861 0xf14 e491db3f 0xf18 bfeb518e 0xf1c fcf048d7
0xf20 d5895373 0xf24 0ff30c9e 0xf28 c470ffcd 0xf2c 663dc342
0xf30 01c36add 0xf34 c0111c35 0xf38 b38afee7 0xf3c cfd5b582e
0xf40 3731f8b4 0xf44 baa8d1a8 0xf48 9c06e811 0xf4c 99a97162
0xf50 27be344e 0xf54 fcb436dd 0xf58 d0f096c0 0xf5c 64c3b5e2
0xf60 c34f929f 0xf64 c77394f9 0xf68 e09720a8 0xf6c 11850ef2
0xf70 3b2ee05d 0xf74 9e617360 0xf78 9d86e1c0 0xf7c c18ea51a
0xf80 012a00bb 0xf84 413b9cb8 0xf88 188a703c 0xf8c d6bae31c
0xf90 c67b34b1 0xf94 b00019e6 0xf98 a2b2a690 0xf9c f02671fe
0xfa0 7a4cf4d1 0xfa4 2dea320e 0xfa8 cd499e72 0xfac f12f3806
0xfb0 4f0cf9f3 0xfb4 39787196 0xfb8 68ddaf47 0xfbc f97161b7
0xfc0 b5683c29 0xfc4 95679e23 0xfc8 853b72f4 0xfcc 69cb55d8
0xfd0 5e4bf6ca 0xfd4 42b3c399 0xfd8 7670c23e 0xfdc e259bc6d
0xfe0 3ae4a16a 0xfe4 809a281e 0xfe8 cbc8b66a 0xfec 467881bb
0xff0 7b9ff5df 0xff4 d2985717 0xff8 54d1a86d 0xffc b5c5cc47

The register values are as follows:
r0:0x0 r1:0x0 r2:0x0 r3:0x0 r4:0x0 r5:0x0 r6:0x0 r7:0x0

The flag value is:0x0000

The instruction pointer value is:0x00000080

Enter an instruction number 1. Load 2. Store 3. Display 4. Exit
```

3) Load Instruction:

```
C:\JSU_Development\projects_mingw\cmpe220g17\Debug\cmpe220g17.exe
0xed0 d5827fab 0xed4 2a56c6ca 0xed8 2f6297ba 0xedc 7a6eea9f
0xee0 70dedf2d 0xee4 c42c5cbd 0xee8 3a96f8a0 0xeec b11418b3
0xef0 608d5733 0xef4 604a2cd3 0xef8 6aabc70c 0xefc e3193bb5
0xf00 153be2d3 0xf04 c06dfdb2 0xf08 d16e9c35 0xf0c 7158be6a
0xf10 41d6b861 0xf14 e491db3f 0xf18 bfeb518e 0xf1c fcf048d7
0xf20 d5895373 0xf24 0ff30c9e 0xf28 c470ffcd 0xf2c 663dc342
0xf30 01c36add 0xf34 c0111c35 0xf38 b38afee7 0xf3c cfd5b582e
0xf40 3731f8b4 0xf44 baa8d1a8 0xf48 9c06e811 0xf4c 99a97162
0xf50 27be344e 0xf54 fcb436dd 0xf58 d0f096c0 0xf5c 64c3b5e2
0xf60 c399993f 0xf64 c77394f9 0xf68 e09720a8 0xf6c 11850ef2
0xf70 3b2ee05d 0xf74 9e617360 0xf78 9d86e1c0 0xf7c c18ea51a
0xf80 012a00bb 0xf84 413b9cb8 0xf88 188a703c 0xf8c d6bae31c
0xf90 c67b34b1 0xf94 b00019e6 0xf98 a2b2a690 0xf9c f02671fe
0xfa0 7a4cf4d1 0xfa4 2dea320e 0xfa8 cd499e72 0xfac f12f3806
0xfb0 4f0cf9f3 0xfb4 39787196 0xfb8 68ddaf47 0xfbc f97161b7
0xfc0 b5683c29 0xfc4 95679e23 0xfc8 853b72f4 0xfcc 69cb55d8
0xfd0 5e4bf6ca 0xfd4 42b3c399 0xfd8 7670c23e 0xfdc e259bc6d
0xfe0 3ae4a16a 0xfe4 809a281e 0xfe8 cbc8b66a 0xfec 467881bb
0xff0 7b9ff5df 0xff4 d2985717 0xff8 54d1a86d 0xffc b5c5cc47

The register values are as follows:
r0:0x0 r1:0x0 r2:0x0 r3:0x0 r4:0x0 r5:0x0 r6:0x0 r7:0x0

The flag value is:0x0000

The instruction pointer value is:0x00000080

Enter an instruction number 1. Load 2. Store 3. Display 4. Exit
1

You have selected Load. Instruction should be of the form lw reg.<off>mem

Please find the acceptable range values:
Registers: r0-r7
Memory Location:1024-4095
lw r1,<3>2297
Instruction to be stored in instruction memory: 0x00103bb5
Load instruction is being performed. The corresponding opcode is 0x00
Register r1 changed from: 0x00000000 to: 0x216a04c

Enter an instruction number 1. Load 2. Store 3. Display 4. Exit
1

You have selected Load. Instruction should be of the form lw reg.<off>mem

Please find the acceptable range values:
Registers: r0-r7
Memory Location:1024-4095
lw r1,<4>2296
Instruction to be stored in instruction memory: 0x001048f8
Load instruction is being performed. The corresponding opcode is 0x00
Register r1 changed from: 0x216a04c to: 0xffefcdab

Enter an instruction number 1. Load 2. Store 3. Display 4. Exit
```

4) Store Instruction:

```
C:\SJSU_Dev\projects_mingw\cmpe220g17\Debug\cmpe220g17.exe

0xfa0 7a4cf4d1 0xfa4 2dea320e 0xfa8 cd499e72 0xfac f12f3806
0xfb0 4f0cf9f3 0xfb4 37787196 0xfb8 68ddafd7 0xfbc f97161b7
0xfc0 b5683c29 0xfc4 95679e23 0xfc8 853b72f4 0xfcc 69cb55d8
0xfd0 5e4bf6ca 0xfd4 42b3c399 0xfd8 7670c23e 0xfdc e259bc6d
0xfe0 3ae4a16a 0xfe4 809a281e 0xfe8 cbc8b66a 0xfec 467881bb
0xff0 7b9ff5df 0xff4 d2985717 0xff8 54d1a86d 0xffc b5c5cc47

The register values are as follows:
r0:0x0 r1:0x0 r2:0x0 r3:0x0 r4:0x0 r5:0x0 r6:0x0 r7:0x0
The flag value is:0x0000
The instruction pointer value is:0x00000080
Enter an instruction number 1. Load 2. Store 3. Display 4. Exit
1
You have selected Load. Instruction should be of the form lw reg,<off>mem
Please find the acceptable range values:
Registers: r0-r7
Memory Location:1024-4095
lw r1,<3>2997
Instruction to be stored in instruction memory: 0x00103bb5
Load instruction is being performed. The corresponding opcode is 0x00
Register r1 changed from: 0x00000000 to: 0x9216a04c
Enter an instruction number 1. Load 2. Store 3. Display 4. Exit
1
You have selected Load. Instruction should be of the form lw reg,<off>mem
Please find the acceptable range values:
Registers: r0-r7
Memory Location:1024-4095
lw r1,<4>2296
Instruction to be stored in instruction memory: 0x001048f8
Load instruction is being performed. The corresponding opcode is 0x00
Register r1 changed from: 0x9216a04c to: 0xffefcdab
Enter an instruction number 1. Load 2. Store 3. Display 4. Exit
2
You have selected Store. Instruction should be of the form sw reg,<off>mem
Please find the acceptable range values:
Registers: r0-r7
Memory Location:1024-4095
sw r1,<0>4092
Instruction to be stored in instruction memory: 0x01100ffc
Store instruction is being performed. The corresponding opcode is 0x01
Memory Address: 0xffc-0xfff changed to: 0xffefcdab
Enter an instruction number 1. Load 2. Store 3. Display 4. Exit
```

5) Instruction Pointer Increment:

```
C:\SJSU_Dev\projects_mingw\cmpe220g17\Debug\cmpe220g17.exe

Enter an instruction number 1. Load 2. Store 3. Display 4. Exit
1
You have selected Load. Instruction should be of the form lw reg,<off>mem
Please find the acceptable range values:
Registers: r0-r7
Memory Location:1024-4095
lw r1,<3>2997
Instruction to be stored in instruction memory: 0x00103bb5
Load instruction is being performed. The corresponding opcode is 0x00
Register r1 changed from: 0x00000000 to: 0x9216a04c
Enter an instruction number 1. Load 2. Store 3. Display 4. Exit
1
You have selected Load. Instruction should be of the form lw reg,<off>mem
Please find the acceptable range values:
Registers: r0-r7
Memory Location:1024-4095
lw r1,<4>2296
Instruction to be stored in instruction memory: 0x001048f8
Load instruction is being performed. The corresponding opcode is 0x00
Register r1 changed from: 0x9216a04c to: 0xffefcdab
Enter an instruction number 1. Load 2. Store 3. Display 4. Exit
2
You have selected Store. Instruction should be of the form sw reg,<off>mem
Please find the acceptable range values:
Registers: r0-r7
Memory Location:1024-4095
sw r1,<0>4092
Instruction to be stored in instruction memory: 0x01100ffc
Store instruction is being performed. The corresponding opcode is 0x01
Memory Address: 0xffc-0xfff changed to: 0xffefcdab
Enter an instruction number 1. Load 2. Store 3. Display 4. Exit
3
You have selected to display the contents of the memory and registers

Values in Instruction Memory
Loc Value Loc Value Loc Value Loc Value
0x80 00103bb5 0x84 001048f8 0x88 01100ffc 0x8c 61ff698f
0x90 02d7e839 0x94 2c53cbc9 0x98 121e3374 0x9c 9e0cf4d5
0xa0 d49fd4a4 0xa4 597e35cf 0xa8 3222f4cc 0xac cfd3902d
0xb0 48d38f75 0xb4 e6d91d2a 0xb8 e5c0f72b 0xbc 78818744
0xc0 0e5f5900 0xc4 d4618dbe 0xc8 7b051507 0xcc 3b53821f
0xd0 187092da 0xd4 6454cebl 0xd8 853e6915 0xdc f8466a04
```

6) Results after Load/Store:

```
C:\JSU_Dev\projects_mingw\cmpe220g17\Debug\cmpe220g17.exe
0xd30 bcc9b100 0xd34 6ec0e848 0xd38 11a8fee0 0xd3c fc0e99e3
0xd40 b0fae8db 0xd44 5d763fd7 0xd48 a81b01be 0xd4c ab2bc3e2
0xd50 3db3aed8 0xd54 74022588 0xd58 695da880 0xd5c 3bf4f98e
0xd60 57157d8d 0xd64 f6a0e47f 0xd68 e7bb0ddc 0xd6c 8ec6232a
0xd70 2d920dce 0xd74 62cd0522 0xd78 f1c186e7 0xd7c c43f6c3d
0xd80 30d557b0 0xd84 7a475015 0xd88 9a3daf76 0xd8c 3e3a3b8a
0xd90 12cd9489 0xd94 3f0bce3e 0xd98 313c5f5e 0xd9c 9ed53b18
0xda0 c4a73ded 0xda4 f255c9c2 0xda8 490bb034 0xdac c46d532b
0xdb0 76ce0cb2 0xdb4 13a3c906 0xdb8 b237faec 0xdbe d1a0ae48
0xdc0 9af1f8ec 0xdc4 65b178ae 0xdc8 7d8cd7hd 0xdcc 2747b335
0xdd0 e0fc23c0 0xdd4 e77d3ea0 0xdd8 f1b3201a 0xddc 66860cf5
0xde0 3a1e5154 0xde4 db430500 0xde8 bd28e0ba 0xdec 6fb5a3cf
0xdf0 d9bfeec0 0xdf4 c2817534 0xdf8 95499990 0xdfc 64719f46
0xe00 2d910dce 0xe04 5cf357e1 0xe08 9247bbd2 0xe0c a5986a21
0xe10 e87fe84e 0xe14 ac97b5eb 0xe18 06302df5 0xe1c d348ad64
0xe20 fab7f701 0xe24 f1a9d878 0xe28 835108ae 0xe2c d0dc5154
0xe30 95a4f3bd 0xe34 ada748ec 0xe38 d837a4f2 0xe3c bcf37f77
0xe40 29f5d696 0xe44 a14dceee 0xe48 18ae9bf5 0xe4c f868a070
0xe50 6896d7e1 0xe54 ce997460 0xe58 92c5c52e 0xe5c 26575da6
0xe60 43b57134 0xe64 75c7826a 0xe68 d9c73c50 0xe6c 25249ffc
0xe70 eabe5c63 0xe74 1655026f 0xe78 bd415952 0xe7c 19648e19
0xe80 cf5e9184 0xe84 73ff3c15 0xe88 4f02b567 0xe8c 60f895c1
0xe90 a2824aeb 0xe94 8ec3ba40 0xe98 e0152805 0xe9c 9dfb5cf9
0xea0 5558fe2d 0xea4 a5de4416 0xea8 01c8cde1 0xeac b0cccc06
0xeb0 184b6820 0xeb4 3ccc63fc 0xeb8 83a8faf0 0xebc bb060d6e
0xec0 5d0a80d8 0xec4 134ae196 0xec8 77824b67 0xecc 1e888af4
0xed0 d5827fab 0xed4 2a56c6ca 0xed8 2f6297ba 0xedc 7a6eeaf9
0xee0 70dedf2d 0xee4 c42c5cbd 0xee8 3a96f8a0 0xeec b11418b3
0xef0 608d5733 0xef4 604a2cd3 0xef8 6aabc70c 0xefc e3193bb5
0xf00 153be2d3 0xf04 c06dfdb2 0xf08 d16e9c35 0xf0c 7158be6a
0xf10 41d6b861 0xf14 e4914b3f 0xf18 bfeb5f8e 0xf1c fcf048d7
0xf20 d5895373 0xf24 0ff30c9e 0xf28 c470ffcd 0xf2c 663dc342
0xf30 01c36add 0xf34 c0111c35 0xf38 b38afee7 0xf3c cfd5b82e
0xf40 3731f8b4 0xf44 baa8d1a8 0xf48 9c06e811 0xf4c 99a97162
0xf50 27be344e 0xf54 fcb436dd 0xf58 d0f096c0 0xf5c 64c3b5e2
0xf60 c399993f 0xf64 c77394f9 0xf68 e09720a8 0xf6c 11850ef2
0xf70 3b2ee05d 0xf74 9e617360 0xf78 9d86e1c0 0xf7c c18ea51a
0xf80 012a00bb 0xf84 413b9cb8 0xf88 188a703c 0xf8c d6bae31c
0xf90 c67b34h1 0xf94 b00019e6 0xf98 a2b2a690 0xf9c f02671fe
0xfa0 7a4ef4d1 0xfa4 20ca320e 0xfa8 cd499e72 0xfac f12f3806
0xfb0 4f0cf9f3 0xfb4 39787196 0xfb8 68ddaaf7 0xfbc f97161b7
0xfc0 b5683c29 0xfc4 95679e23 0xfc8 853b72f4 0xfcc 69cb55d8
0xfd0 5e4bf6ca 0xfd4 42b3c399 0xfd8 7670c23e 0xfdc e259bc6d
0xfe0 3ae4a16a 0xfe4 809a281e 0xfe8 cbc8b66a 0xfec 467881bb
0xff0 7b9ff5df 0xff4 d2985717 0xff8 54d1a86d 0xffc abcdefff

The register values are as follows:
r0:0x0 r1:0xffefcdab r2:0x0 r3:0x0 r4:0x0 r5:0x0 r6:0x0 r7:0x0

The flag value is:0x0000

The instruction pointer value is:0x0000008c

Enter an instruction number 1. Load 2. Store 3. Display 4. Exit
```