Signed arithmetic explanation:

Negative numbers are stored as 2s complement form and we can identify negative numbers by checking the MSB of the binary representation of the number. 16 in 16-bit signed binary form has 0 at the MSB, hence this is a positive number. -32767 in 16-bit signed binary form has 1 at the MSB and hence this is a negative number.

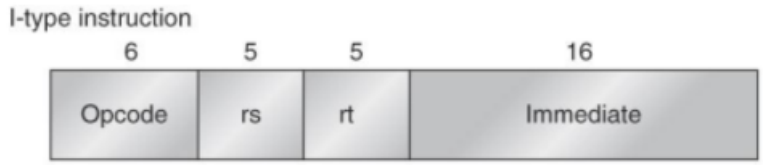
1) **Arithmetic instruction:**

Case a: Add Immediate instruction with positive immediate value

ADDI R17, R18, 16 => R17 = R18 + 16

Since ADDI is an I - type instruction in MIPS, this instruction will be encoded as 06510010 in the memory image

Binary representation: 000001_10010_10001_0000000000010000



Since the MSB of the 16-bit Immediate value is 0, this is a positive number.

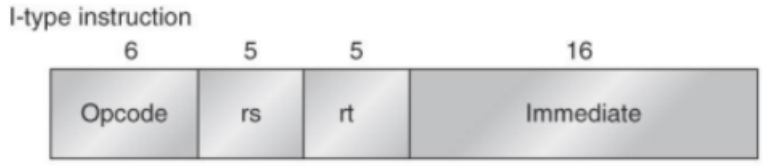
Integer Value = 16

Case b: Add Immediate instruction with negative immediate value

ADDI R17, R18, -32767 => R17 = R18 - 32767

Since ADDI is an I- type instruction in MIPS, this instruction will be encoded as 06518001 in the memory image

Binary representation: 000001_10010_10001_1000000000000001



Since the MSB of the 16-bit Immediate value is 1, this is a negative number and we must perform 2s complement arithmetic.

Note that -32767 is encoded as 1000000000000001 because -32767 is a negative number and negative numbers are stored in 2s complement form.

Conversion of negative number in binary form to an integer value:

Integer value = Unsigned value of 16-bit number - 2^16

Unsigned value of 1000000000000001 is 32769

Integer value = 32769 - 65536 = -32767

(Notes: 32767 in decimal = 0111 1111 1111 1111 in binary => invert each bit: 1000 0000 0000 0000

=> add 1: 1000 0000 0000 0001, therefore -32767 in 2's complement format is 1000 0000 0000 0001).

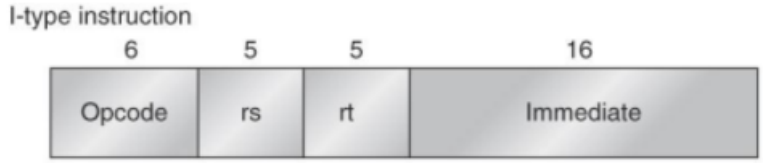
2) **Memory access instruction:**

Case a: Load word instruction with positive immediate value

LDW R17, (R18) 16 => R17 = Mem[R18 + 16]

Since LDW is an I - type instruction in MIPS, this instruction will be encoded as 32510010 in the memory image

Binary representation: 001100_10010_10001_0000000000010000



Since the MSB of the 16-bit Immediate value is 0, this is a positive number.

Integer Value = 16

Assuming R18 = 2000,

Since each instruction in the memory image is 4 byte aligned,

R17 = Memory_image_file[(2000 + 16)/4] , since (2000+16)/4 = 504

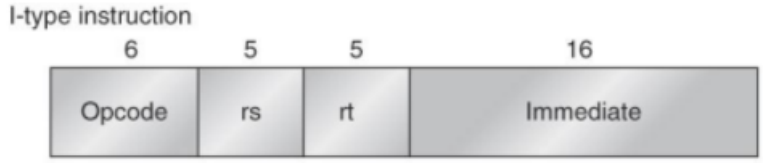
R17 will contain line 504 of the memory image file

Case b: Load word instruction with negative immediate value

LDW R17, R18, -32767 => R17 = Mem[R18 – 32767]

Since LDW is an I- type instruction in MIPS, this instruction will be encoded as 32518001 in the memory image

Binary representation: 001100_10010_10001_1000000000000001



Since the MSB of the 16-bit Immediate value is 1, this is a negative number and we must perform 2s complement arithmetic.

Note that -32767 is encoded as 1000000000000001 because -32767 is a negative number and negative numbers are stored in 2s complement form.

Conversion of negative number in binary form to an integer value:

Integer value = Unsigned value of 16-bit number - 2^16

Unsigned value of 1000000000000001 is 32769

Integer value = 32769 - 65536 = -32767

Assume R18 = 39999

Since each instruction in the memory image is 4 byte aligned,

R17 = Memory_image_file[(39999 - 32767)/4]

R17 will contain line 1808 of the memory image file

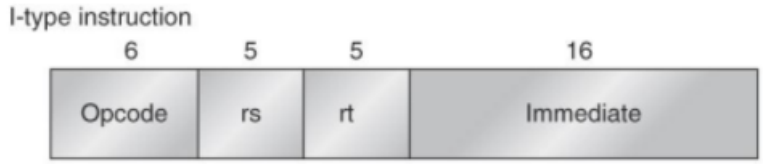
3) **Branch instruction:**

Case a: Branch instruction with positive immediate value

BEQ R17, R18, 16 => PC = PC + 16

Since BEQ is an I - type instruction in MIPS, this instruction will be encoded as 3E510010 in the memory image

Binary representation: 001111_10010_10001_0000000000010000



Since the MSB of the 16-bit Immediate value is 0, this is a positive number.

Integer Value = 16

Assuming R18 = R17,

Since each instruction in the memory image is 4 byte aligned,

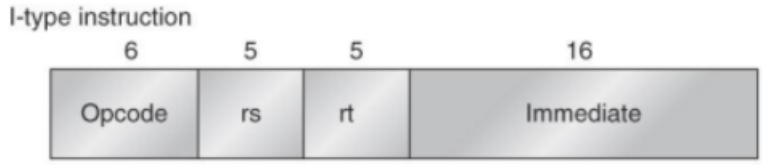
PC = Memory_image_file[PC + 16]

Case b: Branch instruction with negative immediate value

BEQ R17, R18, -2 => PC = PC -2

Since BEQ is an I- type instruction in MIPS, this instruction will be encoded as 3E51FFFE in the memory image

Binary representation: 001111_10010_10001_ 1111111111111110



Since the MSB of the 16-bit Immediate value is 1, this is a negative number and we must perform 2s complement arithmetic.

Note that -2 is encoded as 1111111111111110 because -2 is a negative number and negative numbers are stored in 2s complement form.

Conversion of negative number in binary form to an integer value:

Integer value = Unsigned value of 16-bit number - 2^16

Unsigned value of 1111111111111110 is 65534

Integer value = 65534 - 65536 = -2

Assume R18 = R17

Since each instruction in the memory image is 4 byte aligned,

PC = Memory_image_file[PC - 2]