Another Example

- π, rounded to 24 bits of precision, has:
  - sign = 0 ;
  - *e* = 1 ;
  - *s* = 11001001000011111011011 (including the hidden bit)
  - The sum of the exponent bias (127) and the exponent (1) is 128, so this is represented in single precision format as

0 10000000 10010010000111111011011
(excluding the hidden bit) = 0x40490FDB

- In binary single-precision floating-point, this is represented as *s* = 1.10010010000111111011011 with *e* = 1. This has a decimal value of
- **3.141592**7410125732421875, whereas a more accurate approximation of the true value of π is
- **3.14159265358979323846264338327950**... The result of rounding differs from the true value by about 0.03 parts per million, and matches the decimal representation of π in the first 7 digits. **The difference is the discretization error and is limited by the machine epsilon.**

Why are we doing this?

- Can't use integers for everything
- Trying to cover a much broader range of real values; but something has to give, and it's the precision
- Pi a good example:
  - Whether or not a rational number has a terminating expansion depends on the base.
    - For example, in base-10 the number 1/2 has a terminating expansion (0.5) while the number 1/3 does not (0.333...).
    - In base-2 only rationals with denominators that are powers of 2 (such as 1/2 or 3/16) are terminating. Any rational with a denominator that has a prime factor other than 2 will have an infinite binary expansion.

Special values

- The hardware that does arithmetic on floating point numbers must be constantly checking to see if it needs to use a hidden bit of a 1 or a hidden bit of a 0 (for 0.0)
- Zero could be 0x00000000 or 0x80000000
  - What numbers cannot be represented because of this?

	S	E	F	hidden bit
0.0	0 or 1	all zero	all zero	0
subnormal	0 or 1	all zero	not all zero	0
normalized	0 or 1	>0	any bit pattern	1
+infinity	0	11111111	00000… (0x7f80 0000)	
-infinity	1	11111111	00000… (0xff80 0000)	
NaN*	0 or 1	0xff	anything but all zeros	

* Not a Number

5-bit floating point representation with one sign bit, two exponent bits (k=2) and two fraction bits (n=2); the exponent bias is $2^{2-1}-1 = 1$

bits	e	E	2^E	f	M	2^ExM	V	Decimal
0 00 00	0	0	1	0/4	0/4	0/4	0	0.00
0 00 01	0	0	1	1/4	1/4	1/4	1/4	0.25
0 00 10	0	0	1	2/4	2/4	2/4	1/2	0.50
0 00 11	0	0	1	3/4	3/4	3/4	3/4	0.75
0 01 00	1	0	1	0/4	4/4	4/4	1	1.00
0 01 01	1	0	1	1/4	5/4	5/4	5/4	1.25
0 01 10	1	0	1	2/4	6/4	6/4	3/2	1.50
0 01 11	1	0	1	3/4	7/4	7/4	7/4	1.75
0 10 00	2	1	2	0/4	0/4	8/4	2	2.00
0 10 01	2	1	2	1/4	1/4	10/4	5/2	2.50
0 10 10	2	1	2	2/4	2/4	12/4	3	3.00
0 10 11	2	1	2	3/4	3/4	14/4	7/2	3.50
0 11 00	--	--	--	--	--	--	inf	--
0 11 01	--	--	--	--	--	--	NaN	--
0 11 10	--	--	--	--	--	--	NaN	--
0 11 11	--	--	--	--	--	--	NaN	--

Note the transition between denormalized and normalized
Have to always check for hidden bit

e: the value represented by considering the exponent field to be an unsigned integer

E: the value of the exponent after biasing = e - bias

2^E: numeric weight of the exponent

f: the value of the fraction

M: the value of the significand =1+f ==1.f

2^ExM: the (unreduced) fractional value of the number

V: the reduced fractional value of the number

Decimal: the decimal representation of the number

Denormalized values

- Also called denormal or subnormal numbers
- Values that are very close to zero
- Fill the “underflow” gap around zero
- Any number with magnitude smaller than the smallest normal number
- When the exponent field is all zeros
- E = 1-bias
- Significand M = f without implied leading 1
- Two purposes
  - Provide a way to represent numeric value 0
    - -0.0 and +0.0 are considered different in some ways and the same in others
  - Represents numbers that are very close to 0.0
    - Gradual underflow = possible numeric values are spaced evenly near 0.0

Denormal numbers (cont)

- In a normal floating point value there are no leading zeros in the significand, instead leading zeros are moved to the exponent.
- For example:
  - 0.0123 would be written as $1.23 * 10^{-2}$.
- Denormal numbers are numbers where this representation would result in an exponent that is too small (the exponent usually having a limited range). Such numbers are represented using leading zeros in the significand.