

[illegible]

0) $F = 1$; операция с регистрами $a, b, c, n = c$;

[F	a	b	op		c]
[000	0000	0000	0000	0000 0000	0000]
F0	27	81	81	81	6 561	81	

1) $F = -1$; операция с регистрами a, b и константой $im, n = im$

[F	a	b	op	im]
[000	0000	0000	0000	0000 0000 0000]
F1	27	81	81	81	531 441	

Набор операций с регистрами:

```

ор      код
0      mov a, n R.a := n
1      lsl a, b, n R.a := R.b  $\leftarrow$  n (сдвиг влево на n тритов)
2      ror a, b, n R.a := R.b rot n (вращение вправо на n тритов)
логические операции
3      and a, b, n R.a := R.b & n
4      ann a, b, n R.a := R.b & ~n
5      ior a, b, n R.a := R.b or n

```

6 **xor** a, b, n R.a := R.b xor n
 целочисленная арифметика
 7 **add** a, b, n R.a := R.b + n
 -7 **sub** a, b, n R.a := R.b - n
 8 **mul** a, b, n R.a := R.a x n
 9 **div** a, b, n R.a := R.b div n
 -9 **mod** a, b, n R.a := R.b mod n

2) операция с памятью (адресация 43(27+16) трит, 328 256 967 394 537 000 000 трайт)

[F	a	b	offset]
[000	0000	0000	0000 0000 0000 0000]
F2	27	81	81	43 046 721	

Набор операций с памятью:

F описание
 3 **ldw** R.a := Mem[R.b + off] загрузить слово в регистр a, сдвиг в трайтах
 -3 **stw** Mem[R.b + off] := R.a сохранить слово из регистра a, сдвиг в трайтах
 5 **ldt** загрузить трайт из памяти в регистр a
 -5 **stt** сохранить трайт в память из регистра a

3) операция ветвления/перехода с регистром сдвига "c" (сдвиг в словах, "c" DIV 3)

[F	cond	data	c]
[000	000000	00 0000 0000 0000	0000]
F3	27	729	4 782 969	81	

F описание
 7 **brr** ветвление с переходом по значению регистра и передачей произвольных данных

4) операция ветвления/перехода с абсолютным сдвигом (сдвиг в словах)

[F	cond	offset]
[000	000000	00 0000 0000 0000 0000]
F4	27	729	387 420 489	

F описание
 -7 **brc** ветвление с переходом по константе

Условия перехода [000000]

старший трит (6)

+ оставлять адрес возврата в старшем регистре
 - не оставлять адрес возврата
 0 ошибка

трит 5

+ безусловный переход
 - никогда не переходить
 0 дальнейшие условия

трит 4

сравнить с тритом-регистром NZ и продолжить если равны

трит 3

0 проверяется только строгое соответствие 4 и NZ
 иначе если трит 4 не равен NZ, а NZ = 0 то продолжить если равны трит 3 и трит 4

[43] мнемонически:

[+0] >
 [00] =
 [-0] <
 [++] >=

[0-] #
[--] <=