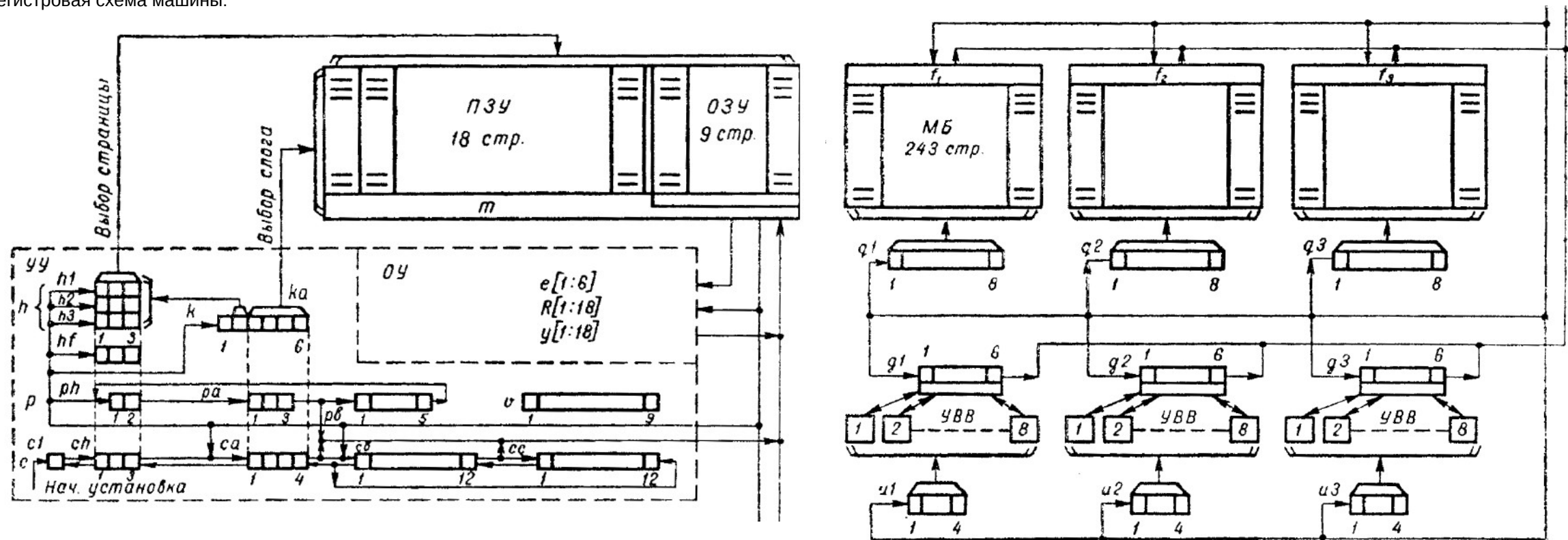


Описание оригинальной системы команд «Сетунь-70» (до внесения модификаций, датируемых 1975 г.)
(с) 2023, Станислав Масловский <stanislav.maslovski@gmail.com>

Регистровая схема машины:



Машинный код «Сетунь-70» представляет собой последовательность 6-tritных трайтов: $k = k_1, k_2, k_3, k_4, k_5, k_6$
Если $k_1, k_2 = 0, 0$, то такой трайт кодирует одну из основных ($k_3 = 0$) или специальных ($k_3 = 1$) операций из нижеследующей таблицы.
В этой таблице, T – вершина стека, S – подвершина стека. Обмен со стеком производится словами по три трайта. Переменная t обозначает первый (старший) трайт T .
Стрелка вниз \downarrow обозначает перемещение указателя стека вниз на одно слово, т. е. после выполнения операции прежняя подвершина стека (S) становится его новой вершиной (T).

Тип	Мнемоника	3-код	9-код	Описание	Комментарии
B1	LST	000---	0ZW	Сдвиг пары $S:Y$ на t разрядов; \downarrow	Сдвиг влево при $t > 0$ и вправо при $t < 0$
B2	COT	000--0	0ZX	Переход по смещению в t , если $ S > 3^{17}/2$ (переполнение); \downarrow	$ca := t$, если переполнение
B3	XNN	000--+	0ZY	Нормализация числа с мантиссой в $T:Y$ и порядком в e	
B4	E-1	000-0-	0ZZ	$e := e - 1$	
B5	E=0	000-00	0Z0	$e := 0$	
B6	E+1	000-0+	0Z1	$e := e + 1$	
B7	T-E	000-+-	0Z2	$T := T - e \times 3^{12}$	
B8	E=T	000-+0	0Z3	$e := t; \downarrow$	
B9	T+E	000-++	0Z4	$T := T + e \times 3^{12}$	
B10	CLT	0000--	00W	Если $S < 0$, то переход по смещению в $t; \downarrow$	$ca := t$, если $S < 0$
B11	CET	0000-0	00X	Если $S = 0$, то переход по смещению в $t; \downarrow$	$ca := t$, если $S = 0$
B12	CGT	0000-+	00Y	Если $S > 0$, то переход по смещению в $t; \downarrow$	$ca := t$, если $S > 0$
B13	T=C	00000-	00Z	Очищает T и помещает в t адрес текущей команды	
B14	R=T	000000	000	$R := T; \downarrow$	
B15	C=T	00000+	001	Безусловный переход в пределах страницы по смещению в $t; \downarrow$	$ca := t$
B16	T=W	0000+-	002	Копирует трайты в вершину стека из ячеек ОЗУ/ПЗУ по ссылке в t	При выходе за пределы страницы генерируется исключение ($w = -29$)
B17	YFT	0000+0	003	Очищает $Y; \downarrow$	
B18	W=S	0000++	004	Копирует трайты из подвершины стека в ячейки ОЗУ по ссылке в $t; \downarrow$	При выходе за пределы страницы генерируется исключение ($w = -29$)
B19	SMT	000+--	01W	Помещает в S результат поразрядного умножения T на $S; \downarrow$	
B20	Y=T	000+-0	01X	$Y := T; \downarrow$	

Instructions

B21	SAT	000+-+	01Y	Помещает в S результат поразрядного порогового сложения T и S; ↓	
B22	S-T	000+0-	01Z	$S := S - T$; ↓	
B23	TDN	000+00	010	$T := -T$	
B24	S+T	000+0+	011	$S := S + T$; ↓	
B25	LBT	000++-	012	$S:Y := S \times 3^{18} + T \times R \times 3^2$; ↓	
B26	L*T	000++0	013	$R := S, S:Y := T \times R \times 3^2$; ↓	
B27	LHT	000+++	014	$S:Y := S \times 3^{18} + Y + T \times R \times 3^2$; ↓	
S1	CG1	00+---	02W	Чтение из первого канала в вершину стека: $T := 0, t := \pm g[1,1:6]$	Плюс, когда $g[-1,7] = 1$, и минус, когда $g[-1,7] = 0$.
S2	CG2	00+--0	02X	Чтение из второго канала в вершину стека: $T := 0, t := \pm g[2,1:6]$	Плюс, когда $g[0,7] = 1$, и минус, когда $g[0,7] = 0$.
S3	CG3	00+--+	02Y	Чтение из третьего канала в вершину стека: $T := 0, t := \pm g[3,1:6]$	Плюс, когда $g[+1,7] = 1$, и минус, когда $g[+1,7] = 0$.
S4	CF1	00+-0-	02Z	Копирует внешнюю страницу $f[-1,q[-1]]$ в 03У по указателю из t; ↓	
S5	CF2	00+-00	020	Копирует внешнюю страницу $f[0,q[0]]$ в 03У по указателю из t; ↓	
S6	CF3	00+-0+	021	Копирует внешнюю страницу $f[+1,q[+1]]$ в 03У по указателю из t; ↓	
S7	LQ1	00+--+	022	$q[-1] := T[5:12], \text{ SUI}T(-1)$	Активизирует асинхронную процедуру поиска страницы на МБ1
S8	LQ2	00+-+0	023	$q[0] := T[5:12], \text{ SUI}T(0)$	Активизирует асинхронную процедуру поиска страницы на МБ2
S9	LQ3	00+-++	024	$q[+1] := T[5:12], \text{ SUI}T(+1)$	Активизирует асинхронную процедуру поиска страницы на МБ3
S10	CP	00+0--	03W	Копирует указатель стека $ph:pa$ в вершину стека	$z := 0; z[5:6] := ph; z[9:11] := pa; T := z$;
S11	EXP	00+0-0	03X	Обменивает текущий указатель стека с резервным	$z[1:5] := p[1:5]; p[1:5] := p[6:10]; p[6:10] := z[1:5]$;
S12	LP	00+0-+	03Y	Загружает значение из вершины стека в указатель стека	$z := T; ph := z[5:6]; pa := z[9:11]$;
S13	CMC	00+00-	03Z	Копирует cb в вершину стека, обменивает содержимое cb и cc	
S14	RMC	00+000	030	Возврат из макрокоманды	$z[1:12] := c[9:20]; c[9:20] := c[21:32]; c[21:32] := z[1:12]; c[1:4] := z[3:6]; c[5:8] := z[9:12]$;
S15	LMC	00+00+	031	Загружает в cb значение из вершины стека, старое $cb \rightarrow cc$; ↓	$z := T; c[21:32] := c[9:20]; c[9:20] := z[1:12]$;
S16	LH1	00+0+-	032	Загружает указатель страниц $h[-1]$ из вершины стека; ↓	$z := T; h[-1] := z[4:6]$;
S17	LH2	00+0+0	033	Загружает указатель страниц $h[0]$ из вершины стека; ↓	$z := T; h[0] := z[4:6]$;
S18	LH3	00+0++	034	Загружает указатель страниц $h[+1]$ из вершины стека; ↓	$z := T; h[+1] := z[4:6]$;
S19	LU1	00+++--	04W	Загружает управляющий регистр первого канала VB ; ↓	$u[-1, 1:4] := t; a[-1] := 1$;
S20	LU2	00+++0	04X	Загружает управляющий регистр второго канала VB ; ↓	$u[1, 1:4] := t; a[1] := 1$;
S21	LU3	00++++	04Y	Загружает управляющий регистр третьего канала VB ; ↓	$u[+1, 1:4] := t; a[+1] := 1$;
S22	LF1	00+++0-	04Z	Выгружает страницу с номером из t во внешнюю память $f[-1,q[-1]]$	
S23	LF2	00+++00	040	Выгружает страницу с номером из t во внешнюю память $f[0,q[0]]$	
S24	LF3	00+++0+	041	Выгружает страницу с номером из t во внешнюю память $f[+1,q[+1]]$	
S25	LG1	00++++-	042	Запись в первый канал из вершины стека; ↓	$\text{TRANSOUT}(-1)$;
S26	LG2	00++++0	043	Запись во второй канал из вершины стека; ↓	$\text{TRANSOUT}(0)$;
S27	LG3	00+++++	044	Запись в третий канал из вершины стека; ↓	$\text{TRANSOUT}(+1)$;

Если $k1, k2, k3 = 0, 0, -1$ то такой трайт кодирует макрооперацию (системный вызов).

Если хотя бы один из тритов $k1, k2$ отличен от нуля, то такой трайт интерпретируется как ссылка на операнд, подлежащий загрузке в вершину стека. Формат ссылки:

$k1$ – задает длину операнда в трайтах, равную $k1+2$

$k2$ – выбирает регистр приписки, используемый для адресации страницы: $h[k2]$

$k[3:6]$ – адрес первого трайта операнда (в пределах страницы)

Поскольку недопустим слог-адрес у которого $k[1:2]=0$, то при использовании регистра приписки $h[0]$ (т.е. $h2$) длина операнда не может быть равной двум слогам.

Операнды в памяти хранятся в порядке от старших разрядов к младшим (big-endian).

В зависимости от значения в регистре $c1$, машина может находиться в трех режимах:

$c1 = -1$: режим пользователя. В этом режиме разрешены команды основного типа (B1-B27) и макрооперации. Прерывания разрешены.

$c1 = 0$: режим макрооперации. В этот режим машина переходит при встрече кода макрооперации. В этом режиме разрешены основные и специальные (S1-S27) операции.

Новые макрооперации в этом режиме запрещены. Прерывания разрешены.

$c1 = +1$: системый режим. В этом режиме доступны основные и специальные операции. Машина исходно стартует в этом режиме. Прерывания и макрооперации запрещены.

Исполнение макрооперации заключается в сохранении текущего значения cb в cc , а значения c в cb , помещении на стек слога $c\ t = m[-13, ka]$, где $ka = k[3:6]$ из кода макрооперации, и перехода к инструкции из $m[-13, -13]$.

Возврат из макрооперации происходит по команде RMC, которая восстанавливает ранее сохраненное значение c из cb .