1 枚目

数の掛け算 1, サブルーチン (subroutine).
$a \times b$
1. a を b 回足す.
2. 筆算
3. シフト命令
4. FFT (Fast Fourier Transformation)

```
 SLL  GR1, 1
 Shift Left Logical
SLL GR1, 1
Shift Right Logical
```

2 枚目

```
  SLA (Shift Left Arithmetic)
SRA (Shift Right Arithmetic)
```

最上位 bit そのまま.
左に 1 シフト は 2 倍.
左に 2 シフト は 2^2 倍.
左に 3 シフト は 2^3 倍.

10 進数を左に 1 シフトすることは 10 倍することにほかならない.
10 進数を左に 2 シフトすることは 10^2 倍することにほかならない.

2 進数 $\sum_{k=0}^{n} a_k 2^k$ を左に 1 シフトすると

$$\sum_{k=0}^{n} a_k 2^{k+1} = 2 \sum_{k=0}^{n} a_k 2^k$$

例題. GR1 を 10 倍しなさい.
$a =$ GR1
$a \times 10 = a \times (2^3 + 2) = a \times 2^3 + a \times 2$

```
LD  GR2, GR1  ; GR2 <-- GR1
SLA GR1, 3
SLA GR2, 1   ; 板書はミスプリ (typo)
ADDA GR1,GR2   ;  GR1 <-- GR1+GR2
```

3 枚目

例題. GR1 を 13 倍せよ.
$a \times 13 = a \times (2^3 + 2^2 + 1)$.

```
LD GR2,GR1
LD GR3,GR1
SLA GR1,3
SLA GR2,2
ADDA GR1,GR2
ADDA GR1,GR3
```

本日のプログラム
CALL 命令. CALL 命令でサブルーチンを呼ぶ.
RET 命令 (return)
サブルーチンとは? asir や C 言語での関数.
GR1 を 10 倍するサブルーチンを作る.

```
TEST START
  LAD  GR1,3
  CALL TEN
( GR1 を出力

  LAD GR1,4
  CALL TEN
  GR1 を出力

  LAD GR1,5
  CALL TEN
  GR1 を出力 ;; 掛け算表が作れる. )

TEN (PUSH 0,GR2)
    LD GR2,GR1
    SLA GR1,3
    SLA GR2,1
    ADDA GR1,GR2
    (POP GR2)
    RET
    END
```

利点 1. プログラムがすっきりする. 再利用できる.
問題点. TEN の中で GR2 の値が変化してしまう.
PUSH, POP で呼び出し元の GR2 を保存.

4 枚目

図が多いので略.

5 枚目

1.

```
LD  GR2, GR1 ; GR2 <- GR1
SLA  GR1,4   ; GR1 を 2^4 倍
SLA  GR2,1   ; GR2 を 2 倍
ADDA GR1,GR2
```

$$\texttt{GR1} \leftarrow \texttt{GR1} \times 2^4 + \texttt{GR1} \times 2^1$$

$2^4 + 2^1 = 18$. 答. 18 倍.

2. 略.

6 枚目

3. 100 倍する.

$$100 = 2^6 + 2^5 + 2^2$$

```
MH PUSH 0, GR2
   PUSH 0, GR3
   LD GR2, GR1
   LD GR3, GR1
   SLA GR1,6
   SLA GR2,5
   SLA GR3,2
   ADDA GR1,GR2
   ADDA GR1,GR3
   POP GR3
   POP GR2
   RET
```

メインプログラムの例.

```
TEST START
 LAD GR1,1
 CALL MH
 RET
 END
```

4. メリット.

1. プログラムがすっきり.
2. 同じことを何度もかかなくてよい.
3. モジュール化, ブラックボックス化.

サブルーチンの働き (仕様) のみ知っていればよい. 中身は知らなくてよい. 分業へ.