Week One - EET3300

Number Systems Conversions

1. Convert 413.1_5 to unsigned binary

converting from base 5 to decimal

413 in binary is:

$(413)_5 = (4 \times 52) + (1 \times 51) + (3 \times 50) = (108)_{10}$

$\frac{108}{2} = 54r0 \leftarrow LSB$

$\frac{54}{2} = 27r0$

$\frac{27}{2} = 13r1$

$\frac{13}{2} = 6r1$

$\frac{6}{2} = 3r0$

$\frac{3}{2} = 1r1$

$\frac{1}{2} = 0r1$

Therefore, $413_5 = 1101100$

0.1_5 in decimal is:

$(0.1)_5 = (1 \times 5^{(-1)}) = (0.2)_{10}$

0.2_{10} to binary is

$\frac{12}{2} = 6r0 \leftarrow LSB$

$\frac{6}{2} = 3r0$

$\frac{3}{2} = 1r1$

$\frac{1}{2} = 0r1$

therefore, $0.1_5 = 0011 recurring$

---

answer: 1101100.0011

2. Express -120_{10} as an 8 bit 2's complement number

$120 \rightarrow 64 + 32 + 16 + 8 = 01111000$

to make negative, flip all bits, then add one

$01111000 \rightarrow 10000111 + 1 \rightarrow answer$ **10001000** ___________

3. Convert 27.77 to unisgned binary number, rounging the non integer component to fit into three spaces

---

4. Convert $AF013C_{16}$ to a base 4 number

$AF013C_{16} \rightarrow (10 \times 16^5) + (15 \times 16^4) + (0 \times 16^3) + (1 \times 16^2) + (3 \times 16^1) + (12 \times 16^0) = (11469116)_1 0$

For fun and to save time, I solved the conversion from base 10 to base 4 below in python, using the division remainder method that I wrote out in full on question 1.

In [3]:
```python
#From decimal to any base via division remainder method
def change_of_base(n, b):
    if n == 0:
        return [0]
    digits = []
    while n:
        digits.append(str(n % b))
        n //= b
    return int("".join(digits[::-1]))

print(f"converting 11349116 to base 4 = {change_of_base(11349116,4)}")
```

```
converting 11349116 to base 4 = 223102301330
```

---

5. Convert 101101.11_2 to a base 7 number

First convert from base 2 to base 10

$00101101 \rightarrow 2^5 + 2^3 + 2^2 + 2^0 = 45$

$0.11 = (1 \times 2^{-1}) + (1 \times 2^{-2}) = (0.75)_{10}$

Then convert to base 7 via the division remainder method:

`print(f"converting 45 to base 7 = {change_of_base(45,7)}")` $\rightarrow 63_7$

0.075 in base 7 by multiplcation remainder method:

0.75 × 7 = **5** + 0.25 $\leftarrow LSB$ 0.25 × 7 = **1** + 0.75
0.75 × 7 = **5** + 0.25
0.25 × 7 = **1** + 0.75
0.75 × 7 = **5** + 0.25
0.25 × 7 = **1** + 0.75
0.75 × 7 = **5** + 0.25
0.25 × 7 = **1** + 0.75
0.75 × 7 = **5** + 0.25
0.25 × 7 = **1** + 0.75

Answer : 63.515

Binary Addition

1.In a 5 bit system, add 15_{10} and -4_{10}

start by converting to 4 bit, twos complement numbers

$15 \rightarrow 01111$
$-4 \rightarrow 11100$
performing binary addition\

Answer is 01011

2.In a 5 bit system, add -13_{10} and 5_{10}

start by converting to 4 bit, twos complement numbers

$-13 \rightarrow 10011$
$5 \rightarrow 00101$
performing binary addition **Answer is 11000 = -8 = -13 + 5**

3.In a 5 bit system, add -6_{10} and 8_{10}

start by converting to 4 bit, twos complement numbers

$-6 \rightarrow 11010$
$8 \rightarrow 01000$
performing binary addition **Answer is 00010 = 2 = -6 + 8**

4.In a 5 bit system, add 12_{10} and 10_{10}

start by converting to 4 bit, twos complement numbers

$12 \rightarrow 01100$
$10 \rightarrow 01010$
Answer is 10110 = -6 != 12+10 = 22

OVERFLOW, because the max number a 5 bit twos complement integer can hold is +/-2^4-1 or +/-15. 12 + 10 is 22

5.In a 5 bit system, add -14_{10} and 7_{10}

start by converting to 4 bit, twos complement numbers

$-14 \rightarrow 10010$
$7 \rightarrow 00111$
Answer is 11001 $\rightarrow$ -00111 = -7 = -14+7

Binary Subtraction

1.In a 5 bit system, subtract 10_{10} and -8_{10}

start by converting to 5 bit, twos complement numbers

$10 \rightarrow 01010$
$-8 \rightarrow 11000$
performing binary subtraction
**Answer is 10010, OVERFLOW because 18 > +/- 2^4 -1 = +/-15

2.In a 4 bit system, subtract -1_{10} and 0_{10}

start by converting to 4 bit, twos complement numbers

$-1 \rightarrow 1111$
$0 \rightarrow 0000$
performing binary subtraction **Answer is 1111= -1 = -1 - 0**

3.In a 6 bit system, subtract -7_{10} and -24_{10}

start by converting to 6 bit, twos complement numbers

$-7 \rightarrow 111001$
$-24 \rightarrow 101000$
performing binary subtraction **Answer is 010001= 17 = -7 - (-24)**

4.In a 4 bit system, subtract -5_{10} and 2_{10}

start by converting to 4 bit, twos complement numbers

$-5 \rightarrow 1011$
$2 \rightarrow 0010$
performing binary subtraction **Answer is 1001 = -7 = -5 -2

5.In a 6 bit system, subtract -7_{10} and 5_{10}

start by converting to 6 bit, twos complement numbers

$-7 \rightarrow 11001$
$5 \rightarrow 00101$
performing binary subtraction **Answer is 10100 $\rightarrow$ -01010 = -12 = -7 - 5**

Binary Multiplication

1. 5 bit system : $10_{10} \times -3_{10} \rightarrow 00010$

OVERFLOW, because -30 does not fit into +/-(2^4-1) or +/-15 2. 4 bit system : $2_{10} \times -4_{10} \rightarrow 1000$ 3. 5 bit system : $9_{10} \times -1_{10} \rightarrow 10111$ 4. 5 bit system : $-8_{10} \times 3_{10} \rightarrow 11000$ OVERFLOW, because -24 does not fit into +/-(2^4 -1) = +/-15 5. 6 bit system : $-6_{10} \times 3_{10} \rightarrow 101110$

Binary Codes

1. Express 352_10 as an 8-4-2-1 BCD Number : **001101010010**
2. ASCII q = 111 0001

In [ ]:

In [ ]:

In [ ]: