6. Gotta have at least one good number problem; let's make it a floating-point question. In the space provided below, show how to add two floating-point numbers together. The two numbers are:

   0 10000100 10101100101101011110000 (call this one X)

   1 01111101 01110101011011010110000 (call this one Y)

   Show how the alignment is done, and any post normalization that may be needed. Also, state any assumptions that you are making in the process.

X = 1.10101100101101011110000 x 2^5

Y = 1.01110101011011010110000 x 2^-2

10000100
-01111111
101 = 5₁₀

01111101
-01111111
11111110 = -2₁₀

Comparing the exponents:

5 - (-2) = 7

Since 7 is positive, Y is smaller, and gets shifted right 7 places

Addition:

101 011 00101 1010 1111 0000
000 0000 1000 1010 1001 0011
101 011 0111 00101 1000 0011

Since there are still 23 bits in the result, the result will not be "normalized" already is

01111111
00000101
10000100 (new exponent)

Answer:

0 | 10000100 | 10101101110010110000011

not quite

7. A program runs in 10 seconds on Computer ONE, which has a 4 GHz clock. In the works is Computer TWO, which should run the same program in 6 seconds. A substantial increase in the clock rate is possible, but because of the way that the increase affects the rest of the CPU design, Computer TWO will require 1.25 times as many clock cycles as Computer ONE for the program. What clock rate should the designer target for the new machine?

$$\text{Speedup} = \frac{10}{6} = \frac{5}{3}$$

$$\frac{(\text{New Clock})}{1.25\,(4\,\text{GHz})} = \frac{5}{3}$$

$$\text{New clock} = \frac{5}{3} \times 5\,\text{GHz} = \boxed{8.33\ \text{GHz}}$$

6. Gotta have at least one good number problem; let's make it a floating-point question. In the space provided below, show how to add two floating-point numbers together. The two numbers are:
   0 10000100 10101100101101011110000 (call this one X)
   1 01111101 01110101011011010110000 (call this one Y)

   Show how the alignment is done, and any post normalization that may be needed. Also, state any assumptions that you are making in the process.

1.10101100101101011110000 = X

diff = 7, need 7 bits

1.1.01110101011011010110000 = Y

10000100 − 1111111 = 100

132

65

X = 110101 100101101011110000

Y = 7. 111011101010110110101100000

V = 127 + X

6 = 127 + X

102

1 25

The alignment is done by shifting the smaller exponent value untill it has the same exponent as the larger one

X = 1.10101100101101011110000000 Exp 5
+ y = 1.11111110111010101101101011 Exp 5

0.1010101110100000101110 1011 Exp 5

0.10101011110100000101110 1011 Exp 5

The number then needs to be normalized so it becomes

1.0101011101000000101110 1011 Exp 4

So the answer is

0 | 10000011 | 01010111010000000101111010

7. A program runs in 10 seconds on Computer ONE, which has a 4 GHz clock. In the works is Computer TWO, which should run the same program in 6 seconds. A substantial increase in the clock rate is possible, but because of the way that the increase affects the rest of the CPU design, Computer TWO will require 1.25 times as many clock cycles as Computer ONE for the program. What clock rate should the designer target for the new machine?

$$10s = \frac{cycles}{4 \times 10^9}$$

4×10^{10} Instructions in program

$$6s = \frac{1.25 \times 4 \times 10^{10}}{X}$$

$$X = \frac{1.25 \times 4 \times 10^{10}}{6s}$$

$$X = \frac{5 \times 10^{10}}{6s}$$

$$X = 8.33 \text{ GHz}$$

They should target a <u>8.33 GHz Clock rate.</u>

8×10^5
$+ 3 \times 10^2$

6. Gotta have at least one good number problem; let's make it a floating-point question. In the space provided below, show how to add two floating-point numbers together. The two numbers are:

exp: 132 0 10000100 10101100101101011110000 (call this one X)

exp: 125 1 01111101 01110101011011010110000 (call this one Y)

Show how the alignment is done, and any post normalization that may be needed. Also, state any assumptions that you are making in the process.

Add: Need to align the numbers first:

shift Y to make exponents equal (don't forget the hidden bit):

1.01110101011011010110000

goes to: 0000010111010101101101011

These #'s now have the same exp, so they can be added, or in this case subtracted:

1.10101100101101011110000
− 0.00000101110101011011010110 11

1.10100000100001010111 0101

no need to adjust the exp.

(assuming we don't want to lose any precision, and the hardware will support the '1' on the end of Y)

~~I'd rather add them; so invert Y:~~

~~1.1010110010110101111 0000~~
~~+ 1.1111010001010100 1001 0100~~
~~11.1010000100001010 0111 0100~~

~~adjust the exponent (~~

X + Y = 0 | 10000100 | 1010000010000101001110 11

sgs SDS

7. A program runs in 10 seconds on Computer ONE, which has a 4 GHz clock. In the works is Computer TWO, which should run the same program in 6 seconds. A substantial increase in the clock rate is possible, but because of the way that the increase affects the rest of the CPU design, Computer TWO will require 1.25 times as many clock cycles as Computer ONE for the program. What clock rate should the designer target for the new machine?

Computer 1	Computer 2
10 seconds	6 seconds
4 GHz $\left(\frac{\text{cycles}}{\text{sec}}\right)$	? GHz

1) $(10\ \text{sec})(4 \times 10^9\ \frac{\text{cycles}}{\text{sec}}) = 4 \times 10^{10}$ cycles

2) $(4 \times 10^{10})(1.25) = \frac{5 \times 10^{10}\ \text{cycles}}{6\ \text{sec}} = 8.33 \times 10^9\ \frac{\text{cycles}}{\text{sec}} = \boxed{8.33\ \text{GHz}}$

6. Gotta have at least one good number problem; let's make it a floating-point question. In the space provided below, show how to add two floating-point numbers together. The two numbers are:
   0 10000100 10101100101101011110000 (call this one X)
   1 01111101 01110101011011010110000 (call this one Y)
   Show how the alignment is done, and any post normalization that may be needed. Also, state any assumptions that you are making in the process.

i) Compare the exponent

$Exponent_X = 128 + 4 = 132$, $132 - 127 = 5$ ($exponet_X$)

$Exponent_Y = 64+32+16+8+4+1 = 125$, $125 - 127 = -2$ ($exponet_Y$)

ii) $X = 2^{-2} \times \{2^7 \times (1.1010 1100 1011 0101111)\}$

$Y = (-1) \times 2^{-2} \times (1.0111 0101 0110 1101011)$

11010110.0101101011110000 000
 − 1.0111010101101101011

11010100.1110010110000010101

exponent = 7   Exponent = 7 + 127 = 134

sign
0 | 10000110 | 11100101100000101010000

7. A program runs in 10 seconds on Computer ONE, which has a 4 GHz clock. In the works is Computer TWO, which should run the same program in 6 seconds. A substantial increase in the clock rate is possible, but because of the way that the increase affects the rest of the CPU design, Computer TWO will require 1.25 times as many clock cycles as Computer ONE for the program. What clock rate should the designer target for the new machine?

Computer$_1$: 10 sec / 4 GHz

$$\text{time} = \frac{\text{clock Cycles}}{4 \times 10^9} = 10 \text{ sec}$$

$$\text{Clock Cycles} = 40 \times 10^9$$

Computer$_2$: 6 sec / X GHz

$$\text{time} = \frac{1.25 \times 40 \times 10^9}{X} = 6 \text{ sec}$$

$$\text{Clock rate: } X = \frac{1.25 \times 40 \times 10^9}{6} \cong 8.334 \text{ GHz}$$

6. Gotta have at least one good number problem; let's make it a floating-point question. In the space provided below, show how to add two floating-point numbers together. The two numbers are:
   0 10000100 10101100101101011110000 (call this one X)
   1 01111101 01110101011011010110000 (call this one Y)
   Show how the alignment is done, and any post normalization that may be needed. Also, state any assumptions that you are making in the process.

$X = (-1)^0 \times (1 + R) \times 2^E$

$E = 10000100 = 132 - \text{bias} \Rightarrow 132 - 127 = 5$ ✓

Finally I'm not gonna use this

$R = (1 - 2^{-1}) + (1 \times 2^{-3}) + (1 \times 2^{-5}) + (1 \times 2^{-6}) + (1 \times 2^{-9}) + (1 \times 2^{-11}) + (1 \times 2^{-12}) + (1 \times 2^{-14}) + (1 \times 2^{-16}) + (1 \times 2^{-17}) + (1 \times 2^{+18}) + (1 \times 2^{-19}) = 0.674650192$

$X = 1.674650192 \times 2^5$

$Y = (-1)^1 \times (1 + R) \times 2^E$

$E = 01111101 = 125 - \text{bias} \Rightarrow 125 - 127 = -2$

$Y = -1.01110101011011010110 \times 2^{-2} = -0.0000001011101010110110101 \times 2^5$

$X = 1.10101100101101011111 \times 2^5$

comp-2 (Y) = 1.1111110100010101001001010 1

(X): + 1.1010110010110101111

11.1010100111001011000 0010 1 01

not quite

$11.1010100111001011000001010 1 \times 2^5 =$

$= 1.11010100111001011000001\boxed{0101} \times 2^6$

23 bits

7. A program runs in 10 seconds on Computer ONE, which has a 4 GHz clock. In the works is Computer TWO, which should run the same program in 6 seconds. A substantial increase in the clock rate is possible, but because of the way that the increase affects the rest of the CPU design, Computer TWO will require 1.25 times as many clock cycles as Computer ONE for the program. What clock rate should the designer target for the new machine?

ONE → 10 secs, 4GHz

TWO → 6 secs, 1.25 clock cycles of ONE

total-time = clock-cycles × cycle-time

ONE: $10 = \frac{\text{clock-cycles}}{4 \cdot 10^9} \Rightarrow \text{clock-cycles} = 4 \cdot 10^{10}$ cycles

TWO: $6 = \frac{1.25 \times \text{clock-cycles}}{\text{clock-rate}} \Rightarrow \text{clock-rate} = \frac{5 \cdot 10^{10}}{6} \approx 8\text{GHz}$

$\frac{5}{6} = 8.33$

exp - bias = ... 125 - 127 = -2 132 - 127 = 5

6. Gotta have at least one good number problem; let's make it a floating-point question. In the space provided below, show how to add two floating-point numbers together. The two numbers are:
   0 10000100 10101100101101011110000 (call this one X) 5659376
   1 01111101 01110101011011010110000 (call this one Y)

   Show how the alignment is done, and any post normalization that may be needed. Also, state any assumptions that you are making in the process. We use 1 bit for the sign, 8 for the exponent and 23 for the mantisa

$-1^S (F+1) \cdot 2^{exp-bias}$

X = 10101100101101011110000 · 2^5

-Y = 01110101011011010110000 · 2^{-2}

1) Adjust to the bigger exponent

-Y = 0 000000 01110101011011010110000 · 2^5

Add

Y = 1 1111111000101010010100100000 · 2^5

2) add

  10101100101101011110000
+ 11111111100010101010010001

  10101100010000000011001

3) round   no time!

7. A program runs in 10 seconds on Computer ONE, which has a 4 GHz clock. In the works is Computer TWO, which should run the same program in 6 seconds. A substantial increase in the clock rate is possible, but because of the way that the increase affects the rest of the CPU design, Computer TWO will require 1.25 times as many clock cycles as Computer ONE for the program. What clock rate should the designer target for the new machine?

$$\text{CPU ex. time} = \frac{\#\text{ Clock cycles for the program 1}}{\text{Clock rate 1}}$$

ONE: $$10 = \frac{\#\text{ Clock cycles 1}}{4 \cdot 10^{9}} \Rightarrow \#\text{ Clock cycles} = 4 \cdot 10^{10}$$

TWO: $$6 = \frac{1'25 \cdot \#\text{ Clock cycles 1}}{\text{Clock rate 2}} \Rightarrow$$

$$6 = \frac{1'25 \cdot 4 \cdot 10^{10}}{\text{Clock rate 2}} \Rightarrow \text{Clock rate 2} = \frac{1'25 \cdot 4 \cdot 10^{10}}{6} \Rightarrow$$

Clock rate 2 = 6'25 MHz

2.3×10^3 + 2.4×10^-2 2300. .024

6. Gotta have at least one good number problem; let's make it a floating-point question. In the space provided below, show how to add two floating-point numbers together. The two numbers are:

0 10000100 10101100101101011110000 (call this one X)

1 01111101 01110101011011010110000 (call this one Y)

Show how the alignment is done, and any post normalization that may be needed. Also, state any assumptions that you are making in the process.

X sign = (+) X exp = 128+4 = 132 − 127 = 5

Y sign = (−) Y exp = 64+32+16+8+4+1 = 125 − 127 = −2

96 + 29

X: 1.10101100101101011110000

− .0101110101011011010110000

Result: 0 10000100 1011000111001011000010

very close

7. A program runs in 10 seconds on Computer ONE, which has a 4 GHz clock. In the works is Computer TWO, which should run the same program in 6 seconds. A substantial increase in the clock rate is possible, but because of the way that the increase affects the rest of the CPU design, Computer TWO will require 1.25 times as many clock cycles as Computer ONE for the program. What clock rate should the designer target for the new machine?

Comp 1	Comp 2
10 s	6 s
4 GHz	1.25 x

clock cycles $= 10s \times \frac{4\times10^9}{s} = 4\times10^{10}$

$4\times10^{10} \cdot 1.25 = 5\times10^{10}$

$\frac{5\times10^{10} \text{ clock cycles}}{X} = 6\ s$

$X = \frac{5\times10^{10}}{6}$

$6\overline{)5.00} = 0.833$

X = 8.33 GHz

6. Gotta have at least one good number problem; let's make it a floating-point question. In the space provided below, show how to add two floating-point numbers together. The two numbers are:

   0 10000100 10101100101101011110000 (call this one X)

   1 01111101 01110101011011001010000 (call this one Y)

   Show how the alignment is done, and any post normalization that may be needed. Also, state any assumptions that you are making in the process.

Sign exp fract

1) look at the exponents notice X is larger than Y hence we will have a positive result

normalize the exponents and add them together

x exp 1000100

Y exp 0111101

(1) 0000001 in excess of 127

x frac: 101 0110 0101 1010 1111 0000

Y frac: 011 1010 1011 0110 1011 0000

excess (1) 001 0001 0001 0001 1010 0000

shift radix point

2) we have 2 over flows in the EXP & fract, so shift radix point

S | exp |

0 | 10000000 | 1 1001 001 0001 001 1010 000

NO

-2.5

$T = \frac{inst}{sec} = \frac{cyc}{sec}$ $T = \frac{inst}{sec} = \frac{cy}{clk}$

7. A program runs in 10 seconds on Computer ONE, which has a 4 GHz clock. In the works is Computer TWO, which should run the same program in 6 seconds. A substantial increase in the clock rate is possible, but because of the way that the increase affects the rest of the CPU design, Computer TWO will require 1.25 times as many clock cycles as Computer ONE for the program. What clock rate should the designer target for the new machine?

clk = 4GHz

Comp 1: T = 10s, 4GHz clk

comp 2: T = 6s 1.25 clk

$10s \times \frac{4E^9}{s} = 4E^{10}$ cycles

$1.25 \times 4E^{10}$

$1.25 \times 4 = 5.00$

$5E^{10}$ cycles

clk rate comp 2 = $\frac{5}{6}E^{12}$ Hz

6. Gotta have at least one good number problem; let's make it a floating-point question. In the space provided below, show how to add two floating-point numbers together. The two numbers are:
   0 10000100 10101100101101011110000 (call this one X)
   1 01111101 01110101011011010110000 (call this one Y)
   Show how the alignment is done, and any post normalization that may be needed. Also, state any assumptions that you are making in the process.

ExpX + ExpY = 1000 0100
why?
+0111 1101
1 1111 1101
overflow

sign bit x + sign bit Y = 0
+ 1
1

manx' + many' {hidden bit}

1.10 10110 0101 1010 1111 0000
+ 1.011 1010 1011 0110 1011 0000

7. A program runs in 10 seconds on Computer ONE, which has a 4 GHz clock. In the works is Computer TWO, which should run the same program in 6 seconds. A substantial increase in the clock rate is possible, but because of the way that the increase affects the rest of the CPU design, Computer TWO will require 1.25 times as many clock cycles as Computer ONE for the program. What clock rate should the designer target for the new machine?

one

10sec @ 4 GHz

TWO

6 sec

1.25(4) = 5

The clk rate the designer should target is ~~5~~GHz