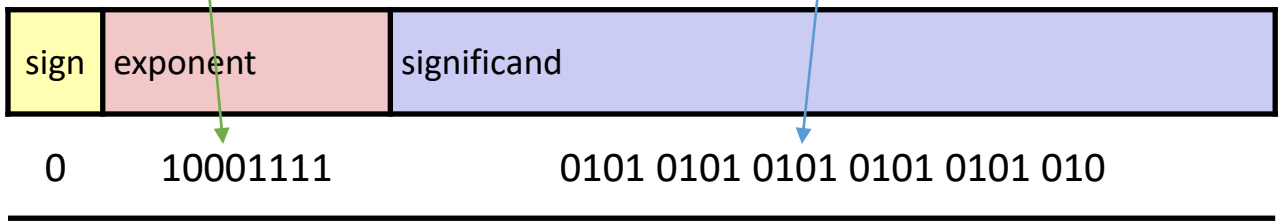
CS 520
IEEE to Integer Conversion

Convert Float to Integer – f2i

- IEEE FP Value: $47AAAAAA_{16}$

→0100 0111 1010 1010 1010 1010 1010 1010



Stored Value = 143

Actual Value = 143– 127 = 16

--> $1.01010101010101010101010 \times 2^{16}$

Move the decimal point right by the exponent value (16)

--> $1010101010101010101.0101010 \times 2^{0}$

Discard the bits to the right of the decimal (since we just want an integer)

Take the shifted bits and pad it with leading zeroes to make it a 32-bit value

--> 0000 0000 0000 0001 0101 0101 0101 0101

--> 0 0 0 1 5 5 5 5 (Hex)

- For negative values, do all the steps above and negate the result

f2i

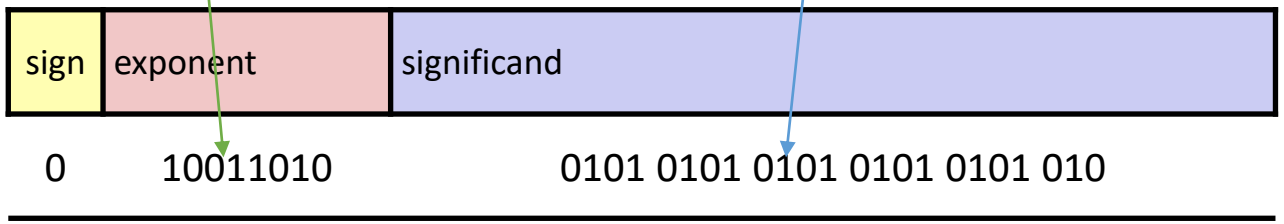
- Implementation in C, one way to do this to capture the significand in an integer (32-bit) in the first 23 bits

- To discard the decimal digits shift right (23 – actual exponent value)

Convert Float to Integer – f2i

- IEEE FP Value: $4D2AAAAA_{16}$

→0100 1101 0010 1010 1010 1010 1010 1010



Stored Value = 154

Actual Value = 154– 127 = 27

--> $1.01010101010101010101010 \times 2^{27}$

--> $10101010101010101010100000.0 \times 2^{0}$ ---> add 4 extra zeroes to the right!**

Take the shifted bits and pad it with leading zeroes to make it a 32-bit value

--> 00001010 1010 1010 1010 1010 1010 0000

--> 0 A A A A A A 0 (Hex)

**Accuracy is lost since we are assuming zeroes

f2i

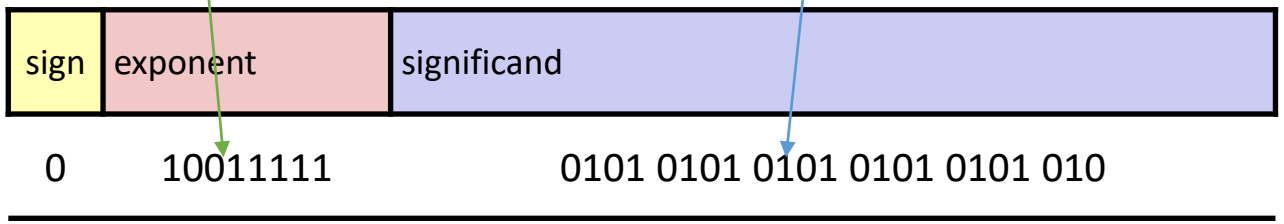
- Implementation in C, one way to do this is to once again capture the significand in an integer (32-bit) in the first 23 bits

- Since our exponent value is more than 23 we will need to shift left (exponent value - 23) to add those trailing zeroes

Convert Float to Integer – f2i

- IEEE FP Value: $4FAAAAAA_{16}$

→0100 1111 1010 1010 1010 1010 1010 1010



Stored Value = 159

Actual Value = 159– 127 = 32

--> $1.01010101010101010101010 \times 2^{32}$

To shift left 32 positions we will cause an overflow the integer 32-bit container!

- Floating point value is too big to be represented
- On the hardware (Intel-IA 32) it uses the biggest possible negative number as a way to set the error flag
- Most negative value is 80000000_{16}

f2i – other errors

- Exponent all zeroes or de-normalized value – very close to zero
  - Return zero
- Exponent all zeroes
  - Shifting decimal point to the right will make it very close to zero
  - Return zero
- Exponent all ones (NaN,Infinity)
  - Return the most negative number (80000000)