

2024 年度

情報科学実験 02

レポート課題

実験テーマ

KUE-CHIP を用いた公開鍵暗号通信の実装(1)

実験実施日: 2024 年 10 月 10 日

レポート提出日: 2024 年 10 月 15 日

提出期限: 2024 年 10 月 17 日

報告者: J2200071 齊藤 隆斗

実験グループ: A 班

要 目

この文書は筑波大学情報学群情報科学類の卒業研究論文のサンプルである。このファイルを書き換えて、このサンプルと同様の書式の論文を作成できる。

目次

第 1 章	実験内課題	1
1.1	予習問題(前期の復習)	1
1.2	課題 01	1
1.3	課題 02	2
1.4	課題 03	3
第 2 章	考察	6
参考文献	7

第 1 章 実験内課題

1.1 予習問題(前期の復習)

下の手順を参考に 2 つの整数 x, n に対し $\text{Modular(剰余)}\text{Mod}[x, n]$ を求めるアセンブリ語プログラムを作成し、命令毎の動作を説明せよ。

このプログラムでは以下のような入出力を想定する。

入力

180: (X)**H (自然数: 1 バイト)

182: (N)**H (自然数: 1 バイト)

出力

ACC: $\text{Mod}[X, N]$

与えられた問題を解くプログラムは以下のようになった。

```
00: 65    LD ACC, (80)
01: 80
02: 6D    LD IX, (82)
03: 82
04: A1    SUB ACC, IX
05: 32    BZP 02
06: 02
07: B1    ADD ACC, IX
08: 0F
```

プログラムの説明

プログラムの手順としては、 X から n を引いていき、それが負数になったら、 n を足すことによって剰余を求めるというものである。 X から n を引く際に、繰り返す回数は自然数であるから、 X は $(n \times (\text{自然数}) + \text{整数})$ という形で表せる。最後に n を足す理由は、剰余は n より小さく、かつ非負である必要があるからである。

1.2 課題 01

$\text{Mod}[ax]$ を求めるプログラムを作成し、KUECHIP で実行せよ。

このプログラムでは以下のような入出力を想定する。

入力

180: (X)**H (乗数)
182: (N)**H
1B0: (A)**H *(被乗数)

出力

ACC: Mod[AX,N]

与えられた問題を解くプログラムは以下のようになった.

```
00: C0    EOR ACC, ACC
01: C9    EOR IX,  IX
02: B5    ADD ACC, (80)
03: 80
04: A5    SUB ACC, (82)
05: 82
06: 32    BZP 04
07: 04
08: B5    ADD ACC, (82)
09: 82
0A: BA    ADD IX,  1
0B: 01
0C: FD    CMP IX,  (B0)
0D: B0
0E: 3A    BN 02
0F: 02
10: 0F    HLT
```

プログラムの説明

このプログラムでは、Mod 演算が足し算に対して不変であることを利用する.すなわち、

$$S_0 = 0, S_{i+1} = \text{Mod}[S_i + x, n]$$

としたとき

$$\text{Mod}[ax, n] = S_a$$

を利用する.

よって、期待する出力である Mod[AX,N]を得るためには、 S_i の*i*がAとなるまで

$$S_{i+1} = \text{Mod}[S_i + x, n]$$

を計算していけば良い. このプログラムでは IX で S_i の *i* をカウントし、ACC に S_i を格納して計算を行っている. ここで、予習問題にて作成した Mod[x,n]を求めるプログラムを利用し、Mod[$S_i + x, n$]を求めている.

1.3 課題 02

3つの整数 x, k, n に対し、Mod[x^k, n]を計算するプログラムを実装せよ.

このプログラムでは以下のような入出力と作業領域を想定する.

入力

180: (X)
181: (K)
182: (N)

作業領域

1B0: (W1)**H (作業領域 1: 途中経過 $A = \text{Mod}[X^p, N]$, 被乗数)

1C0: (W2)**H (作業領域 2: X の乗算を実行した回数 p)

出力

ACC: $\text{Mod}[X^K, N]$

与えられた問題を解くプログラムは以下のようになった.

```
00: 65    LD ACC, (80)
01: 80
02: 6A    LD IX, 1
03: 01
04: 7D    ST IX, (C0)
05: C0
06: 75    ST ACC, (B0)
07: B0
08: C0    EOR ACC, ACC
09: C9    EOR IX, IX
0A: B5    ADD ACC, (80)
0B: 80
0C: A5    SUB ACC, (82)
0D: 82
0E: 32    BZP 0C
0F: 0C
10: B5    ADD ACC, (82)
11: 82
12: BA    ADD IX, 1
13: 01
14: FD    CMP IX, (B0)
15: B0
16: 3A    BN 0A
17: 0A
18: 6D    LD IX, (C0)
19: C0
1A: BA    ADD IX, 1
1B: 01
1C: FD    CMP IX, (81)
1D: 81
1E: 3A    BN 04
1F: 04
20: 0F    HLT
```

プログラムの説明

1.4 課題 03

課題 02 をプログラム領域のアドレス 080H 以下に実装し、分岐命令 BA を使い、ルーチンとして 2 つの数値を連続して暗号化し、メモリに保存するプログラムを作成せよ.

このプログラムでは以下のような入出力と作業領域を想定する.

入力

170: (D) 02H 0FH (データ)
181: (K) 05H (公開鍵)
182: (N) 5BH (共通鍵)

作業領域

150: (LN)**H (作業領域: 残り繰り返し回数)
180: (X)**H (作業領域: 暗号化するデータの一時保存)

出力

190: (EN) 20H 47H (暗号化データ)

与えられた問題を解くプログラムは以下のようになった.

```
00: 6A    LD IX, 1
01: 01
02: 7D    ST IX, (50)
03: 50
04: 67    LD ACC, (IX+70)
05: 70
06: 75    ST ACC, (80)
07: 80
08: 30    BA 80
09: 80
0A: 6D    LD IX, (50)
0B: 50
0C: 77    ST ACC, (IX+90)
0D: 90
0E: AA    SUB IX, 1
0F: 01
10: 32    BZP 02
11: 02
12: 0F    HLT
```

Subroutine

```
80: 65    LD ACC, (80)
81: 80
82: 6A    LD IX, 1
83: 01
84: 7D    ST IX, (C0)
85: C0
86: 75    ST ACC, (B0)
87: B0
88: C0    EOR ACC, ACC
89: C9    EOR IX, IX
8A: B5    ADD ACC, (80)
8B: 80
8C: A5    SUB ACC, (82)
8D: 82
8E: 32    BZP 8C
8F: 8C
90: B5    ADD ACC, (82)
91: 82
92: BA    ADD IX, 1
93: 01
94: FD    CMP IX, (B0)
```

```
95: B0
96: 3A    BN 8A
97: 8A
98: 6D    LD IX, (C0)
99: C0
9A: BA    ADD IX, 1
9B: 01
9C: FD    CMP IX, (81)
9D: 81
9E: 3A    BN 84
9F: 84
A0: 30    BA 0A
A1: 0A
```


第 2 章 考察

参考文献