

1.10 "1" C

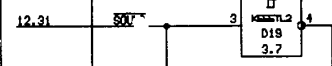


● 2013年12月

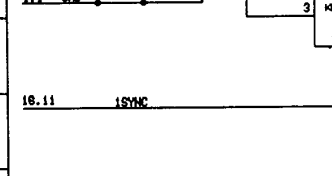
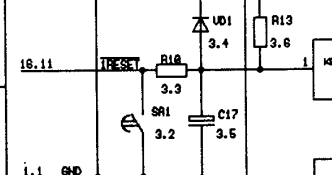
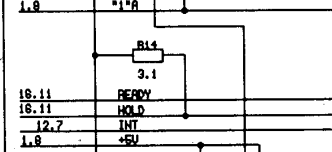
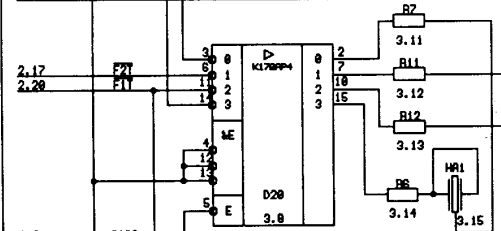
						Et3.059.001 \ 3 B	List
Izmlist # Dokum.	Podp.	Data					2
OFF NIITG MBU	20.01.89	kopiroval				format R2	

Et3.059.001 \3

16.11	D0	1
16.11	D1	2
16.11	D2	3
16.11	D3	4
16.11	D4	5
16.11	D5	6
16.11	D6	7
16.11	D7	8

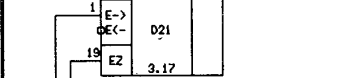


5.8 CSDEV



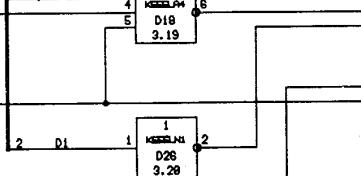
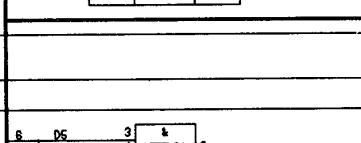
2.21 CS8  
1.18 \*1°C  
16.11 DBIN

8	D7	2
7	D6	3
6	D5	4
5	D4	5
4	D3	6
3	D2	7
2	D1	8
1	D0	9



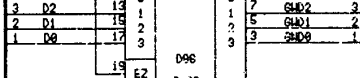
1 E->  
OE(-) D21  
EZ 3.17

20	CPU	2R
(-)	MEMO-PROM	
1	D0	10
2	D1	9
3	D2	8
4	D3	7
5	D4	6
6	D5	5
7	D6	4
8	D7	3
9	D8	2
10	D9	1
11	D10	11
12	D11	12
13	D12	13
14	D13	14
15	D14	15
16	D15	16
17	D16	17
18	D17	18
19	D18	19
20	D19	20
21	D20	21
22	D21	22
23	D22	23
24	D23	24
25	D24	25
26	D25	26
27	D26	27
28	D27	28
29	D28	29
30	D29	30
31	D30	31
32	D31	32
33	D32	33
34	D33	34
35	D34	35
36	D35	36
37	D36	37
38	D37	38
39	D38	39
40	D39	40
41	D40	41
42	D41	42
43	D42	43
44	D43	44
45	D44	45
46	D45	46
47	D46	47
48	D47	48
49	D48	49
50	D49	50
51	D50	51
52	D51	52
53	D52	53
54	D53	54
55	D54	55
56	D55	56
57	D56	57
58	D57	58
59	D58	59
60	D59	60
61	D60	61
62	D61	62
63	D62	63
64	D63	64
65	D64	65
66	D65	66
67	D66	67
68	D67	68
69	D68	69
70	D69	70
71	D70	71
72	D71	72
73	D72	73
74	D73	74
75	D74	75
76	D75	76
77	D76	77
78	D77	78
79	D78	79
80	D79	80
81	D80	81
82	D81	82
83	D82	83
84	D83	84
85	D84	85
86	D85	86
87	D86	87
88	D87	88
89	D88	89
90	D89	90
91	D90	91
92	D91	92
93	D92	93
94	D93	94
95	D94	95
96	D95	96
97	D96	97
98	D97	98
99	D98	99
100	D99	100



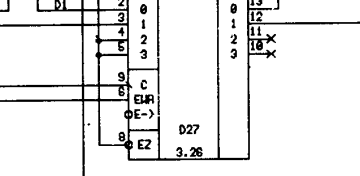
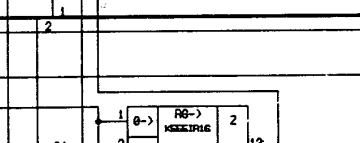
2 D1 1  
1 D2 2  
2 D3 3  
3 D4 4  
4 D5 5  
5 D6 6  
6 D7 7  
7 D8 8  
8 D9 9  
9 D10 10  
10 D11 11  
11 D12 12  
12 D13 13  
13 D14 14  
14 D15 15  
15 D16 16  
16 D17 17  
17 D18 18  
18 D19 19  
19 D20 20  
20 D21 21  
21 D22 22  
22 D23 23  
23 D24 24  
24 D25 25  
25 D26 26  
26 D27 27  
27 D28 28  
28 D29 29  
29 D30 30  
30 D31 31  
31 D32 32  
32 D33 33  
33 D34 34  
34 D35 35  
35 D36 36  
36 D37 37  
37 D38 38  
38 D39 39  
39 D40 40  
40 D41 41  
41 D42 42  
42 D43 43  
43 D44 44  
44 D45 45  
45 D46 46  
46 D47 47  
47 D48 48  
48 D49 49  
49 D50 50  
50 D51 51  
51 D52 52  
52 D53 53  
53 D54 54  
54 D55 55  
55 D56 56  
56 D57 57  
57 D58 58  
58 D59 59  
59 D60 60  
60 D61 61  
61 D62 62  
62 D63 63  
63 D64 64  
64 D65 65  
65 D66 66  
66 D67 67  
67 D68 68  
68 D69 69  
69 D70 70  
70 D71 71  
71 D72 72  
72 D73 73  
73 D74 74  
74 D75 75  
75 D76 76  
76 D77 77  
77 D78 78  
78 D79 79  
79 D80 80  
80 D81 81  
81 D82 82  
82 D83 83  
83 D84 84  
84 D85 85  
85 D86 86  
86 D87 87  
87 D88 88  
88 D89 89  
89 D90 90  
90 D91 91  
91 D92 92  
92 D93 93  
93 D94 94  
94 D95 95  
95 D96 96  
96 D97 97  
97 D98 98  
98 D99 99  
99 D100 100

8	D7	2
7	D6	3
6	D5	4
5	D4	5
4	D3	6
3	D2	7
2	D1	8
1	D0	9



1 E->  
OE(-) D21  
EZ 3.17

25	A0	1
26	A1	2
27	A2	3
28	A3	4
29	A4	5
30	A5	6
31	A6	7
32	A7	8
33	A8	9
34	A9	10
35	A10	11
36	A11	12
37	A12	13
38	A13	14
39	A14	15
40	A15	16
41	A16	17
42	A17	18
43	A18	19
44	A19	20
45	A20	21
46	A21	22
47	A22	23
48	A23	24
49	A24	25
50	A25	26
51	A26	27
52	A27	28
53	A28	29
54	A29	30
55	A30	31
56	A31	32
57	A32	33
58	A33	34
59	A34	35
60	A35	36
61	A36	37
62	A37	38
63	A38	39
64	A39	40
65	A40	41
66	A41	42
67	A42	43
68	A43	44
69	A44	45
70	A45	46
71	A46	47
72	A47	48
73	A48	49
74	A49	50
75	A50	51
76	A51	52
77	A52	53
78	A53	54
79	A54	55
80	A55	56
81	A56	57
82	A57	58
83	A58	59
84	A59	60
85	A60	61
86	A61	62
87	A62	63
88	A63	64
89	A64	65
90	A65	66
91	A66	67
92	A67	68
93	A68	69
94	A69	70
95	A70	71
96	A71	72
97	A72	73
98	A73	74
99	A74	75
100	A75	76



2 D1 1  
1 D2 2  
2 D3 3  
3 D4 4  
4 D5 5  
5 D6 6  
6 D7 7  
7 D8 8  
8 D9 9  
9 D10 10  
10 D11 11  
11 D12 12  
12 D13 13  
13 D14 14  
14 D15 15  
15 D16 16  
16 D17 17  
17 D18 18  
18 D19 19  
19 D20 20  
20 D21 21  
21 D22 22  
22 D23 23  
23 D24 24  
24 D25 25  
25 D26 26  
26 D27 27  
27 D28 28  
28 D29 29  
29 D30 30  
30 D31 31  
31 D32 32  
32 D33 33  
33 D34 34  
34 D35 35  
35 D36 36  
36 D37 37  
37 D38 38  
38 D39 39  
39 D40 40  
40 D41 41  
41 D42 42  
42 D43 43  
43 D44 44  
44 D45 45  
45 D46 46  
46 D47 47  
47 D48 48  
48 D49 49  
49 D50 50  
50 D51 51  
51 D52 52  
52 D53 53  
53 D54 54  
54 D55 55  
55 D56 56  
56 D57 57  
57 D58 58  
58 D59 59  
59 D60 60  
60 D61 61  
61 D62 62  
62 D63 63  
63 D64 64  
64 D65 65  
65 D66 66  
66 D67 67  
67 D68 68  
68 D69 69  
69 D70 70  
70 D71 71  
71 D72 72  
72 D73 73  
73 D74 74  
74 D75 75  
75 D76 76  
76 D77 77  
77 D78 78  
78 D79 79  
79 D80 80  
80 D81 81  
81 D82 82  
82 D83 83  
83 D84 84  
84 D85 85  
85 D86 86  
86 D87 87  
87 D88 88  
88 D89 89  
89 D90 90  
90 D91 91  
91 D92 92  
92 D93 93  
93 D94 94  
94 D95 95  
95 D96 96  
96 D97 97  
97 D98 98  
98 D99 99  
99 D100 100

1	A0	16.11
2	A1	16.11
3	A2	16.11
4	A3	16.11
5	A4	16.11
6	A5	16.11
7	A6	16.11
8	A7	16.11
9	A8	16.11
10	A9	16.11
11	A10	16.11
12	A11	16.11
13	A12	16.11
14	A13	16.11
15	A14	16.11
16	A15	16.11

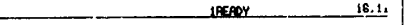
16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



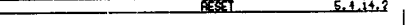
16.11 A0-A15 5.6.8.12.13.14



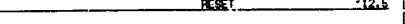
16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



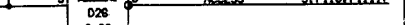
16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14



16.11 A0-A15 5.6.8.12.13.14

Et3.059.001 \3

5.1	MR8
5.1	MR1
5.8	MR2
5.8	MR3
5.9	MR4
5.9	MR5
5.13	MR6
5.13	MR7
2.22	MR8
2.21	CR8
5.12	IE
3.18	D8
3.18	D1
3.18	D2
3.18	D3
3.18	D4
3.18	D5
3.18	D6
3.18	D7

D8	2	D	RAM	
MR3	12	A	0	
MR4	11	1		
MR5	10	2		
MR6	13	3		
MR1	7	4		
MR2	6	5		
MR8	5	6		
MR7	9	7		
MR8	4	CR8		
CR8	15	OCRS	D83	
IE	3	EUR	4.1	

2 14 MD8

D1	2	D	RAM	
MR3	12	A	0	
MR4	11	1		
MR5	10	2		
MR6	13	3		
MR1	7	4		
MR2	6	5		
MR8	5	6		
MR7	9	7		
MR8	4	CR8		
CR8	15	OCRS	D82	
IE	3	EUR	4.2	

2 14 MD1

D2	2	D	RAM	
MR3	12	A	0	
MR4	11	1		
MR5	10	2		
MR6	13	3		
MR1	7	4		
MR2	6	5		
MR8	5	6		
MR7	9	7		
MR8	4	CR8		
CR8	15	OCRS	D81	
IE	3	EUR	4.3	

2 14 MD2

D3	2	D	RAM	
MR3	12	A	0	
MR4	11	1		
MR5	10	2		
MR6	13	3		
MR1	7	4		
MR2	6	5		
MR8	5	6		
MR7	9	7		
MR8	4	CR8		
CR8	15	OCRS	D80	
IE	3	EUR	4.4	

2 14 MD3

D4	2	D	RAM	
MR3	12	A	0	
MR4	11	1		
MR5	10	2		
MR6	13	3		
MR1	7	4		
MR2	6	5		
MR8	5	6		
MR7	9	7		
MR8	4	CR8		
CR8	15	OCRS	D79	
IE	3	EUR	4.5	

2 14 MD4

D5	2	D	RAM	
MR3	12	A	0	
MR4	11	1		
MR5	10	2		
MR6	13	3		
MR1	7	4		
MR2	6	5		
MR8	5	6		
MR7	9	7		
MR8	4	CR8		
CR8	15	OCRS	D78	
IE	3	EUR	4.6	

2 14 MD5

D6	2	D	RAM	
MR3	12	A	0	
MR4	11	1		
MR5	10	2		
MR6	13	3		
MR1	7	4		
MR2	6	5		
MR8	5	6		
MR7	9	7		
MR8	4	CR8		
CR8	15	OCRS	D77	
IE	3	EUR	4.7	

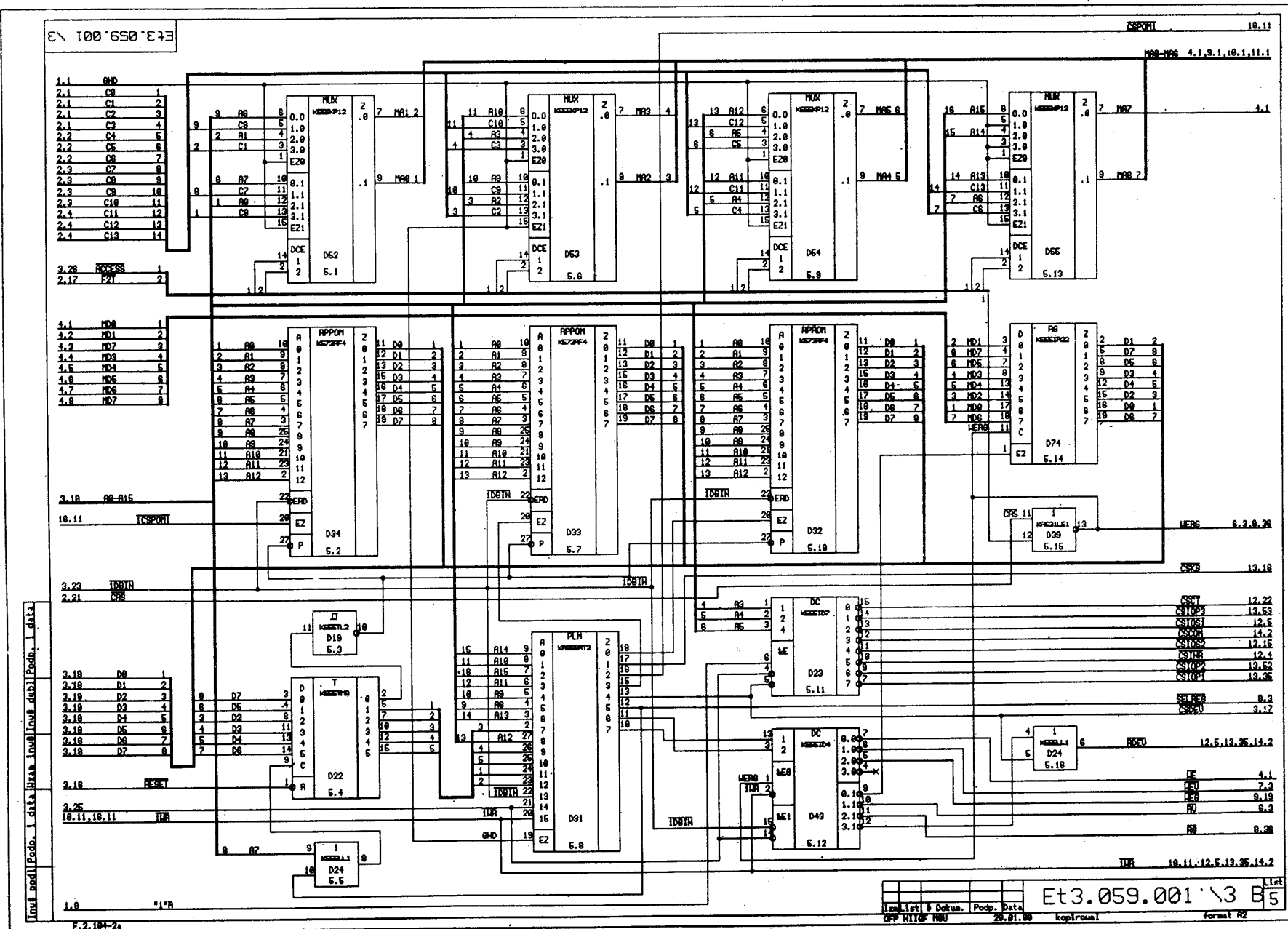
2 14 MD6

D7	2	D	RAM	
MR3	12	A	0	
MR4	11	1		
MR5	10	2		
MR6	13	3		
MR1	7	4		
MR2	6	5		
MR8	5	6		
MR7	9	7		
MR8	4	CR8		
CR8	15	OCRS	D76	
IE	3	EUR	4.8	

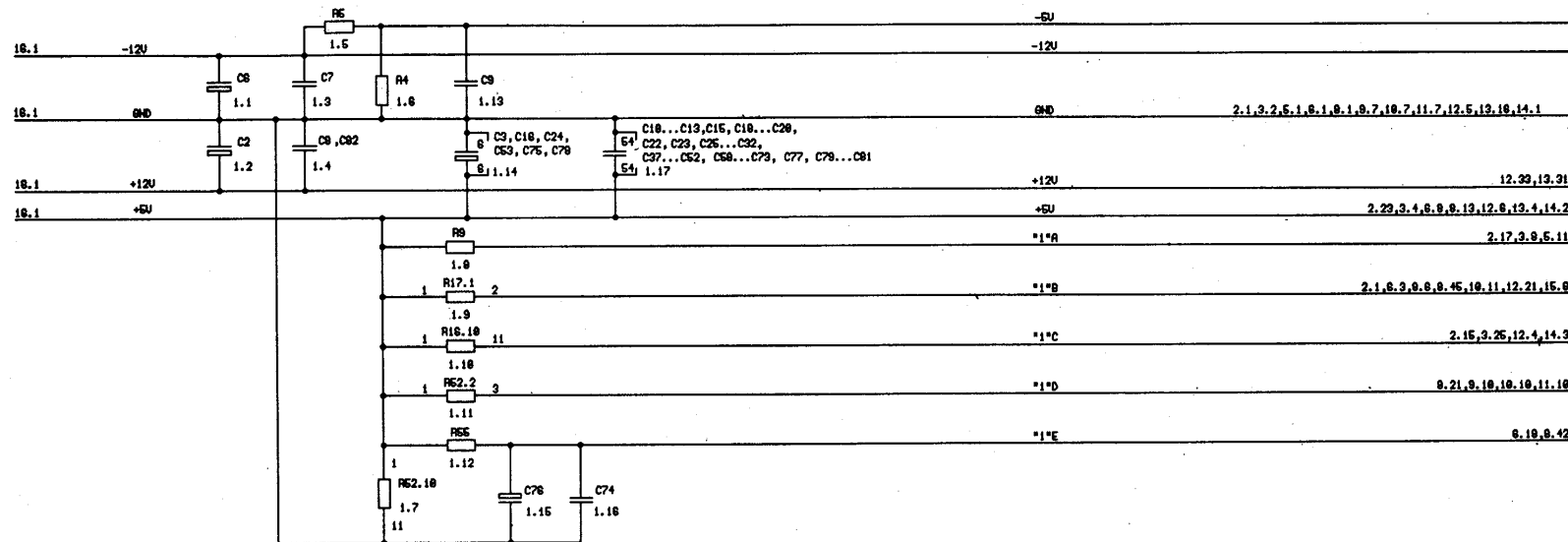
2 14 MD7

MD8	5.14
MD1	5.14
MD2	5.14
MD3	5.14
MD4	5.14
MD5	5.14
MD6	5.14
MD7	5.14

load modl Podb. data base modl modl Podb. data



ET3.059.001 \3

[illegible]

Et3.059.001 \3 B

Plata

### Shema :lektrikeskaq principlalxnaq

Lit.	Massa Masctab
------	---------------

List 1	Listou 15
--------	-----------

—

1st	Dokum.	Podp.	Data
Razrab	Skurikhin		
Prover	Roi		
Tkontr			
Nar.	KB	Savrillov	
Nkontr		Mikhailova	
Utu.		Rahimov	
OFF NITC HBU			26.01.

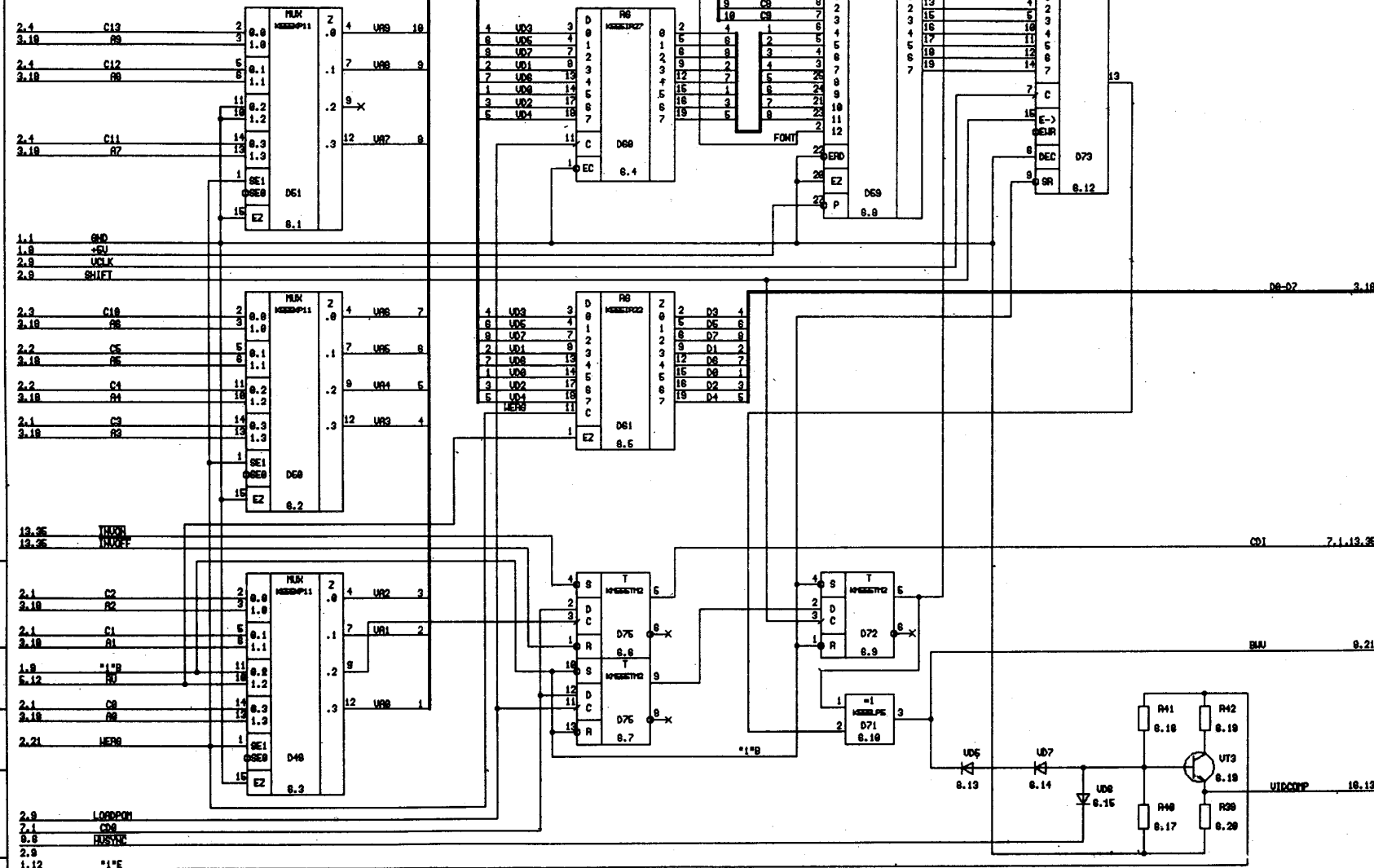
20.01.99 koplrowal

Format A2

Et3.059.001 \3

U08-U09 7.1

2.2.2.3 CS-C9  
13.38 FONT  
2.1-2.9 U08-U07



2.9 L08P0H  
2.1 C09  
9.8 BUSYH  
2.9  
1.12 \*1"E

Et3.059.001 \3

CD0 6.2

8.6	CD1	15	D	RAM	15120PLM	2	7
8.3	UR0	1	A	0			
8.3	UR1	2	A	1			
8.3	UR2	3	A	2			
8.2	UR3	4	A	3			
8.2	UR4	5	A	4			
8.2	UR5	6	A	5			
8.2	UR6	7	A	6			
8.1	UR7	8	A	7			
8.1	UR8	9	A	8			
8.1	UR9	10	A	9			
3.18	D0	11	ERD	OEUR	D62	7.1	
3.18	D1	12	EZ				
3.18	D2	13					
3.18	D3	14					
3.18	D4	15					
3.18	D5	16					
3.18	D6	17					
3.18	D7	18					

15	D	RAM	15120PLM	2	7
1	A	0			
2	A	1			
3	A	2			
4	A	3			
5	A	4			
6	A	5			
7	A	6			
8	A	7			
9	A	8			
10	A	9			
11	ERD	OEUR	D62	7.1	
12	EZ				
13					
14					
15					

13	D2	15	D	RAM	15120PLM	2	7
1	A	0					
2	A	1					
3	A	2					
4	A	3					
5	A	4					
6	A	5					
7	A	6					
8	A	7					
9	A	8					
10	A	9					
11	ERD	OEUR	D68	7.4			
12	EZ						
13							
14							
15							

16	D5	15	D	RAM	15120PLM	2	7
1	A	0					
2	A	1					
3	A	2					
4	A	3					
5	A	4					
6	A	5					
7	A	6					
8	A	7					
9	A	8					
10	A	9					
11	ERD	OEUR	D65	7.7			
12	EZ						
13							
14							
15							

11	D8	15	D	RAM	15120PLM	2	7
1	A	0					
2	A	1					
3	A	2					
4	A	3					
5	A	4					
6	A	5					
7	A	6					
8	A	7					
9	A	8					
10	A	9					
11	ERD	OEUR	D78	7.2			
12	EZ						
13							
14							
15							

14	D3	15	D	RAM	15120PLM	2	7
1	A	0					
2	A	1					
3	A	2					
4	A	3					
5	A	4					
6	A	5					
7	A	6					
8	A	7					
9	A	8					
10	A	9					
11	ERD	OEUR	D67	7.5			
12	EZ						
13							
14							
15							

17	D6	15	D	RAM	15120PLM	2	7
1	A	0					
2	A	1					
3	A	2					
4	A	3					
5	A	4					
6	A	5					
7	A	6					
8	A	7					
9	A	8					
10	A	9					
11	ERD	OEUR	D64	7.8			
12	EZ						
13							
14							
15							

12	D1	15	D	RAM	15120PLM	2	7
1	A	0					
2	A	1					
3	A	2					
4	A	3					
5	A	4					
6	A	5					
7	A	6					
8	A	7					
9	A	8					
10	A	9					
11	ERD	OEUR	D69	7.3			
12	EZ						
13							
14							
15							

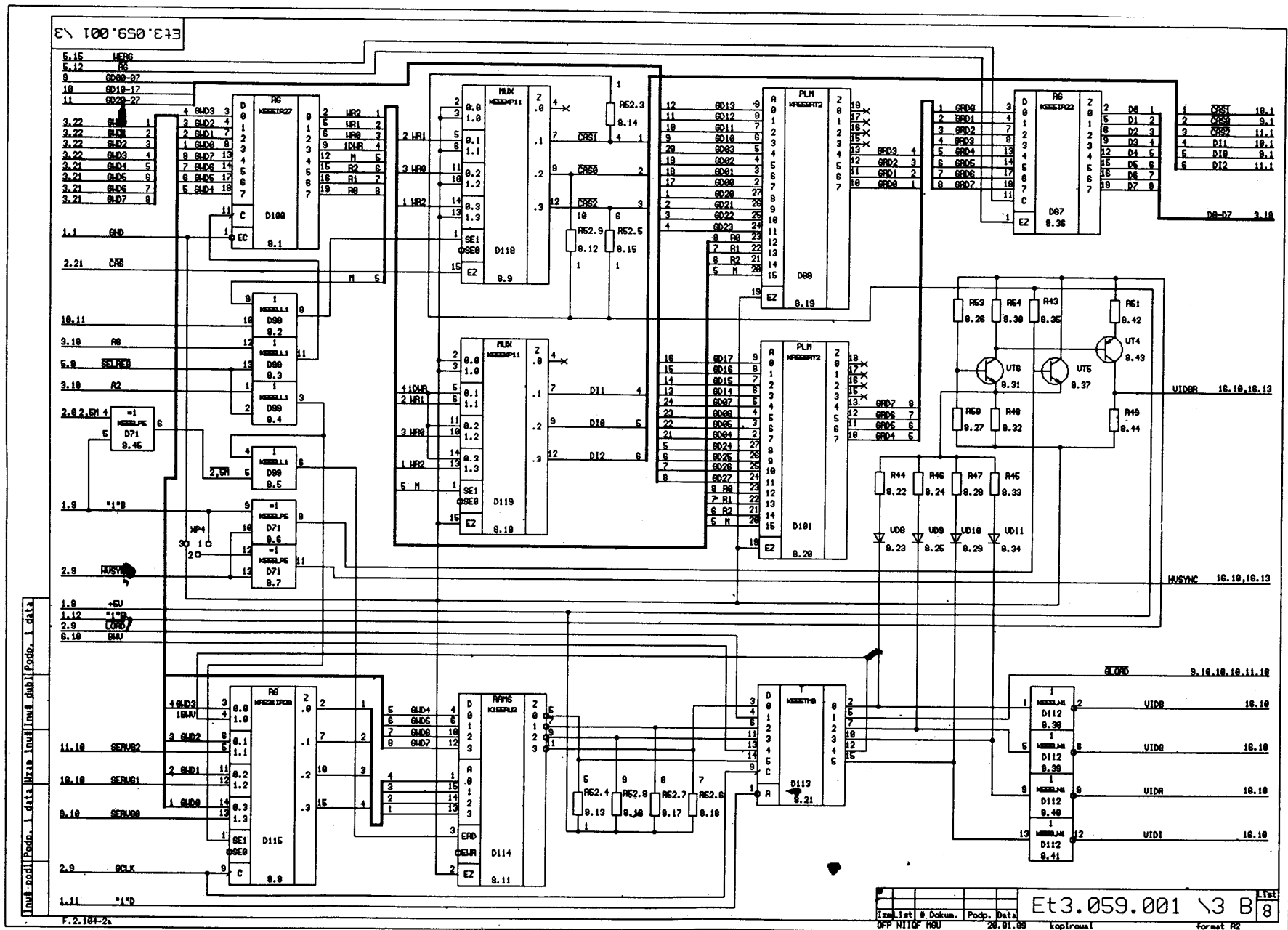
15	D4	15	D	RAM	15120PLM	2	7
1	A	0					
2	A	1					
3	A	2					
4	A	3					
5	A	4					
6	A	5					
7	A	6					
8	A	7					
9	A	8					
10	A	9					
11	ERD	OEUR	D68	7.6			
12	EZ						
13							
14							
15							

18	D7	15	D	RAM	15120PLM	2	7
1	A	0					
2	A	1					
3	A	2					
4	A	3					
5	A	4					
6	A	5					
7	A	6					
8	A	7					
9	A	8					
10	A	9					
11	ERD	OEUR	D63	7.9			
12	EZ						
13							
14							
15							

lini modl Podo. i dat. Uzam Inuflnuh duh Podo. i data

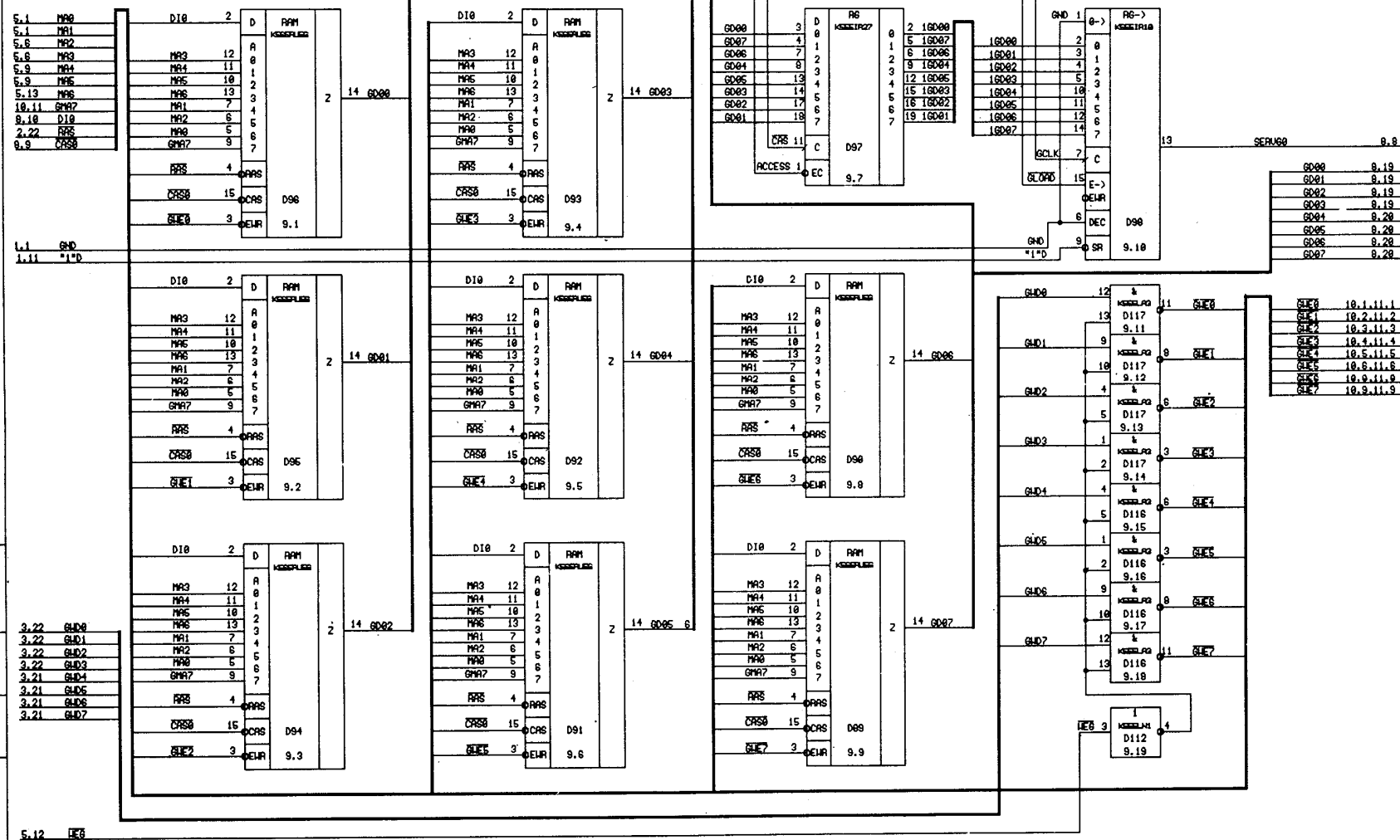
5.12 NEU  
2.9 CSU

UD0-UD7 6.4



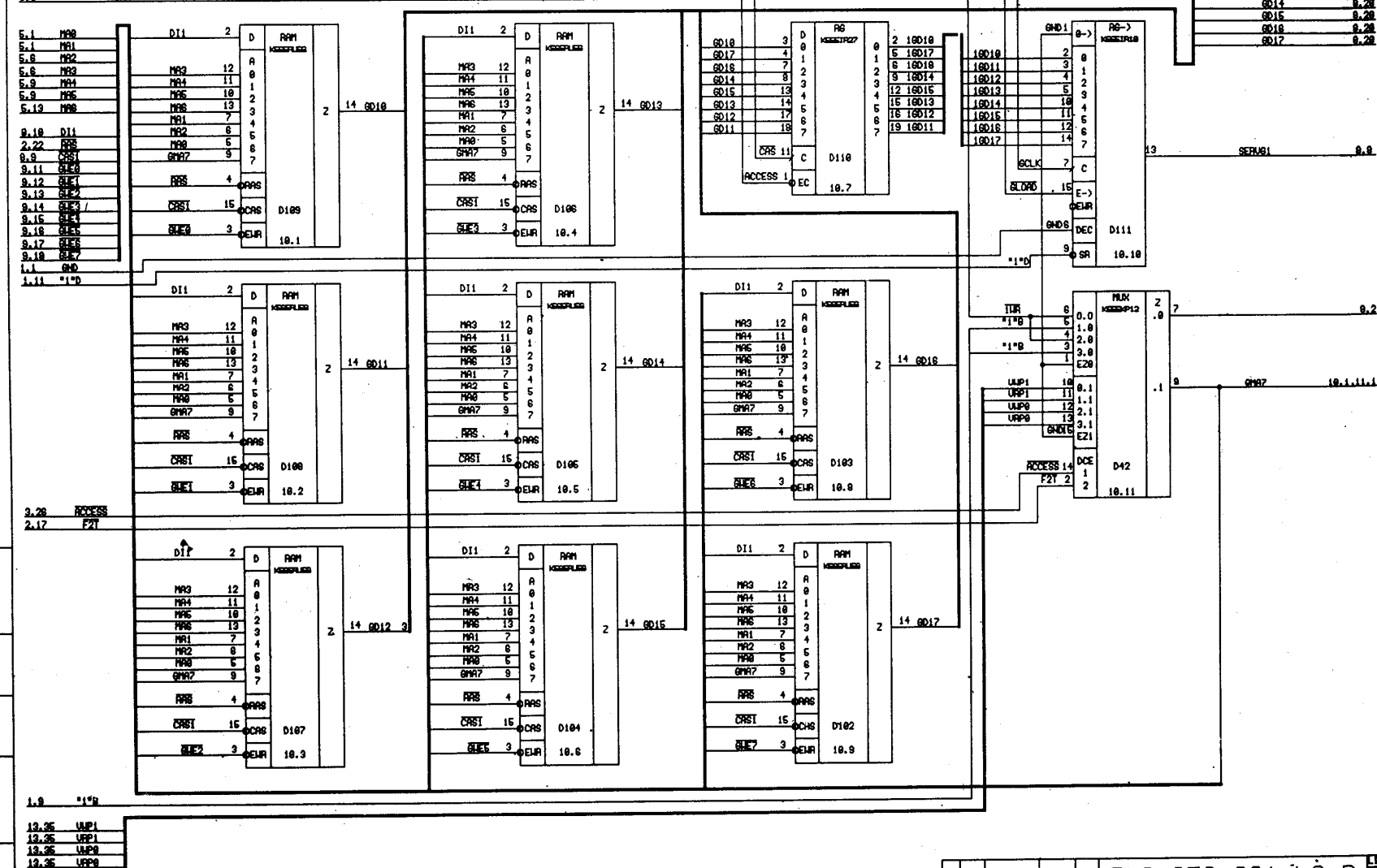


2.9	GCLK
8.21	GLRD
2.21	CAS
3.28	ACCESS



5.12 156

2.9	BCLK
9.21	BLOAD
2.21	CAS
3.28	ACCESS
5.6	TIME



Inu# podl	Podo.	I data	Uzao	Inu# dubl	Podo.	I data
-----------	-------	--------	------	-----------	-------	--------

1.9	"1"R
13.35	UMP1
13.35	UPP1
13.35	UMP0
13.35	UPP0

2304-25

Et3.059.001 \3 B		11
Imm list # Dokum.	Podp. Data	
OP HIGG HBU	22.10.90	11
kopiroval		
format R2		

Et3.059.001 \3

2.9 GCLK  
9.21 SLOD  
2.21 CRS  
3.28 ACCESS

5.1 MA8	D12	2	D	RAM	14	6020
5.1 MA1						
5.8 MA2						
5.8 MA3						
5.9 MA4						
5.9 MA5						
5.13 MA6						
10.11 GMA7						
9.10 D12						
2.22 RAS						
9.9 CRS2						
9.11 SLE0						
9.12 SLE1						
9.13 SLE2						
9.14 SLE3						
9.15 SLE4						
9.16 SLE5						
9.17 SLE6						
9.18 SLE7						
1.1 SLO						
1.11 *1D						

5.1 MA8	D12	2	D	RAM	14	6023
5.1 MA1						
5.8 MA2						
5.8 MA3						
5.9 MA4						
5.9 MA5						
5.13 MA6						
10.11 GMA7						
9.10 D12						
2.22 RAS						
9.9 CRS2						
9.11 SLE0						
9.12 SLE1						
9.13 SLE2						
9.14 SLE3						
9.15 SLE4						
9.16 SLE5						
9.17 SLE6						
9.18 SLE7						
1.1 SLO						
1.11 *1D						

5.1 MA8	D12	2	D	RAM	14	6024
5.1 MA1						
5.8 MA2						
5.8 MA3						
5.9 MA4						
5.9 MA5						
5.13 MA6						
10.11 GMA7						
9.10 D12						
2.22 RAS						
9.9 CRS2						
9.11 SLE0						
9.12 SLE1						
9.13 SLE2						
9.14 SLE3						
9.15 SLE4						
9.16 SLE5						
9.17 SLE6						
9.18 SLE7						
1.1 SLO						
1.11 *1D						

5.1 MA8	D12	2	D	RAM	14	6026
5.1 MA1						
5.8 MA2						
5.8 MA3						
5.9 MA4						
5.9 MA5						
5.13 MA6						
10.11 GMA7						
9.10 D12						
2.22 RAS						
9.9 CRS2						
9.11 SLE0						
9.12 SLE1						
9.13 SLE2						
9.14 SLE3						
9.15 SLE4						
9.16 SLE5						
9.17 SLE6						
9.18 SLE7						
1.1 SLO						
1.11 *1D						

5.1 MA8	D12	2	D	RAM	14	6021
5.1 MA1						
5.8 MA2						
5.8 MA3						
5.9 MA4						
5.9 MA5						
5.13 MA6						
10.11 GMA7						
9.10 D12						
2.22 RAS						
9.9 CRS2						
9.11 SLE0						
9.12 SLE1						
9.13 SLE2						
9.14 SLE3						
9.15 SLE4						
9.16 SLE5						
9.17 SLE6						
9.18 SLE7						
1.1 SLO						
1.11 *1D						

5.1 MA8	D12	2	D	RAM	14	6024
5.1 MA1						
5.8 MA2						
5.8 MA3						
5.9 MA4						
5.9 MA5						
5.13 MA6						
10.11 GMA7						
9.10 D12						
2.22 RAS						
9.9 CRS2						
9.11 SLE0						
9.12 SLE1						
9.13 SLE2						
9.14 SLE3						
9.15 SLE4						
9.16 SLE5						
9.17 SLE6						
9.18 SLE7						
1.1 SLO						
1.11 *1D						

5.1 MA8	D12	2	D	RAM	14	6026
5.1 MA1						
5.8 MA2						
5.8 MA3						
5.9 MA4						
5.9 MA5						
5.13 MA6						
10.11 GMA7						
9.10 D12						
2.22 RAS						
9.9 CRS2						
9.11 SLE0						
9.12 SLE1						
9.13 SLE2						
9.14 SLE3						
9.15 SLE4						
9.16 SLE5						
9.17 SLE6						
9.18 SLE7						
1.1 SLO						
1.11 *1D						

5.1 MA8	D12	2	D	RAM	14	6022
5.1 MA1						
5.8 MA2						
5.8 MA3						
5.9 MA4						
5.9 MA5						
5.13 MA6						
10.11 GMA7						
9.10 D12						
2.22 RAS						
9.9 CRS2						
9.11 SLE0						
9.12 SLE1						
9.13 SLE2						
9.14 SLE3						
9.15 SLE4						
9.16 SLE5						
9.17 SLE6						
9.18 SLE7						
1.1 SLO						
1.11 *1D						

5.1 MA8	D12	2	D	RAM	14	6026
5.1 MA1						
5.8 MA2						
5.8 MA3						
5.9 MA4						
5.9 MA5						
5.13 MA6						
10.11 GMA7						
9.10 D12						
2.22 RAS						
9.9 CRS2						
9.11 SLE0						
9.12 SLE1						
9.13 SLE2						
9.14 SLE3						
9.15 SLE4						
9.16 SLE5						
9.17 SLE6						
9.18 SLE7						
1.1 SLO						
1.11 *1D						

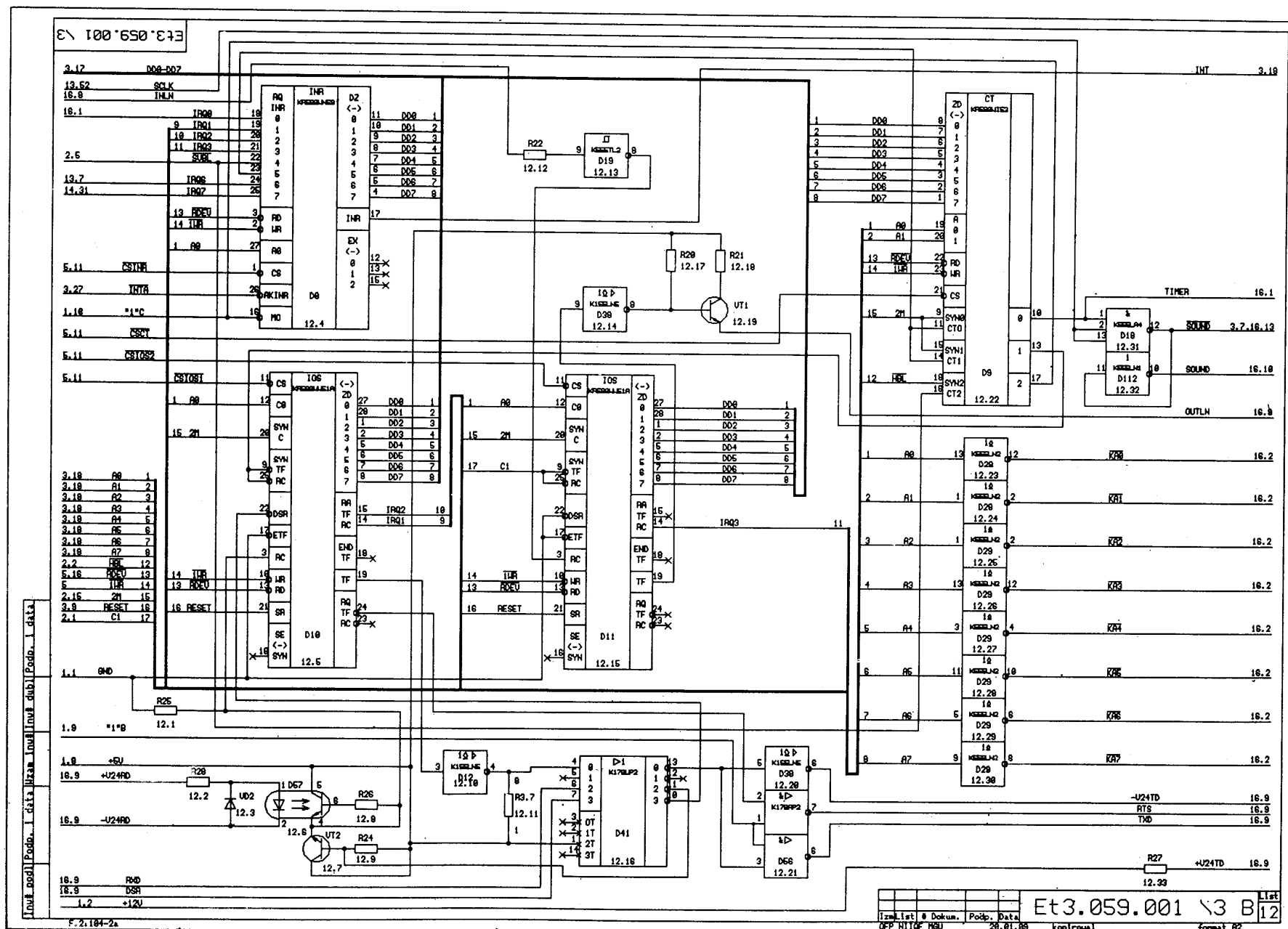
5.1 MA8	D12	2	D	RAM	14	6027
5.1 MA1						
5.8 MA2						
5.8 MA3						
5.9 MA4						
5.9 MA5						
5.13 MA6						
10.11 GMA7						
9.10 D12						
2.22 RAS						
9.9 CRS2						
9.11 SLE0						
9.12 SLE1						
9.13 SLE2						
9.14 SLE3						
9.15 SLE4						
9.16 SLE5						
9.17 SLE6						
9.18 SLE7						
1.1 SLO						
1.11 *1D						

6020	8.18
6021	8.18
6022	8.18
6023	8.18
6024	8.18
6025	8.18
6026	8.18
6027	8.18

Inst. mod. Pod. 1 data. Izam. Inst. Inst. dubl. Pod. 1 data.

F.2.104-2a

Et3.059.001 \3 B11  
Izm. List. 8 Dokum. Pod. Data  
OFF HITC-18U 22.10.98 kopiroval format R2



3.100.650.873

3.17	DDA-DD7	1
3.18	AD	2
3.19	AD	3
3.20	AD	4
3.21	AD	5
3.22	AD	6
3.23	AD	7
3.24	AD	8
3.25	AD	9
3.26	AD	10
3.27	AD	11
3.28	AD	12
3.29	AD	13
3.30	AD	14
3.31	AD	15
3.32	AD	16
3.33	AD	17
3.34	AD	18
3.35	AD	19
3.36	AD	20
3.37	AD	21
3.38	AD	22
3.39	AD	23
3.40	AD	24
3.41	AD	25
3.42	AD	26
3.43	AD	27
3.44	AD	28
3.45	AD	29
3.46	AD	30
3.47	AD	31
3.48	AD	32
3.49	AD	33
3.50	AD	34
3.51	AD	35
3.52	AD	36
3.53	AD	37
3.54	AD	38
3.55	AD	39
3.56	AD	40
3.57	AD	41
3.58	AD	42
3.59	AD	43
3.60	AD	44
3.61	AD	45
3.62	AD	46
3.63	AD	47
3.64	AD	48
3.65	AD	49
3.66	AD	50
3.67	AD	51
3.68	AD	52
3.69	AD	53
3.70	AD	54
3.71	AD	55
3.72	AD	56
3.73	AD	57
3.74	AD	58
3.75	AD	59
3.76	AD	60
3.77	AD	61
3.78	AD	62
3.79	AD	63
3.80	AD	64
3.81	AD	65
3.82	AD	66
3.83	AD	67
3.84	AD	68
3.85	AD	69
3.86	AD	70
3.87	AD	71
3.88	AD	72
3.89	AD	73
3.90	AD	74
3.91	AD	75
3.92	AD	76
3.93	AD	77
3.94	AD	78
3.95	AD	79
3.96	AD	80
3.97	AD	81
3.98	AD	82
3.99	AD	83
4.00	AD	84
4.01	AD	85
4.02	AD	86
4.03	AD	87
4.04	AD	88
4.05	AD	89
4.06	AD	90
4.07	AD	91
4.08	AD	92
4.09	AD	93
4.10	AD	94
4.11	AD	95
4.12	AD	96
4.13	AD	97
4.14	AD	98
4.15	AD	99
4.16	AD	100

2.4	UOL	1
2.5	CD1	2
2.6	ADPC	3
2.7	ADPC	4
2.8	ADPC	5
2.9	ADPC	6
2.10	ADPC	7
2.11	ADPC	8
2.12	ADPC	9
2.13	ADPC	10
2.14	ADPC	11
2.15	ADPC	12
2.16	ADPC	13
2.17	ADPC	14
2.18	ADPC	15
2.19	ADPC	16
2.20	ADPC	17
2.21	ADPC	18
2.22	ADPC	19
2.23	ADPC	20
2.24	ADPC	21
2.25	ADPC	22
2.26	ADPC	23
2.27	ADPC	24
2.28	ADPC	25
2.29	ADPC	26
2.30	ADPC	27
2.31	ADPC	28
2.32	ADPC	29
2.33	ADPC	30
2.34	ADPC	31
2.35	ADPC	32
2.36	ADPC	33
2.37	ADPC	34
2.38	ADPC	35
2.39	ADPC	36
2.40	ADPC	37
2.41	ADPC	38
2.42	ADPC	39
2.43	ADPC	40
2.44	ADPC	41
2.45	ADPC	42
2.46	ADPC	43
2.47	ADPC	44
2.48	ADPC	45
2.49	ADPC	46
2.50	ADPC	47
2.51	ADPC	48
2.52	ADPC	49
2.53	ADPC	50
2.54	ADPC	51
2.55	ADPC	52
2.56	ADPC	53
2.57	ADPC	54
2.58	ADPC	55
2.59	ADPC	56
2.60	ADPC	57
2.61	ADPC	58
2.62	ADPC	59
2.63	ADPC	60
2.64	ADPC	61
2.65	ADPC	62
2.66	ADPC	63
2.67	ADPC	64
2.68	ADPC	65
2.69	ADPC	66
2.70	ADPC	67
2.71	ADPC	68
2.72	ADPC	69
2.73	ADPC	70
2.74	ADPC	71
2.75	ADPC	72
2.76	ADPC	73
2.77	ADPC	74
2.78	ADPC	75
2.79	ADPC	76
2.80	ADPC	77
2.81	ADPC	78
2.82	ADPC	79
2.83	ADPC	80
2.84	ADPC	81
2.85	ADPC	82
2.86	ADPC	83
2.87	ADPC	84
2.88	ADPC	85
2.89	ADPC	86
2.90	ADPC	87
2.91	ADPC	88
2.92	ADPC	89
2.93	ADPC	90
2.94	ADPC	91
2.95	ADPC	92
2.96	ADPC	93
2.97	ADPC	94
2.98	ADPC	95
2.99	ADPC	96
3.00	ADPC	97
3.01	ADPC	98
3.02	ADPC	99
3.03	ADPC	100

1.1 GND

1.2 +12V

16.2	2000	1
16.2	2000	2
16.2	2000	3
16.2	2000	4
16.2	2000	5
16.2	2000	6
16.2	2000	7
16.2	2000	8
16.2	2000	9
16.2	2000	10
16.2	2000	11
16.2	2000	12
16.2	2000	13
16.2	2000	14
16.2	2000	15
16.2	2000	16

2.19	AD	11
2.20	AD	12
2.21	AD	13
2.22	AD	14
2.23	AD	15
2.24	AD	16
2.25	AD	17
2.26	AD	18
2.27	AD	19
2.28	AD	20
2.29	AD	21
2.30	AD	22
2.31	AD	23
2.32	AD	24
2.33	AD	25
2.34	AD	26
2.35	AD	27
2.36	AD	28
2.37	AD	29
2.38	AD	30
2.39	AD	31
2.40	AD	32
2.41	AD	33
2.42	AD	34
2.43	AD	35
2.44	AD	36
2.45	AD	37
2.46	AD	38
2.47	AD	39
2.48	AD	40
2.49	AD	41
2.50	AD	42
2.51	AD	43
2.52	AD	44
2.53	AD	45
2.54	AD	46
2.55	AD	47
2.56	AD	48
2.57	AD	49
2.58	AD	50
2.59	AD	51
2.60	AD	52
2.61	AD	53
2.62	AD	54
2.63	AD	55
2.64	AD	56
2.65	AD	57
2.66	AD	58
2.67	AD	59
2.68	AD	60
2.69	AD	61
2.70	AD	62
2.71	AD	63
2.72	AD	64
2.73	AD	65
2.74	AD	66
2.75	AD	67
2.76	AD	68
2.77	AD	69
2.78	AD	70
2.79	AD	71
2.80	AD	72
2.81	AD	73
2.82	AD	74
2.83	AD	75
2.84	AD	76
2.85	AD	77
2.86	AD	78
2.87	AD	79
2.88	AD	80
2.89	AD	81
2.90	AD	82
2.91	AD	83
2.92	AD	84
2.93	AD	85
2.94	AD	86
2.95	AD	87
2.96	AD	88
2.97	AD	89
2.98	AD	90
2.99	AD	91
3.00	AD	92
3.01	AD	93
3.02	AD	94
3.03	AD	95
3.04	AD	96
3.05	AD	97
3.06	AD	98
3.07	AD	99
3.08	AD	100

1.3 +5V

F.2.104-2a

1.1 GND

1.2 +12V

1.3 +5V

1.4 +5V

1.5 +5V

1.6 +5V

1.7 +5V

1.8 +5V

1.9 +5V

1.10 +5V

1.11 +5V

1.12 +5V

1.13 +5V

1.14 +5V

1.15 +5V

1.16 +5V

1.17 +5V

1.18 +5V

1.19 +5V

1.20 +5V

1.21 +5V

1.22 +5V

1.23 +5V

1.24 +5V

1.25 +5V

1.26 +5V

1.27 +5V

1.28 +5V

1.29 +5V

1.30 +5V

1.31 +5V

1.32 +5V

1.33 +5V

1.34 +5V

1.35 +5V

1.36 +5V

1.37 +5V

1.38 +5V

1.39 +5V

1.40 +5V

1.41 +5V

1.42 +5V

1.43 +5V

1.44 +5V

1.45 +5V

1.46 +5V

1.47 +5V

1.48 +5V

1.49 +5V

1.50 +5V

1.51 +5V

1.52 +5V

1.53 +5V

1.54 +5V

1.55 +5V

1.56 +5V

1.57 +5V

1.58 +5V

1.59 +5V

1.60 +5V

1.61 +5V

1.62 +5V

1.63 +5V

1.64 +5V

1.65 +5V

1.66 +5V

1.67 +5V

1.68 +5V

1.69 +5V

1.70 +5V

1.71 +5V

1.72 +5V

1.73 +5V

1.74 +5V

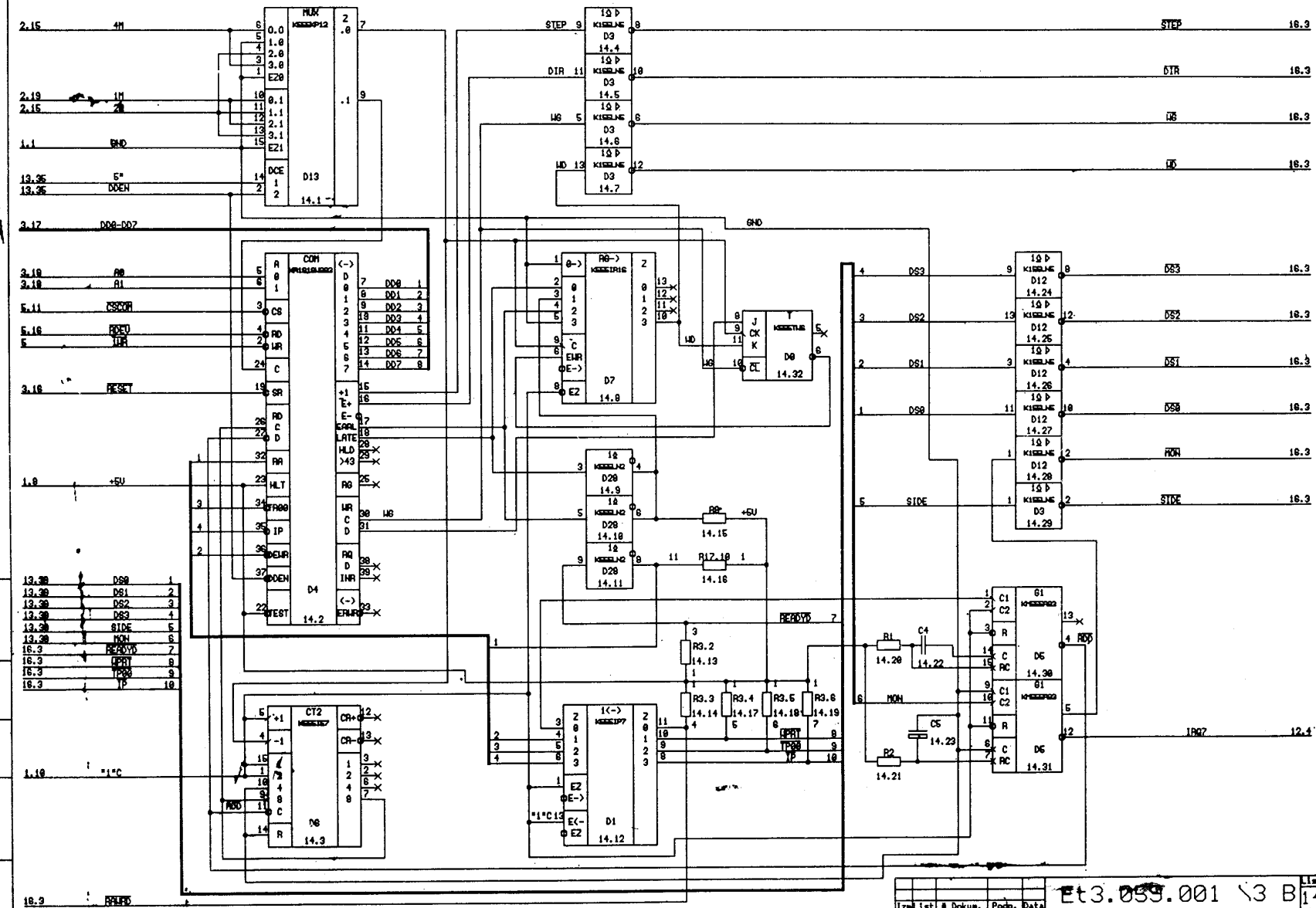
1.75 +5V

1.76 +5V

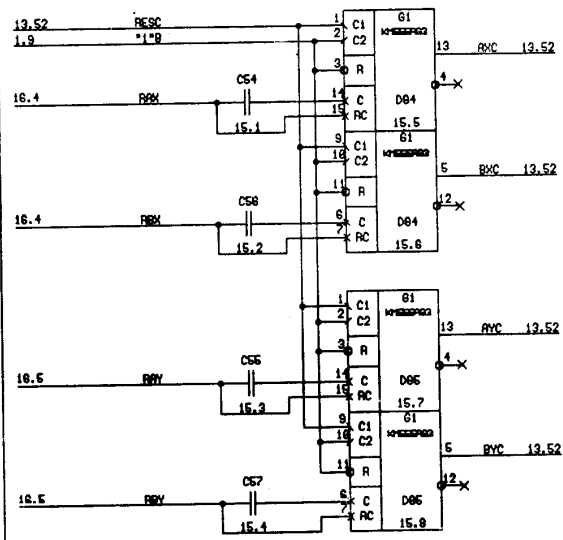
1.77 +5V

1.78 +5V

1.79 +5V



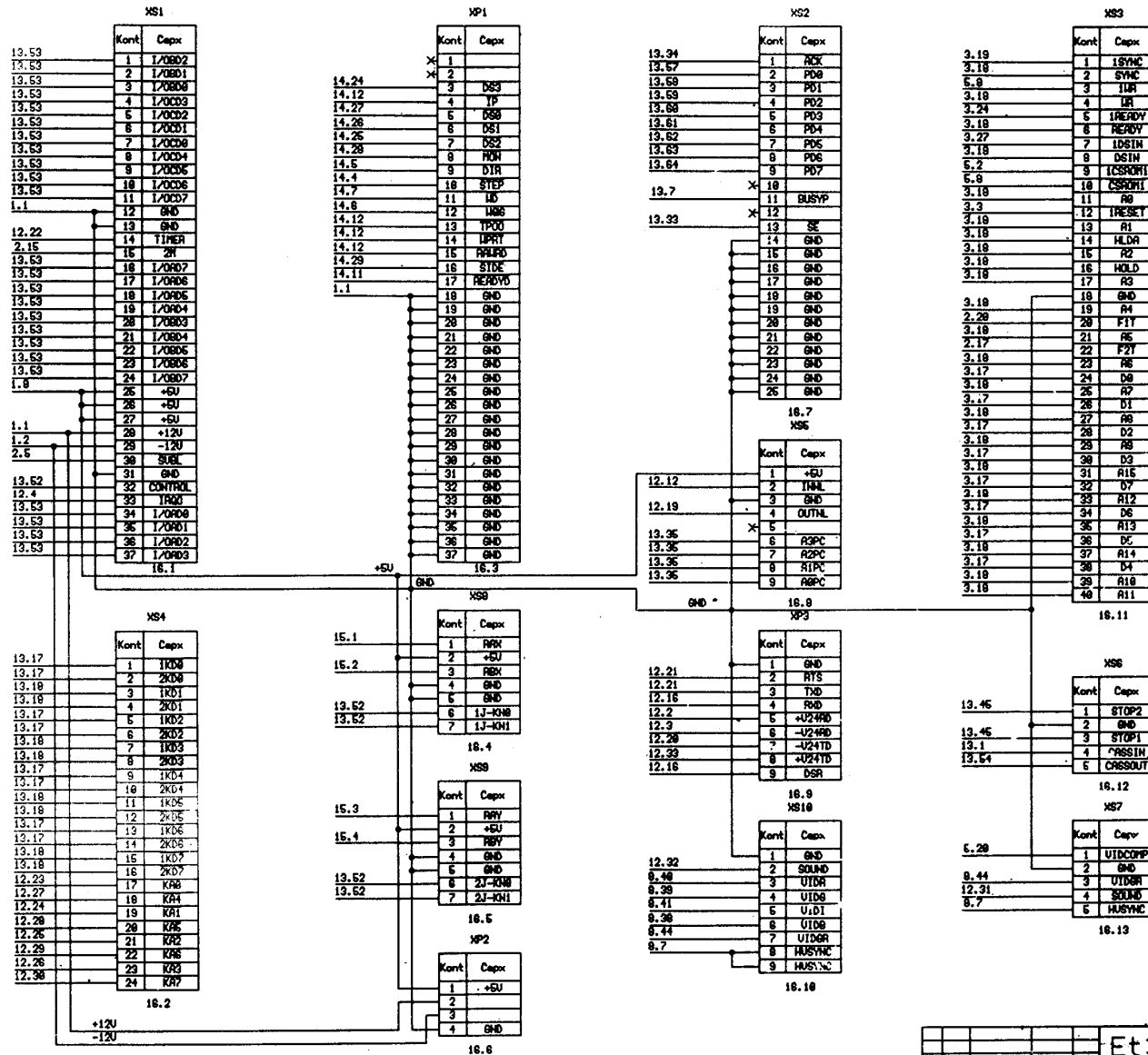
Et3.059.001 \3



Inv# podl Podc. I diti Uzam Inv# Inv# dubl Podc. I data

F.2.104-2a

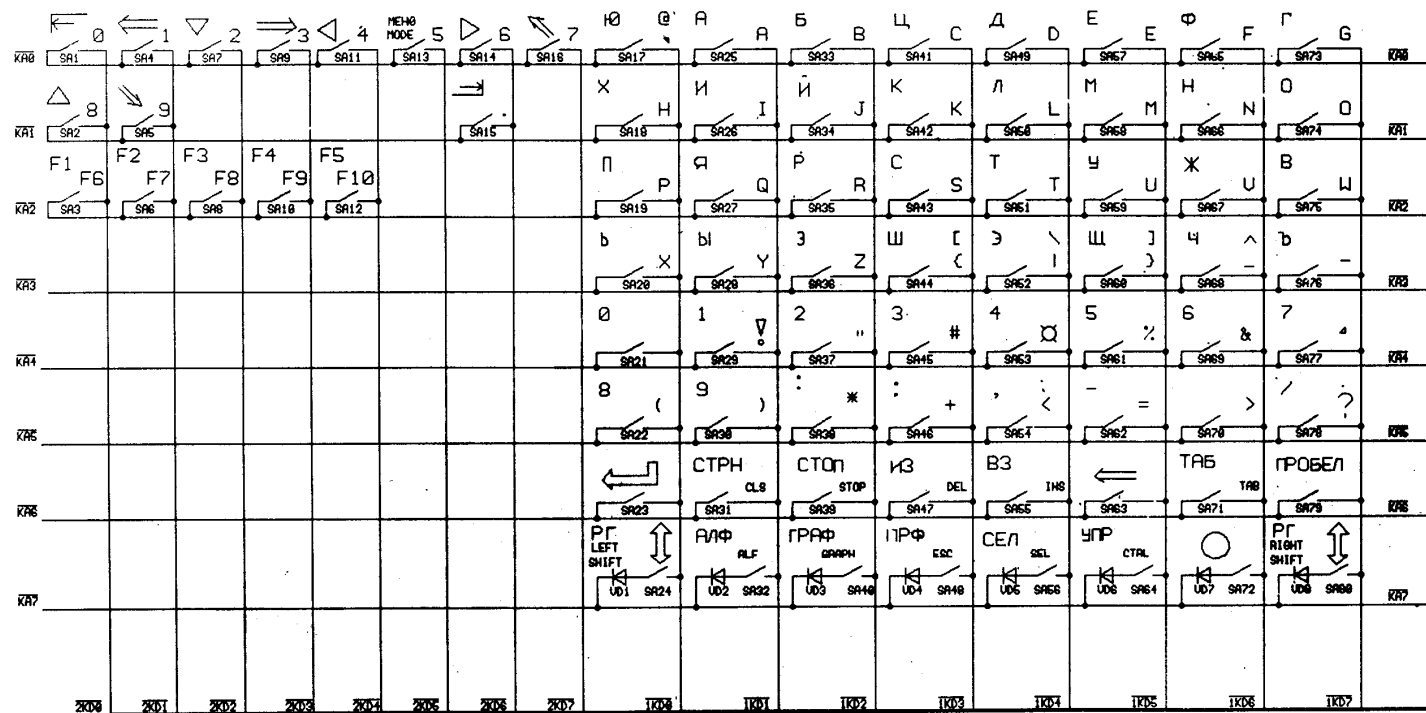
Et3.059.001





Справ. # Перу. др. л. 1000

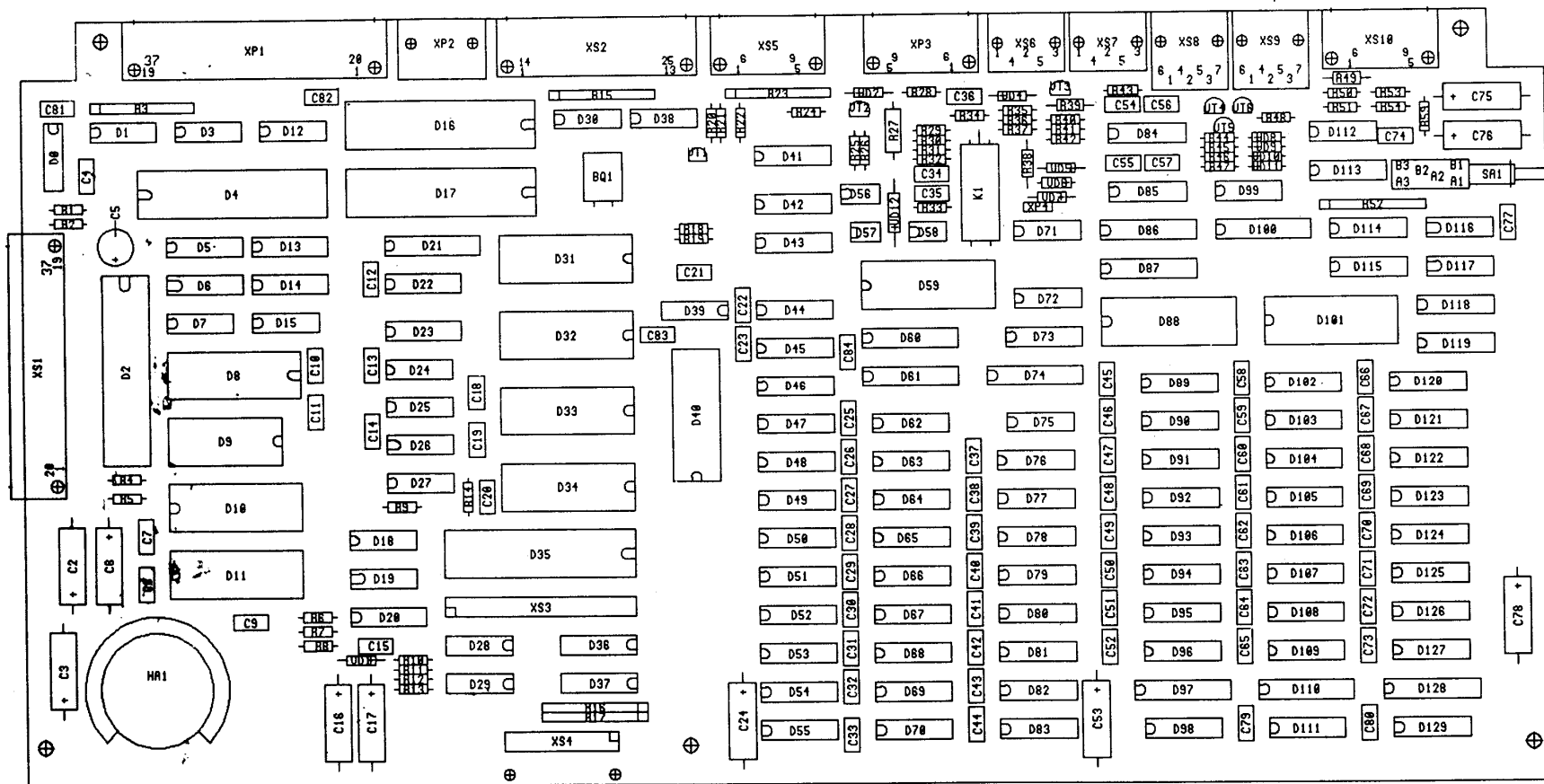
Имя подл. Подп. I. data Имя инв. инв. дубл. Подп. I. data

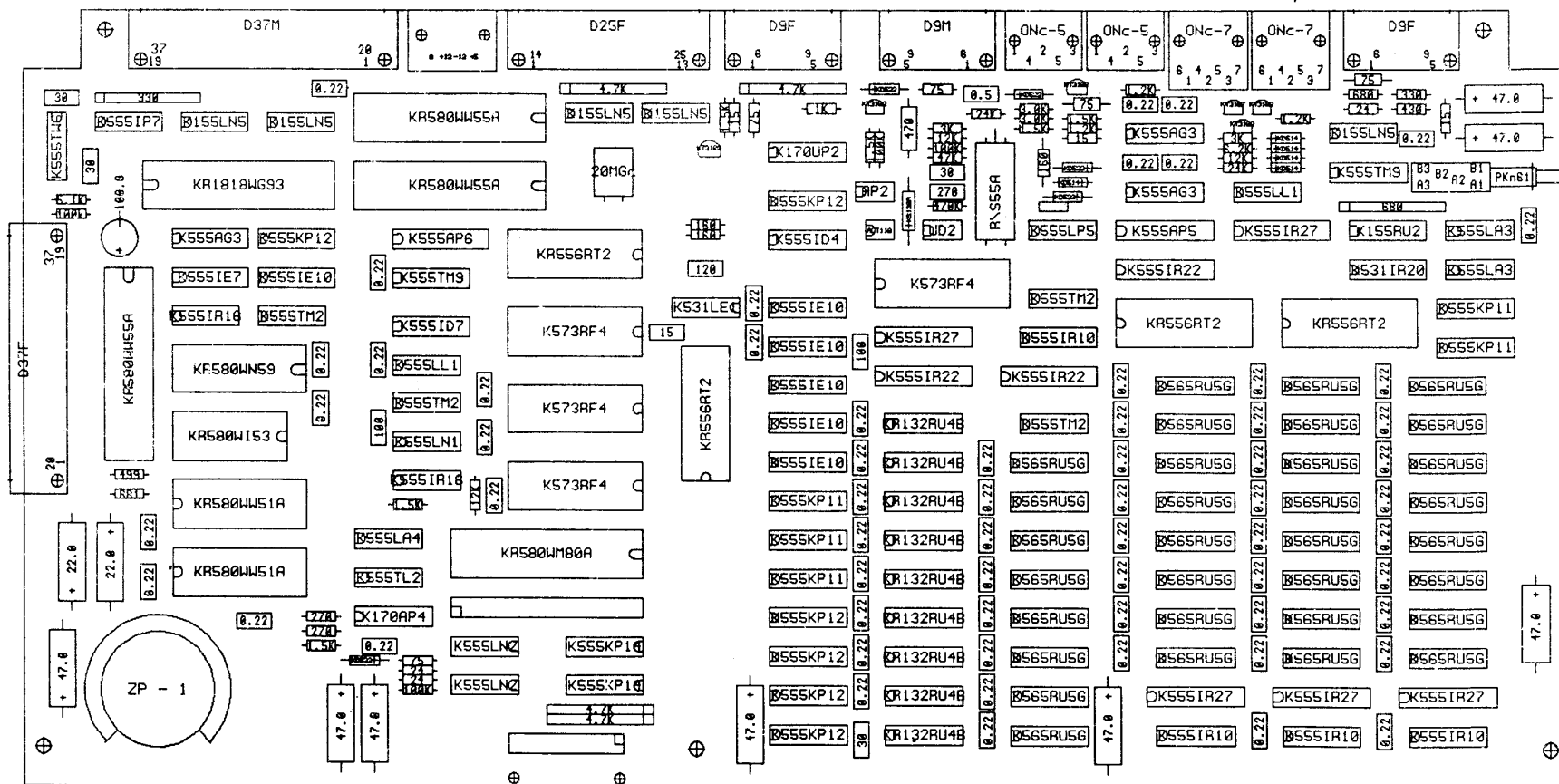


Конт.	Спра.
1	1000
2	2000
3	1001
4	2001
5	1002
6	2002
7	1003
8	2003
9	1004
10	2004
11	1005
12	2005
13	1006
14	2006
15	1007
16	2007
17	KR0
18	KR1
19	KR2
20	KR3
21	KR4
22	KR5
23	KR6
24	KR7

Изм. лист	№ докум.	Подп. дата	Лит.	Масштаб	Масштаб
Разработчик					
Проверен					
Исполнитель					
Наименование					
Исполнитель					

22.10.94 копировал формат: A2





Временные диаграммы  
делителя на 164

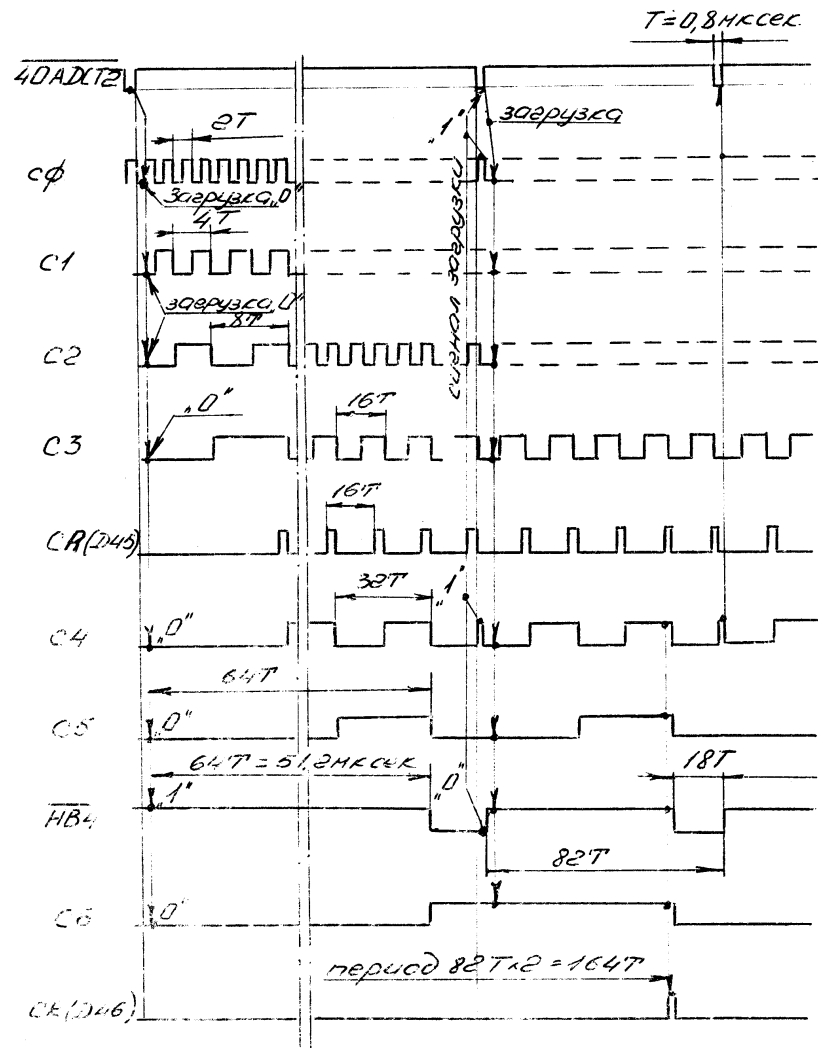


Рис. 1

Временная диаграмма  
делителя на 156

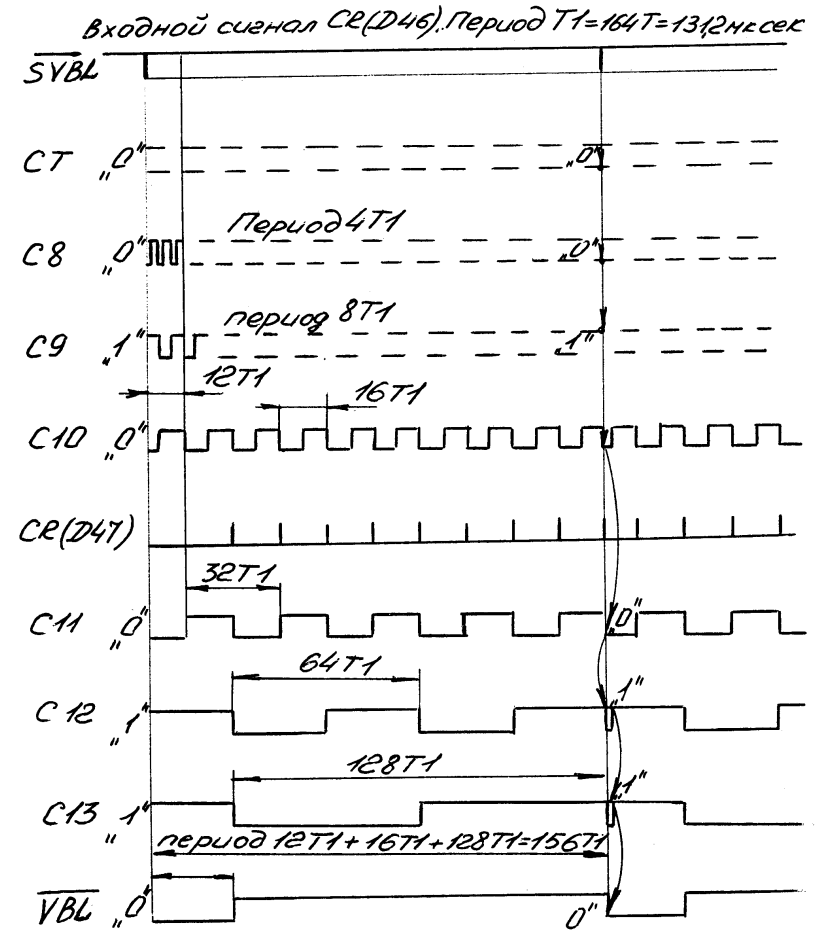


Рис. 2

Временные диаграммы формирователя  
импульсов  $F1$ ,  $F2$  и сигналов  $\overline{RAS}$  и  $\overline{CAS}$

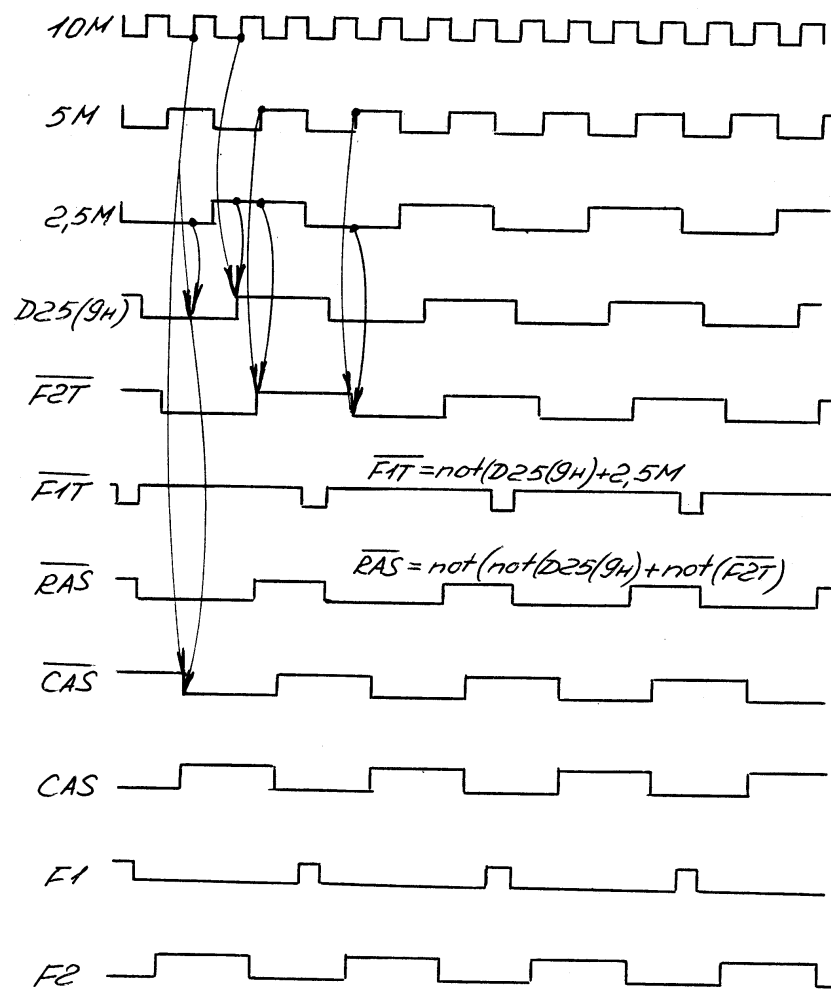


Рис. 3

Временные диаграммы формирователя  
тактовых сигналов для контролера НГМД

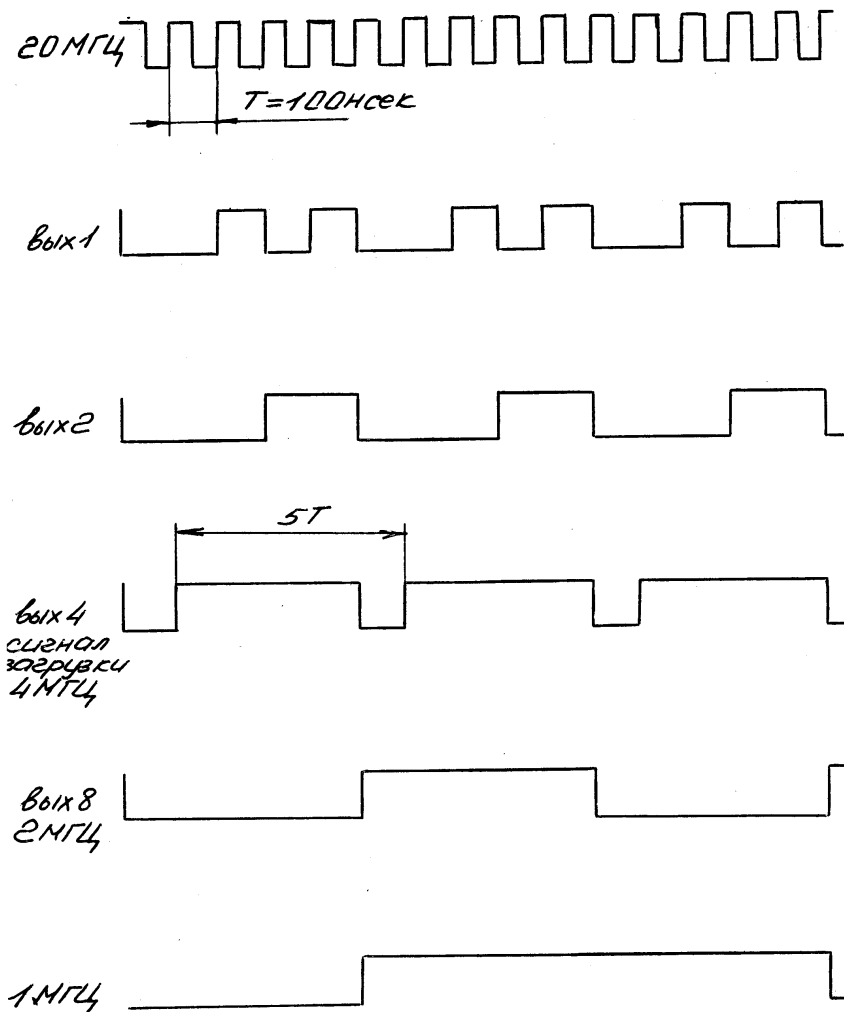


Рис. 4

Временные диаграммы машинных циклов М1, М2, М3

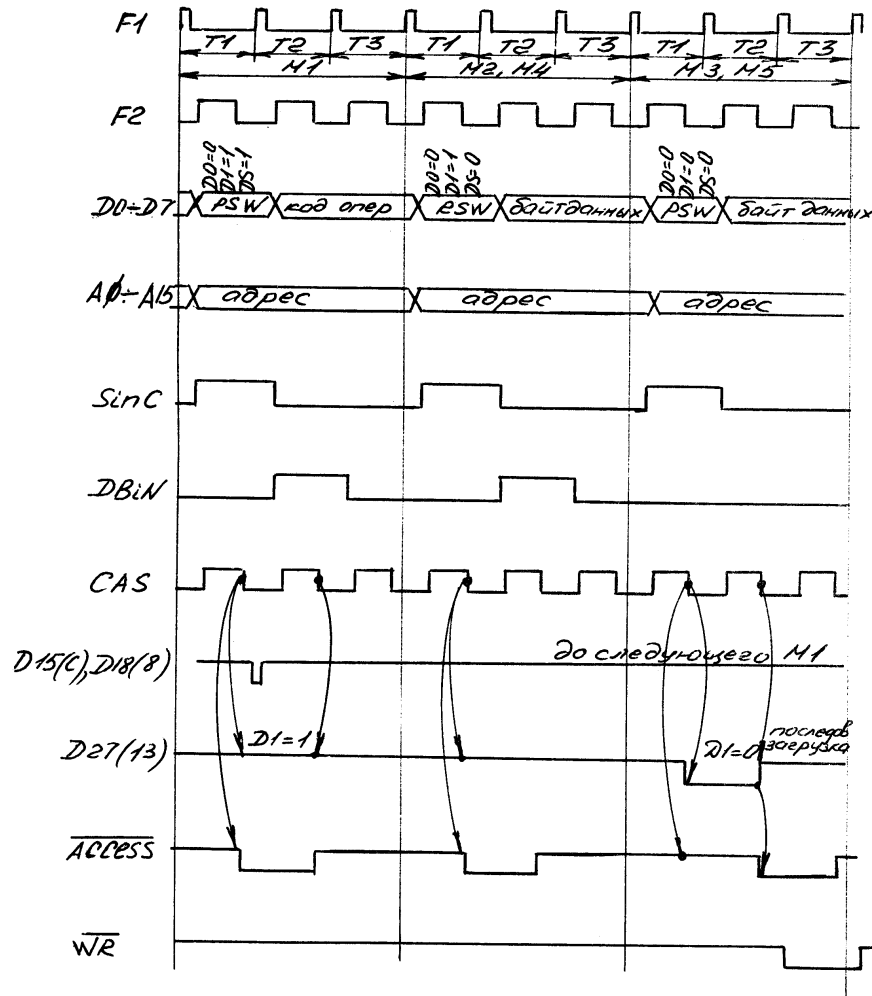


Рис. 5

Временные диаграммы работы ОЗУ  
Запись, чтение, регенерация

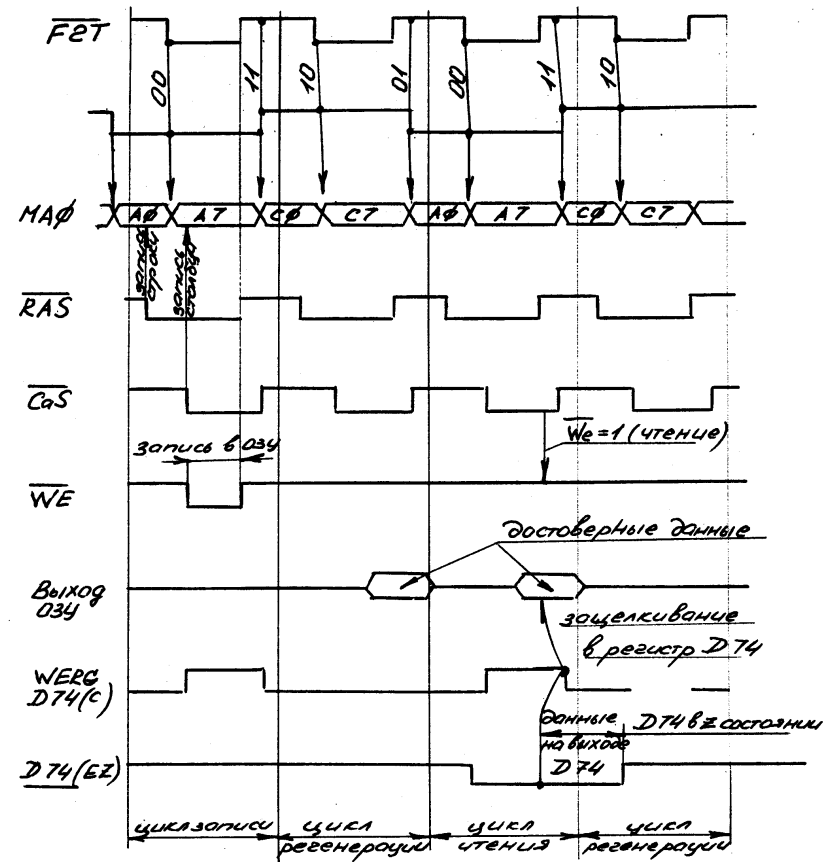


Рис. 6

Временные диаграммы работы  
АЦЗУ (запись, чтение, процессором)

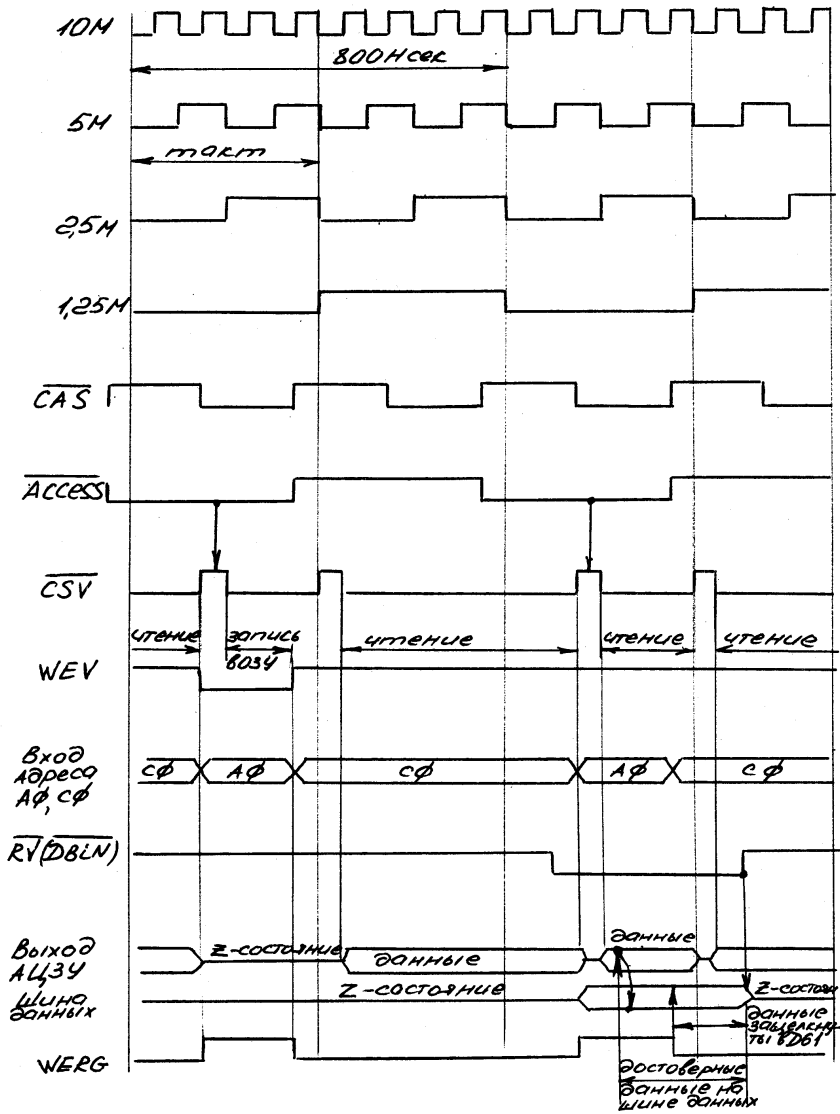


Рис. 7

Временные диаграммы  
получения изображения

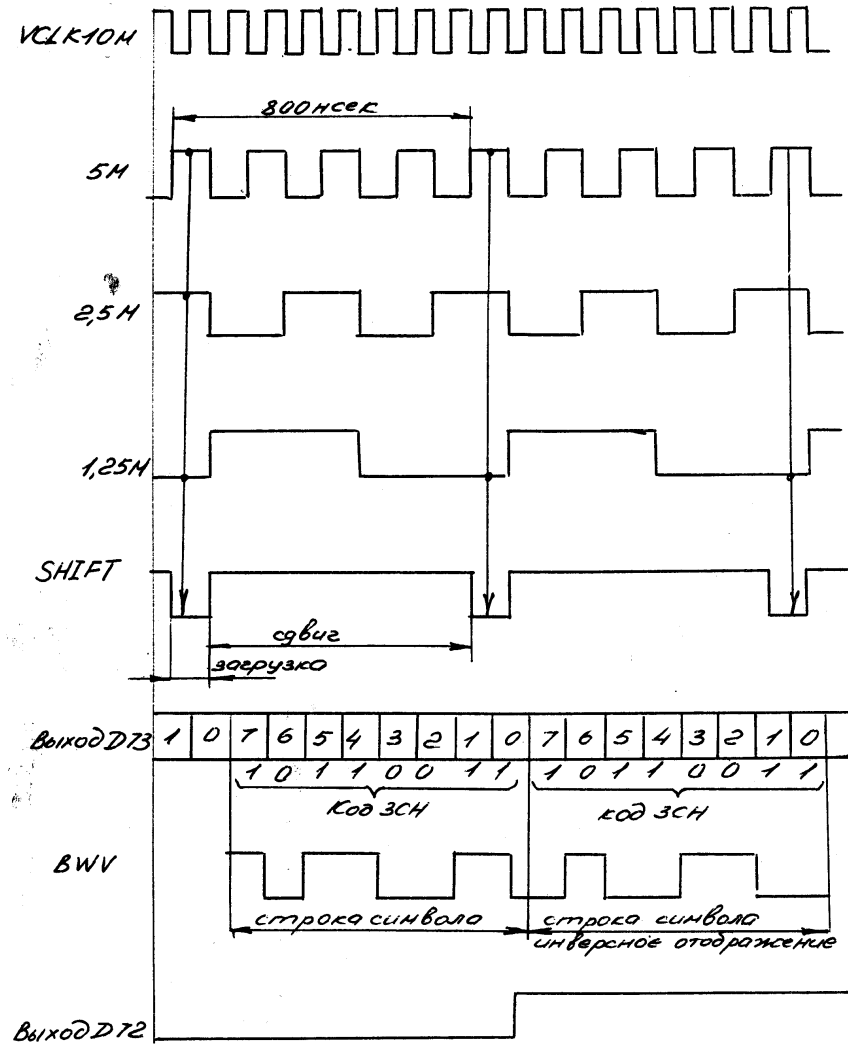


Рис. 8

Временные диаграммы  
сканирования АЦЗУ

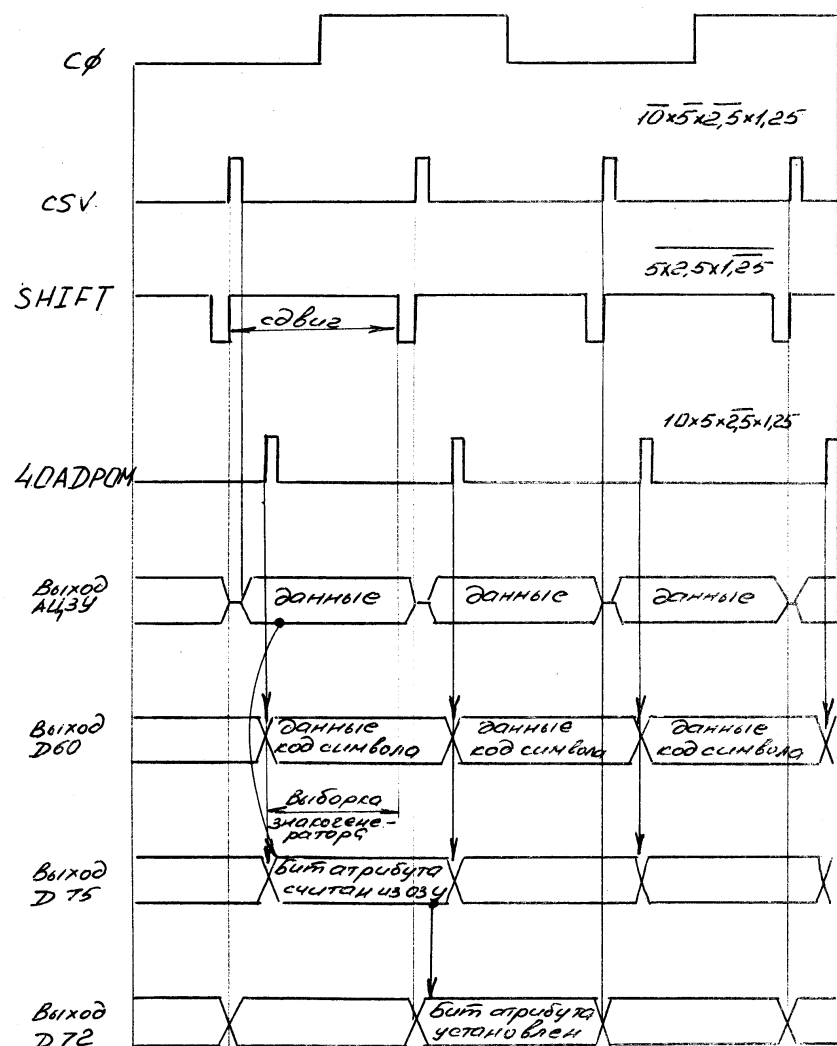


Рис. 9

Временные диаграммы адресации  
столбцов ГЗУ при записи и чтении

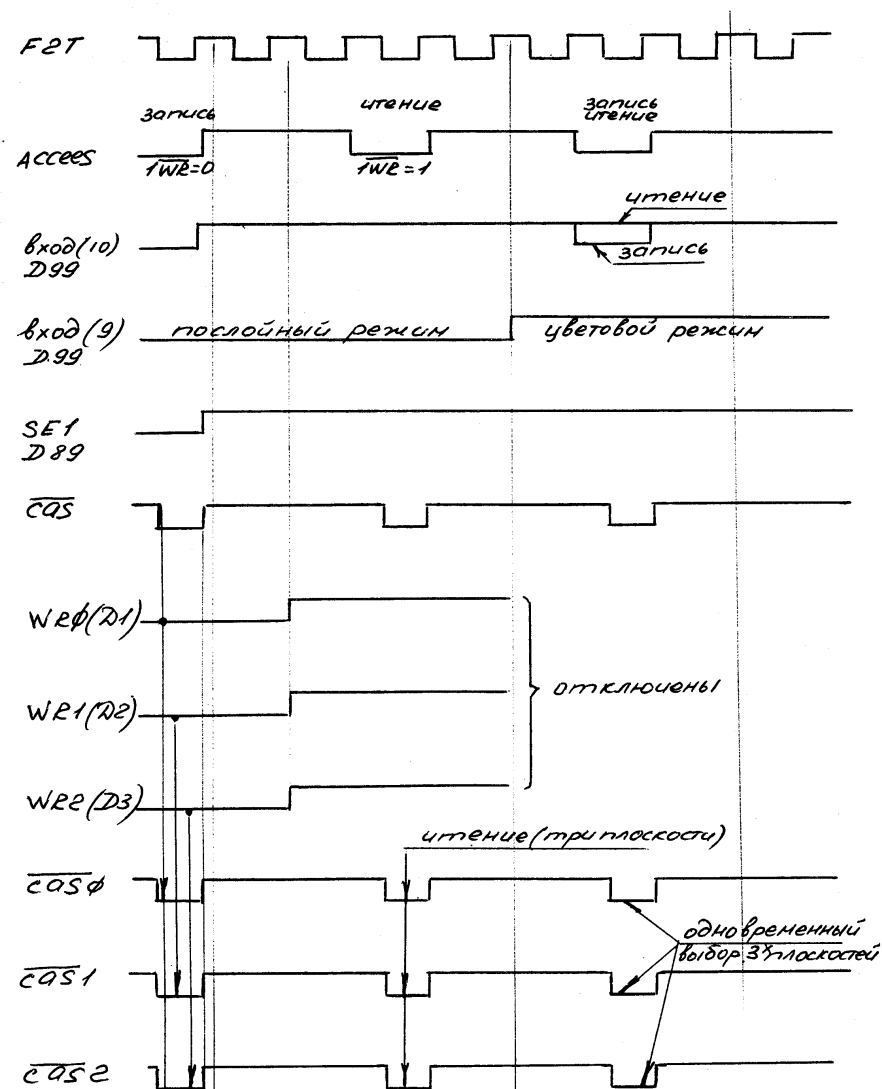
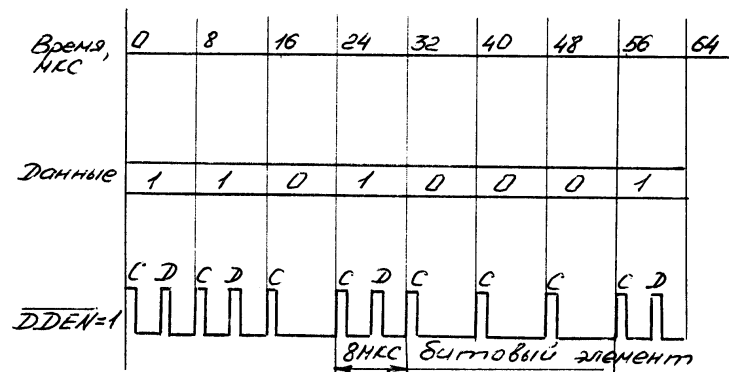


Рис. 10

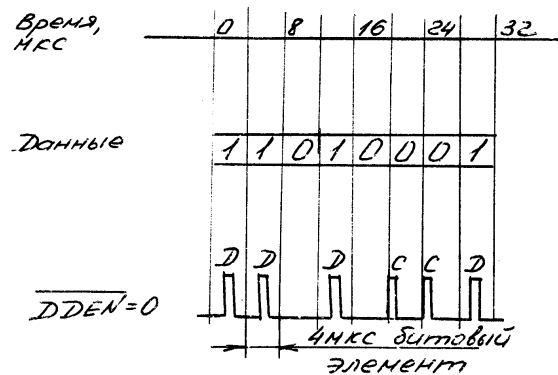


# Временные диаграммы кодирования данных

Удлинная плотность записи FM.

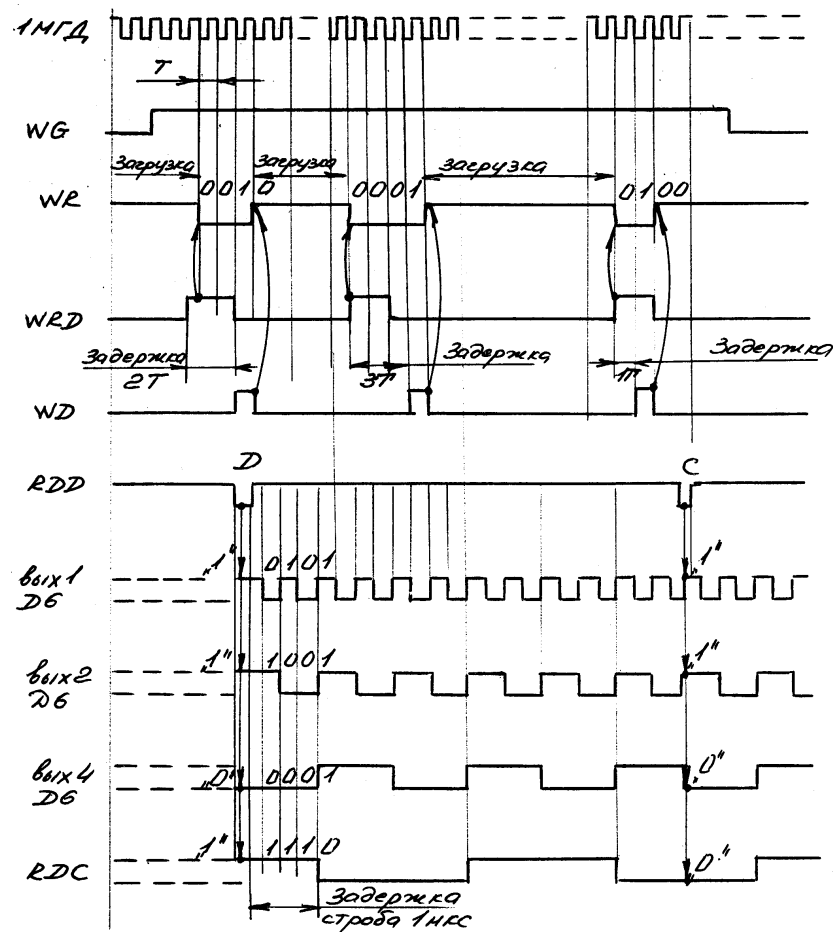


Двойная плотность записи, МФМ



Puc. 11

Временные диаграммы  
схемы компенсации и стробиро-  
вания данных при чтении.



Puc. 12

# КОРВЕТ

## ТЕХНИЧЕСКОЕ ОПИСАНИЕ

### 1. ВВЕДЕНИЕ

Если у Вас нет опыта в цифровой технике, но есть желание разобраться в КОРВЕТЕ досконально, то отбросьте все сомнения и беритесь за дело.

Лучше предварительно обзавестись книжкой В.Л.Шило "Популярные цифровые микросхемы". Несомненно пригодится любой справочник по микропроцессорному комплекту КР580. Если дополнительную литературу достать не удастся, не беда, при обсуждении схем узлов КОРВЕТА мы приводим дополнительную информацию о работе микросхем.

Итак давайте попробуем разобраться в принципиальной схеме КОРВЕТА. Сначала разберемся, как устроен альбом схем. На листе 1 приведена разводка питания. Здесь также приведены все типы микросхем, используемые в схеме, и номера ножек микросхем, на которые подается питающее напряжение.

Собственно схема КОРВЕТА начинается с листа 2. Чтобы разбираться в альбоме, нужно уяснить следующие нехитрые правила.

Микросхемы маркируются двойной нумерацией (например, на листе 2 микросхема КР556РТ2 имеет номера D40 и 2.9. По номеру D40 можно определить на карте размещения элементов в конце альбома расположение микросхемы на плате. Номер 2.9 означает, что микросхему можно найти на листе 2 и ее порядковый номер 9).

На любом листе альбома цепи входных сигналов изображаются всегда слева, а выходных – справа. Маркировка этих цепей также двойная и состоит из названия сигнала и номера микросхемы, от которой (или на которую) поступает сигнал.

На этом комментарий к альбому мы закончим и приступим непосредственно к самой схеме.

Принципиальной особенностью КОРВЕТА является жесткая синхронизация всех процессов. Специальное устройство – синхронизатор вырабатывает все необходимые синхросигналы. Поэтому изучение схемы КОРВЕТА мы начнем с синхронизатора.

### 2. СИНХРОНИЗАТОР

Схема синхронизатора приведена на листе 2 альбома. Синхронизатор состоит из задающего генератора, двадцатиразрядного счетчика и схем формирования тактовых сигналов для других узлов ПЭВМ.

Задающий генератор частотой 20 Мгц. реализован на микросхеме КР531ЛЕ1 (D39(2.8, 2.14) и содержит кварцевый резонатор BQ1.

Счетчик собран на пяти микросхемах К555ЛЕ10 (D44, D45, D46, D47 и D48).

Микросхема D44 это четырехразрядный двоичный счетчик, работающий как делитель частоты задающего генератора. На выходах счетчика формируются частоты: 10 Мгц (14 ножка), 5 Мгц (13 ножка), 2.5 Мгц (12 ножка) и 1.25 Мгц (11 ножка).

Частота 1.25 Мгц поступает на тактовые входы восьмиразрядного счетчика D45, D46, с выхода переноса (15 ножка) которого выходит сигнал частотой 7.622 Кгц, т.е. счетчик D45, D46 делит частоту 1.25 Мгц на 164. Какие же формируются сигналы на остальных выходах счетчика? Обратимся к рисунку 1, на котором изображены временные диаграммы сигналов на всех выходах счетчика.

Рассмотрение начнем с момента времени, когда на счетчик приходит сигнал загрузки LOADCT2 (активный уровень низкий, т.е. при низком уровне сигнала происходит загрузка). Длительность сигнала 0.8 мсек. и совпадает с периодом  $T = 0.8$  мсек. тактовой частоты счетчика (1.25 Мгц).

Как видно из схемы, названия многих сигналов помечены чертой сверху. Этот прием применяется для того, чтобы указать, что активный уровень сигнала является низким. В наших рассуждениях мы не будем применять черту для обозначения названий сигналов. При этом давайте договоримся, что сигнал, прошедший через инвертор, не меняет своего названия. Там, где в схеме важно подчеркнуть, что сигнал низкого уровня активный, мы будем это делать словами. Такой подход даст нам определенные преимущества в будущем при использовании логических формул для сигналов.

Условия формирования сигнала LOADCT2 мы рассмотрим чуть позже, а пока, внимательно посмотрев на схему, мы можем сказать, что в счетчик загружаются данные:  $C0-C5 = "0"$  и  $HBL = "1"$ . Что же загружается в старший разряд счетчика (сигнал C6)? Очень просто: то значение, которое было на выходе в момент загрузки, т.е. сигнал C6 при загрузке не меняется. Положим для определенности  $C6 = "0"$ .

После окончания сигнала загрузки, счетчик переходит в режим счета, поэтому на выходах счетчика будут формироваться периодические сигналы типа "МЕАНДР" с периодами:  $C0 - 2T$ ;  $C1 - 4T$ ;  $C2 - 8T$ ;  $C3 - 16T$ ;  $C4 - 32T$ ;  $C5 - 64T$ . Через время  $64T = 51.2$  мсек. сигналы  $C0 - C5$  будут такие же, как после

загрузки, а сигналы HBL и C6 инвертируются. Эта ситуация отражена на рисунке. Ждем еще  $16T = 12.8$  мксек. Сигналы с периодом кратным  $16T$  не изменяются, т.е.  $C0, C1, C2$  и  $C3 = "0"$ ;  $C4 = "1"$ ;  $C5 = "1"$ ;  $HBL = "0"$  и  $C6 = "1"$ . Теперь самое время пояснить, как формируется сигнал загрузки. Из схемы этого понять нельзя, т.к. все происходит в программируемой матрице D40.

Нам нужна карта прошивки микросхемы D40. Об этом мы поговорим позже, а пока дадим формулу:

$$LOADCT2 = \text{not}(C * C4 * \text{not}HBL)$$

Смысл формулы очень простой. Чтобы получить сигнал  $LOADCT2$ , нужно логически перемножить сигналы  $C0, C4$  и  $\text{not}HBL$ , т.е. свернуть с помощью логической операции "И" сигналы  $C0, C4$  и проинвертированный сигнал  $HBL$ , и полученный результат проинвертировать. Другими словами сигнал  $LOADCT2$  формируется тогда, когда  $C0 = "1"$ ,  $C4 = "1"$  и  $HBL = "0"$ . Через время  $80T$  после начала загрузки у нас ситуация:  $C0 = "0"$ ,  $C4 = "1"$  и  $HBL = "0"$ . Ясно, что нужно подождать еще половину периода сигнала  $C0$ , т.е.  $T$ , и сформируется сигнал загрузки, длительность которого будет равна половине периода сигнала  $C0$ . Далее картина будет в основном повторяться. Исключение будет составлять сигнал переноса  $CR$  (D46). Мы предоставляем читателю самому проанализировать, как формируется сигнал  $CR$  (D46) и убедиться, что его период равен  $164T$ , т.е. счетчик все-таки делит входную частоту на 164.

Набив руку на счетчике D45, D46, попробуем самостоятельно разобраться в счетчике D47, D48. На его вход (вход разрешения счета - D47(ножка 10)) поступает сигнал  $CR$ (D46) с частотой 7.622 КГц (период  $T1 = 164T = 131.2$  мксек.). В момент загрузки в счетчик заносится число 01100100B = 100D. Счетчик начинает считать импульсы  $CR$ (D46) и досчитав до 255, формирует сигнал на выходе счетчика  $CR$ (D48), при этом на остальных выходах установятся единицы. Нетрудно сообразить, что сигнал  $CR$ (D48) появится через  $255 - 100 + 1 = 156$  периодов  $T1$ . Этот же сигнал, пройдя через инвертор D26 и получив название  $SVBL$ , является сигналом загрузки счетчика. Мы получили, что частота сигнала  $SVBL$  в 156 раз меньше входной частоты 7.622 КГц и равна 50 Гц.

После загрузки счетчика в четырех его младших разрядах будет число 0100B = 4D, поэтому сигнал  $CR$ (D47) появится через  $16 - 4 = 12$  периодов  $T1$  после начала счета, а сигнал  $VBL$  станет сигналом высокого уровня через  $12 + 16 = 28$  периодов  $T1$ .

Теперь нетрудно определить параметры сигнала  $VBL$ . Они приведены на рисунке 2.

Наконец мы разобрались с сигналами, которые формирует двадцатиразрядный счетчик. Назначение этих сигналов мы будем выяснять по мере обсуждения тех узлов "КОРВЕТА", в которых они используются, а сейчас разберемся с микросхемой KP556PT2 (D40).

Эта микросхема называется программируемой логической матрицей или ПЛМ и позволяет проводить над входными сигналами логические операции AND, OR и NOT.

Мы уже знаем, как "изготавливается" сигнал  $LOADCT2$ . Теперь приведем рецепты всех сигналов, выходящих из ПЛМ (D40).

0. CSV =  $\text{not}10M * \text{not}5M * \text{not}2.5M * 1.25M + 10M * 5M * \text{not}2.5M * \text{not}ACCESS$
1. GCLK = 10M
2. SHIFT =  $\text{not}(HBL * 5M * 2.5M * \text{not}1.25M * \text{not}MODSEL * VBL + HBL * 5M * 2.5M * \text{not}1.25M * MODSEL * VBL * \text{not}C0)$
3. LOADPOM =  $10M * 5M * \text{not}2.5M * 1.25M$
4. HVSNC =  $\text{not}(\text{not}HBL * VBL * \text{not}C3 * C2 + \text{not}HBL * \text{not}VBL * C11 * \text{not}C3 * C2 + \text{not}HBL * \text{not}VBL * \text{not}C10 * C9 * \text{not}C11 * \text{not}C3 * C2 + \text{not}HBL * \text{not}VBL * C10 * \text{not}C9 * \text{not}C11 * \text{not}C3 * C2 + HBL * \text{not}VBL * C10 * C9 * \text{not}C11)$
5. LOADCT2 =  $\text{not}(\text{not}HBL * C4 * C0)$
6. LOAD =  $\text{not}(HBL * 5M * 2.5M * \text{not}1.25M * VBL)$
7. VCLK =  $10M * \text{not}MODSEL + 1M * 5M * MODSEL$

Вообще-то этой информации достаточно, чтобы составить карту прошивки ПЛМ, но мы также приведем и ее в таблице 1.

В таблице закодирована информация, как программировать ПЛМ. Процесс программирования заключается в пережигании плавких перемычек.

Рассмотрим устройство матрицы. Каждая ПЛМ состоит из 48 матриц "AND", которые называются термами, имеющие номера от 0 до 47. Каждая матрица "AND" (терм) имеет 32 входа. Все входы всех термов подключены к 16 входным линиям ПЛМ. Таким образом, каждый вход ПЛМ подключен к паре входов каждого терма (один вход прямой, а другой инверсный) через плавкие перемычки. Если целы обе перемычки для какого-нибудь терма, то этот терм никогда не будет выбран, т.к. одна из двух линий всегда будет в состоянии "НОЛЬ". Эта ситуация обозначается знаком "0".

Если удалена одна перемычка, например, соединяющая инверсную линию, то поданный на вход сигнал высокого уровня может активизировать терм. Эта ситуация обозначается символом "Н". В противном случае (удалена "прямая" перемычка) ситуация обозначается символом "L". Если удалить обе перемычки, то активность терма не будет зависеть от данного входа. В этом случае ставят знак "--".

Теперь мы уже сможем понять половину таблицы 1. Например, посмотрим, как подключен ко входам ПЛМ терм N4. Предварительно заметим, что вход 0 соответствует самому правому символу в терме, а вход 15 - левому. Легко определяем из таблицы, что к четвертому терму подключены 2 входа: вход 14 (удалена инверсная перемычка) и вход 10 (удалена прямая перемычка). Остальные входы отключены от четвертого терма.

Выход с каждого терма (всего 48 выходов) можно подать

через плавкие перемычки в различных комбинациях на матрицу "OR", состоящую из 8 линий по 48 входов каждая (одна линия соответствует одному выходу). Наличие входной перемычки обозначается символом "А", отсутствие символом ".". Добавив, что выход 0 соответствует самому правому символу таблицы для матрицы "OR", а выход 7 самому левому, предлагаем вам убедиться, что формулы соответствуют таблице.

Чтобы закончить с матрицами, осталось разобраться с одной строчкой таблицы перед термом ноль. Дело в том, что выход матрицы "OR" поступает на выход ПЛМ через элемент "исключающее или". Второй вход элемента соединен через плавную перемычку с общей шиной. Если эта перемычка цела, элемент не меняет полярность входного сигнала, т.е. активное состояние выхода - "1". Такая ситуация обозначается символом "Н". В противном случае активное состояние выхода - "0" и используется символ "L". Таким образом первая строчка таблицы кодирует полярность выходных сигналов ПЛМ и обозначается "POL" (полярность).

Вот теперь мы знаем о ПЛМ все, верней почти все. Как видно, вышеприведенные формулы для сигналов и карта прошивки просто разные языки описания одних и тех же процессов.

Чтобы закончить рассмотрение работы синхронизатора, нам осталось рассмотреть формирователь сигналов "F1" и "F2" частотой 2.5 Мгц и амплитудой 12 вольт для микропроцессора КР580ВМ80А, формирователь сигналов выбора строк "RAS" и столбцов "CAS" частотой 2.5 Мгц для динамического ОЗУ и ГЗУ (графического ЗУ) и формирователь тактовых сигналов для контроллера НГМД частотой 4, 2 и 1 Мгц.

Мы надеемся, что читатель стал уже настолько опытен, что в состоянии сам разобраться в работе этих формирователей.

Для подсказки сделаем небольшое замечание по поводу режима работы микросхем D25 и D72. Они включены, как D - триггер (на входы S и R подана одновременно "1"). С приходом положительного фронта тактирующего импульса на вход С данные с входа D поступают на прямой и инверсный выходы триггера и сохраняются на них до прихода следующего положительного перепада тактового сигнала. Те читатели, у которых возникли трудности в этом месте, могут посмотреть рисунки 3 и 4.

Разобравшись с синхронизатором, займемся теперь процессором.

### 3. ПРОЦЕССОР

На листе 3 показана схема процессора. Процессор имеет шестнадцатиразрядную шину адреса, двунаправленную восьмиразрядную шину данных и несколько входных и выходных управляющих сигналов. На входы F1 и F2 процессора поступают из синхронизатора синхроимпульсы. Эти синхроимпульсы мы уже знаем, они задают тактовую частоту процессора.

Микропроцессор выполняет команды по машинным циклам. Число циклов, необходимое для выполнения команды, зависит от ее типа и может быть от одного до пяти. Машинные циклы выполняются по машинным тактам. Число тактов в цикле также определяется кодом выполняемой команды и может быть от трех до пяти. Длительность такта равна периоду тактовой частоты сигналов F1 и F2, т.е. 400 нсек. Начало такта определяется положительным фронтом импульса F1.

Работа микропроцессора управляется входными и выходными сигналами. Входные сигналы следующие:

- READY - готовность
- HLD - захват
- INT - запрос прерывания
- RESET - установка в исходное состояние

Выходные сигналы:

- WAIT - ожидание
- HLDA - подтверждение захвата
- SYNC - сигнал синхронизации
- DBIN - сигнал чтения
- WR - сигнал записи (активн. уровень низк.)

Дотошный читатель уже сообразил, что некоторые управляющие сигналы идут в "никуда", т.е. только на разъем XS3. Разъем XS3 введен в схему для возможности подключения ПЭВМ к тестируемому оборудованию. При нормальной работе ПЭВМ он не используется (к нему ничего не подключается).

Для того, чтобы управляющие сигналы не терялись и "КОРВЕТ" мог нормально работать, на разъеме XS3 должны быть установлены пять перемычек, замыкающие попарно следующие сигналы:

1SYNC	-	SYNC
1WR	-	WR
1READY	-	RADY
1DBIN	-	DBIN
1CSROM1	-	CSROM1

Перед тем, как перейти к дальнейшему анализу схемы, сделаем несколько общих замечаний о работе процессора.

В начале каждого машинного цикла процессор вырабатывает сигнал SYNC длительностью равной периоду тактовой частоты (в нашем случае 400 нсек.). Точное время появления сигнала SYNC зависит от конкретной микросхемы. Оговаривается, что SYNC

может появиться не позже, чем через 120 нсек. после переднего фронта сигнала F2.

Сигнал RESET переводит процессор в исходное состояние, снятие сигнала запускает процессор из исходного состояния, т.е. процессор начинает выполнение первой команды с первого цикла и первого такта (T1) в первом цикле. Длительность сигнала RESET должна быть не менее трех периодов тактовой частоты.

В такте T1 каждого машинного цикла процессор выдает на адресную шину адрес ячейки памяти, к которой он хочет обратиться (после сигнала RESET адрес ячейки будет 0000H). В этом же такте процессор выдает на шину данных слово состояния, которое "говорит" о том, что собирается делать процессор в данном цикле.

В такте T2 процессор принимает информацию по шине данных и анализирует состояние сигнала READY. Если READY = "1", то процессор продолжает работу и переходит к такту T3, если READY = "0", то процессор переходит в режим "Ожидание" и формирует сигнал WAIT высокого уровня (WAIT = "1").

В такте T3 при условии, что READY = "1", процессор передает информацию по шине данных и анализирует сигнал HOLD. Если HOLD = "1", то по окончании T3 процессор переходит в состояние захвата и подтверждает это высоким уровнем сигнала HLDA (HLDA = "1"). При этом процессор "отключается" (переходит в высокоомное состояние) от шин данных/адреса и переводит выходные управляющие сигналы, кроме HLDA и WR, в состояние низкого уровня.

Если процессор смог благополучно выполнить такт T3 первого цикла команды, то он переходит ко второму машинному циклу. И так такт за тактом, цикл за циклом, команда за командой процессор выполняет программу.

У читающего должен возникнуть естественный вопрос, как разобраться в такой путанице, поскольку произвольная команда может выполняться и за один цикл и за пять, а цикл может занимать от трех до пяти тактов. Правда есть сигнал SYNC, привязанный к началу каждого цикла, но этого, конечно, мало.

Выше мы упомянули о том, что в первом такте каждого машинного цикла процессор выдает на шину данных слово состояния, которое кодирует тип цикла. Число типов машинных циклов десять (M1 - M10).

M1 - прием кода команды в регистр команд.

M2 - цикл чтения ЗУ по содержимому программного счетчика или содержимому одного из регистров BC, DE, HL.

M3 - цикл записи в ЗУ по содержимому программного счетчика или содержимому одного из регистров BC, DE, HL.

M4 - цикл чтения стека по содержимому указателя стека.

M5 - цикл записи в стек по содержимому указат. стека.

M6 - цикл ввода информации в аккумулятор из внешнего устройства.

M7 - цикл ввода информации в аккумулятор из внешнего устройства.

M8 - цикл прерывания (прием кода команды RST или CALL из контроллера прерываний).

M9 - цикл останова

M10 - цикл прерывания при останове (прием кода команды RST или CALL при выводе микропроцессора из режима "Останов" по прерыванию).

Таким образом в такте T1 каждого машинного цикла сообщается о типе выполняемого цикла. Мы приводим наименования сигналов слова состояния и соответствие их машинным циклам в таблице 2.

Вот теперь мы достаточно подготовлены, чтобы обратить свой взор на лист 3 альбома схем.

Прежде всего отметим, что в формировании управляющих сигналов участвуют разряды D0, D1, и D5 шины данных. Как вы уже знаете, в "Корвете" обращение к устройствам ввода/вывода (YBB) происходит как к ячейкам памяти. Поэтому команды IN и OUT не используются, и связанные с их выполнением циклы M6 и M7 нас интересовать не будут. Нас так же не будут интересовать циклы, связанные с разрядами D2, D3, D4, D6 и D7.

Внимательно посмотрев на таблицу 2, мы можем сказать, что нам нужно рассмотреть циклы M1-M5, M8 и M10. Причем, с точки зрения схемотехники "Корвета", M2 совпадает с M4, а M3 с M5. На рис. 5 приведены временные диаграммы управляющих сигналов для циклов M1, M2 и M3. Прокомментируем коротко эти сигналы.

Сигналы SINC, DBIN и WR формируются по фронту F2 в тактах T1, T2, T3 соответственно и имеют длительность 400 нсек. Слово состояния на шине данных сопровождается сигналом SINC. Адрес на шине адреса сохраняется до следующего цикла.

Сигнал ACCESS формируется с помощью сдвигового регистра D27, который работает следующим образом. Если на входе EWR сигнал высокого уровня, то данные загружаются в регистр с параллельных входов 0, 1, 2, 3 по отрицательному перепаду на входе C. Низкий уровень сигнала на входе EWR вызывает загрузку данных с последовательного входа 0 --> (1 ножка) синхронно с каждым отрицательным перепадом на входе C. При этом на выходе 0 (13 ножка) появляется сигнал, соответствующий последовательному входу, на выходе 1 (12 ножка) появляется сигнал, сдвинутый с выхода 0, на выходе 2 сигнал, сдвинутый с выхода 1, а на выходе 3 с выхода 2. Таким образом данные, загружаемые с последовательного входа, сдвигаются относительно выходов регистра. Этой информации достаточно, чтобы построить все временные диаграммы на рисунке 5.

Обратите внимание, что сигнал доступа к памяти ACCESS при чтении формируется в такте T2, а при записи в такте T3 и практически совпадает по времени с сигналом WR.

Рассмотрим теперь сигнал lREADY, который через разъем XS3 поступает на вход RA (готовность) процессора. Он формируется на выходе микросхемы D24 (ИЛИ). На вход D24 подаются два сигнала CSDEV и WAIT. CSDEV - это сигнал

обращения к UBB (при обращении он имеет низкий уровень). Когда обращения к UBB нет, то на выходе lREADY сигнал высокого уровня и процессор, анализируя этот сигнал в такте T2, не переходит в режим ожидания. При обращении к UBB низкий уровень сигнала CSOEV появляется на выходе D24 и процессор переходит в режим ожидания, устанавливая на выходе WI сигнал WAIT высокого уровня. Сигнал WAITE формирует сигнал lREADY высокого уровня, поэтому, пропустив после такта T2 один такт – такт ожидания, процессор продолжает работу с выполнения такта T3. Таким образом при обращении к UBB процессор автоматически вставляет один такт ожидания между тактами T2 и T3. Эта хитрость применена для надежного обращения к UBB. На рисунке 5 такт ожидания не показан. Заметим только, что в случае появления такта ожидания сигнал DBIN автоматически расширяется до 800 нсек.

Еще одно замечание по поводу обращения к UBB. Связь с UBB организована через двунаправленный буфер D21 (лист 3).

При отсутствии обращения к UBB высокий уровень на входе EZ микросхемы D21 удерживает ее входы/выходы в состоянии высокого импеданса. В машинном цикле ввода информации (M3) сигнал DBIN через инвертор D26 (3.23) поступает на вход управления передачи и переводит D21 в режим ввода информации в процессор. Если сигнал DBIN отсутствует, то при выборе микросхемы она переключается в режим вывода информации.

Нам осталось разобраться с обработкой прерываний, т.е. с сигналом INTA. При формировании контроллером прерываний D8(12.4) сигнала INT процессор заканчивает выполнение команды и начинает выполнять трехбайтную команду "CALL", которую он принимает из контроллера прерываний, при условии, что прерывания разрешены командой "EI".

Как это делается? Попробуем разобраться без наглядной иллюстрации. В первом цикле процессор выполняет цикл M10, который отличается от M1 наличием единицы в разряде D0 шины данных. Кроме этого вводится такт ожидания, потому при обращении к контроллеру прерываний вырабатывается сигнал CSDEV. Положительным фронтом сигнала на тактовом входе С триггера D15 (3.25) единица в D0 устанавливает выход триггера D15 (5 ножка) в единицу. На рисунке 5 сигнал на тактовом входе D15 показан.

Изменить состояние D15 теперь можно двумя способами. Первый: с приходом следующего положительного перепада на тактовый вход сигнал на линии D0 должен низкий уровень. Но такая ситуация может наступить после выполнения полностью команды "CALL". Второй способ: подать на вход R (RESET) сигнал низкого уровня. Из рисунка 5 видно, что это возможно при выполнении цикла записи (M3 или M5).

Итак, триггер D15 находится в состоянии единица. Сигнал высокого уровня с выхода триггера поступает на дешифратор адресов, запрещая его работу, и на микросхему D18 (3.27) – И. Поэтому сигнал INTA будет инвертированным сигналом DBIN. Сигнал INTA выбирает контроллер прерываний и в цикле M10

в процессор вводится код команды CALL из контроллера.

Команда "CALL" выполняется за пять машинных циклов, из которых первые три связаны с вводом команды в процессор, а два последних с записью в стек текущего адреса прерванной команды. Первый цикл (M10), мы уже разобрали. Затем выполняется два раза подряд цикл M8, отличающийся от M2 только единицей в разряде D0 шины данных. В первом цикле M8 читается младший байт адреса, а во втором цикле M8 – старший байт адреса перехода на программу обработки прерываний. В конце цикла M8 будет вырабатываться сигнал INTA, который и обеспечивает чтение из контроллера.

В четвертом и пятом циклах будет выполняться цикл M5, обеспечивающий запись в стек старшего и младшего байта адреса прерванной программы. Отрицательный сигнал с выхода регистра D27 (3.26) сбросит триггер D15 в нуль. Поэтому в четвертом и пятом циклах команды CALL сигнал INTA формироваться не будет.

Кажется нам удалось обойтись без временных диаграмм. Чтобы совсем закончить с листом 3, добавим несколько слов о кнопке SA1, с помощью которой формируется сигнал RESET. Главным элементом в цепи формирования сигнала RESET является микросхема D19 (3.9). Это триггер Шмидта, который обладает гистерезисными свойствами. Поэтому через 0,5 сек после включения питания, или отпускания кнопки RESET, сигнал скачком принимает значение низкого уровня и разрешает процессору работать.

Теперь самое время перейти к схеме дешифрации адресов и задания конфигурации памяти. Смотрите на листе 5 микросхемы D22(5.4), D24(5.5) D23(5.11), D31(5.8), D43(5.12), D24(5.16). Процессор имеет диапазон адресов 0000H – FFFFH, т.е. 64 К, а "Корвет" может иметь: ПЗУ – 96 К, ОЗУ – 64 К, АЦЗУ – 1К, ГЗУ – 192 К и UBB – 2К, т.е. 355 К. По желанию программиста к процессору можно подключить некоторую комбинацию частей разных видов памяти.

Рассмотрим механизм переключения адресов. Но сначала разберемся, как работают микросхемы D23 и D43. Это дешифраторы. Начнем с дешифратора K555ИД4. Он состоит из двух дешифраторов. Первый дешифратор имеет выходы 0.0, 1.0, 2.0, 3.0, второй – 0.1, 1.1, 2.1, 3.1. Двухразрядный адресный вход 2.1 подключен к обоим дешифраторам. Каждый дешифратор имеет свои управляющие входы, первый – &E0, второй – &E1.

Если работа дешифратора разрешена, то двоичный код на входе (вход 1 младший разряд) преобразуется в сигнал на линии с номером, равным входному коду, только в инверсном виде. Когда дешифратор отключен, на выходных линиях присутствуют сигналы высокого уровня. Управление дешифратором производится следующим образом:

Для первого дешифратора: дешифратор выключен, если сигнал на ножке 1 высокого уровня, а на ножке 2 низкого. Эта ситуация отражена на схеме:

(1) and (not(2)) = "1"

Для второго дешифратора, пользуясь схемой мы можем написать следующее условие разрешения работы:

not (14) and (not(15)) = "1"

Работу D43 можно охарактеризовать и другими словами. Например, пусть WERG – сигнал разрешения первого дешифратора. Тогда сигнал 1WR демultipлексруется в соответствии с адресами на адресных входах. Очевидно, что можно назвать сигнал 1WR разрешением, тогда демultipлексироваться будет сигнал WERG. Что же происходит в нашем случае? Ответ зависит от контекста схемы, т. е. от того, что она делает. Как раз к этому мы собираемся приступить. Но закончим с дешифраторами.

Как работает K555ИД7? Вы уже сами можете ответить на этот вопрос, глядя на условное обозначение дешифратора на схеме. Выходные линии помечены кружком, значит выходы инверсные. Адресный вход имеет три линии, а выходных восемь. Следовательно трехразрядный код преобразуется в напряжение низкого уровня на выходной линии, соответствующей коду. Управление дешифратором производится по трем входам и по формуле

(6) and (not(4)) and (not(5)) = "1"

Кстати, теперь можно понять, как сигнал высокого уровня с триггера D15 (3.25) отключает дешифраторы.

Итак, как работает механизм переключения памяти? Основным элементом является ПЛМ D31, в которой "защиты" различные варианты карты памяти. Карту прошивки ПЛМ D31 мы приводим в таблице 3.

Использование регистра D22 на входе ПЛМ обеспечивает переконфигурацию памяти в процессе выполнения программы. Регистр D22 и есть "знаменитый" системный регистр, с которым работают программисты. Запись данных в системный регистр осуществляется с шины данных положительным сигналом выборки регистра SELREG. Сигнал RESET сбрасывает регистр в нуль.

Маленькое замечание: сигнал SELREG попадает на регистр через элемент "ИЛИ" – D24 (5.5), на вход которого заведен также сигнал A7. Поэтому выбор регистра осуществляется, когда на адресной линии A7 низкий уровень напряжения. Обратите внимание, что на входы ПЛМ поступают только старшие восемь разрядов с шины адреса, поэтому сигнал SELREG никак не связан с младшими адресными линиями. Теперь мы можем объяснить, почему младший байт адреса системного регистра 7FH. Просто для адресации регистра на линии A7 должен быть нуль. Поскольку к другим адресным линиям могут быть подключены другие устройства, поэтому для устранения путаницы на остальных линиях нужно установить сигналы высокого уровня. Это как раз и будет код

7FH.

Мы уже обладаем достаточной информацией, чтобы по карте прошивки ПЛМ – D31 составить варианты карты адресов и определить соответствующее каждому варианту число в системном регистре. Работа эта нудная и большой необходимости в ее продельвании нет. Ответ, который мы получим, если сделаем такую работу, должен совпасть с результатами предыдущей главы.

Выходы из ПЛМ можно почти все проследить на листе 5. Исключение составляет сигнал CSKB – сигнал обращения к клавиатуре.

При обращении процессора к UBB на входе 4 ПЛМ появляется сигнал CSDEV, который разрешает работу дешифратора D23 (5.11). В зависимости от состояния разрядов адресной шины A3, A4 и A5 на выходах дешифратора будут формироваться сигналы выборки адресуемого UBB. Ниже мы приводим перечень устройств ввода-вывода:

CSCT – таймер (00H)  
SCIOP3 – параллельный интерфейс N 3 (08H)  
SCIOS1 – последовательный интерфейс N 1 (10H)  
CSCOM – контроллер НГМД (18H)  
CSIOS2 – последовательный интерфейс N 2 (20H)  
CSINA – контроллер прерываний (28H)  
CSIOP2 – параллельный интерфейс N 2 (30H)  
CSIOP1 – параллельный интерфейс N 1 (38H)

В скобках приведены младшие байты адреса. Вы сами легко можете проверить правильность адресов. Например: сигнал CSCOM выходит с линии 3, т. е. на входе дешифратора был код 3. Чтобы получить настоящий адрес нужно код сдвинуть на три разряда влево, что эквивалентно умножению на восемь, т.е. получаем 24D или 18H.

Заканчивая с работой дешифратора, перечислим сигналы, формируемые дешифратором D43(5.12):

0.0 WE – сигнал записи в ОЗУ  
1.0 WEV – сигнал записи в АЦЗУ  
2.0 WEG – сигнал записи в ГЗУ  
3.0 – не используется  
0.1 – сигнал чтения ОЗУ (активный уровень низкий)  
1.1 RV – сигнал чтения АЦЗУ  
2.1 RG – сигнал чтения ГЗУ  
3.1 – сигнал "складывается" с сигналом CSDEV на элементе "ИЛИ" – D24, на выходе которого формируется сигнал RDEV – сигнал чтения UBB.

Какова же длительность этих сигналов? Из временных диаграмм работы процессора (рис.5) мы помним, что адрес держится на шине несколько тактов, а управляющие сигналы 1DBIN и WERG имеют длительность порядка такта. Поэтому можно

говорить, что сигналы 1DBIN и WERG демultipлексуются в соответствии с кодом на адресных входах дешифраторов D23 и D43. Но это уж кому как нравится. Можно демultipлексировать, а можно и дешифровать. Главное здесь в том, что длительность сигналов записи с выходов дешифратора D43 равна длительности сигнала WERG, а длительность сигналов чтения равна длительности сигнала 1DBIN.

На этом мы закончим с дешифратором и перейдем к ОЗУ.

#### 4. ОЗУ

В "Корвете" ОЗУ занимает 64К и выполнено на восьми микросхемах КР565РУ5Г динамического типа D76 - D83 (4.1 - 4.8). Схема ОЗУ приведена на листе 4.

Как обычно сделаем несколько предварительных замечаний. Микросхема КР565РУ5Г имеет объем памяти 64К бит. Поэтому, чтобы организовать ОЗУ на 64К байт, используется восемь микросхем.

Такие микросхемы имеют весьма малый размер, но за это приходится платить тем, что каждые 30 миллисекунд нужно восстанавливать или, как обычно принято, регенерировать содержимое микросхемы. Как нужно работать с микросхемой? Вы уже заметили, что адресных входов только восемь. Нам же нужна адресация в пространстве адресов 64 К. Микросхема организована в виде матрицы однобитовых ячеек памяти размером 256x256. Матрица имеет восемь входов для выбора строк и восемь входов для выбора столбцов. Чтобы выбрать нужную ячейку сигнала, на адресный вход подается восемь младших разрядов адреса для выборки нужной строки и код адреса запоминается по спаду сигнала RAS. Затем на эти же входы подаются восемь старших разрядов адреса для выборки столбца и запоминается по спаду сигнала CAS. Микросхема ОЗУ работает в одном из трех режимов: запись, чтение и регенерация. Чтобы дальше обсуждать работу ОЗУ, нам нужно теперь разобраться с механизмом адресации строк и столбцов. Этим занимаются четыре мультиплексора D52 - D55 (смотри лист 5).

Микросхема К555КП12 представляет из себя два мультиплексора. Поэтому мы имеем восемь выходов с четырех мультиплексоров, как раз столько, сколько нужно для адресации строки или столбца. Управляются мультиплексоры сигналами F2T и ACCESS, которые задают двухразрядный код (ACCESS- младший бит). Микросхемы все время выбраны, т. к. входы разрешения EZ0 и EZ1 заземлены. Поэтому в любой момент времени к выходу подключена входная линия с номером, равным коду на входе управления.

Теперь вы можете сами сообразить какие сигналы приходят на адресные входы ОЗУ. На рис. 6 показаны временные диаграммы сигналов F2T и ACCESS, MA0, RAS и CAS, и показаны моменты времени, когда происходит выборка строки и столбца. Что происходит на остальных выходах мультиплексоров (сигналы MA1 - MA7), мы приводим в таблице 4.

Теперь, глядя на таблицу 4 и рис. 7, попытаемся понять как работает ОЗУ. Если мы не обращаемся к памяти, то сигнал ACCESS отсутствует и в каждом такте на адресных входах ОЗУ присутствуют синхросигналы C1 - C13. В соответствии со значениями этих сигналов производится выборка ячейки памяти для регенерации. Как происходит регенерация? Она производится автоматически после выбора ячейки памяти.

Когда мы рассматривали синхронизатор, то подробно



анализировали временные диаграммы синхросигналов. Теперь нетрудно сообразить, что за период сигнала C13, равный 20 мсек., последовательно будут выбраны все ячейки ОЗУ по семи адресным линиям. Почему семь? Так уж устроена микросхема. Нетрудно сообразить, сканирование по восьми адресным линиям потребует слишком много времени и данные за это время не сохранятся.

Таким образом, когда к ОЗУ не обращаются, все время происходит регенерация. Если обращение к памяти происходит, то появляется сигнал ACCESS и тогда на ОЗУ подается адрес с шины адреса. При записи данные через вход D записываются сигналом WE. При чтении данные с выхода ОЗУ по сигналу WERG проходят на выход регистра D74(5.14) и защелкиваются в нем по спаду сигнала WERG. Процессор в это время по сигналу DBIN принимает данные, установленные регистром D74. После прекращения сигнала DBIN регистр D74 переходит в высокоомное состояние.

Остается только добавить, что адресный мультиплексор обслуживает также и ГЗУ.

Разобравшись с работой ОЗУ у нас не должно возникнуть проблем с работой ПЗУ.

## 5. ПЗУ

ПЗУ содержит три микросхемы емкостью по 8 К.

## 6. А/Ц ДИСПЛЕЙ

В предыдущей главе мы уже рассказывали об устройстве а/ц дисплея. Под этим термином понимается весь комплекс аппаратуры, обеспечивающий вывод изображения на экран телевизора, включая и сам телевизор – монитор. Мы обсудим только схему, занимающуюся формированием видеосигнала. Такое устройство обычно называется контроллером отображения алфавитно-цифровой информации. То, что происходит с видеосигналом в телевизоре (мы будем также употреблять термины дисплей и монитор), мы не рассматриваем. Для нас вполне достаточно знать, что в дисплее осуществляется растровый процесс получения изображения. В качестве дисплея для "Корвета" годится любой бытовой телевизор или специальный монитор, работающий в телевизионном стандарте.

Итак, контроллер а/ц дисплея должен вырабатывать видеосигнал, обеспечивающий на экране изображение размером 512x256 точек. Каждый символ изображается с помощью матрицы 8x16 точек. Всего на экране размещается 64x16 знакомест.

Мы уже знаем, что контроллер имеет ОЗУ, емкостью 1К x 9 бит, которое еще называется АЦЗУ – Алфавитно-Цифровое Запоминающее Устройство. Схема АЦЗУ приведена на листе 7 и состоит из девяти микросхем KP132PY4. Каждая KP132PY4 – это ОЗУ статического типа (не заботимся о регенерации данных!) с организацией 1 К x 1 бит. Микросхема имеет десять адресных входов, на которые подаются сигналы VA0 – VA9, один вход данных (D), один выход (Z) и два управляющих входа (EZ и ERD/ERW). При EZ = "1" микросхема находится в режиме хранения. В этом режиме состояния адресных входов и входа данных безразличны, выход находится в высокоимпедансном состоянии (Z- состоянии). При EZ = "0" (сигнал выборки микросхемы) в зависимости от сигнала ERD/EWR микросхема находится в режиме записи или чтения. При ERD/EWR = "0" режим записи. На адресных входах установлен адрес записываемой ячейки. Значение на входе данных записывается в ячейку. Выход в Z-состоянии. При ERD/EWR = "1" режим чтения. На адресных входах адрес считываемой ячейки. Состояние входа данных безразлично. На выходе содержимое адресуемой ячейки.

Этой информации вполне достаточно, чтобы самому построить временные диаграммы работы АЦЗУ. Но сначала разберемся с адресными сигналами, которые поступают с выхода десяти мультиплексоров, реализованных с помощью трех микросхем K555KP11 (D49-D51) (см. лист 6).

У нас уже имеется некоторый опыт в понимании работы мультиплексоров. Нетрудно сообразить, что мультиплексируются десять младших адресных линий (A0-A9) и сигналы C0-C5, C10-C13. Например, на выходы VA0 приходит сигнал A0 или C0 в зависимости от уровня сигнала SE1/SE0. Так же легко сообразить, что микросхемы выбраны все время, поскольку входы EZ заземлены. И,

наконец, тоже очевидно, что при SE1/SE0 = "0" к выходу подключается

линия 0, а при SE1/SE0 = "1", линия 1.

Сигнал WERG, поступающий одновременно на входы SE1/SE0 всех микросхем, мы уже знаем. Сигнал WEV (сигнал записи в АЦЗУ) тоже нам уже знаком. А вот про сигнал CSV мы только знаем, что он вырабатывается в ПЛМ D40 (2.9). Но не зря же столько усилий было потрачено на устройство ПЛМ. Возвращаемся назад и выписываем его логическую формулу:

$$\text{CSV} = 10M*5M*2.5M*1.25M + \\ + 10M*5M*2.5M*ACCESS$$

Теперь читателю предлагается самому построить временные диаграммы сигналов CSV, WEV, VA0, VD0, RV, D0 и убедиться, что рисунок 7 соответствует действительности. Не поленитесь еще раз посмотреть на рисунок 6 (цикл чтения).

Разумеется, у вас возник вопрос о периодической компоненте сигнала SCV, не связанной с появлением сигнала ACCESS, т.е. с режимом записи или чтения, и о сигналах C0-C5, C10-C13. Это связано с режимом отображения содержимого АЦЗУ на экране. Прежде, чем перейти к схеме отображения информации, разберемся с битом атрибута.

Из предыдущей главы мы уже знаем, что битом атрибута управляют разряды 4 и 5 регистра управления (относительный адрес регистра управления 3AH), а состояние бита атрибута фиксируется в разряде 3 регистра с относительным адресом 38H. В схеме состоянием бита атрибута управляют сигналы INVON (установка бита) и INVOFF (сброс бита). Убедитесь, что они поступают с порта программируемого регистра D17 (13.35) и что в предыдущей главе информация дана правильно (сброс - бит 4, установка - бит 5).

Теперь проверим по схеме, как действует нехитрый комментарий по поводу этих битов в предыдущей главе. Сигналы INVON и INVOFF поступают на S и R входы триггера инверсии D75 (6.6).

С таким триггером мы уже имели дело. При подаче сигнала низкого уровня на вход S, триггер устанавливается в единицу, т. е. INVON = "0" устанавливает бит атрибута равным единице. Сигнал INVOFF = "0" приходит на R вход триггера и сбрасывает его в ноль, т. е. бит атрибута в этом случае равен нулю. Ситуация, когда на входах R и S сигналы низкого уровня, запрещена для триггера TM2. Если на R и S входах сигналы высокого уровня, то по положительному фронту сигнала на C входе данные с D входа поступают на выход. На C вход с мультиплексора D49 приходит сигнал RV (сигнал чтения АЦЗУ) и защелкивает считанный из АЦЗУ бит атрибута в триггере инверсии.

Мы получили те же самые правила управления битом атрибута, что и в предыдущей главе. Более того, можно добавить, что запись в АЦЗУ при установленных в единицу

сигналах INVON и INVOFF может привести к непредсказуемому результату, т. к. триггер инверсии может оказаться в любом состоянии. Чтобы этого избежать, сигнал CDI с выхода триггера, кроме АЦЗУ поступает в порт A (бит 3) программируемого регистра D17 (13.35). Теперь у программиста есть возможность в сомнительных случаях считать состояние триггера инверсии.

Наконец мы можем поговорить о схеме отображения информации, которая размещена на листе 6 и состоит из регистра D60 (6.4), знакогенератора D59 (6.8), сдвигового регистра D73 (6.12), элемента "исключающее или" D71 (6.10) и смесителя информационного сигнала BWV с синхросигналом HVSNZNC, выполненного на транзисторе VT3 (6.19).

Выходной видеосигнал VIDCOMP с транзистора поступает на разъем XS7 (16.3), куда должен подключаться черно-белый монитор, называемый иногда ВКУ (Видео-Контрольное Устройство).

Как происходит формирование видеосигнала? Для этой цели служит сдвиговый регистр D73 (6.12). Работает он следующим образом.

При высоком уровне сигнала на входе E-->/EVR данные из регистра выдвигаются последовательно на выход синхронно с передним фронтом тактирующего сигнала на входе C. При низком уровне сигнала E-->/EVR восьмиразрядный код из знакогенератора по положительному фронту сигнала на входе C загружается в регистр. Вход DEC регистра заземлен, поэтому в режиме сдвига первым будет выдвигаться старший бит 7 загруженного кода, остальные биты сдвигаются, а в освободившийся младший разряд загружаются единицы с входа 0-->.

Что же происходит на самом деле? Для этого нам нужны временные диаграммы управляющих работой регистра сигналов VCLK и SHIFT. Логические формулы этих сигналов мы уже знаем:

$$\text{VCLK} = 10M*MODSEL + 10M*5M*MODSEL$$

$$\text{SHIFT} = \text{not}((\text{HBL}*5M*2.5M*\text{not}(1.25)*MODSEL*\text{VBL}) + \\ + (\text{HBL}*5M*2.5M*1.25M*MODSEL*\text{VBL}*C0))$$

Сигналы, из которых комбинируются VCLK и SHIFT мы знаем, кроме сигнала MODSEL. Это сигнал режима отображения. MODSEL = "0" - режим 64 символа в строке, MODSEL = "1" - режим 32 символа в строке. Этим сигналом программист может управлять с помощью уже известного нам регистра управления. Давайте сначала разберемся, что происходит в режиме 64 символа в строке, поэтому будем считать, что MODSEL = "0", тогда:

$$\text{VCLK} = 10M$$

$$\text{SHIFT} = \text{not}(\text{HBL}*5M*2.5M*1.25*\text{VBL})$$

Сделаем еще упрощение. Импульс HBL (строчный гасящий импульс) сформирован для того, чтобы его низкий уровень гасил

луч на экране при переходе на новую строку. Аналогично действует сигнал VBL (кадровый гасящий импульс), только при переходе к новому кадру. Низкие уровни сигналов HBL и VBL на обратном ходе луча будут устанавливать сигнал SHIFT в единицу, тем самым блокируя загрузку кода в регистр D73. Поэтому мы рассмотрим временные диаграммы работы сдвигового регистра для сигнала

$SHIFT = \text{not}(5M \cdot 2.5 \cdot 1.25M)$

После сделанных упрощений вы сами легко постройте временные диаграммы сигналов VCLK и SHIFT. Для ленивых читателей диаграммы приведены на рисунке 8. С помощью комментария к работе сдвигового регистра Вы сможете понять, что происходит на его выходе. С выхода регистра сигнал поступает на микросхему D71 (6.10). Это элемент "исключающее ИЛИ". На выходе этого элемента высокий уровень возникает тогда и только тогда, когда на выходах сигналы имеют разные уровни. Поэтому бит атрибута с выхода триггера D75 (6.7) по положительному фронту сигнала SHIFT записывается в триггер D72 (6.9), выход которого подключен к одному из выходов элемента D71. Если бит атрибута установлен в ноль, то сигнал с выхода регистра D73 без искажения переходит на выход, если бит атрибута установлен в единицу, то сигнал BWVC выхода D71 будет инвертирован относительно входного сигнала и на экране дисплея будет инверсное изображение символа.

Далее сигнал BWV поступает на эмиттерный повторитель, реализованный на транзисторе VT3 (6.19). На этом транзисторе сигнал BWV смешивается с синхросигналом HVSYNС и получается видеосигнал VIDCOMP.

Нам осталось в работе контроллера АЦЗУ понять, как происходит выборка кода из знакогенератора для загрузки в сдвиговый регистр и как нужно программировать знакогенератор, чтобы на экране отображались красивые цифры и буквы.

В общих чертах задача ясна. Нужно считать код символа из АЦЗУ и подать его на адресные входы знакогенератора. Для представления кода символа нужно восемь разрядов, а ПЗУ знакогенератора имеет тринадцать адресных входов. Куда же подать код символа? Давайте вспомним, что на изображение символа необходимо 16 строк, которые адресуются четырьмя линиями. После некоторых размышлений нетрудно сообразить, что 4 младших разряда адреса должны использоваться для выбора номера строки символа. Следующие 8 разрядов должны использоваться для задания кода символа. Остается еще один старший разряд адреса. Но, если Вы помните, в знакогенераторе находятся два набора символов. Они выбираются старшим разрядом адреса, на который поступает сигнал FONT с регистра управления D17 (13.35) (бит 2).

Нам осталось понять какими сигналами производить выборку АЦЗУ. Обратимся к схеме на листе 6. Мы уже знаем, что

в отсутствии сигнала записи WEV в АЦЗУ, т.е. когда WEV имеет высокий уровень, АЦЗУ находится в режиме чтения, а читаемый адрес определяется сигналами C0-C5 и C10-C13. Сигналы C0-C5 выбирают шесть младших разрядов адреса АЦЗУ.

Давайте займемся арифметикой. Период сигнала C0 = 1.6 мксек, т.е. 0.8 мксек младший разряд адреса АЦЗУ равен нулю, следующие 0.8 мксек – единица, еще через 0.8 мксек опять нуль и т.д.. Следовательно, младший адрес A0 сканируется с периодом 0.8 мксек, A1 с периодом 1.6 мксек и т.д., т.е. сканирование памяти идет с той же частотой, что и загрузка бита в сдвиговый регистр. Остается добавить, что шесть младших разрядов адреса АЦЗУ как раз определяют номер знакоместа в строке. (от 0 до 63).

Четыре сигнала C6 – C9 для маркирования АЦЗУ не используются. Они определяют текущий номер строки в символе и заведены на младшие адресные входы знакогенератора. Сигналы C10 – C13 определяют номер знакоместа по вертикали (от 0 до 15) и поступают на старшие четыре адресных входа АЦЗУ.

При сканировании АЦЗУ данные, т.е. код символа, защелкиваются в регистре D60 (6.4) положительным фронтом сигнала LOADPOM. На рис.9 без комментариев приводятся необходимые временные диаграммы.

Давайте закрепим наши рассуждения и для примера запрограммируем латинскую букву "А" в знакогенератор. ASCII код буквы равен 41H. Нам нужно придумать где будут "гореть" точки, изображающие букву, определить содержание шестнадцати байт, отвечающих за изображение буквы и определить адрес каждого байта в ПЗУ знакогенератора. Для этого подготовим таблицу из восьми столбцов и шестнадцати строк. Строки нумеруем от нуля сверху вниз, а столбцы нумеруем от нуля справа налево. Тогда номер столбца будет соответствовать номеру бита в байте, а номер строки относительно адресу строки.

Заготовив такую таблицу, нарисуем в ней букву, т.е. поставим точки в клетках таблицы, чтобы получилась буква "А". Теперь против каждой строки выпишем содержимое байта. Каждая строка символа соответствует байту, хранящемуся в ПЗУ. Чтобы определить адрес каждого байта, смещаем код буквы в H-формате на одну позицию влево и в младший разряд добавляем номер строки.

Результат такой процедуры мы приводим в таблице 5. Если Вы хотите сделать в своем компьютере другие начертания букв, то Вы должны составить 512 таких картинок по 256 на набор и полученную информацию "зашить" в ПЗУ.

Нам осталось разобраться, как осуществляется процесс отображения расширенных букв (32 символа в строке). Мы уже знаем, что переключение производится сигналом MODSEL. При высоком уровне сигнала MODSEL, тактовая частота сигналов VCLK и SHIFT уменьшается в два раза, увеличивая время выдвигания байта данных из сдвигового регистра тоже в два раза. В этом случае половина ячеек АЦЗУ не отображается.

Теперь можно перейти к изучению контроллера отображения графической информации.

## 7. ГРАФИЧЕСКИЙ ДИСПЛЕЙ

Схема контроллера имеет ряд интересных решений, что делает графику "Корвета" достаточно быстрой и красочной.

Схема приведена на листах 8, 9 и 10. Основным узлом контроллера, как Вы уже видите, является ОЗУ, емкостью 192 К, которое мы будем называть графическим запоминающим устройством (ГЗУ).

С ОЗУ мы уже разбирались, поэтому в ГЗУ мы проследим только за управляющими сигналами. Регенерация данных в ГЗУ осуществляется точно так же, как в ОЗУ. Блок формирования изображения в принципе устроен так же, как в контроллере АЦЗУ за исключением специфики цветного изображения и еще ряда тонких моментов.

Давайте сначала разберемся с памятью. Как программисты, мы знаем, что к адресному пространству "Корвета" подключается всегда 16К ГЗУ. Этого как раз достаточно, чтобы отобразить на экране 512 \* 256 точек. Так же нам известно, что вся емкость ГЗУ разбита на плоскости по 64К каждая. В свою очередь каждая плоскость разбита на четыре страницы по 16К. Следовательно к адресному пространству подключается всегда одна из четырех страниц, при этом отображаться на экране может содержание другой страницы.

Попытаемся разобраться, как это происходит. Обратимся к листу 11. На нем представлена плоскость N2 ГЗУ, емкостью 64К. Нужно сообразить, что регистр D128 (11.7) защелкивает данные из ГЗУ, которые затем загружаются в сдвиговый регистр D129 (11.10) для формирования изображения.

Плоскости N0 и N1 ГЗУ устроены аналогично. Как происходит обращение к ГЗУ? Сигнал выборки строк нам уже знаком. Чем же отличается сигнал выборки столбцов CAS2 (на других плоскостях CAS0 и CAS1) от известного нам сигнала CAS. По схеме определяем, что эти сигналы идут с мультиплексора D118 (8.9) (лист 8). Мультиплексируются сигналы WR0, WR1, WR2 и GND (земля), которые идут с регистра D100 (8.1). Это как раз и есть знаменитый регистр цвета, содержание которого определяет режим работы графики.

Запись в регистр производится положительным фронтом сигнала на входе С. Убедитесь, что сигнал SELREG синхронизован с сигналом процессора 1WR (запись) и попадает на вход С регистра, если A6 = "0". Условие A6 = "0" как раз и определяет относительный адрес регистра цвета BFH (все восемь младших адресных линий, кроме шестой, должны иметь высокий уровень, а шестая - низкий). Вернемся опять к мультиплексору D118. Сигнал мультиплексирования есть результат операции "ИЛИ" над сигналом M из регистра цвета и сигналом мультиплексора D42 (10.11). Обратите внимание, что с этого же мультиплексора D42 выходит сигнал GMA7 и используется для адресации седьмой строки и седьмого столбца в ОЗУ.

Теперь мы готовы нарисовать временные диаграммы

сигналов CAS0, CAS1 и CAS2, что и сделано на рис. 10.

Теперь прокомментируем рисунок. Когда бит 7 в регистре цвета установлен в единицу (цветовой режим), мультиплексор D118 выдает все три сигнала CAS одновременно, т.е. адресация строк и столбцов идет одновременно во всех плоскостях. Для полноты картины разберемся с сигналом GMA7, который формируется на другом выходе мультиплексора D42 (10.11).

С нашим, уже богатым опытом, не составляет труда сообразить, что в отсутствие сигнала ACCESS, т.е. когда он имеет высокий уровень, мультиплексироваться будут сигналы VRP1 и VRP0, причем VRP1 отвечает за выбор строки, а VRP0 за выбор столбца. При наличии сигнала ACCESS, т.е. при записи или чтении в ГЗУ сигнал VWP1 будет выбирать строку, а VWP0 – столбец.

Сигналы VWP и VRP можно программно устанавливать регистром управления изображением (порт C, D17 (13.35)). Теперь мы знаем, за что отвечает каждый бит этого регистра. Обратите внимание, что биты 0,1 задают номер страницы памяти, которая отображается на экране. Для записи или чтения ГЗУ нужно устанавливать номер страницы с помощью битов 6 и 7. В ситуации, когда бит 7 в регистре цвета равен 0 (послойный режим), за выбор памяти отвечают биты 1, 2, 3 в регистре цвета (сигналы GWD1, GWD2 и GWD3, поступающие с шины данных через буфер D86 (лист 3)).

Обратите внимание, как происходит выбор микросхемы ГЗУ в режиме записи (вход EWR). На них поступает логическая комбинация сигналов WEG и одного из разрядов шины данных, т.е. переводятся в режим записи только те микросхемы, для которых в байте, приготовленном для записи в ГЗУ, соответствующие биты установлены в единицу. Теперь нам ясно, что кроется за фразами, типа: "Байт, записываемый в ГЗУ представляет из себя маску".

С адресацией ГЗУ мы разобрались. Что же туда записывается? Это уже совсем просто. Достаточно посмотреть на мультиплексор D119(8.10) и Вы тут же сами ответите на этот вопрос. В цветовом режиме, когда сигнал M высокого уровня, в плоскость 0 записывается бит 1 регистра цвета (сигнал WR0), в плоскость 1 – бит 2 и в плоскость 2 – бит 3. В этом случае биты 1,2 и 3 можно рассматривать, как логический номер цвета (от 0 до 7), что и делалось в предыдущей главе. В послойном режиме во все плоскости на вход D поступает значение бита 0 в регистре цвета.

Как происходит чтение из ГЗУ? С адресацией памяти мы уже разобрались с помощью рисунка 10. При чтении выбираются все плоскости. Из схемы на листе 8 мы видим, что данные из всех плоскостей поступают на входы двух ПЛМ – D88 (8.19) и D101 (8.20) вместе с сигналами R0, R1, R2 и M регистра цвета. С выхода ПЛМ данные стандартно (смотри работу ОЗУ) защелкиваются в регистре D87 (8.36).

Что же происходит в ПЛМ? Из предыдущей главы мы познакомились с процессом чтения с точки зрения программиста. Сейчас мы с этим вопросом разберемся капитально. Для этого в таблице 6 приводится карта прошивки. Она одинаковая для обеих микросхем. Вы уже обнаружили, что на D88 заводятся младшие адреса, а на D101 – старшие. Процедура обработки сигналов одинаковая в каждой микросхеме и, более того, одинакова для каждого разряда данных из ГЗУ.

Из таблицы мы можем составить логическую формулу сигнала, например для разряда 0 (выход 7).

$$\begin{aligned} \text{GRD0} = & \text{M}^* \text{R0}^* \text{GD00} + \text{M}^* \text{R0}^* \text{GD00} + \text{M}^* \text{R1}^* \text{GD10} + \text{M}^* \text{R1}^* \text{GD10} + \\ & + \text{M}^* \text{R2}^* \text{GD20} + \text{M}^* \text{R2}^* \text{GD20} + \\ & + \text{M}^* \text{R0}^* \text{GD00} + \text{M}^* \text{R1}^* \text{CD10} + \text{M}^* \text{R2}^* \text{GD20}. \end{aligned}$$

Теперь займемся анализом. Из структуры формулы видно, что имеются два случая. M = "1" (цветовой режим) а M = "0" (послойный). Давайте начнем с M = "0". В этом случае в формуле остается мало членов:

$$\text{GRD0} = \text{R0}^* \text{GD00} + \text{R1}^* \text{GD10} + \text{R2}^* \text{CD20}.$$

Здесь все ясно. Биты 4, 5 и 6 в регистре цвета позволяют выбирать интересующую нас плоскость. Для этого нужный бит нужно установить в единицу. Рассмотрим теперь цветовой режим (M = "1"). Т.к. все плоскости в этом режиме эквивалентны, то будем анализировать одну плоскость, например 0, тогда:

$$\text{GRD0} = \text{R0}^* \text{GD00} + \text{R0}^* \text{GD00}.$$

Чтобы проще и быстрее понять смысл действий над содержимым плоскости 0, составим таблицу 7.

Теперь вспомним, что записывается в ГЗУ в цветовом режиме. Биты 1, 2, 3 определяли логический номер цвета. Пусть мы записали номер цвета 6 (110B), т.е. в плоскость 0 записали ноль, в плоскость 1 и 2 по единице. Теперь при чтении мы тоже установим в битах 4, 5 и 6 то же число 110B. Тогда плоскость 0 даст выход 0 в соответствии с первой строчкой таблицы. Плоскости 1 и 2 тоже дадут нулевой вклад, но уже в соответствии с последней строчкой таблицы.

Таким образом в ПЛМ происходит аппаратное сравнение считанного логического номера цвета с заданным в битах 4, 5 и 6 регистра цвета. В случае совпадения в соответствующем бите считанного байта будет 0, что означает, что цвет точки совпадает с заданным в регистре цвета.

Нам осталось разобраться, как формируется изображение на дисплее. Данные с ГЗУ при сканировании выдвигаются из сканируемых регистров с частотой сигнала GCEK, равной 10 МГц. Механизм загрузки сдвиговых регистров точно такой же, как в

АЦЗУ, поэтому на нем мы не останавливаемся. А вот с выходов сдвиговых регистров сигналы поступают в регистр D115 (8.8), который вместе с микросхемой D114 (8.11) и D113 (8.21) образует просмотровую таблицу. Назначение этого устройства мы уже знаем. Оно позволяет логическому номеру цвета присваивать физический цвет. Давайте посмотрим, как это происходит.

Сначала разберемся в микросхемах. Начнем с регистра. Регистр KP531IP20 – четырехразрядный регистр. Его входы организованы, как два порта данных, по четыре проводника в каждом. Коммутация осуществляется сигналом на входе SE1/SE0. При высоком уровне этого сигнала подключены линии SERV0, SERV1, SERG2 и 1BWV, при низком – четыре младших разряда шины данных.

На выходе регистра данные появляются при отрицательном перепаде на входе С.

Теперь, как Вы уже догадались, память. Микросхема K155PY2 действительно память статического типа с организацией 16x4 бит.

Для понимания работы микросхемы приведем таблицу 8.

И, наконец, триггер D113 – это шесть триггеров. Вход R – общий сброс при низком уровне сигнала. В нашем случае сброс не используется. Защелкивание в триггер происходит по положительному фронту сигнала на входе С.

Теперь Вы уже сами в состоянии понять работу просмотровой таблицы. В режиме отображения информации D114 находится в режиме чтения. На ее адресные входы поступают три сигнала с трех плоскостей ГЗУ и сигнал с АЦЗУ (1BWV). На выходе D114 будут появляться данные, записанные в ней, которые с частотой 10 МГц (частота сигнала VCLK) будут защелкиваться в триггере.

После триггера сигналы через инвертор D122 поступают на разъем XS10 цветного ВКУ. Инвертор D122 нужен, чтобы исправить инверсию сигнала после считывания его из памяти D114.

Для примера рассмотрим случай, когда из ГЗУ считана точка с логическим цветом 7. Следовательно, в соответствующий момент времени на адресных входах будет код 7. (полагаем, что сигнал 1BWV низкого уровня – в этом месте экрана нет символа). Следовательно на выходе D114 может появиться все, что угодно. Все зависит от того, что мы записали раньше в D114 по адресу 7. Допустим, что по адресу 7 было ранее записано число 5, т.е. сигнал высокого уровня появится на выходах 0 и 2 микросхемы D114. Следовательно цвет точки будет реализован с помощью сигналов VIDI (интенсивности) и VIDR (красный) и будет красным. Если бы по адресу 7 лежало число 4, то цвет точки был бы черным, т.к. сигнал VIDI был бы нулевым, верней бы была изображена "красная" точка нулевой интенсивности.

Из сказанного ясно, что программирование просмотровой таблицы требует тщательности. При некоторых вариантах программирования могут происходить нежелательные явления с Вашим изображением на ВКУ.

Нам осталось, учитывая богатый накопленный опыт анализа схем, уточнить, как происходит запись в просмотровую таблицу. Адресация осуществляется по линиям сигналов SELREG и A2, следовательно относительный адрес просмотровой таблицы FBH. При обращении к просмотровой таблице происходит одновременное переключение D114 в режим записи сигналом низкого уровня с элемента "ИЛИ" D99 (8.5) и регистра D115 (8.8) на младшие разряды шины данных.

Результаты наших рассуждений сведены в таблицу 9.

В заключение сделаем еще несколько замечаний. Сигнал HVSYNC с помощью перемычки XP4 (лист 8) может быть проинвертирован. Для этого нужно соединить т.1 и т.2 перемычкой. Если перемычка будет соединять т.3 и т.2, то сигнал HVSYNC не инвертируется. Это сделано для удобства сопряжения "Корвета" с ВКУ. Некоторые ВКУ требуют инверсного сигнала HVSYNC.

На этом мы закончим разбор работы ГЗУ и перейдем к схеме контроллера дисководов.

## 8. КОНТРОЛЛЕР ДИСКОВОДОВ

Контроллер дисководов также называется контроллером накопителя на гибких магнитных дисках (НГМД).

Схема приведена на листе 14 и обеспечивает подключение к микроЭВМ до четырех накопителей диаметром 3.5; 5.25 или 8 дюймов. Возможно осуществление режимов односторонней и двусторонней записи с одинарной или двойной плотностью. Контроллер реализован на базе БИС KP1818BG93 (D4 (14.2)) и порта В адаптера параллельного интерфейса D17 (13.35), запрограммированного на вывод.

Интерфейс D17 (13.35) нам уже знаком, только теперь используется его другой порт. Адресация D4 (14.2) идет по линиям CSCOM, A0 и A1. Сигналы 1WR (запись) и RDEV (чтение) осуществляют связь процессора с контроллером по буферизированной шине данных.

Давайте сначала разберемся с сигналами, которыми управляет регистр D17 (13.35). Сигналы DS0, DS1, DS2 и DS3 отвечают за выбор накопителя. Установка единицы в соответствующем разряде регистра сформирует соответствующий сигнал низкого уровня на выходе контроллера.

Здесь уместно добавить, что НГМД требует, чтобы активные уровни всех сигналов были низкие. Поэтому выходные сигналы как правило инвертируются.

Сигнал SIDE адресует сторону выбранного дисковода. Установка нуля выбирает сторону "0", установка единицы – сторону "1".

Сигнал MON используется для включения двигателя НГМД. Установка единицы в разряде 5 регистра D17 запускает ждущий одновибратор D5 (14.31), выход которого формирует сигнал MON. Времязадающая цепочка C5, R2 подобрана так, что через три секунды одновибратор возвращается в исходное состояние, формируя сигнал MON высокого уровня длительностью три секунды. Одновибратор D5 имеет одно полезное свойство, он перезапускаемый, т.е., если до окончания выходного импульса одновибратора на его входе опять создать положительный перепад потенциала, то выходной импульс с этого момента опять будет продолжаться три секунды. Таким образом сигнал MON можно сделать активным сколь угодно долго, перезапуская одновибратор с периодом перезапуска меньше трех секунд.

Если одновибратор возвращается в исходное состояние, то на его инверсном выходе формируется сигнал прерывания IRQ7 от выключения двигателя, поступающий в контроллер прерываний D8(12.4). Процессор программно обслуживает этот запрос.

Оставшиеся два старших разряда в регистре управления D17 определяют режим работы НГМД. Из-за большого разнообразия различных типов применяемых НГМД необходимо уметь настраивать контроллер на различные режимы работы. Как видно из схемы (лист 14) сигналы DDEN (бит 6) и 5" (бит 7) управляют работой мультиплексора D13 (14.1) и выбирают нужную тактовую частоту

как для БИС D4, так и для счетчика D6 (14.3). В таблице 10 дается описание различных режимов работы НГМД и зависимость их от сигналов DDEN и 5".

Таким образом с помощью регистра D17 можно программно выбрать накопитель, сторону диска, режим работы и включить мотор. Все остальные манипуляции производятся с БИС D4 и это уже обсуждалось в предыдущей главе. Здесь мы обсудим метод записи на диск. Как видно из таблицы 10 в колонке "Метод записи" стоят сочетания букв FM и MFM. Это аббревиатура от английских слов Frequency Modulation (частотная модуляция) и Modify Frequency Modulation (модифицированная частотная модуляция).

В зависимости от уровня напряжения на линии DDEN контроллер кодирует загруженный байт данных по методу частотной модуляции FM (при высоком уровне) или по методу модифицированной частотной модуляции MFM (при низком уровне) и в последовательном коде выводит его на выход WRD микросхемы контроллера. Временные диаграммы кодирования данных приведены на рисунке 11. При обоих способах кодирования в последовательном потоке информации содержатся данные ("D") и синхросигналы ("C"). При способе кодирования FM каждый битовый элемент содержит бит данных и синхросигнал. При способе кодирования MFM синхросигнал помещается в битовом элементе только в том случае, если данные предыдущего и текущего битового элемента равны нулю.

Рассмотрим теперь, как выполняются команды контроллера. Управление перемещением головки осуществляется контроллером по линиям STEP и DIR. Высокий уровень сигнала на линии DIR (или низкий после инвертора D3(14.5) задает направление перемещения головки к центру диска, низкий уровень – наоборот, от центра. Перемещение головки осуществляется низким уровнем сигнала (после инвертора D3(14.4)) на один шаг. Чтобы сместить головку на несколько шагов, контроллер должен менять уровень сигнала STEP через интервалы времени, определяемые командой процессора.

Физическое начало каждой дорожки отмечается на линии IP низким уровнем напряжения длительностью не менее 10 мкс при каждом обороте диска. Сигнал IP используется для синхронизации контроллера. Низкий уровень напряжения на линии TP00 (нулевая дорожка) сигнализирует контроллеру, что головка записи-воспроизведения находится над нулевой дорожкой.

Как осуществляется запись данных на диск? В начале операции записи микропроцессор анализирует состояние входных линий готовности READYD и защиты записи WPRT и приступает к записи только при низком уровне напряжения на входе READYD и высоком уровне напряжения на входе WPRT. Запись данных контроллер выполняет, используя выходные линии микросхемы KP1818BG93 WR, C и D, EARL, LATE.

Если имеет место запись на диск, контроллер формирует на выходе разрешения записи WG низкий уровень напряжения на все

время операции записи. При этом первый байт данных должен быть загружен с шины данных в контроллер до установления низкого уровня напряжения на линии WG.

Данные записи WRD поступают в НГМД через схему цифровой компенсации данных, выполненную на микросхемах D0(14.32) и D7(14.8). Микросхема D7 это четырехразрядный сдвиговый регистр. Сигналом высокого уровня на входе EZ его выходы все время подключены. Если на входе EWR сигнал высокого уровня, то по отрицательному фронту импульса на входе С происходит параллельная загрузка. При EWR = "0" происходит тактируемый сдвиг содержимого регистра в сторону старших разрядов, а в младший разряд загружаются данные с входа 0→ (в нашем случае ноль). Сигнал на вход управления работой регистра EWR поступает с инверсного выхода триггера D0. Попробуем теперь разобраться в работе схемы.

Сигнал разрешения записи WG сбрасывает своим низким уровнем триггер D0 и все время держит его в состоянии сброса. При этом на инверсном выходе триггера сформирован сигнал высокого уровня, который "держит" регистр D7 в режиме параллельной загрузки. Если производится запись на диск, то сигнал WG становится сигналом высокого уровня, разрешая триггеру работать. Поскольку в старший разряд регистра D7 все время загружается ноль (соответствующий вход заземлен), то на К входе триггера будет сигнал низкого уровня. При передаче данных первый же импульс на выходе WRD контроллера, поступивший на J вход триггера, запишет в него ноль на инверсном входе и переведет регистр в режим сдвига.

В режиме сдвига при условии, что на вход J данные больше не поступают, первая же единица на выходе сдвигового регистра опрокинет триггер в исходное состояние, который в свою очередь переведет регистр в режим загрузки. Таким образом триггер D0 синхронизирует работу сдвигового регистра с тактовой частотой, опрокидываясь импульсом данных. С выхода регистра D7 задержанные данные через инвертор D3(14.7) поступают по линии данных в НГМД.

Чтобы разобраться с задержкой данных в регистре D7, займемся тем, что загружается в регистр. Сначала предположим, что на выходах LATE и EARL сигналов нет, т.е. присутствуют напряжения низкого уровня, тогда в регистр с помощью инверторов D28(14.9) и D29(14.10) будет записано число 0010B и данные на выходе WD задержатся на два такта. Сигнал данных может сопровождаться или сигналом EARL(раньше) или сигналом LATE(позже). Очевидно, при сигнале EARL сигнал WD задержится на один такт (в регистр будет загружено число 0100B), а при сигнале LATE на три такта (в регистр будет загружено число 0001B).

Таким образом схема компенсации изменяет задержку выдачи данных в НГМД в зависимости от напряжений на выходах контроллера EARL и LATE.

Зачем нужна компенсация данных? Прежде всего отметим,

что она используется только при методе записи MFM. В этом случае для надежной записи необходима коррекция данных, причем дисководы разных фирм требуют разной величины временной коррекции. В нашем случае схема рассчитана на дисководы фирмы TEAC и величина коррекции равна 250 нсек. Поэтому при использовании дисководов других фирм могут возникнуть затруднения. Мы надеемся, что наши читатели смогут победить эту проблему. Для этого нужно знать рекомендуемую величину коррекции для конкретного дисковода.

Нам осталось разобраться с чтением данных. Этот процесс осуществляется из адресуемого НГМД при низком уровне напряжения на входе READYD.

Данные чтения в формате FM или MFM поступают по линии RAWRD через микросхему D1(14.12) на вход одновибратора D5(14.30). С выхода одновибратора отрицательный импульс длительностью 150 нс поступает на вход данных RD D микросхемы контроллера.

Для разделения потока информации на данные и синхрои́мпульсы используется счетчик D6(14.3), который формирует стробирующие сигналы, поступающие на вход RD C микросхемы контроллера.

При отсутствии сигналов RAWRD счетчик, тактируемый частотой 4 МГц или 2 МГц в зависимости от входов управления мультиплексора D13(14.1), формирует "окна" номинальной длительности 2 мкс при частоте 4 МГц.

При поступлении сигнала RAWRD отрицательный импульс с выхода одновибратора D5 загружает в младшие разряды счетчика код 011 сохраняя состояние старшего разряда без изменения. Поскольку счетчик считает на уменьшение, то через четыре такта уровень сигнала на выходе счетчика обязательно изменится. В результате с приходом импульса данных или синхрои́мпульса счетчик с задержкой в четыре такта начинает "отсчет времени". Разделение потоков данных и синхрои́мпульсов (стробирование данных) происходит из-за разного временного интервала между импульсами.

На этом обсуждение контроллера НГМД мы закончим. Для тех, кто ничего не понял в том, как осуществляется запись на диск и чтение с него, в качестве утешения рекомендуем обратиться к рисунку 12. На нем приведены временные диаграммы работы контроллера.

Мы подробно разобрали работу основных и сложных узлов "КОРВЕТА". С остальными периферийными устройствами мы поступим более экономно и ограничимся краткими комментариями.



## 9. ОСТАЛЬНЫЕ ПЕРИФЕРИЙНЫЕ УСТРОЙСТВА

Контроллер прерываний выполнен на микросхеме KP580BH59 (D8 (лист 12)). Контроллер позволяет обслуживать 8 уровней прерываний от адаптеров периферийных устройств с фиксированными или циклически меняющимися приоритетами, маскировать отдельные уровни.

Приоритеты запросов распределены следующим образом:

уровень 0 – запрос от дополнительных внешних устройств, подключенных к разъему РАСШИРЕНИЕ;

уровень 1 – запрос от адаптера последовательного интерфейса (готовность приема);

уровень 2 – запрос от адаптера последовательного интерфейса (готовность передачи);

уровень 3 – запрос от адаптера локальной сети;

уровень 4 – запрос сигналом кадрового гасящего импульса VBL с частотой 50 Гц;

уровень 5 – запрос от канала 2 программируемого таймера;

уровень 6 – запрос от интерфейса принтера;

уровень 7 – запрос от контроллера НГМД.

Программируемый таймер выполнен на микросхеме KP580BI53 (D9, лист 12) и имеет три канала. Канал 0 используется для формирования звуковых колебаний программируемого тона. Канал 1 задает скорость передачи данных по последовательному интерфейсу (до 9600 бит/с). Канал 2 предназначен для генерации прерываний через фиксированные промежутки времени. На вход С канала 2 поступает сигнал HBL с периодом 65.6 мксек. Режим работы канала 2 задается по шине данных в регистре режима, а коэффициент деления частоты – в счетчике канала, адресуемых по линиям CSCT, A0, A1.

Адаптер клавиатуры выполнен на двух микросхемах K555KP14 D37(13.18), D36(13.17) (лист 13) с использованием инвертора K555ЛН2 D28, D29 (лист 12). Адаптер обеспечивает программное сканирование и декодирование простейшей матрицы контактов-замыкателей. Дребезг контактов устраняется программно. Микропроцессор опрашивает группу из восьми клавиш, адресуя ее логической единицей на шине адреса и считывая сигналом чтения CSKB через мультиплексоры D37, D36. Нажатое

состояние клавиши дает высокий уровень напряжения в соответствующем разряде считанного на шине данных байта данных. Сигнал A8 шины адреса через инвертор D28(13.2) подается на входы SE мультиплексоров D36, D37 и управляет считыванием группы клавишей из основного поля при низком уровне напряжения на линии A8 или из дополнительных полей – при высоком. Опрос одновременно нескольких групп клавишей с целью выявления хотя бы одной нажатой клавиши микропроцессор выполняет, выставляя высокий уровень напряжения в соответствующих разрядах шины адреса. При этом за одну операцию чтения микропроцессор может проанализировать матрицу 8 x 8 на наличие хотя бы одной нажатой клавиши. По адресу группы и байту считанных данных микропроцессор программно формирует адрес и считывает из ПЗУ знакогенератора символ нажатой клавиши. Подробнее смотри предыдущую главу.

Адаптер последовательного интерфейса предназначен для подключения к микроЭВМ дополнительных внешних устройств, а в случае использования ее в качестве графического терминала – для связи со старшей ЭВМ.

Адаптер построен на базе микросхемы KP580BB51A (D10, лист 12) и реализует либо интерфейс типа токовой петли 20 мА (ИРПС) с использованием оптрона D57 и микросхем D12, D41, D38, либо интерфейс типа RS232 с использованием микросхемы D41, транзистора VT2 и микросхем D12, D56.

Способ обмена – асинхронный, двунаправленный, скорость передачи до 9600 бит/сек. Скорость задается программно установкой коэффициента деления частоты сигнала синхронизации приемника и передатчика. Этот сигнал поступает на микросхему D10 с канала 1 таймера D9.

Микропроцессор может осуществлять ввод символа в адаптер по запросу, генерируемому на выходе RATF микросхемы D10. Сигнал высокого уровня на этом выходе поступает на второй уровень прерываний БИС D8. Уровень напряжения автоматически меняется на низкий, когда символ загружается в адаптер. Передатчик выводит битовые посылки по спаду сигналов синхронизации в последовательном формате на выход TF микросхемы D10 и далее на линии -V24TD или TXD. Причем первыми идут младшие биты.

Приемник принимает с линий +V24RD, -V24RD или RXD последовательные биты по фронту сигналов синхронизации. После приема всего символа на выходе RARC микросхемы D10 генерируется высокий уровень сигнала запроса на обслуживание (уровень прерываний 1). Линии готовности DSR и RTS используются в зависимости от программ пользователя.

Адаптер локальной сети построен на основе БИС приемо-передатчика KP580BB51A (D11, лист 12). Способ передачи информации в линии связи принят последовательный асинхронный с контролем байтов на четность. Скорость передачи фиксированная

19600 бит/с. Обмен информацией в сети осуществляется под управлением компьютера ПК8020 с помощью опроса состояний компьютеров ПК8010. Последние имеют в сети индивидуальные номера от 1 до 15. Для одновременной циркулярной загрузки системных программ имеется групповой номер 0, на который реагируют все персональные компьютеры ПК8010.

Номер персонального компьютера (рабочего места) задается установкой перемычек на ответной части соединителя локальной сети и определяться программно через порт А микросхемы D17(13.35).

Топологически сеть представляет собой двунаправленную магистраль, соединяющую все рабочие места комплекса. Линия связи выполнена в виде витой пары и прокладывается стационарно. При работе с адаптером локальной сети необходимо установить перемычку между контактами 04 и 02 соединителя адаптера.

Адаптер кассетного накопителя позволяет под управлением программы производить запись и чтение информации с магнитной ленты со скоростью до 2400 бит/с. Входная информация с кассетного накопителя поступает через усилитель D58 (13.9, 13.22) на вход порта "А" микросхемы KP580BB55A (D17(13.35)). Выходной сигнал адаптера CASSOUT поступает на кассетный накопитель с выходов порта "С" (1,0) микросхемы D16(13.52). Выходной сигнал управления реле "кратковременный стоп" поступает с выхода "2" порта "С" микросхемы D16(13.52) через усилитель K155ЛН5 (D38 (13.32)). Блочный способ записи с использованием специальных кодов дает возможность получать высокую надежность записи.

Адаптер печатающего устройства (ПУ) осуществляет вывод данных на печать и реализует необходимое подмножество цепей параллельного интерфейса типа ИППР-М(CENTRONICS). Это подмножество включает восемь однонаправленных цепей данных с высоким активным уровнем напряжения и цепи сигналов стробирования "SE", выбора ПУ "ACK" и сигнала "BUSYP".

Адаптер содержит регистр данных, реализованный на базе порта "А" микросхемы D16 (13.52), регистр управления, реализованный на базе порта "С" микросхемы D16 (разряды 4 и 5), регистр состояния, реализованный на базе порта "А" микросхемы D17 (13.35) (разряд 2). Микропроцессор осуществляет передачу данных в ПУ, предварительно анализируя состояние линии "BUSYP" через регистр состояния, адресуя его по линиям "CSIOP1", "A0", "A1". При наличии признака готовности ПУ (низкий уровень напряжения на линии "BUSYP") микропроцессор записывает байт данных в регистр данных, адресуя его по линиям "CSIOP2", "A0", "A1", устанавливает и сбрасывает бит стробирующего сигнала. Предусмотрена возможность обслуживания ПУ по прерыванию, для чего сигнал "BUSYP" по линии "IRQ6" передается в контроллер прерывания. Микропроцессор программно решает, когда обслужить этот запрос.

Адаптер параллельного интерфейса на основе микросхемы KP580BB55A (D2 (13.53)) обеспечивает расширение функциональных возможностей компьютера. Цепи программируемых портов ввода/вывода непосредственно выведены на соединитель. Это обеспечивает большую гибкость в управлении подключаемыми устройствами, так как позволяет свободно выбирать режим работы и направление передачи сигналов по линиям, но при этом требует обязательного применения согласующих элементов или дополнительных адаптеров в непосредственной близости от соединителя. На этот же соединитель выведены сигналы частотой 2 МГц и выход канала 0 программируемого таймера D9 (12.22).

Синтезатор звука вырабатывает звуки заданной частоты и может быть использован для встраивания в программы музыкальных мелодий. Высота звука и длительность звучания задаются программно с помощью канала 0 программируемого таймера (D9, лист 12).

Для задания тональности звучания канал 0 программируется как делитель частоты 2 МГц, подаваемой на вход С канала. Микропроцессор задает по ШД коэффициент деления частоты, который зависит от требуемой тональности. Управление пьезоэлектрическим излучателем звука HA1 (лист 3) осуществляется по линии SOUND через схему совпадения D18 (12.31), на второй вход которой по линии стробирования звука SCLK подается сигнал разрешения (высокий уровень напряжения) или запрета (низкий уровень напряжения) звучания. Сигнал подается с выхода "3" порта С микросхемы D16 (13.52). Установку или сброс этого разряда микропроцессор выполняет программно. Сигнал управления излучателем формируется по амплитуде с помощью усилителя K170АП4 (D20, лист 3).

В заключение несколько слов о блоке питания. Для надежной работы годится любой источник питания, имеющий напряжения:

+ 5В	- 3.0А
+12В	- 2.0А
-12В	- 0.5А

Питание системного блока осуществляется от источника постоянного напряжения через вилку XP2 (лист 16).

На этом мы заканчиваем комментирование периферийных устройств и обсуждение схемотехники КОРВЕТА.

Мы надеемся, что информация этой главы будет полезной для многих любознательных пользователей и воодушевит их на пути изучения цифровой техники.

ТАБЛИЦА 1

ТАБЛИЦА ПРОШИВОК В ПЛМ ДЛЯ ФОРМИРОВАНИЯ УПРАВЛЯЮЩИХ СИГНАЛОВ АЦЗУ /* VIDEO.PLM */		
Номер терма	Матрица "AND"	Матрица "OR"
		pol hlllhllhh
0	1-----h---h	..a.....
1	-hhlh-----	....a...
2	-lllh-----	.....a
3	-hhl---l-----	.....a
4	-h---l-----	a.....
5	-hh--h-----	a.....
6	h-hhllh-----	....a..
7	h-hhlhh-----l	....a..
8	h-hhl-h-----	.a.....
9	-h-----	.....a.
10	1----h-----lh--	...a....
11	1----l---h-lh--	...a....
12	1----l-lhl-lh--	...a....
13	1----l-hll-lh--	...a....
14	h----l-hhl-----	...a....

ТАБЛИЦА 2

С Л О В О   С О С Т О Я Н И Я								
Тип	D7	D6	D5	D4	D3	D2	D1	D0
цикла	чтение	ввод	M1	вывод	подтв.	стек	запись	подтв.
					оста- нова			преры- вания
M1	1	0	1	0	0	0	1	0
M2	1	0	0	0	0	0	1	0
M3	0	0	0	0	0	0	0	0
M4	1	0	0	0	0	1	1	0
M5	0	0	0	0	0	1	0	0
M6	0	1	0	0	0	0	1	0
M7	0	0	0	1	0	0	0	0
M8	0	0	0	0	0	0	1	1
M9	1	0	0	0	1	0	1	0
M10	0	0	1	0	1	0	1	1

ТАБЛИЦА 3

ТАБЛИЦА ПРОШИВОК В ПЛИМ ДЛЯ ФОРМИРОВАНИЯ КАРТЫ ПАМЯТИ /* DEC.PLM */		
Номер терма	Матрица "AND"	Матрица "OR"
		pol hh111111
0	-h-----	aa.a....
1	-1-1-11h1hhhh111	aa.a....
2	-1--11hh-hhhhh1h	aa.a....
3	-1--1h1h-hhhhh1h	aa.a....
4	-1--h1hh-h1hhhhh	aa.a....
5	-1--1hhh-h1hhhhh	aa.a....
6	-111-11h1h-1h111	aa....a.
7	-11-11hh-h-1hh1h	aa....a.
8	-11-1h1h-h-1hh1h	aa....a.
9	11-1-11h1h1hh111	aaa....
10	11--11hh-h1hhhh1h	aaa....
11	11--1h1h-h1hhhh1h	aaa....
12	11--hh1h-hhhhhh1	aaa....
13	11--h1hh-hhhhhh1	aaa....
14	11--1hhh-hhhhhh1	aaa....
15	-1-1-11h1h--h1h1	a.....
16	-1--11hh-h--hhhh	a.....
17	-1--1h1h-h--hhhh	a.....
18	1h-----	aaa....
19	-1--hh-----h-h	.a.....
20	-1--h1h-----l-h	.a.....
21	-1--h-1-----h-h	.a.....
22	h1-1-----1---1-1	aa...a..
23	h1-----11---1-1	aa...a..
24	h1-1-1111h---1-1	aa.a....
25	h1-1-11h1h--11-1	aa.a....
26	h1-1-1--hh---1-1	aa.a....
27	h1-1-h---h---1-1	aa.a....
28	h1-1-h--11---1-h	aa....a

ТАБЛИЦА 4.

Обращение к памяти			Нет обращения к памяти	
Сигнал	Адрес строки	Адрес столбца	Адрес строки	Адрес столбца
MA0	A00	A07	C00	C07
MA1	A01	A08	C01	C08
MA2	A02	A09	C02	C09
MA3	A03	A10	C03	C10
MA4	A04	A11	C04	C11
MA5	A05	A12	C05	C12
MA6	A06	A13	C06	C13
MA7	A14	A15	0	0

ТАБЛИЦА 5

Номер строки	Номер столбца	Байт ПЗУ	Относительный адрес строки	ASCII код символа	Адрес байта в ПЗУ
ки	7 6 5 4 3 2 1 0	ПЗУ	адрес строки	символа	в ПЗУ
0		00H	0	41H	0410H
1		00H	1	41H	0411H
2		00H	2	41H	0412H
3	@	10H	3	41H	0413H
4	@   @   @	38H	4	41H	0414H
5	@   @   @   @	6CH	5	41H	0415H
6	@   @       @   @	C6H	6	41H	0416H
7	@   @       @   @	C6H	7	41H	0417H
8	@   @   @   @   @   @	FEN	8	41H	0418H
9	@   @       @   @	C6H	9	41H	0419H
10	@   @       @   @	C6H	A	41H	041AH
11	@   @       @   @	C6H	B	41H	041BH
12		00H	C	41H	041CH
13		00H	D	41H	041DH
14		00H	E	41H	041EH
15		00H	F	41H	041FH

ТАБЛИЦА 6

ТАБЛИЦА ПРОШИВОК В ПЛМ ДЛЯ ФОРМИРОВАНИЯ ДАННЫХ ИЗ ГЗУ /* COLOR.PLM */			
Номер терма	Матрица "AND"	Матрица "OR"	
		pol	hhhhhhh
0	h--l-----h---		...aaaaa
1	h--h-----l---		...aaaaa
2	h-l-----h---		...aaaaa
3	h-h-----l---		...aaaaa
4	hl--h-----		...aaaaa
5	hh--l-----		...aaaaa
6	h--l-----h---		..a.aaaa
7	h--h-----l---		..a.aaaa
8	h-l-----h---		..a.aaaa
9	h-h-----l---		..a.aaaa
10	hl---h-----		..a.aaaa
11	hh---l-----		..a.aaaa
12	h--l-----h---		.a...aaaa
13	h--h-----l---		.a...aaaa
14	h-l-----h---		.a...aaaa
15	h-h-----l---		.a...aaaa
16	hl---h-----		.a...aaaa
17	hh---l-----		.a...aaaa
18	h--l-----h---		a...aaaa
19	h--h-----l---		a...aaaa
20	h-l-----h---		a...aaaa
21	h-h-----l---		a...aaaa
22	hl---h-----		a...aaaa
23	hh---l-----		a...aaaa
24	l--h-----h---		...aaaaa
25	l-h-----h---		...aaaaa
26	lh--h-----		...aaaaa
27	l--h-----h---		..a.aaaa
28	l-h-----h---		..a.aaaa
29	lh---h-----		..a.aaaa
30	l--h-----h---		.a...aaaa
31	l-h-----h---		.a...aaaa
32	lh---h-----		.a...aaaa
33	l--h-----h---		a...aaaa
34	l-h-----h---		a...aaaa
35	lh---h-----		a...aaaa

ТАБЛИЦА 7.

R0	GD00	GRD0
0	0	0
0	1	1
1	0	1
1	1	0

ТАБЛИЦА 8

EZ	ERD/EWR	D(0-3)	Выход	Режим работы
1	X	X	Z-состояние	Хранение
0	0	1	Z-состояние	Запись
0	1	X	Данные в инверсном коде	Считывание

ТАБЛИЦА 9.

Ш и н а   д а н н ы х							
7	6	5	4	3	2	1	0
B	R	G	I	BWV	пл.N2	пл.N1	пл.N0
Подключаемый сигнал				Адрес в просмотрной таблице			

Т А Б Л И Ц А 10

Режим работы			Логический		Частота сигналов	
Диаметр	Плотность	Метод	уровень на		МГц	
диска	записи	записи	линиях			
в			5"	DDEN	Вх С	Вх"-1"
дюймах					ВГ93	ИЕ7
5.25	Двойная	MFM	0	0	1	4
	Запрещено		0	1	2	0 !!!!
3.5	Одинарная	FM	1	0	1	2
8	Одинарная	FM	1	0	1	2

Карты памяти

			TR80	ROMB1	ROMB2	ODOSA	HDOS	BASIC	BASG	DOSA	DOSG1
C000-EFFF	FC00-EFFF	FF00-EFFF	RAM	TRAM	GRAM	TRAM	TRAM	TRAM	GRAM	REGS	REGS
		FE00-EFFF								DEVS	DEVS
		FD00-EDEF								RAM	RAM
		FC00-FCFF									
		FB00-FBFF		DEVS		DEVS	DEVS	DEVS			
		FA00-FAFF		REGS		REGS	REGS	REGS			
		F800-F9FF		KB		KB	KB	KB			
		C000-F7FF		RAM	RAM	RAM	RAM	RAM	RAM	RAM	GRAM
		BF00-BFFF									
		8000-BEFFFF									
	4000-7FFF	6000-7FFF									
		4000-5FFF									
	2000-3FFF	3C00-3FFF	TRAM	ROM2	TRAM	RAM	RAM	ROM3	ROM3	RAM	GRAM
		3B00-3BFF	DEVS		DEVS						
		3A00-3AFF	REGS		REGS						
		3800-39FF	KB		KB						
		3000-37FF	ROM2 6k		ROM2 6k						
		2000-2FFF									
		0000-1FFF	ROM1	ROM1	ROM1		ROM1	ROM1	ROM1		RAM

RAM - основная память; GRAM - ОЗУ графического контроллера; TRAM - ОЗУ алфавитно-цифрового контроллера; ROM# - блоки ПЗУ; DEVS - устройства ввода-вывода; REGS - регистры; KB - клавиатура.