

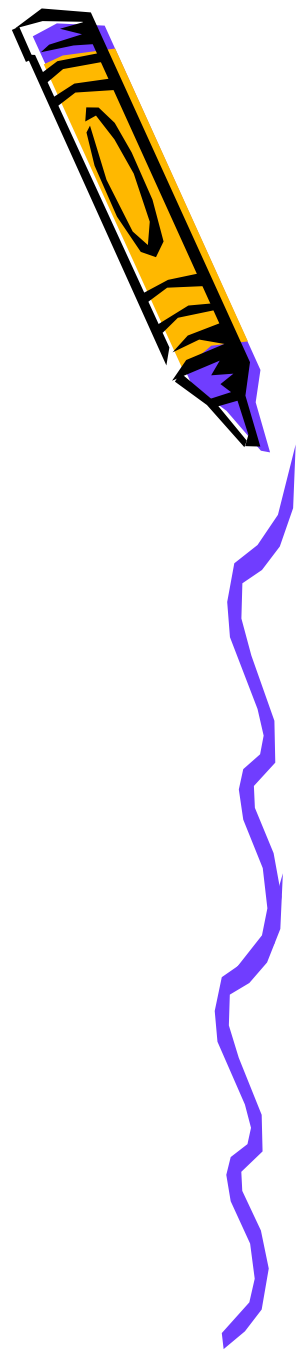


Computer Architecture

Hossein Asadi
Department of Computer Engineering
Sharif University of Technology
asadi@sharif.edu

Today's Topics

- Fixed Point
- Floating Point



Copyright Notice



- Parts (text & figures) of this lecture adopted from:
 - Computer Organization & Design, The Hardware/Software Interface, 3rd Edition, by D. Patterson and J. Hennessey, MK publishing, 2005.
 - "Computer Organization & Design" handouts, by Prof. Kumar, UIUC, Fall 2007.



Real Numbers in Computers



- Fixed-Point Representation
 - Example: $d_{23}d_{22}\dots d_1d_0.f_0f_1f_2f_3f_4f_5f_6f_7$
 - 24-bit: integer bits
 - 8-bit: fraction bits
- Application
 - Used in CPUs with no floating-point unit
 - Embedded microprocessors and microcontrollers
 - Digital Signal Processing (DSP) applications



Real Numbers in Computers



- Fixed-Point Representation
 - Pros
 - Simple hardware
 - Fast computation
 - Cons
 - Low precision
 - Small range



Real Numbers in Computers



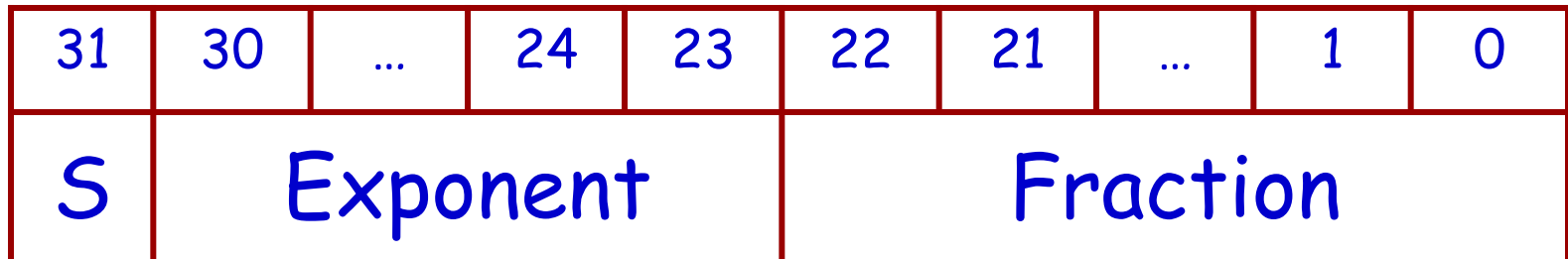
- Floating-Point Representation
 - Scientific notation in base 2
 - $1.xxxxxx_{\text{two}} * 2^{yyyy}$



Floating-Point Notation



- FP Notation Consists of:
 - Fraction (F): 23 bits
 - Exponent (E): 8 bits
 - Sign bit (S)
 - Also called, single precision floating-point
- $N = (-1)^S * F * 2^E$



Floating-Point Notation (cont.)



- Pros (compared to fixed-point)
 - Very Wide Range
 - More precision bits
- Cons (compared to fixed-point)
 - Arithmetic operation more complicated
 - HW more complicated
 - More time-consuming

31	30	...	24	23	22	21	...	1	0
S	Exponent				Fraction				



Floating-Point Notation (cont.)

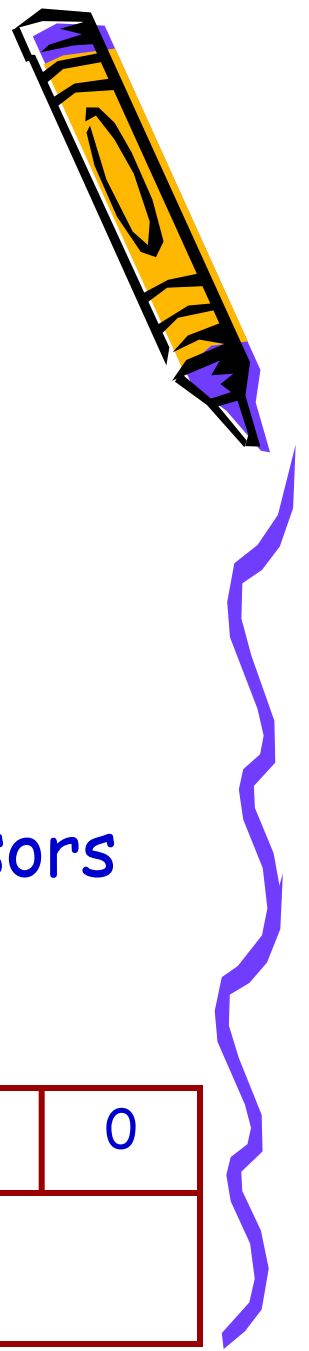


- Precision versus Range
 - Wider range \rightarrow less precision?
 - More precision \rightarrow smaller range?

31	30	...	24	23	22	21	...	1	0
S	Exponent				Fraction				



Floating-Point Notation (cont.)



- IEEE 754 FP Standard
 - $N = (-1)^S * (1 + F) * 2^E$
 - Significand: $1 + F$
 - Fraction: F
 - Used in MIPS and most microprocessors

31	30	...	24	23	22	21	...	1	0
S	Exponent				Fraction				



Floating-Point Notation (cont.)

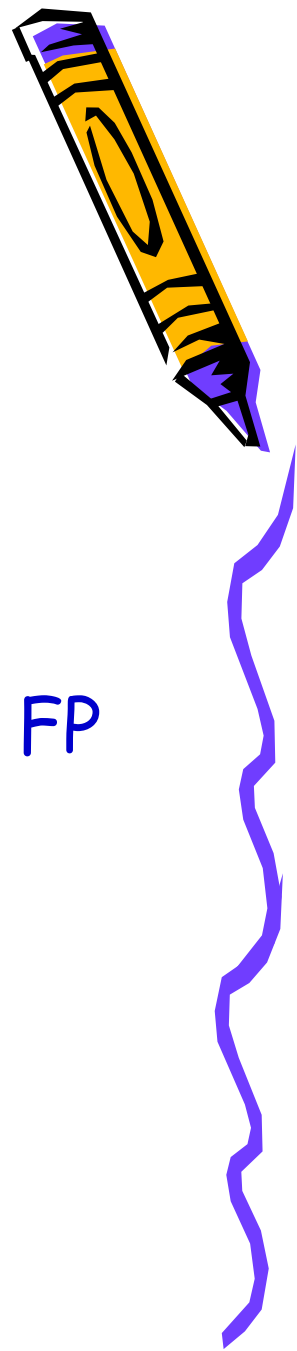


- Overflow:
 - Can we have overflow in FP notation?
 - Exponent too large to fit in "Exponent" field
- Underflow:
 - Non-zero fraction so small to represent
 - Negative exponent too large to fit

31	30	...	24	23	22	21	...	1	0
S	Exponent				Fraction				



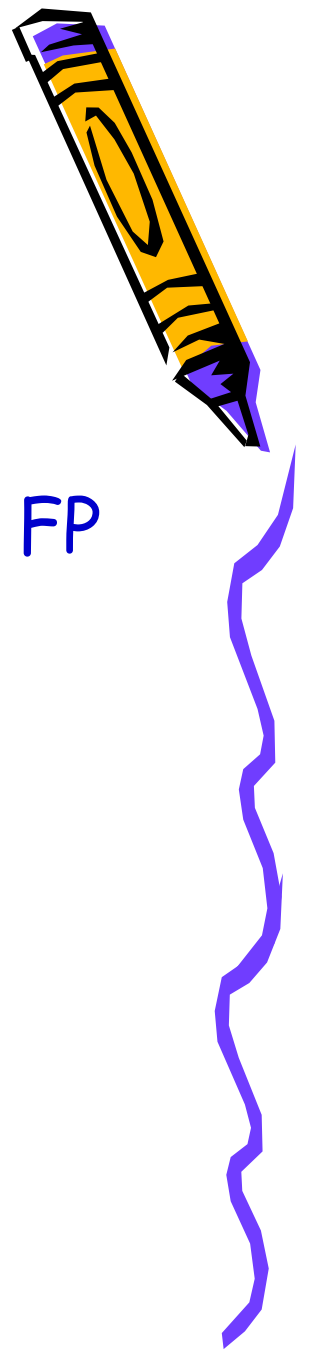
Floating-Point Notation (cont.)



- Biased-Notation in Exponent Field
 - Used in IEEE 754 FP Standard
 - In order to compare FP numbers faster
 - Uses a bias of 127 in single-precision FP
 - $N = (-1)^S * (1 + F) * 2^{(E-bias)}$



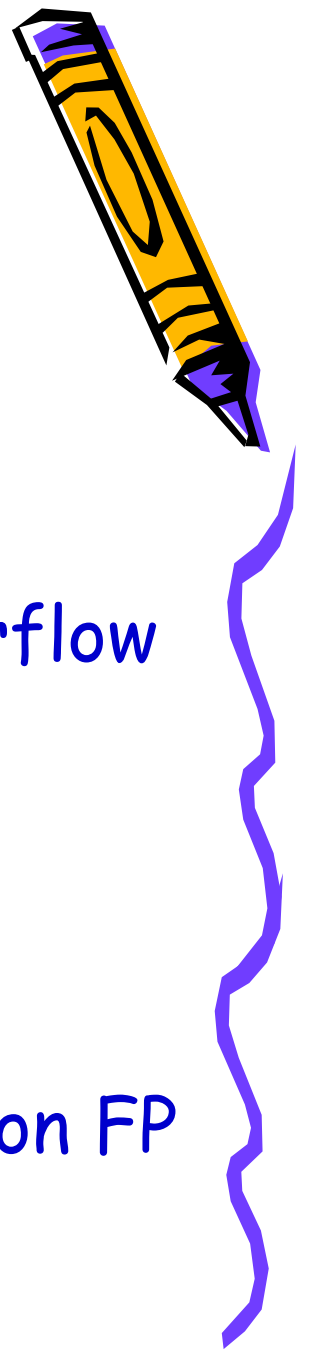
Floating-Point Notation (cont.)



- Biased-Notation in Exponent Field
 - Uses a bias of 127 in single-precision FP
 - $N = (-1)^S * (1 + F) * 2^{(E-bias)}$
 - 0 reserved
 - (-126) represented by $-126+127 = 1$
 - (-1) represented by $-1+127 = 126$
 - (0) represented by $0+127 = 127$
 - (+1) represented by $1+127 = 128$
 - (+127) represented by $127+127 = 254$
 - 255 reserved



Floating-Point Notation (cont.)



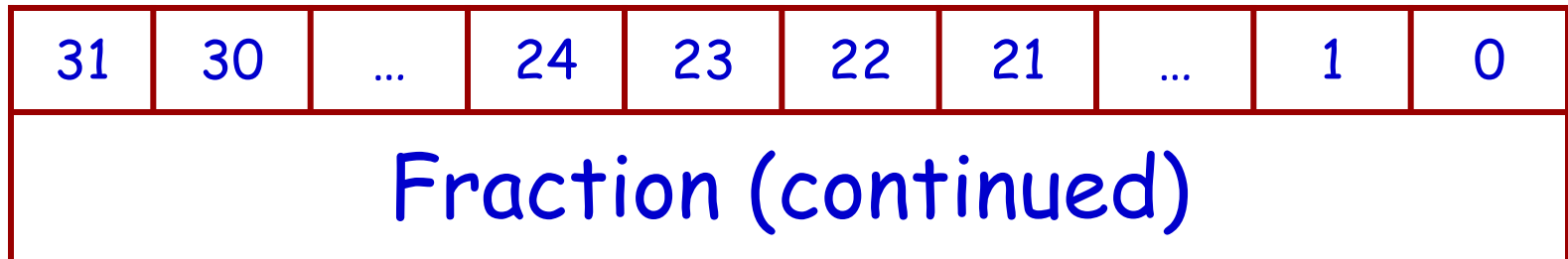
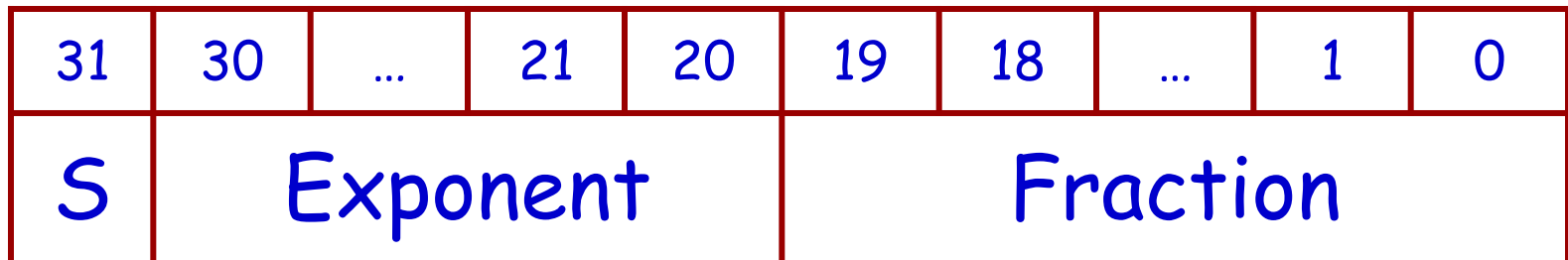
- Double-Precision Floating-Point
 - Uses two words
 - Reduces chances of overflow & underflow
 - Format
 - Fraction (F): 52 bits
 - Exponent (E): 11 bits
 - Sign bit (S)
 - Uses a bias of 1023 in double-precision FP



Floating-Point Notation (cont.)



- Double-Precision Floating-Point
 - Fraction (F): 52 bits
 - Exponent (E): 11 bits
 - Sign bit (S)



Floating-Point Notation (cont.)



Single Precision		Double Precision		Object Represented
Exponent	Fraction	Exponent	Fraction	
0	0	0	0	0
0	Nonzero	0	Nonzero	Denormalized
1-254	Anything	1-2046	Anything	FP No
255	0	2047	0	Infinity
255	Nonzero	2047	Nonzero	NaN



Floating-Point Notation (cont.)



- $N = (-1)^S * (1 + F) * 2^E$
- Questions on Single Precision FP:
 - Smallest positive number?
 - $1.0000\ 0000\ 0000\ 0000\ 0000\ 000_{\text{two}} * 2^{-126}$
 - Smallest absolute negative number?
 - $-1.0000\ 0000\ 0000\ 0000\ 0000\ 000_{\text{two}} * 2^{-126}$

31	30	...	24	23	22	21	...	1	0
S	Exponent				Fraction				



Floating-Point Notation (cont.)



- $N = (-1)^S * (1 + F) * 2^E$
- Questions on Single Precision FP:
 - Largest positive number?
 - $1.1111\ 1111\ 1111\ 1111\ 1111\ 111_{\text{two}} * 2^{+127}$
 - Largest absolute negative number?
 - $-1.1111\ 1111\ 1111\ 1111\ 1111\ 111_{\text{two}} * 2^{+127}$

31	30	...	24	23	22	21	...	1	0
S	Exponent				Fraction				



Floating-Point Notation (cont.)



- Denormalized Numbers
 - Smallest positive normalized number
$$= 1.0000\ 0000\ 0000\ 0000\ 0000\ 000_{\text{two}} * 2^{-126}$$
$$= 1_{\text{two}} * 2^{-126}$$
 - Smaller positive numbers using exponent 0
$$= 0.0000\ 0000\ 0000\ 0000\ 0000\ 001_{\text{two}} * 2^{-126}$$
$$= 1_{\text{two}} * 2^{-149}$$



Floating-Point Notation (cont.)



- Practice:

- Represent following number in IEEE 754 single-precision FP

- (-0.75)

$$= -\frac{3}{4} = -3 * 2^{-2} = -11_{\text{two}} * 2^{-2} = -0.11_{\text{two}}$$

$$= -1.1_{\text{two}} * 2^{-1} = -1.1_{\text{two}} * 2^{127-1} = -1.1_{\text{two}} * 2^{126}$$

31	30	...	24	23	22	21	...	1	0
S	Exponent				Fraction				
1	01111110				10000000000000000000000000000000				



Floating-Point Notation (cont.)



- FP Addition

- Example:

- $1.000_{\text{two}} * 2^{-1} + -1.110_{\text{two}} * 2^{-2}$

$$\begin{aligned} & 1.0000_{\text{two}} * 2^{-1} \\ + & -0.1110_{\text{two}} * 2^{-1} \\ = & 0.0010 * 2^{-1} \\ = & 1.0 * 2^{-4} \end{aligned}$$



Floating-Point Notation (cont.)



- Another Practice:

- Convert (7.75) in IEEE 754 single-precision FP

$$= 7 + \frac{3}{4} = 111_{\text{two}} * 2^0 + 11_{\text{two}} * 2^{-2} =$$

$$= 1.11_{\text{two}} * 2^2 + 0.0011_{\text{two}} * 2^2$$

$$= 1.1111_{\text{two}} * 2^2$$

$$= 1.1111_{\text{two}} * 2^{2+127} = 1.1111_{\text{two}} * 2^{129}$$

31	30	...	24	23	22	21	...	1	0
S	Exponent				Fraction				
0	10000001				11110000000000000000000000000000				



Backup

