CS200 - Computer Organization

Data Internals - Part1

Aravind Mohan

Allegheny College

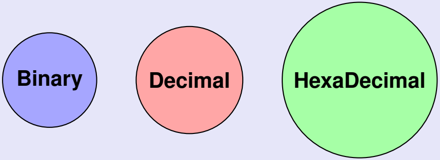
October 5, 2021

Motivation to learn data representation

- How is data represented internally?
- Examine how data is referenced inside a Program?

How is Data represented internally?

- **Binary:** is readable by hardware.
- **Decimal:** is readable by human beings.
- **HexaDecimal:** is readable by storage devices.



Binary Vs Decimal

- What does the number $(123)_{10}$ mean?
  - $100 + 20 + 3$
  - $1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$
- What does the number $(1001010)_2$ mean?
  - $1000000 + 1000 + 10$
  - $1 \times 2^6 + 1 \times 2^3 + 1 \times 2^1$
  - $2^6 + 2^3 + 2^1 = 64 + 8 + 2 = 74$

Data referencing in a C Program

```
1   #include <stdio.h>
2   int main(){
3       int alpha = 100;
4       printf("%d", alpha);
5   }
```

Q_1: What happens when line 3 is executed?

Q_2: What happens when line 4 is executed?

Q_3: How is data referenced in Memory when lines 3 and 4 are executed?

Decimal to Binary

Q_1: What happens when line 3 is executed?

- Divide repeatedly by 2 and retain the remainders. Continue until the quotient = 0.
- For example, 245_{10}

- $245 \div 2 = 122$	R = 1 LSB
- $122 \div 2 = 61$	R = 0
- $61 \div 2 = 30$	R = 1
- $30 \div 2 = 15$	R = 0
- $15 \div 2 = 7$	R = 1
- $7 \div 2 = 3$	R = 1
- $3 \div 2 = 1$	R = 1
- $1 \div 2 = 0$	R = 1 MSB

- Solution is 11110101_2

Q_2: What happens when line 4 is executed?

- Starting with the most significant bit (left to right), repeatedly multiply by 2, adding each bit as we move along.
- For example, 1010111_2
  - $(0+1) \times 2 = 2$
  - $(2+0) \times 2 = 4$
  - $(4+1) \times 2 = 10$
  - $(10+0) \times 2 = 20$
  - $(20+1) \times 2 = 42$
  - $(42+1) \times 2 = 86$
  - $(86+1) = 87$

  - Solution is 87_{10}

Hexa Decimal To Binary

Q_3: How is data referenced in Memory when lines 3 and 4 are executed?

- Expand each hexadecimal digit into the corresponding 4 binary digits:
- For example: $(1234AF0C)_{16}$
  - 0001 0010 0011 0100 1010 1111 0000 1100
  - Solution: $1234AF0C_{16}$ = $00010010001101001010111100001100_2$

Binary to Hexa Decimal

Q_3: How is data referenced in Memory when lines 3 and 4 are executed?

- Create groups of 4 bits (LSB to MSB), and translate each group to its corresponding Hex:
- For example: 11001011101_2

- 110	0101	1101_2
- 6	5	D_{16}

- Solution: $11001011101_2 = 65D_{16}$

Q_3: How is data referenced in Memory when lines 3 and 4 are executed?

- Starting with the most significant digit, repeatedly multiply by 16, adding each digit as we move along.
- For example, $24E_{16}$
  - $(0 + 2) \times 16 = 32$
  - $(32 + 4) \times 16 = 576$
  - $(576 + 14(E)) = 590$
  - Solution is 590_{10}

Decimal to Hexa Decimal

Q_3: How is data referenced in Memory when lines 3 and 4 are executed?

- Divide repeatedly by 16 and retain the remainders. Continue until the quotient = 0.
- For example, 53241_{10}

- $53241 \div 16 = 3327$	R = 9	LSB
- $3327 \div 16 = 207$	R = 15(F)	
- $207 \div 16 = 12$	R = 15(F)	
- $12 \div 16 = 0$	R = 12(C)	MSB
- Solution is $CFF9_{16}$		

Try out!

Class Activity:
Upload your solution to your class repo (git) to get class participation credits.

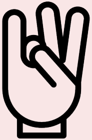
- Convert $(10101010)_2$ to Decimal
- Convert $(87)_{10}$ to Binary
- Convert $(DECAF)_{16}$ to Decimal
- Convert $(1234567)_{10}$ to HexaDecimal
- Convert $(3A8D2)_{16}$ to Binary
- Convert $(1110100101001010010 1)_2$ to HexaDecimal

Fractions: Binary Vs Decimal

- Starting with the least significant bit, divide the value by 2 and add the next bit. Continue to the binary point.
- For example, 0.01101_2
  - $(0+1)/2 = 1/2$
  - $(1/2+0)/2 = 1/4$
  - $(1/4+1)/2 = 5/8$
  - $(5/8+1)/2 = 13/16$
  - $(13/16+0)/2 = 13/32$

  - Solution: $0.01101_2 = 13/32$

Fractions: Decimal Vs Binary



- Multiply repeatedly by 2 and subtract the whole numbers until the multiplicand = 0.
- For example, 0.6875_{10}

  - $0.6875 \times 2 = 1.375$ Most Significant Bit
  - $0.375 \times 2 = 0.75$
  - $0.75 \times 2 = 1.5$
  - $0.5 \times 2 = 1.0$ Least Significant Bit

  - Solution is $0.6875_{10} = 0.1011_2$

Signed Binary

```
1  #include <stdio.h>
2  int main(){
3      int alpha = 10;
4      int beta = 3;
5      int gamma = 5;
6      alpha += beta;
7      alpha -= gamma;
8      printf("%d\n", alpha);
9  }
```

Q_1: What happens when lines 6, 7,and 8 are executed?

How does Binary (2 bit) Add Work?

$\oplus$

Rules:

1. 0 + 0 = 0
2. 0 + 1 = 1
3. 1 + 0 = 1
4. 1 + 1 = 0 with carry (1)

a	b	output	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

How does Binary (3 bit) Add Work?

$\oplus$

Rules:

1. $0 + 0 + 0 = 0$
2. $0 + 0 + 1 = 1$
3. $0 + 1 + 0 = 1$
4. $0 + 1 + 1 = 0$ with carry (1)
5. $1 + 0 + 0 = 1$
6. $1 + 0 + 1 = 0$ with carry (1)
7. $1 + 1 + 0 = 0$ with carry (1)
8. $1 + 1 + 1 = 1$ with carry (1)

How does Binary (3 bit) Add Work?

a	b	carry in	output	carry out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Try out

How does computer's execute:

- (10 + 3)
- (8 + 7)
- (5 + 6)

How does Binary Subtraction Work?

- What does (5 - 2) mean to computers?

- (5 - 2) = (5 + (-2))

  How do we represent (-2) in binary?

How is Signed Data represented internally?

- In decimal we are quite familiar with placing a "-" sign in front of a number to denote that it is negative.
- The same is true for binary numbers a computer won't understand that.
- What happens in memory then?

Binary Negative Numbers

- There are several representations
  - Signed magnitude
  - One's complement
  - Two's complement
- Two's complement is the system used in microprocessors
- Most significant bit becomes important

Signed Magnitude

- Represent the decimal number as binary.
- Left bit (MSB) used as the sign bit.
- Only have 7 bits to express the number.

$12_{10} = 00001100$

$-12_{10} = 10001100$

Signed Magnitude Limitation

- What is -7 in signed magnitude? (duplicates)
- How does computer's execute (5 - 2) using signed magnitude?

One's Complement

- Method: Invert the ones and zeros

  $11_{10} = 00001011$

  $-11_{10} = 11110100$
- 0 in MSB implies positive
- 1 in MSB implies negative

One’s Complement Limitation

- What is 1111 in one’s complement? (duplicate)

Two’s Complement

- Method: Take the one’s complement and add 1 to the result. **most stable**

11_{10} = 00001011

-11_{10} = 11110100 one’s comp

-11_{10} = 11110101 two’s comp

Reading Assignment

Section 1.10 in **PH**

Questions

Do you have any questions from this class discussion?