MACROS FOR PALM

BRF	(BRANCH FORWARD)	THERE	ADDI 0,
BRB	(BRANCH BACKWARD)	BACK	SUBI 0,

(max 128 ADDRESSES)

BRNCH THERE
+ LDHI R0, R0
+ DC AL2 (THERE - $$$)

LCON (LOAD CONSTANT) R1, 25

+ LDHI R1, R0, 2
+ DC H '25'

HADD R2, R1

	R1	R2
	1 2 3 4	5 6 7 8
ADD R2, R1	1 2 3 4	5 6 A C
HTL R1, R2	1 2 5 6	5 6 A C
ADDS 1 R1, R1	0 0 6 8	5 6 A C
LTH R2, R1		6 8 A C

for address conversion (R1 will be destroyed)

it only doesn't work if FF are in R1 = FF..

RAM { I CYCLE
      E "

ROS { ICYCLE ROS
      E " RAM

LAD R1, THERE

+ LDHI R1, R0, 2

+ DC AL2 (THERE)

LAD R1, THERE

+ EMIT R1, X '10'
- LTH R1, R1
+ EMIT R2, '00'

<u>ADD</u>   R1, R2

R1   1 2 3 4
R2   A B C D
R1   ~~13~~ 0 1

<u>ADDS 1</u>   R1, R2

R1   R2
1 2 3 4   A B C D
    └──── + ────┘
R1   00 DF

HTL   LTH

HTL   R1   R2
                                    } FOR ADDING 16 BIT's
LTH   R1   R2

LDHI R1, R2, 2   R1   R2
LDHI | D | R1 | R2 | 1 2 3 4   5 6 7 8
                                   A B C D

                                   5 6 7 8 | A B C D |

        FOR I/O OPERATIONS HERE THROUGH THE MEMORY

SETI $\xrightarrow{\text{OPPOSIT}}$ CLRI

JUMP (ONLY OVER 1 INSTRUCTION)

```
        EMIT    R1, X'7A'  ———          R1  =  AB1B

    ┌── JEQ     R1, R2                   R1  =  AB7A
    │        ↓
    │   EMIT    R1, X'3A'
    └─→ LDHI    R3, ———
```
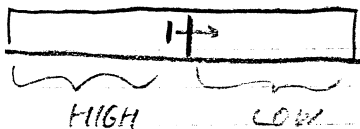

```
        EMIT     R2, 12
                                    JUMP    R1        R2
        JHAM     R1, R2          →  FF ,,       ,, FF
```

JUMP ADDRESS NORMALLY IN R1

FOR EXAMPLE

```
    R1 = 1002  ———→   1000   EMIT R1, 12
    R2 = 1004         1002   JHAM R1, R2
                      1004
```

SHIFTR

```
┌──────────┬──────────┐
│      1 ┼→ │          │
└──────────┴──────────┘
  ⌣⌣⌣⌣⌣      ⌣⌣⌣⌣⌣
  HIGH         LOW
```

HIGH ORDER BYTE NOT AFFECTED
LOW ORDER BIT OF HIGH ORDER BYTE IS SHIFTED TO THE HIGH ORDER BIT
OF LOW ORDER BYTE