Brandon Vo, Amy Qi, Andrea Yangco, Diane Nguyen

1. Explain the difference between: general purpose computing and embedded Microsystems. Give 5 differences between the two.

A) Performance - General purpose computing is usually much faster and cares less about balancing system size and system. General purpose will have multi core, GPU, an OS for concurrency, etc.

B) Cost - A general purpose computing system will usually be much more expensive, having tons of transistors and integrated circuits, a potential GPU, multiple cores, etc.

C) Power and heat - While both may need or may not need cooling on the specific application, embedded microsystems usually run on relatively low voltages and wattages, like 10W, general purpose computers may vary hugely. A top end GPU just by itself will run at 450W, which requires a huge amount of cooling.

D) Abstraction - Embedded microsystems run on low-level assembly languages like x86, AVR, etc. They may also run on C with a compiler and have a real-time OS with modern day engineering, but that is usually the best they can do. General purpose computers will usually have an OS and abilities to run very high level languages like Python and Javascript, which require pretty much 0 knowledge of hardware interfacing.

E) Size - Embedded microsystems are usually very small. They can range from a microcontroller with peripheral board to a very small microprocessor. General purpose computers have less size restraints and can be huge if using high wattage components that need tons of cooling.

2. What is a bus? Explain how the data bus and address bus are used.

A) A bus is a physical connection (wire for example) that transfers data to the computer at a constant frequency, also called bus width. The address bus is the bus that transfers the memory addresses for reading and writing to the processor. The data bus has the data that was read or written.

3. Your friend doesn't understand how a 4,096MB video file fits on 4GB flash drive when the video file seems 96MB larger. How would you explain it to him?

A) A 4GB representation is not just 4GB. It is a shortening that we use because it's easier to say and use. 4GB is actually 4,294,967,296 bytes, which is 4294MB. This is larger than the video file. I would explain to him by telling him to look up how much bytes the rounded size actually is and then convert to MB to compare.

****The final answers will be highlighted in blue for the following questions.****

4. Explain the similarity between -16 and 16 in signed binary as (a) 8-bit, (b)16-bit, (c)32-bit. Binary is split into 4 pairs (hex-like) for easier reading/writing

A) 8-bit
   a) 16 is 0b0001 0000
      -16 is 0b1111 0000

B) 16-bit
   a) 16 is 0b0000 0000 0001 0000
      -16 is 0b1111 1111 1111 0000

C) 32-bit
   a) 16 is 0b0000 0000 0000 0000 0000 0000 0001 0000
      -16 is 0b1111 1111 1111 1111 1111 1111 1111 0000

In general, the positive number 16 will always stay the same (extended with 0's on the left to fit the number of bits), as the 5th bit will always be equal to 16. For a signed -16, we extend all the extra bits with 1's when going to 16-bit and 32-bit. As an 8-bit number, they are pretty similar in the ease of calculation. The main problem with the -16 is finding which combination of numbers and bits solves the following: -128 + bit_values = -16. We could use 2's complement to do this easier, but for argument's sake, it is generally easier to determine the positive. This becomes more problematic with more bits since we must solve the equation: -32,768 + bit_values = -16. Thankfully for the number 16, there is an easy and clear pattern for determining the negative version in binary.

5. Show signed and unsigned range for a 6 bit number.

A) 6 bit unsigned number:
   a) Max = 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0
      = 32 + 16 + 8 + 4 + 2 + 1 = 63
      Min = 000000 = 0

Range = 0 to 63

B) <u>6 bit signed number:</u>

a) Max = 0 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0
= 0 + 16 + 8 + 4 + 2 + 1
= 31

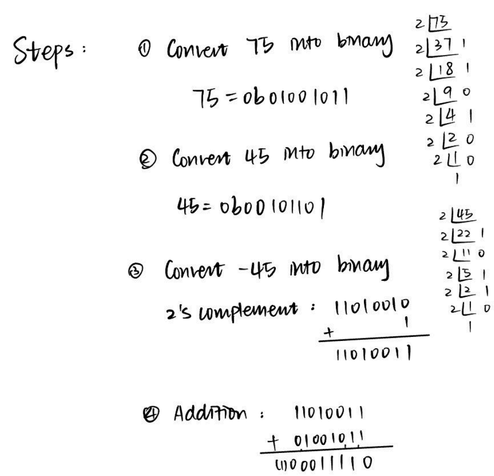
Min = -2^5 + 0 + 0 + 0 + 0 + 0 + 0
= -32

Range = -32 to 31

6. Perform that following subtraction as 8 bit signed addition. Show the step by step results (convert the numbers to binary and show the results). Comment on the correctness of your answers: (a) 75 – 45, (b) -60 – 120, (c) 64 + 64, (d) 45 – 75, (e) 156 – 50

A) 75 - 45
= 0b01001011 + 0b11010011
= 0b00011110 (correct 8-bit answer)
= 0b(1)00011110 (original 9-bit answer with overflow)

The answer is correct because the decimal calculation result (30) equals the binary calculator result in 8 bits. In this case, the overflow can be ignored or it will affect the correct final result.



B) -60 - 120
= 0b11000100 + 10001000
= 0b01001100 (incorrect 8-bit answer)
= 0b(1)01001100 (correct 9-bit answer)

This number is incorrect due to the fact -180 can only be represented by 9 bits. The range of a signed 8-bit number is -127 to127. The 8-bit answer is equal to 76, which is different from the correct answer, -180.

Steps:

① Convert -60 into binary

60 = 0b00111100

2|60
2|30 0
2|15 0
2|7 1
2|3 1
2|1 1
1

2's complement:

$$\begin{array}{r} 11000011 \\ +\quad 1 \\ \hline 11000100 \end{array}$$

-60 = 0b11000100

② Convert -120 into binary

2|120
2|60 0
2|30 0
2|15 0
2|7 1
2|3 1
2|1 1
1

120 = 0b01111000

2's complement:

$$\begin{array}{r} 10000111 \\ +\quad 1 \\ \hline 10001000 \end{array}$$

-120 = 0b10001000

③ Addition:

$$\begin{array}{r} 11000100 \\ +10001000 \\ \hline (1)01001100 \end{array}$$

C) 64 + 64
= 0b01000000 + 0b01000000
= 0b10000000

This answer is incorrect. In this case, the decimal calculation result is 128, but the binary calculation result is -128. The two results don’t match each other since it’s a signed addition.

Steps:

① convert 64 to binary

2|64
2|32 0
2|16 0
2|8 0
2|4 0
2|2 0
2|1 0
1

64 = 0b010000000

② Addition

$$\begin{array}{r} 01000000 \\ +\ 01000000 \\ \hline 10000000 \end{array}$$

D) 45 - 75
= 0b00101101 + 0b10110101
= 0b11100010
This answer is correct because the binary number is equal to the decimal calculation result (-30) and can be represented by 8 bits.

Steps: ① convert 45 to binary
as shown in (A)
45 = 0b00101101
② convert -75 to binary
75 = 0b01001011
2's complement: 10110100
+ 1
= 10110101
③ Addition: 10110101
+ 00101101
= 11100010

E) 156 - 50
= 0b10011100 + 0b11001110
= 0b01101010 (correct 8-bit answer)
= 0b(1)01101010 (9-bit answer with overflow)

The answer is correct because the decimal calculation result (106) equals the binary calculator result in 8 bits. In this case, the overflow can be ignored or it will affect the correct final result.

Steps: ① Convert 156 into binary

2 | 156			2 | 50	
2 | 78	0		2 | 25	0
2 | 39	0		2 | 12	1
2 | 19	1		2 | 6	0
2 | 9	1		2 | 3	0
2 | 4	1		2 | 1	1
2 | 2	0		1	
2 | 1	0			
1				

156 = 0b10011100
② Convert -50 into binary
50 = 0b00110010
2's complement: 11001101
+ 1
= 11001110
③ Addition: 11001110
10011100
= (1)01101010

7. Convert the following binary values into real decimals: (a) 10011.0011, (b) 110110.110110

A) 10011.0011
= 2^4 + 2^1 + 2^0 + 2^-3 + 2^-4
= 16 + 0 + 0 + 2 + 1 + 0 + 0 + 1/8 + 1/16
= 19.1875

B) 110110.110110

= 2^5 + 2^4 + 2^2 + 2^1 + 2^-1 + 2^-2 + 2^-4 + 2^-5

= 32 + 16 + 0 + 4 + 2 + 0 + ½ + ¼ + 0 + 1/16 + 1/32 + 0

= 54.84375

8. Convert the following number into binary real values (fractions up to 15 bits): (a) 23.45, (b) 67.891 **add steps later**

A) 23.45

= 10111.011100110011001

= 23 + ¼ + ⅛ + 1/16 + ….

Steps : ① integer part (23)

23 = 0b10111

2	23	
2	11	1
2	5	1
2	2	1
2	1	0
		1

② Fraction Part (.45)

0.45 × 2 = 0.90	0.90 × 2 = 1.80	0.80 × 2 = 1.60	0.60 × 2 = 1.20	0.20 × 2 = 0.40	0.40 × 2 = 0.80	0.80 × 2 = 1.60	0.60 × 2 = 1.20
0.20 × 2 = 0.40	0.40 × 2 = 0.80	0.80 × 2 = 1.60	0.60 × 2 = 1.20	0.20 × 2 = 0.40	0.40 × 2 = 0.80	0.80 × 2 = 1.60	

0.45 = 0b0.011100110011001

23.45 = 10111.01110011001101

B) 67.891

= 1000011.111001000001100

= 64 + 2 + 1 + ½ + ¼ + ⅛ + ….

Steps : ① integer part (67)

67 = 0b1000011

2	67	
2	33	1
2	16	1
2	8	0
2	4	0
2	2	0
2	1	0
		1

② Fraction part (.891)

0.891 × 2 = 1.782	0.782 × 2 = 1.564	0.564 × 2 = 1.128	0.128 × 2 = 0.256	0.256 × 2 = 0.512	0.512 × 2 = 1.024	0.024 × 2 = 0.048	0.048 × 2 = 0.096
0.096 × 2 = 0.192	0.192 × 2 = 0.384	0.384 × 2 = 0.768	0.768 × 2 = 1.536	0.536 × 2 = 1.072	0.072 × 2 = 0.144	0.144 × 2 = 0.288	

0.891 = 0b0.111001000001100

67.891 = 1000011.111001000001100

9. Convert the following values to 1.7 and 1.15 format: (a) 1.5, (b) 0.1, (c) -1.5, (d) -0.1

A) 1.5
   a) 1.7 format: 1.5 x 128 = 192
      i) Convert 192 to binary: 192%2 = 0; 96%2 = 0; 48%2 = 0; 24%2 = 0; 12%2 = 0; 6%2 = 0; 3%2 = 1; 1 left → 0b11000000
      ii) Prove the binary answer: (2^7+2^6)/128 = 1.5
   b) 1.15 format: 1.5 x 2^15 = 49152
      i) Convert 49152 to binary: 49152%2 = 0; 24576%2 = 0; 12288%2 = 0; 6144%2 = 0; the rest will follow the same method until the number cannot be divided by 2 anymore → 0b1100000000000000
      ii) Prove the binary answer: (2^15+2^14)/2^15 = 1.5

B) 0.1
   a) 1.7 format: 0.1 x 128 = 12.8 (decimal) ~= 12
      i) Convert 12 to binary: 12%2 = 0; 6%2 = 0; 3%2 = 1; 1 left; → 0b00001100
      ii) Prove the binary answer: (2^2+2^3)/128 = 0.09375 ~= 0.1

   b) 1.15 format: 0.1 x 2^15 = 3276.8 ~= 3276
      i) Convert 3276 to binary: 3276%2 = 0; 1638%2 = 0; 819%2 = 1; the rest will follow the same method until the number cannot be divided by 2 anymore → 0b0000110011001100
      ii) Prove the binary answer: (2^2+2^3+2^6+2^7+2^10+2^11)/2^15 = 0.0999755859 ~= 0.1

C) -1.5
   a) 1.7 format: -1.5 x 128 = number outside range (min value is -1)
   b) 1.15 format: → -1.5 x 2^15 = number outside range (min value is -1)

D) -0.1
   a) 1.7 format: -0.1 x 128 = -12.8 ~= -12
      i) Convert -12 to binary: First, we will convert 12 into binary like what we did in B). Then, we will do the 2's complement on the binary number by flipping the 1s and 0s and adding 1. Finally, we will have -12 in binary → 0b11110100
      ii) Prove the binary answer: (2^2+2^4+2^5+2^6-2^7)/128 = -0.09375

   b) 1.15 format: -0.1 x 2^15 = -3276.8 ~= -3276

i) Convert -3726 to binary: First, we will convert 3276 into binary like what we did in B). Then, we will do the 2's complement on the binary number by flipping the 1s and 0s and adding 1. Finally, we will have -3276 in binary → 0b1111001100110100

ii) Prove the binary answer: (2^2+2^4+2^5+2^8+2^9+2^12+2^13+ 2^14-2^15)/2^15 = -0.0999755859 ~= -0.1