Azwad Shameem

Assignment 2
CS 30100

1. What is *p* for your system?

The *p* I am using for this system is 22. One of the requirements of the system is that it has a machine epsilon equal to or less than 1.11 x 10^{-16}. The equation for machine epsilon is the following, $\boldsymbol{\varepsilon = ½(\beta^{1-p})}$. Since we know the formula for machine epsilon, it is possible for us to plug in **β** and **ε** and solve for *p*. We know that **β** is 6 since we are creating a floating point system based on HexBits and by requirement we know the machine epsilon must be 1.11 x 10^{-16} or less. After solving for p, we get 1.11 x 10^{-16} = ½(6^{1-p}) and p is equal to about 21.1165. Clearly, we cannot use 21.1165 because we can't have .1165 digits, so we will try out 21. However, after trying out 21 in the equation $\boldsymbol{\varepsilon = ½(\beta^{1-p})}$, we get about ½($6^{1-21}$) = 1.36 x 10^{-16}, which is more than the machine epsilon allowed. This time we will try out p = 22, which leads us to about ½(6^{1-22}) = 2.2793 x 10^{-17}, clearly this is lower than the machine epsilon given and therefore, is usable.

2. What is *q* for your system?

The *q* for my system is 378. One of the requirements of the system is that it the largest number that can be represented in your system must be greater than or equal to the largest binary 64 number that can be represented in the IEEE Standard-754, which is +1.7976931348623157x10^{308}. In order to find the largest positive floating number is by using the equation $\boldsymbol{\beta}^{emax}$ ($\boldsymbol{\beta}$-$\boldsymbol{\beta}^{1-p}$). Utilizing the equation for largest positive floating number we can obtain the e_{max} by substituting the **β** for 6, since we are using a HexBit floating point system, and p as 22 because we have decided on that precision from question 1 and set it equal to +1.7976931348623157x10^{308}. As a result, we get +1.7976931348623157x10^{308} = $6^{\mathbf{emax}}$ (6-6^{1-p}) as the equation and we end up getting an e_{max} of about 398.48, so for simplicity we can round up to 399 since we know it has to be equal to or greater than. Also to view *q* as an integer we have the inequality of e_{min} <= *q* + *p*-1 <= e_{max}, which tells us that q can be at most 399 = *q* + *p*-1 and we know p is 22 so, 399 = *q* + 22 - 1, which leads us to 399 - 21 = q = 378.

3. What is the offset for the exponent for your system?

The offset for the exponent for the system is 799, [0...799]. We have the requirement that the representation of the exponent must use an offset (rather than an explicit sign). Therefore, we know we have to use the offset representation with e_{max} and e_{min} which is 399 and -399 respectively. As a result we have [-399...399] which can give us [0...798] and we know that the exponent must use an offset by the requirement so we use [0...799].

4. What is the wobble of your system?

The system has a wobble of 1.899×10^{-17}. To find the wobble we use the inequality equation for Err_{rel} is $½(\boldsymbol{\beta}^{-p}) <= Err_{rel} <= ½(\boldsymbol{\beta}^{1-p})$. We know that **β** is 6 because we are creating a HexBit floating point system and we choose *p* as 22 from question 1. After substituting the numbers into the inequality we obtain $3.799 \times 10^{-18} <= Err_{rel} <= 2.279 \times 10^{-17}$. To find wobble we find the variation in relative error by doing $2.279 \times 10^{-17} - 3.799 \times 10^{-18} = 1.899 \times 10^{-17}$.

5. What is the machine epsilon of your system?

The machine epsilon of our system is 2.2793×10^{-17}. To find the machine epsilon we use the equation $\boldsymbol{\varepsilon = ½(\beta^{1-p})}$ and substitute the values we already know, **β** as 6 and *p* as the answer choice from question one, 22. Essentially the equation gives us, $½(6^{1-22}) = 2.2793 \times 10^{-17}$ and we know this machine epsilon is allowed because we know that our machine epsilon needs to be equal or lower than 1.11×10^{-16} and clearly 2.2793×10^{-17} is smaller than 1.11×10^{-16}.