|  |
| --- |
| **Họ và tên:** Kiều Nguyễn Thiên Phúc  **Môn:** Kiến trúc máy tính và hợp ngữ  **Khoa:** Công nghệ thông tin  **Lớp:** COMP101106 |

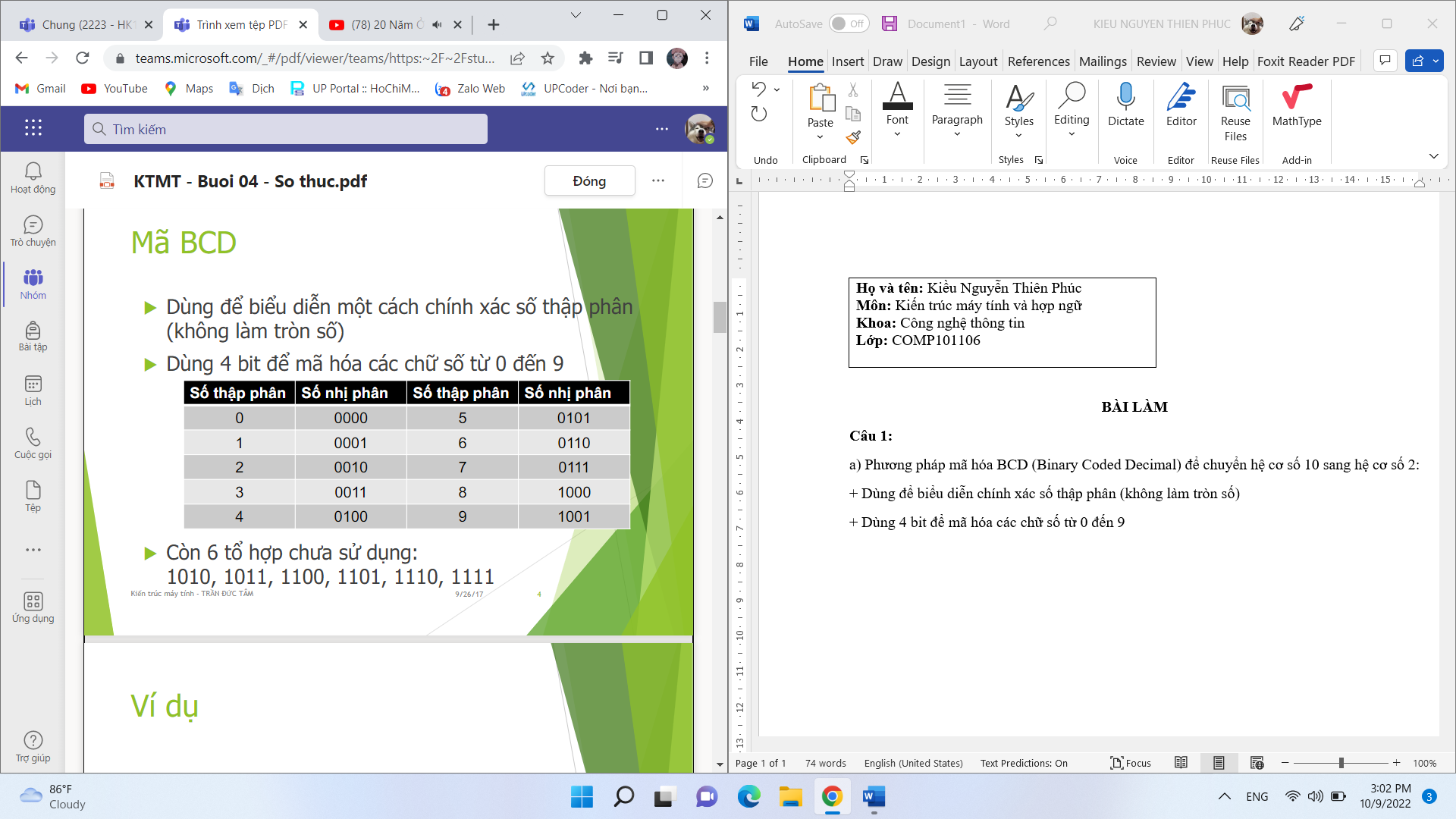
**BÀI LÀM**

**Câu 1:**

a) Phương pháp mã hóa BCD (Binary Coded Decimal) để chuyển hệ cơ số 10 sang hệ cơ số 2:

+ Dùng để biểu diễn chính xác số thập phân (không làm tròn số)

+ Dùng 4 bit để mã hóa các chữ số từ 0 đến 9



+ Ví dụ:

35 10 = 0011 0101 (BCD)

61 10 = 0110 0001 (BCD)­­

35.61 10 = 0011 0101.0110 0001 (BCD)

b) Tràn số (overflow) là hiện tượng xảy ra khi kết quả của phép tính vượt quá độ chính xác giới hạn cho phép (của máy tính).

Tràn số chỉ xảy ra khi cộng (hoặc trừ) 2 số cùng dấu.

+ Đối với số không dấu: Hiện tượng tràn số xảy ra khi kết quả bị dư ra bit nhớ

Ví dụ: 1111 + 0011 = 10010 (overflow – kết quả là dãy 5 bits, nhưng giới hạn cho phép chỉ là 4 bits)

+ Đối với số có dấu: Hiện tượng tràn số xảy ra khi dương cộng dương ra âm, âm cộng âm ra dương

Ví dụ: 0111 + 0111 = 1110 (overflow, 7 + 7 = 14, nhưng kết quả lại có bit msb là 1, bit 1 thể hiện số âm)

**Câu 2:**

a) (160)8 = 1\*82 + 6\*81 + 0\*80 = (112) 10

112 / 2 = 56 dư **0**

56 / 2 = 28 dư **0**

28 / 2 = 14 dư **0**

14 / 2 = 7 dư **0**

7 / 2 = 3 dư **1**

3 / 2 = 1 dư **1**

1 / 2 = 0 dư **1**

Vậy (160) 8 = (112) 10 = (0111 0000) 2

b) (11011.011) 2 = 1\*2 4  + 1\*2 3 + 1\*2 1 + 1\*2 0 + 1\*2 -2 + 1\*2 -3 = (27.375) 10

27 / 16 = 1 dư **11**

1 / 16 = 0 dư **1**

=> phần nguyên: 1B

0.375 x 16 = **6**.0

=> phần thập phân = 6

Vậy (11011.011) 2 = (1B.6) 16

**Câu 3:**

a) 63 10 x 11 10

63 10 = 0011 1111 2

11 10 = 0000 1011 2

Đặt M = 0000 1011 2

|  |  |  |  |
| --- | --- | --- | --- |
| C | A | Q |  |
| 0 | 0000 | 0011 1111 | = 1; +M |
| 0 | 1011 | 0011 1111 |  |
| 0 | 0101 | 1001 1111 | Shift; =1; +M |
| 1 | 0000 | 1001 1111 |  |
| 0 | 1000 | 0100 1111 | Shift; =1; +M |
| 1 | 0011 | 0100 1111 |  |
| 0 | 1001 | 1010 0111 | Shift; =1; +M |
| 1 | 0100 | 1010 0111 |  |
| 0 | 1010 | 0101 0011 | Shift; =1; +M |
| 1 | 0101 | 0101 0011 |  |
| 0 | 1010 | 1010 1001 | Shift; =1; +M |
| 1 | 0101 | 1010 1001 |  |
| 0 | 1010 | 1101 0100 | Shift; = 0 |
| 0 | 0101 | 0110 1010 | Shift; = 0 |
| 0 | 0010 | 1011 0101 |  |

Vậy kết quả của phép tính 63 10 \* 11 10 = 0011 1111 2 \* 0000 1011 2 = 0000 0010 1011 0101 2

b) 92 10 – 17 10

92 10 = 0101 1100 ­2­­

17 10 = 0001 0001

M = 0101 1100 2

Bù 2 của N = 1110 1111

Tổng = 1 0100 1011

Bỏ 28 = 1 0000 0000

Kết quả = 0100 1011

c) – 117 10 / 11 10

Ta có: 117 10 = 0111 0101 2

11 10  = 0000 1011 ­2

Đặt M = 0000 1011 2

|  |  |  |
| --- | --- | --- |
| A | Q | M = 0000 1011 |
| 0000 0000 | 0111 0101 | Initial value |
| 0000 0000 | 1110 1010 | Shift left |
| 1111 0101 |  | A = A - M |
| 0000 0000 | 1110 1010 | Restore (Count = 7) |
| 0000 0001 | 1101 0100 | Shift left |
| 1111 0110 |  | A = A - M |
| 0000 0001 | 1101 0100 | Restore (Count = 6) |
| 0000 0011 | 1010 1000 | Shift left |
| 1111 1000 |  | A = A – M |
| 0000 0011 | 1010 1000 | Restore (Count = 5) |
| 0000 0111 | 0101 0000 | Shift left |
| 1111 1100 |  | A = A – M |
| 0000 0111 | 0101 0000 | Restore (Count = 4) |
| 0000 1110 | 1010 0000 | Shift left |
| 0000 0011 |  | A = A – M |
| 0000 0011 | 1010 0001 | Set Q0 = 1 (Count = 3) |
| 0000 0111 | 0100 0010 | Shift left |
| 1111 1100 |  | A = A – M |
| 0000 0111 | 0100 0010 | Restore (Count = 2) |
| 0000 1110 | 1000 0100 | Shift left |
| 0000 0011 |  | A = A – M |
| 0000 0011 | 1000 0101 | Set Q0 = 1 (Count = 1) |
| 0000 0111 | 0000 1010 | Shift left |
| 1111 1100 |  | A = A – M |
| 0000 0111 | 0000 1010 | Restore (Count = 0) |

Vậy – 117 10 / 11 10  = 0111 0101 2  / 0000 1011 2 được 0000 1010 2 dư 0000 0111 2

**Câu 4:**

**a) -111.612510**

X = - 111.612510 = - 1101111. 1001110012 = - 1.101111 100111001 x 26

Vì X < 0 nên S = 1

E = 6 => e = E + 127 = 6 + 127 = 13310 = 1000 0101 2

M = 1.101111 10011001 => m = 1011111001110011001

X = 1 1000 0101 101 1111 0011 1001 1001 1001 2

**b) 1010.0110**

X = 1010.0110

= 1111110010. 000100101000112

= 1.11111001000010010100011 x 29­

Vì X > 0 nên S = 0

E = 9 => e = E + 127 = 9 + 127 = 136 10 = 1000 1000 2

M = 1.11111001000010010100011

=> m = 11111001000010010100011

X = 0 1000 1000 111 1100 1000 0100 1010 0011

**c) 0100 001011010001 1101 0000 0000 0000 2**

S = 0 → số dương

e = 1000 01012 = 13310

E = 133 - 127 = 6

X = 1. 111 100 1000 0000 1010 0011 x 26 = 1111100.1000000010100011

= 124.5024871826171875 10

Vậy: 0100 0010 1101 0001 1101 0000 0000 2 = 124. = 124.502487182617187510