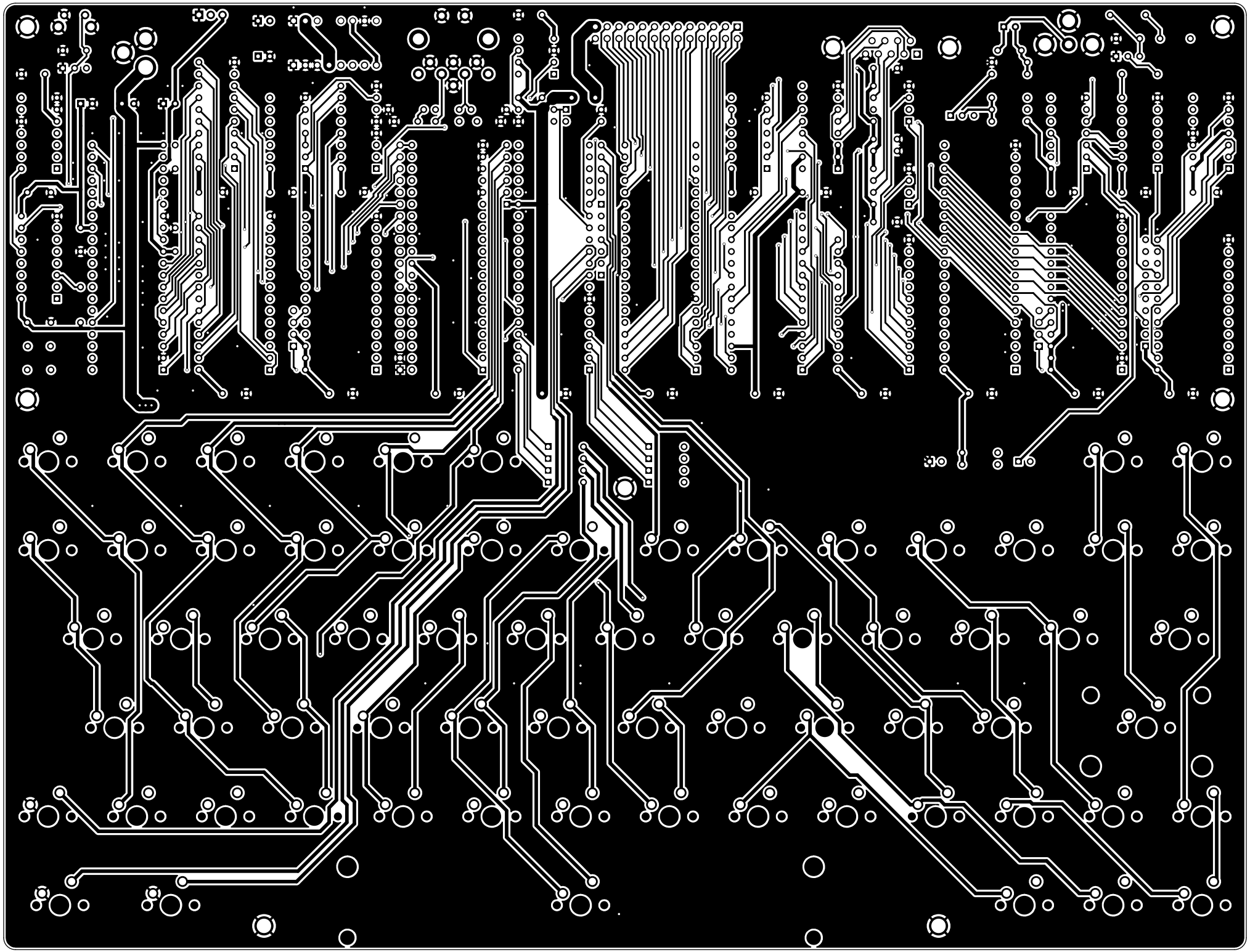

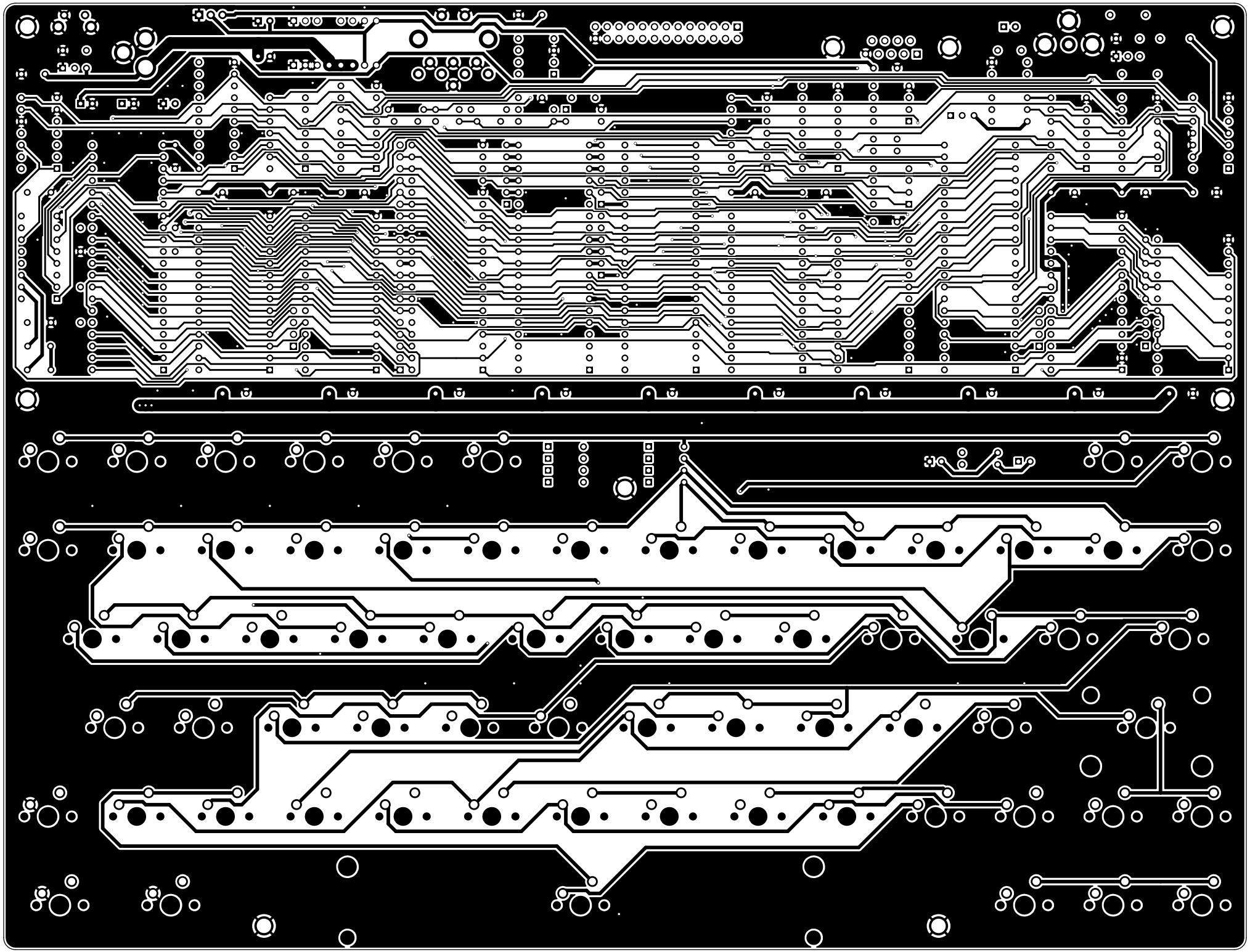

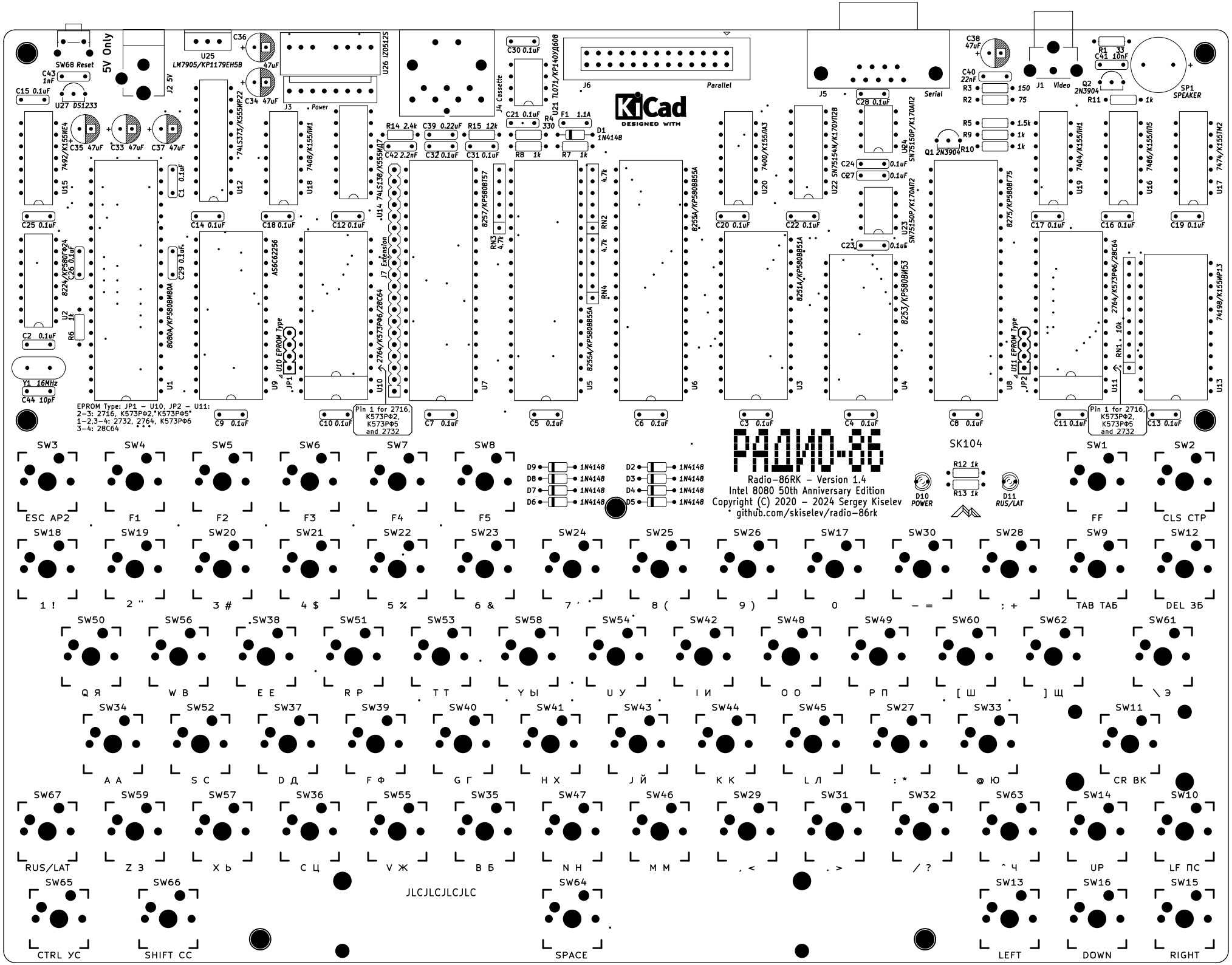
5V Only
SW68 Reset
C43 1nF
C15 0.1uF
U27 DS1233
J2 5V
U25
LM7905/KP1179EH5B
C36
47uF
C34 47uF
U26 IZ05125
J3
Power
J4 Cassette
C30 0.1uF
U21 TL071/KP140УД608
J6
Parallel
J5
Serial
C38 47uF
C40 22nF
R3 150
R2 75
J1 Video
R1 33
C41 10nF
Q2 2N3904
R11 1k
SP1 SPEAKER
C21 0.1uF
R4 330
F1 1.1A
D1 1N4148
R14 2.4k
C39 0.22uF
R15 12k
C42 2.2nF
C32 0.1uF
C31 0.1uF
R8 1k
R7 1k
C28 0.1uF
R5 1.5k
R9 1k
R10 1k
Q1 2N3904
C35 47uF
C33 47uF
C37 47uF
U15 7492/К155ИЕ4
U12 74LS373/К555ИР22
U18 7408/К155ЛИ1
U14 74LS138/К555ИД7
U20 7400/К155ЛА3
U22 SN75154N/К170УП2B
U24 SN75150P/К170АП2
U19 7404/К155ЛН1
U16 7486/К155ЛП5
U17 7474/К155ТМ2
C25 0.1uF
C14 0.1uF
C18 0.1uF
C12 0.1uF
C20 0.1uF
C22 0.1uF
C24 0.1uF
C27 0.1uF
C17 0.1uF
C16 0.1uF
C19 0.1uF
C1 0.1uF
8257/КР580ВТ57
RN3 4.7k
8255A/КР580ВВ55A
8275/КР580ВГ75
U23 SN75150P/К170АП2
C23 0.1uF
8224/КР580ГФ24
C26 0.1uF
C29 0.1uF
A56C62256
J7 Extension
RN2 4.7k
RN4 4.7k
8251A/КР580ВВ51A
8253/КР580ВИ53
2764/К573РФ6/28C64
74198/К155ИР13
U2 1k
R6
U1 8080A/КР580ВМ80A
U9 U10 EPROM Type
JP1
U10 2764/К573РФ6/28C64
U7
U5 8255A/КР580ВВ55A
U6
U3
U4
U8 U11 EPROM Type
JP2
U11 2764/К573РФ6/28C64
RN1 10k
U13
C2 0.1uF
Y1 16MHz
C44 10pF
EPROM Type: JP1 – U10, JP2 – U11:
2–3: 2716, К573РФ2,*К573РФ5*
1–2,3–4: 2732, 2764, К573РФ6
3–4: 28C64
C9 0.1uF
C10 0.1uF
Pin 1 for 2716,
К573РФ2,
К573РФ5
and 2732
C7 0.1uF
C5 0.1uF
C6 0.1uF
C3 0.1uF
C4 0.1uF
C8 0.1uF
C11 0.1uF
Pin 1 for 2716,
К573РФ2,
К573РФ5
and 2732
C13 0.1uF
SW3 SW4 SW5 SW6 SW7 SW8
D9 1N4148
D8 1N4148
D7 1N4148
D6 1N4148
D2 1N4148
D3 1N4148
D4 1N4148
D5 1N4148
РАДИО-86
Radio-86RK – Version 1.4
Intel 8080 50th Anniversary Edition
Copyright (C) 2020 – 2024 Sergey Kiselev
github.com/skiselev/radio-86rk
SK104
R12 1k
R13 1k
D10 POWER
D11 RUS/LAT
SW1 SW2
ESC AP2 F1 F2 F3 F4 F5 FF CLS CTP
SW18 SW19 SW20 SW21 SW22 SW23 SW24 SW25 SW26 SW17 SW30 SW28 SW9 SW12
1 ! 2 " 3 # 4 $ 5 % 6 & 7 ' 8 (9) 0 – = : + TAB ТАБ DEL ЗБ
SW50 SW56 SW38 SW51 SW53 SW58 SW54 SW42 SW48 SW49 SW60 SW62 SW61
Q Я W В E Е R Р T Т Y Ы U У I И O О P П [Ш] Щ \ Э
SW34 SW52 SW37 SW39 SW40 SW41 SW43 SW44 SW45 SW27 SW33 SW11
A А S С D Д F Ф G Г H Х J Й K К L Л : * @ Ю CR ВК
SW67 SW59 SW57 SW36 SW55 SW35 SW47 SW46 SW29 SW31 SW32 SW63 SW14 SW10
RUS/LAT Z З X Ь C Ц V Ж B Б N Н M М , < . > / ? ^ Ч UP LF ПС
SW65 SW66 SW64 SW13 SW16 SW15
JLCJLCJLCJLC
CTRL УС SHIFT СС SPACE LEFT DOWN RIGHT

Memory Map:

```
0000h -  7FFFh:  RAM (32K); 7600h - 7FFFh: Display and Monitor Memory
8000h -  BFFFh:  8255A PPI (Keyboard, Cassette, Sound Control)
9000h -  9FFFh:  8253 PIT (Sound, USART Clock)
0A000h - 0AFFFh: 8255A PPI (Parallel Interface)

0B000h - 0BFFFh: 8251A USART
0C000h - 0CFFFh: 8275 CRT Controller
0D000h - 0DFFFh: Extension Connector
0E000h - 0EFFFh: 8257 DMA Controller (Write Only)
0E000h - 0FFFFh: ROM (Read Only)
```

Monitor Commands – Memory:

```
D<Start_Address>,<End_Address> – Display memory content in hexadecimal
L<Start_Address>,<End_Address> – Display memory content in ASCII
F<Start_Address>,<End_Address>,<Value> – Fill memory with a value

M<Address> – Modify memory content
T<Start_Address>,<End_Address>,<Destination_Start_Address> – Copy memory block to destination
C<Start_Address>,<End_Address>,<Destination_Start_Address> – Compare memory block with destination
S<Start_Address>,<End_Address>,<Value> – Search memory for a value

R<ROM_Start_Address>,<ROM_End_Address>,<Destination_Start_Address> – Read from ROM connected to Parallel interface to memory
```

Monitor Commands – Run Control and Registers:

```
G<Start_Address>[,End_Address] – Run code, optionally stop at the specified address
X – Display and modify registers
```

Monitor Commands – Cassette Input\Output:

```
O<Start_Address>,<End_Address>[,Speed] – Write memory to cassette. Default speed is 1Dh \ 1200 bps
I[<Offset>][,Speed] – Read data from to cassette memory at specified offset
```

Credits:

Schematic and firmware are based on Radio Magazine publications
April 1986 to June 1989: archive.radio.ru\web\ and on
Alexey Khudyakov's Radio-86RK-SRAM project: radio86rk.pbworks.com

Monitor Subroutines:

```
0F803h – Keyboard input – Output: A – character
0F806h – Cassette input – Input: A=0FFh – with sync
         A=08h – no sync; Output: A – data

0F809h – Print to screen – Input: C – character
0F80Ch – Cassette output – Input: C – data
0F812h – Query keyboard – Output: A=00h – key not
         pressed; A=0FFh – key pressed

0F815h – Print to screen in hex – Input: A – data
0F818h – Print string – Input: HL – string address
0F81Bh – Get key – Output: A=0FFh – key not pressed
         A=0FEh – Rus\Lat; otherwise A – key code

0F81Eh – Get cursor – Output: H – row, L – column
0F821h – Read screen – Output: A – char at cursor
0F824h – Read from cassette – Input: HL – offset
         Output: HL – start; DE – end; BC – chk.sum

0F82Ah – Calculate checksum – Input: HL – start;
         DE – end; Output: BC – checksum
0F82Dh – Initialize CRT (after cassette I/O)
0F830h\0F833h – Get\Set RAM top – Output\Input: HL
         – address
```