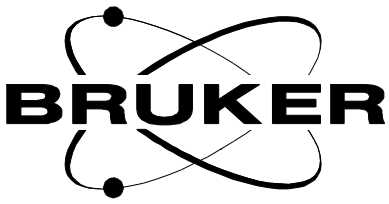


B-HPCU

High Power Control Unit
User Manual

Version 005

BRUKER

The information in this manual may be altered without notice.

BRUKER accepts no responsibility for actions taken as a result of use of this manual. BRUKER accepts no liability for any mistakes contained in the manual, leading to coincidental damage, whether during installation or operation of the instrument.

This manual was written by

Günther Hiß

Rheinstetten, Germany

P/N: Z31314
DWG-Nr: 1024005

Contents

Contents

B-HPCU diagrams

1

Figure 1.1. B-HPCU Wiring Diagram of intelligent (master/slave) Units

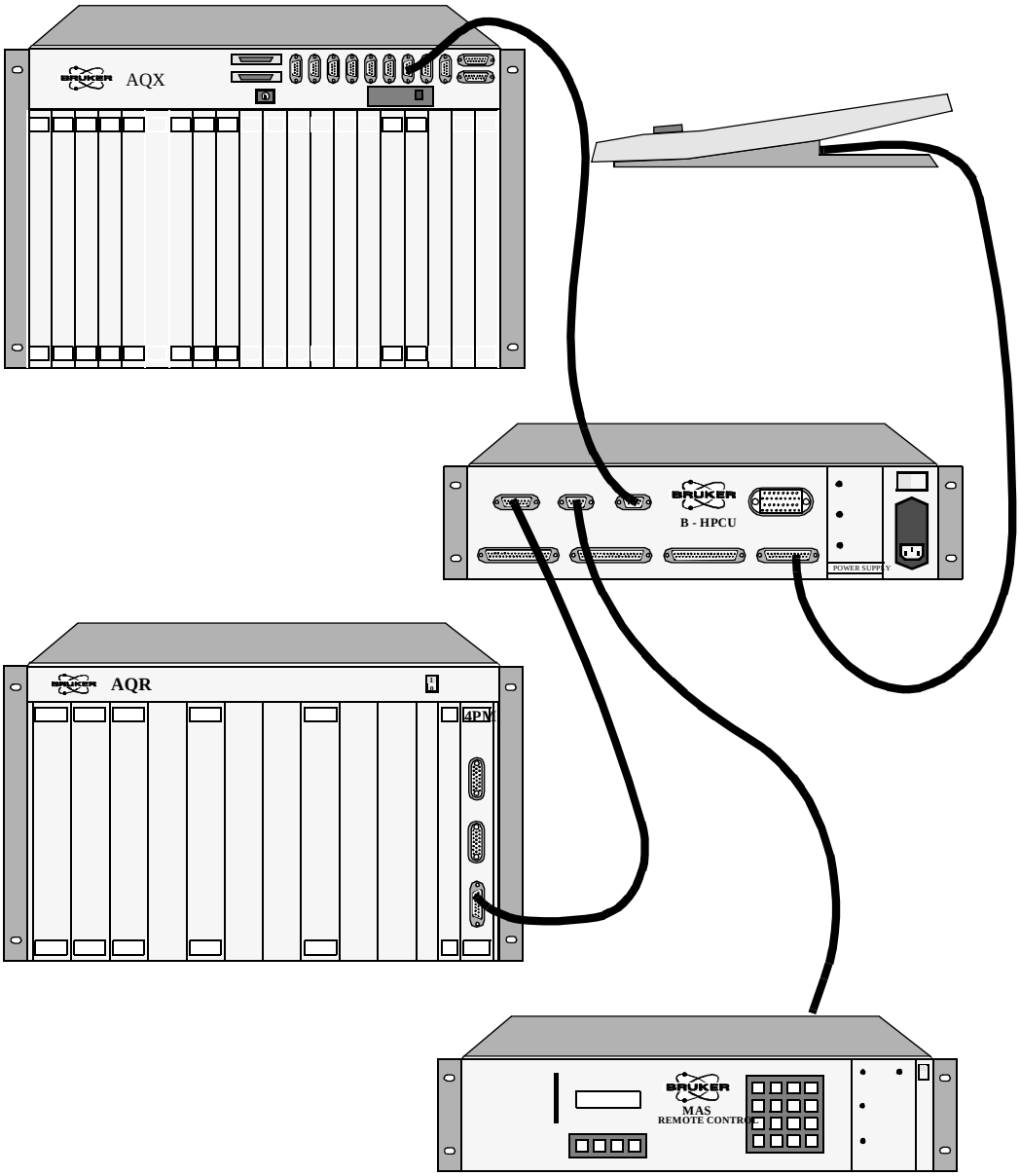


Figure 1.2. B-HPCU Block Diagram

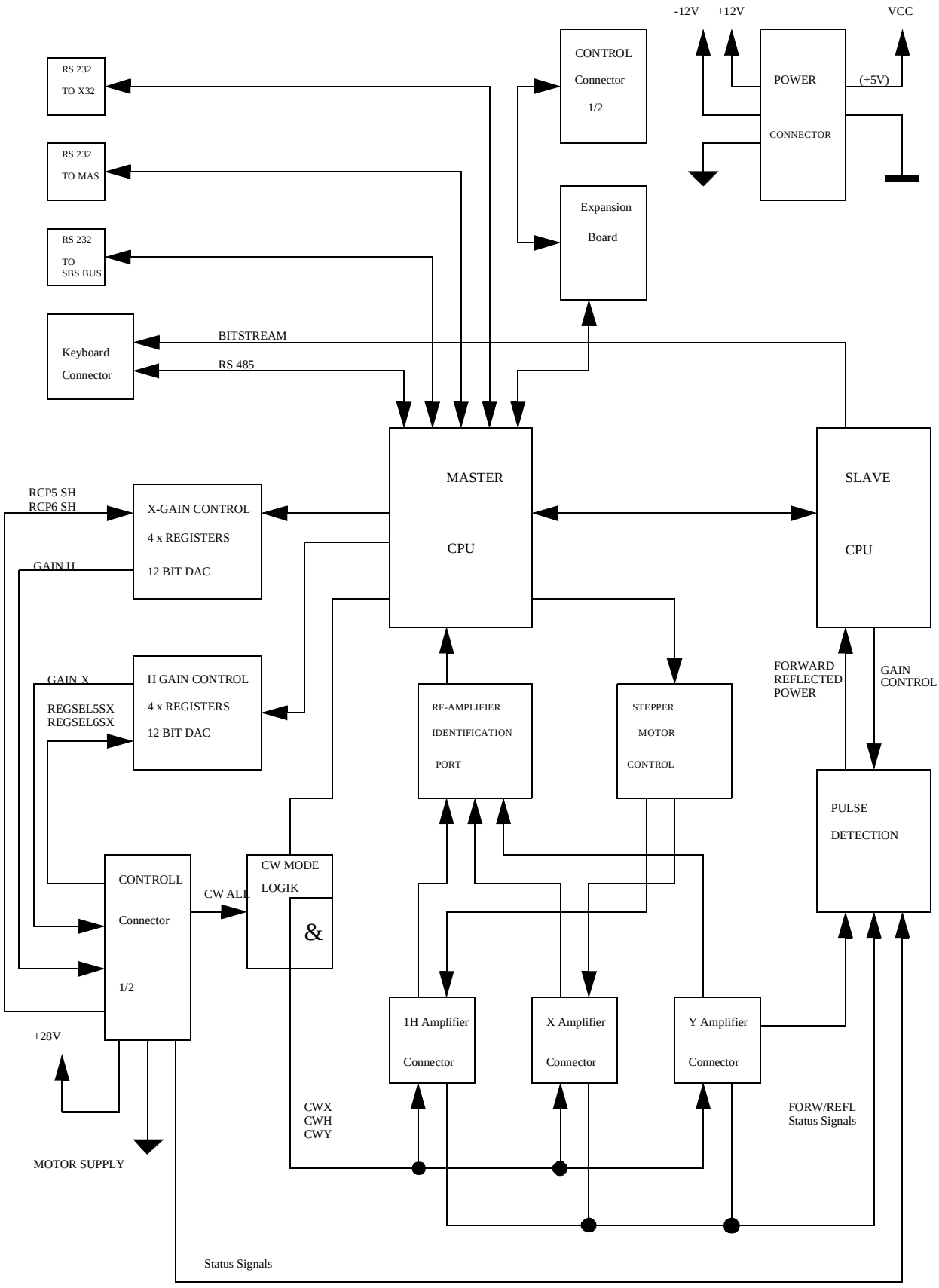
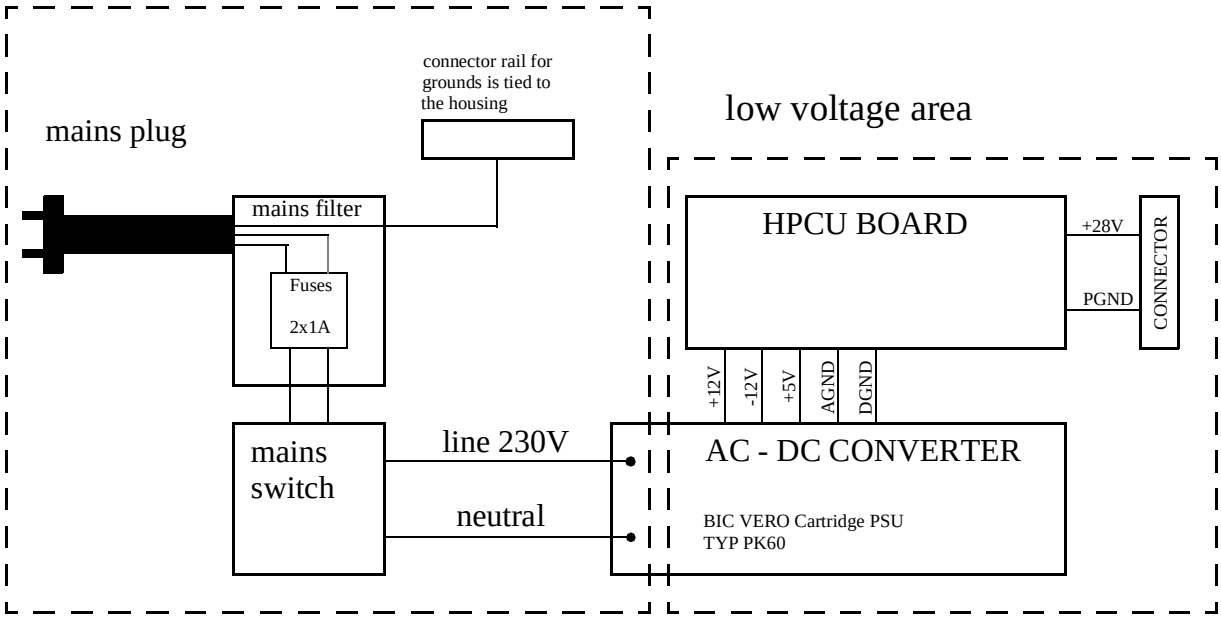


Figure 1.3. B-HPCU Power Supply Block Diagram



B-HPCU diagrams

B-HPCU hardware description 2

The B-HPCU hardware is documented in 24 schematic sheets. These documents are divided into functional modules. The schematic sheet name of a module is located under the module in the root sheet. This sheet name can be found at the top of the title block on the corresponding schematic sheet.

Power supply 2.1

Located in the housing of the HPCU there is an AC/DC switching power supply (BIC VERO PK 60). It delivers +5V/6A, +(12-15)V/2A and -(12-15)V/0.5A. It must be adjusted to +5V, -12V and +12.3V. An additional voltage of +(24-28V) must be supplied. It can either be connected to the 34 pole Burndy (+28V at pin EE, PGND at pin KK) or to the X2 connector on the PCB. At present the connection to the Burndy is used. Connecting this voltage to X2 is an option for service or if future versions of the HPCU are to generate all voltages internally.

CPU link 2.2

The HPCU hardware is controlled by two 80C515AN microcontrollers: a master and a slave CPU. The schematics are split into two main modules: the Motor/ Interface module and the Bitstream module (sheet 1). The microcontrollers are joined via the 8 bit parallel bus CPLID[0..7], and one handshake signal: / CPLISTR.

CPU link protocol *2.2.1*

All commands, and acknowledges have the following structure.

Table 2.1. CPU Link Commands, and Acknowledges Structure

Byte	Number	Meaning
length = n	1 (n=1)	message length
data	2 (n=2)	data byte
data	3 (n=3)	data byte
...	...	...
...	...	...
data	n	data byte

« 8 Bit »

The master first sends the length of the command and then the command and data. The slave performs the command and then sends the acknowledge (first the length and then data). If the slave does not recognise a command, it returns an acknowledge with the length 00hex.

The master writes the byte to CPLID[0..7] and generates a high/low/high transition on/CPLISTRB. This causes an interrupt at the slave CPU. The slave reads the byte on CPLID[0..7] and generates a high/low/high transition at /CPLISTRB also. This causes an interrupt at the master CPU. The master now knows that the slave has accepted the byte and repeats this procedure for the other bytes in the command. When the master has sent all bytes in the command, it switches CPLID[0..7] to 0xFFhex.

Now the slave writes the first acknowledge byte (length) to CPLID[0..7] and generates a high/low/high transition at /CPLISTRB. This causes an interrupt to the master CPU. The master reads the byte from CPLID[0..7] and generates a high/ low/high transition at /CPLISTRB also. This now causes an interrupt to the slave CPU. The slave now knows that the master has accepted the byte and repeats this procedure for the other bytes in the acknowledge. When the slave has sent all bytes of the acknowledge, it switches CPLID[0..7] to 0xFFhex.

If the slave CPU does not respond to an acknowledge the master CPU performs a hardware reset to the slave CPU with the signal /CPLIRESET.

Bit stream module *2.3*

The bitstream module (sheet 2) consists of three parts: Bit Stream CPU (Slave CPU), Keyboard I/O and the Pulse Detection.

Bit stream CPU *2.3.1*

This module (sheet 3) shows the slave CPU kernel. The CPU (U26) 80C515AN runs at 12 MHz. U25, an 27C512, is an EPROM storing the boot software. It is only erasable by UV light. This software is necessary for a download to the FLASH EPROM (U44). The FLASH EPROM stores the firmware (application software). New releases can simply be downloaded, even by the customer. This IC needs the +12V to erase its contents. This is generated from the +12.3V supply by a very low drop regulator U102 MAX 667. This voltage is deliberately generated by the regulator to protect the FLASH IC, as the switching power supply can be trimmed manually by a screw driver from about 11V to 15,5V, which can destroy the FLASH. In hardware version A a Zener diode was used instead of the MAX667. In this case the power supply voltage must not exceed 12V.

Jumper J3 can be used to select the boot EPROM size. Connecting pin 1 and 2 together selects a 27C256 with 32KByte. Connecting pin 2 and 3 together selects a 27C512 with 64KByte. Pin 1 of jumper J3 is marked on the silk screen on the PCB.

IC, U6, generates a hardware reset when either the signal /CPLIRESET becomes „LOW“ (performed by the master CPU) or when the supply voltage VCC drops below 4,75V.

The PAL U46 generates all chip select, read and write signals needed for the PROMs in the application and download mode.

The slave CPU has no external RAM. The internal RAM of the slave CPU used is 256 byte.

IC U45 demultiplexes the data and address bus of CPU.

U7 is a precision 5V reference voltage for the A/D converter of the CPU.

Pulse detection *2.3.2*

The pulse detection module is split into H, X and Y Pulse Detection. The amplitude of all (H, X, and Y) detected pulses can be modified remotely with the HPCU keyboard and E^2POTs. This allows an independent full scale adjustment to the bar graph LED on the keyboard for each forward or reflected power channel. This is realized electrically with the E^2POTS (digitally controlled Electrically Erasable POTentiometer). The wiper of these potentiometers can be programmed in 100 steps from 101 Ohm until 10K Ohm. To control the amplitude of the pulses these potentiometers are located in the feedback loop of Ups U60, U79, U80 (sheet 5, 6 and 7).

The E^2POTs U57.. U63 (X9MME) each have an up/down, increment and an active low chip select input. They are all controlled by the slave CPU. All up/down signals are connected to port pin P16 and all increment inputs are connected to port pin P35. The chip select signals are generated by the slave CPU with three port pins P30..P31 and the demultiplexer U47 (74HCT138 sheet 4), in order that only one E^2POT can be selected at the any time. The resistor value (wiper position) will be stored in the E^2POT when the chip select signal goes “HIGH“. This value is stored even when the power is removed.

H and X Pulse detection

The schematics for forward and reflected power detection of H and X are identical (sheet 6 and 7). Only X power (sheet 7) is shown. All input voltages are positive.

For signals below 0.7V the OP-amp U96 works as a nonlinear feedback amplifier due to two anti-parallel diodes (D46 and D47) instead of a resistor in the feedback loop. Its gain is limited by resistor R86. The capacitors C209, C210 and the resistors IR14C, R86 and R125 prevent the OP-amp from overshooting and oscillating. For input voltages over 0.7V the OP-amp works as a voltage follower + 0.7V.

Capacitor C184 is used to hold (store) the pulses' maximum voltage. If the voltage stored on C184 is 0.7V below the output voltage of OP-amp, U96, the transistor, T10, pumps current directly from the +12V supply into C184 until the input voltage of forward X is reached (equals output voltage of U96 + 0,7V). The current pumped into C184 is limited by R56.

The current through R56 causes a voltage drop on the base of T3. For fast switching of T3 the diode D31 and capacitor C138 are inserted. This triggers the monostable U43. The monostable output HOLDX becomes „LOW", and the FET T15 is switched off. After the monostable times out, HOLDX goes „HIGH" and FET T15 discharges the hold capacitor C138.

The OP-amp, U80A, buffers the voltage on the hold capacitor. Its gain can be adjusted in the range from 1.2 until 5.3. To get a full scale level on the keyboard bar graph, the output voltage of this OP-amp should be 5V.

The monostable is also be triggered if a new voltage is loaded into the hold capacitor C183.

The input voltage of Reflected H In power is negative, and has very low amplitudes. An inverting amplifier with a gain of -10 has been inserted into the signal path. The following Sample & Hold stage operates as described previously.

Y Pulse detection

The Y pulse detection for forward and reflected power each consists of a simple non-inverting OP-amp U79. Their gain can be adjusted via the E^2POT from 1.2 to 5.3.

Keyboard I/O *2.3.3*

This module contains the HPCU keyboard interface and the peripherals needed to read the amplifier status signals.

The signals BLANK H, BLANK X, BLANK Y are short pulses. They are stretched by the monostables U97 and U98 to allow them to be seen on the keyboard. Signal HVY is read over P67 by the CPU. All other status signals are read by the slave CPU over port P4 via the ICs U90 and U91 (74HC541). The enable signals for U90 and U91 are generated directly by the CPU.

All active low input signals are pulled down and all active high signals are pulled up with 4k7 resistors, to prevent unintended logic states.

The HPCU keyboard gets a bitstream with 40 bits to show the amplifier status, forward and reflected power. Each bit takes a single clock pulse, and all 40 bits (a

block) require one strobe pulse. This strobe latches (updates) the display data in the keyboard.

The control of the HPCU via keyboard uses an asynchronous serial interface. The master CPU has such an internal UART. Signal KBTXD is the transmit data line from the CPU and signal KBRXD carries the data received from the keyboard. The protocol used meets the BRUKER SBS BUS requirements (see software description).

Two of the keyboard switches are handled directly (not via SIO) by the slave CPU. These are Toggle and Fine. These signals are connected to interrupt inputs on the CPU.

All signals, SIO, bitstream and switches, between HPCU and keyboard meet RS485 requirements. This is realized with U100 (75174) for HPCU outputs, and U99 (75175) for HPCU inputs. All differential RS485 signals are terminated with resistors.

The keyboard receives a supply voltage of about 10.9V. This voltage is generated with two diodes from the +12.3V HPCU supply. The keyboard does not require a stabilized supply. It has an internal regulator.

All keyboard signals and the supply voltage are fed to a female 25 pin Mini-D connector.

Motor / Interface *2.4*

This module (sheet 9) consist of the Main CPU, Motor Control, RS232/SBS Interfaces, Gain Control and an Expansion Slot module. Most of the chip select signals used in this module are generated in the PAL, U41, 22V10. The signal /IOENABLE enables access to the hardware in the application mode. In the boot mode, this signal disables hardware access because the FLASH EPROM occupies the external memory space. There is only one exception: The /X32ENABLE signal enables the access to the serial interface regardless of the mode selected. This is necessary because the master CPU gets the download information over the serial interface.

Main CPU *2.4.1*

IC U32 an 80C515AN is the master CPU of the B-HPCU (sheet 15). It has access to about 32KByte RAM (U10) and to 64KByte firmware located in U24 (application mode). In boot mode the CPU only has access to the 64KByte boot EPROM (U11) and to the 64KByte FLASH EPROM mapped into the external RAM area. The voltage needed to erase the FLASH is generated in the Bit Stream CPU module.

The IC U13 is I^2C Bus E^2PROM, controlled by the master CPU. It can store 2KByte system information without battery backup. SDA is the serial data line and SCL is the clock signal.

IC U26 74HCT245 is a bus transceiver which decouples the CPU kernel from the other peripherals.

The PAL U40 22V10 generates all chip selects, read and write signals used on this sheet. Additionally the signals /X32ENABLE and /IOENABLE are generated (see above).

IC U38 is used as a simple driver to increase the fan out of some important signals.

IC U12 generates a hardware reset when either the reset button (to rear of unit) is pressed or when the supply voltage VCC drops below 4,75V. The connector X1 is currently not populated but it would allow a reset switch to be brought to the front panel.

Jumper J2 can be used to select the boot EPROM size. Connecting pin 1 and 2 together selects a 27C256 with 32KByte. Connecting pin 2 and 3 together selects a 27C512 with 64KByte. Pin 1 of jumper J2 is marked on the silk screen on the PCB.

IC U37 (74HC573) demultiplexes the data and address bus of CPU.

Q1 is a 12MHz quartz oscillator driving both CPUs the master and the slave. Q2 is an 8MHz quartz used to drive the serial interfaces.

The use of jumper J5 can modify the state of port pin P17. This port pin is used to switch between the boot- and application software. After reset this pin is „HIGH“ by default. When the CPU switches to the application, it sets the P17 to „LOW“. If jumper J5 is set, the boot programme will be ignored. After a reset the CPU starts to work in the application software (firmware).

NOTE: If jumper J5 is set, no download can be done!

To avoid problems, the header for jumper J5 is not factory populated. J5 should only be used for tests, to see if the CPU is the boot or download mode. It can also be used for temporary work with the HPCU, if a boot programme does not work together with a new downloaded application programme, until the problem is fixed.

Gain Control *2.4.2*

The Gain Control module (sheet 10) is split into Logic Control, Registers Gain H and X and into a Digital to Analog section module.

Digital to Analog Section

On this module (sheet 13) are located two DACs and their output stages. The structure of the X and H channels is identical, so only the H channel is described.

U9 is a 12Bit DAC DAC80CBI-I (BURR BROWN) with a current output of 0-2mA. It has its own reference voltage of 6.3V and resistors for internal current/voltage conversion. The internal resistors used are 5k*, 3k*, and 2k*. Only this type of DAC is allowed to be used. Other (so called) compatible replacements can have a different internal reference voltage (affecting different current sources) and associated internal impedances. The HPCU also needs external resistors (R115 and R116) to generate output voltages between 0-2.5V. If a replacement DAC has a different current source, the output voltage will differ.

The OP-amp AD846 is used together with the feedback resistors R115 and R116 as a current to voltage converter. The absolute feedback resistor value is calculated by R internal parallel R external = 1.25k*. This results in a full scale voltage of 2mA * 1.25k* = 2.5V. The capacitor C4 prevents overshoot.

Changing any part in the analog signal path (DAC, OP-amp, resistors) requires a new offset and full scale adjustment. It is important that offset is adjusted before

full scale. Before any adjustment, the B_HPCU should be allowed time to warm up.

For H channel offset adjustment, apply the digital input code 0xFFFhex to give 0V from the DAC. This can be done either manually with the HPCU-Keyboard via FH GAIN 1 switch, entering the value 0 with the code wheel, or by software, sending the H 1 W 0 0 0 0 0 10 hex command via X32 RS232 link. In both cases, hard or software, make sure that RCP5_SH and RCP6_SH are both „LOW“. Now turn potentiometer P4 until the voltage at test point TP9 is 0,0000V.

Now the full scale adjustment can be done. Push FH GAIN 1 switch on the keyboard and enter 4095 with the code wheel, or send the H 1 W 0 0 0F hex FF hex 0 10 hex command via X32 RS232 link. Now turn potentiometer P3 until the voltage at test point TP9 is 2,5000V.

For X channel offset adjustment, apply the digital input code 0xFFFhex to give 0V from the DAC. This can be done either manually by the HPCU-Keyboard via FX GAIN 1 switch, entering the value 0 with the code wheel, or by software, sending the H 5 W 0 0 0 0 0 10 hex command via X32 RS232 link. In both cases, hard- or software solution, make sure that REGSEL5SX and REGSEL6SX are both „LOW“. Now turn potentiometer P2 until the voltage at test point TP10 is 0,0000V.

Now the full-scale adjustment for X channel can be done. Push FX GAIN 1 switch on the keyboard and enter 4095 with the code wheel, or send the H 5 W 0 0 0F hex FF hex 0 10hex command via X32 RS232 link. Now turn potentiometer P1 until the voltage at test point TP10 is 2,5000V.

Registers GAIN H and X

The modules for Registers Gain H (sheet 11) and Registers Gain X (sheet 12) have the same structure, so only Registers Gain H is described.

This module consist of 4 register pairs, each containing one 8 Bit register 74ALS574 and one 4 Bit register 74LS173. Each register pair stores 12 bit DAC data, provided at D[0..11]. The clock and enable signals are generated for each pair separately in the Logic Control module (sheet 14). The enable and clock signals are used for both ICs in the pair. Only one register pairs' output can be activated at any time. If all register pairs outputs are disabled, resistor networks with 4,7k* provide a true „HIGH-LEVEL“ for DACs. In this case a DAC outputs 0V

Logic Control

The signals RCP5_SH, RCP&_SH, REGSEL5SX and REGSEL6SX are terminated with a 50* resistor (sheet 14). These signals drive the positive input of differential line receiver U5 a 75175. This chip is used as a simple comparator. The negative inputs of these receivers are driven by 1.21V reference voltage generated by U52, R22 and R23. The receiver chip has an input sensitivity of * 200mV and an additional hysteresis of 50mV for increased noise immunity.

The switching time over full scale is * 1*s measured at the analog output voltage.

This means:

Low-level input voltage max.:	1.01V
High-level input voltage min.:	1.41V
Undefined input voltage range:	1.06V - 1.36V (due to hysteresis)

The comparator outputs RCP[0..3] are fed to a PAL U35 (18P8). It acts as a multiplexer with an enable and additional common input select. The /ENABLE (P42) and INPUT SELECT (P43) signals are generated by the master CPU.

If /ENABLE is „HIGH“, all PAL outputs are „HIGH“. This means, all DAC registers outputs are disabled; the DACs output 0V. If /ENABLE is „LOW“, one of /ENX[0..3] and one of /ENH[0..3] outputs are „LOW“; one DAC register from each of H and X channels is selected.

If the INPUT SELECT signal (P43) is „HIGH“, the state of RCPH[1..0] and RCPX[1..0] selects the ENH[0..3] and ENX[0..3] outputs. If INPUT SELECT is „LOW“ the CPU signals P4[1..0] are taken to select ENH[0..3] and ENX[0..3] outputs, common.

The following tables show the relationship between RCP5_SH, RCP6_SH and the DAC register selected.

Table 2.2. Relationship between RCP5_SH,RCP6_SH and DAC.

RCP6_SH (RCPH1)	RCP6_SH (RCPH0)	Keyboard switch FH GAIN 1	Keyboard switch FH GAIN 2	Keyboard switch FH GAIN 3	Keyboard switch FH GAIN 4
0	0	selected	---	---	---
0	1	---	selected	---	---
1	0	---	---	selected	---
1	1	---	---	---	selected

The following tables show the relationship between REGSEL5SX, REGSEL6SX and the selected DAC Register.

Table 2.3. Relationship between REGSEL5SX,REGSEL6SX and DAC.

REGSEL6SX (RCPX1)	REGSEL5SX (RCPX0)	Keyboard switch FX GAIN 1	Keyboard switch FX GAIN 2	Keyboard switch FX GAIN 3	Keyboard switch FX GAIN 4
0	0	selected	---	---	---
0	1	---	selected	---	---
1	0	---	---	selected	---
1	1	---	---	---	selected

The PAL IC U39 generates the clock signals to store the DAC data in its corresponding registers (see module Registers Gain H and X, sheet 11 and 12). The CPU used, has an 8Bit data bus. Because of the 12Bit DACs, 4 further bits must be stored elsewhere to simulate a 12Bit data bus. This is done with register U22 an 74LS173. If a 12Bit data word is to be written into one of the 8 DAC register pairs, the highest 4 bits D[8..11] must first be written into U22 (pre-stored). The next write cycle, writes the lower 8 bits from the CPU data bus together with those 4 pre-stored bits into the desired register pair. PAL U39 also generates the clock signal for U22.

RS232/SBS Interfaces *2.4.3*

This RS232/SBS Interfaces module consists of the serial RS232 to X32 link, a serial RS232 link to the MAS and serial SBS Bus interface based on RS485 to the 4 Channel Modulator.

RS232 to X32

The serial link between X32 and B-HPCU (sheet 19) via RS232 is galvanically isolated. This is done with the optocouplers U75, U85 HPCL-2731 and U78 an MCT6. The galvanically isolated voltages +/-12V are provided by U42, a DC/DC converter.

The IC U76 converts the interface output voltages to RS232 requirements. The diodes D32, D33 and D34 protect the optocoupler inputs.

U56 is the UART NS16550 with an internal baud rate generator. It has a 16 byte FIFO buffer for receiving and transmitting data. It is programmed to 9600 baud, 8 data bits, one stop bit and no parity. The baud rate is derived from an 8MHz crystal oscillator located on sheet 15. It generates an interrupt to the master CPU either when the receiver FIFO trigger level is reached, or when a complete byte has just left the transmitter output shift register.

All handshake outputs signals are supported. DTR shows to the host, that the B-HPCU is on-line (present). RTS shows that the B-HPCU is able to receive data. It is controlled by two signals. /RXRDY (UART pin 32) goes „LOW“, when the receive FIFO trigger level is reached to stop the host CPU transmitting more data. Output /RTS can disable the host CPU, by programming this pin to „HIGH“. However the RTS signal is not used by X32 software yet because the B-HPCU and the X32 use a software protocol. The input handshake signals are also not used, but they can be programmed if necessary.

The jumper, J4, can be used to connect AGND (header pin 2) and XGND (header pin 1) together to forego the galvanic isolation. This jumper must not be in place during normal use of the B-HPCU. Setting this jumper is only intended to allow an easier test of the serial link.

Jumper J1 is used in the boot software. If set, the CPU does not jump into the application software. It waits in the boot software for a new download. It should only be set for test or if a defect application software was successfully downloaded, which itself does not allow another new download with a correct software. The bit used for jumper J1 can be read via the UART. Another digital input signal DIN0, fed to the Expansion Slot module, can also be read by the UART.

RS232 to MAS

IC U55 (sheet 17) is an UART NS16550 with internal baud rate generator. It has a 16 byte FIFO buffer for receiving and transmitting data. It is programmed to 9600 baud, 8 data bits, 8th. data bit always „HIGH“, one stop bit and no parity. The baud rate is derived from an 8MHz crystal oscillator located on sheet 15. It generates an interrupt to the master CPU either when the receiver FIFO trigger level is reached; a complete byte has just left the transmitter output shift register; or when a hardware input handshake signal DSR2 or CTS2 changes its logical state from „HIGH to LOW“ or „LOW to HIGH“.

IC U84 converts all serial out and input signals between TTL and RS232 requirements.

All handshake output signals are supported. DTR2 tells the MAS, that the B-HPCU is on-line (present). RTS2 shows that the B-HPCU is able to receive data. It is controlled by two signals. /RXRDY (UART pin 32) goes immediately „LOW“, when the receive FIFO trigger level is reached to stop the MAS transmitting more data. Output /RTS2 disables the MAS, by programming this pin to „HIGH“.

The B-HPCU only sends data to the MAS when the handshake signal CTS2 is active (ca. +12V). The DSR2, RI2 and DCD2 input signals are ignored.

The UART is used to store the levels of the CWX and CWY (Continuous Wave X/ Y) signals. These signals are gated with the CWALL signal. The gated signals are fed to the 25pin amplifier connectors. The CW X/Y signals are „LOW“ when CWALL is „LOW“ too. If CWALL is switched to „HIGH“, CW X/Y get the levels preset into the UART (/OUT1 and /OUT2).

SBS Bus Interface

The IC, U54, (sheet 18) is a UART, NS16550, with internal baud rate generator. It has a 16 byte FIFO buffer for receiving and transmitting data. It is programmed to 9600 baud, 8 data bits, one stop bit and no parity. The baud rate is derived from an 8MHz crystal oscillator located on sheet 15. It generates an interrupt to the master CPU either when the receiver FIFO trigger level is reached, or a complete byte has just left the transmitter output shift register.

ICs U70 and U77 convert all serial out- and input signals between TTL and RS485 requirements. The transmit and receive signals are terminated by resistors R35, R36, R37 and R59.

The UART /RTS output is used to enable/disable the transmitter U70. If it is disabled, its outputs are in a high impedance condition.

The UART /DTR output is used to wake up a sleeping slave device on the SBS Bus.Diodes D24 and D25 and resistor R50 protect transistor T5 from over voltages and short circuits. The /RI input pin can be used to read back the wake up signal, /WUP, in test mode. Signal TXDTEST can be used in test mode to read back the level of the TXD+/- signal on the SBS Bus.

The UART is used to store the levels of the CWH and CWA (Continuous Wave H/ Auxiliary) signals. These signals are gated with the CWALL signal. The gated CWH signal is fed to the 25pin amplifier connector H. The CW H/A signals are „LOW“ when CWALL is „LOW“, too. If CWALL is switched to „HIGH“, CW H/A get the levels preset into the UART (/OUT1 and /OUT2).

The SBS Bus slaves can be supplied with +12V direct from the SBS Bus. This is useful for galvanically isolated SBS Bus slaves.

Motor Control *2.4.4*

The Motor Control module (sheet 20) consists of the 1H-, X Amplifier Motor Control, and the Amplifier Control Connectors modules.

1H and X Amplifier Motor Control

These modules include drivers for 4 independent stepper motors, one for the X amplifier (sheet 23) and three for the 1H amplifier (sheet 21). The motors move the capacitors in the amplifiers. These 4 drivers have all the same schematic. The function is therefore described only for 1H-input tuning.

The 4 bit register U49 stores the information for the IC U65:

D0	enable / disable the motor (current)	„HIGH“ enabled
D1	motor rotation direction	(depends on connection of windings)
D2	full / half step rotation	„HIGH“ half step
D3	motor control circuit L297 reset	„LOW“ reset

The IC U65 is a motor driver control chip. An on-chip PWM chopper circuit permits switched mode control of the current in the motor windings. It is used together with the monolithic bridge driver IC U87 an L298. Resistor R57 and capacitor C141 determine the chopper rate of about 50kHz. This frequency is fed to all L297 chips via their SYNC in-/output.

The current through the motor windings is measured using 1* resistors R89 and R90. The reference voltage for the chopper comperator is generated with R19, R20, R21 and T1. This reference voltage is 0.49V when the motors turn, supplied by R20 and R21 (T1 switched off). The corresponding motor current is 490mA in each winding. If the motor stops turning, transistor T1 switches R19 in parallel to R20 after few seconds. The reference voltage becomes 0,26V. This reduces the current limit to 260mA per winding. This current will be enough to hold the motor in position after all steps are done. The current limit setting is controlled by the master CPU via port pin P44 signal MOPOWSEL.

IC U65, a dual full-bridge driver, can drive up to 2A (DC) into a motor winding. The emitters of the lower transistors of each bridge are connected together for easy sensing of the winding current. Its total power dissipation is 25W so it is mounted, together with another driver, onto a heatsink. The driver output stages are protected by soft recovery diodes, type BYW72.

The motors can all be switched off by the B-HPCU keyboard with the switch named MOTOR POWER. This is electrically done with 3 transistors. T16 and T17 are used as high-side switches. They are paralleled to reduce the on resistance. Transistor T4 generates the control signal for the high-side switches with help of R72 and R81.

The voltage, fed to the motor drivers, can be measured with help of R73, R74 and C177, via an analog input in the master CPU. Diode D37 is used to protect this input.

Amplifier Control Connectors

This module (sheet 22) shows all three amplifier connectors 37pin Mini D.

X11 is used for Y amplifier. There are only amplifier status signals and one identifier signal connected.

X9 is used for the 1H amplifier. Three stepper motors can be controlled via this connector, MHIN[0..3],MHOUT[0..3] and MAUX[0..3]. The third motor MAUX[0..3] is for future use. The signals LIGHTPOT (output tuning) and LIGHTB1 (input tuning) are used to find the zero position of the amplifier output tuning motor. If the amplifier

is a non cavity type, a light barrier is connected to the LIGHTPOT signal. LIGHTB1 signal is connected to a light barrier all the time, in both cavity and non cavity amplifiers.

In cavity amplifiers, a linear potentiometer is used to find the absolute position of the output tuning motor after power on. The analog potentiometer signal passes through an OP-amp IC (U83A). This OP-amp adds an offset voltage to the position signal to compensate for the mechanical adjustment offset. The offset

voltage added can be controlled via the B-HPCU keyboard (in the service menu) and U82 an E^2POT to adjust the zero position.

The max. position adjustment can also be done with the E^2POT, U81, and the B-HPCU keyboard. The output voltage of OP-amp, U83A, is fed to the resistor network R62, R69 and U81. Changing the resistance of U81 will modify the voltage fed to the analog inverter OP-amp, U83C. The output voltage of U83C corresponds to the motor position. It can be read by a master CPU's analog input.

The zero position is necessary to assure that the auto tuning always starts tuning from the same position. In cavity type amplifiers the stepper motors cause a linear movement. To prevent mechanical damage, limit switches are located at the min. and max. limits. These switches can be read via signals LIMSW[0..3].

The X amplifier must be connected to X10. MXOUT[0..3] controls the X output tuning motor. The MAUX[0..3] motor driver signals are also fed to this connector, in case they are required in any future version.

All limit switches, light barriers and the ID bit of the Y amplifier can be read in by IC U68. The other ID bits of the 1H and X amplifiers can be read in by IC U69. To avoid undefined logic levels, these inputs are pulled up by resistor networks IR8 and IR9.

The 28V required for the stepper motor is supplied to the PCB over pins TP36 and TP42.

An additional analog/digital input SENS2 is connected to amplifier connector X9 and X10 for possible future use.

The table shows all identifier codes of X and 1H amplifiers.

Table 2.4. Identifier Codes of X and 1H Amplifiers.

Identifier code digital	Identifier code analog	Amplifier type
0	-	200 MHz X-BB,
1	-	300 MHz X-BB,
2	-	360 MHz X-BB,
3	-	400 MHz X-BB,
4	-	future used amplifier,
5	-	future used amplifier,
6	0	100MHz 1H/X amplifier,
6	1..9	future used amplifier,
7	-	AMT amplifier (no motors),
8	-	200MHz 1H/19F amplifier,
9	-	300 MHz 1H/19F,
10	-	360 MHz 1H/19F,
11	-	400 MHz Cavity 1H/19F,

Table 2.4. Identifier Codes of X and 1H Amplifiers.

Identifier code digital	Identifier code analog	Amplifier type
12	-	500 MHz Cavity 1H/19F,
13	-	600 MHz Cavity 1H/19F,
14	0	750 MHz Cavity 1H/19F
14	1	400 MHz Cavity 1H/19F
14	2..9	future used amplifier,
15	-	amplifier missing,

The 1H amplifier type can be read in by the signal AMPHID[0..3]. The X amplifier type can be read in by the signal AMPHID[4..7]. All amplifiers with digital ID 6 (X amplifiers) have an additional analog identifier. This ID can be read in via Sens0 at connector X10 pin 3. All amplifiers with digital ID 0Ehex (1H amplifiers) also have an additional analog identifier. This ID can be read in via Sens1 at connector X9 pin 13.

Expansion Slot *2.4.5*

The expansion slot (sheet 24) is a 96pin VG connector. The CPU data bus, /RD,/ WR and addresses A[0..7] are fed to this connector. Signal /F0 together with / IOENABLE enables the access to external hardware connected.

Part of the PAL listing:

```
$(A[7..0]),        IOENABLE,     RD, WR : F0;   address range
20H..FFH,              0     ,   -   ,  -   : 0 ;   8020H..80FFH
 REST                                          : 1   ; ---
```

A hardware reset can be performed with signal /RESET during power on. Three further analog AN[0..2] and digital DIN[0..3] input signals can be processed by the master CPU. Signal P14 can be used as a digital in/output or as a „HIGH/LOW“ transition interrupt input. All voltages used on the PCB are also fed to the connector. Expansion Board

The B-HPCU Expansion board (H5764) can be used to control the signals HP Relay H-, HP Relay X-, HP Relay Y-, HP Relay Z- and HP Rack On/Off. This board can be plugged into all B-HPCU Electronic racks (H5336) with ECL02 and serial number 0046. A cable (HZ04003) connects the expansion board with the 34pin Burndy connector at the B-HPCU’s front panel. The expansion board contains a EEPROM chip U3 with the BBIS data version 2.

To allow software control of the HP Relay / On /Off signals the following B-HPCU firmware is needed:

Master boot firmware: 921124 or newer.

Master application firmware: 940527 or newer.

The boot and application firmware of the slave CPU are not relevant.

The HP Relay /On/Off signals are generated with an I2C bus IC PCF8574. The I2C bus protocol must be used to change the output information of this chip. This „complicated protocol“ was chosen, to prevent data loss (wrong output states) caused by unintended CPU resets, software errors or spices. The output signals of this chip are converted to +/-12V levels to obtain high noise immunity.

The clock signal for the I2C bus is generated with CPU’s D0 at address 807Fhex. It is stored in the D type Flip-flop U4 74HC74. The I2C data signal is performed by P14. IC U7 and U6 are used for address decoding. data 3 (n=3) data byte

Summary of jumper settings

3

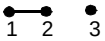
The default factory setting is marked with.

Jumper J1: 

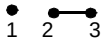
select EPROM 27C256 (stay in boot software)

select EPROM 27C512 (application software allowed)

Further details see chapter: RS232 to X32.

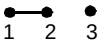
Jumper J2: 

select EPROM 27C256 (master boot software)

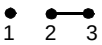


select EPROM 27C512 (master boot software)

Further details see chapter: Main CPU.

Jumper J3: 

select EPROM 27C256 (slave boot software)



select EPROM 27C512 (slave boot software)

Further details see chapter: Bit stream CPU.

Jumper J4: 

RS232 interface to X32 not galvanically isolated

galvanically isolated RS232 interface to X32

Further details see chapter: RS232 to X32.

Jumper J5: 

select EPROM 27C256 (ignore master boot software)

select EPROM 27C512 (perform master boot software)

Further details see chapter: RS232 to MAS.

Summary of jumper settings

Connectors' pinout

4

Figure 4.1. RS232 Connector to X32 9 pin male -Mini D-

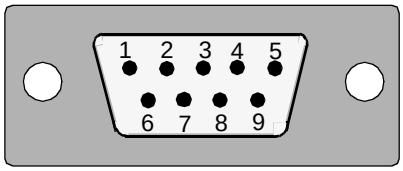


Table 4.1. RS232 Connector Pinout to X32

Pin	
1	-
2	/RxD
3	/TxD
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	-

Figure 4.2. RS232 Connector to MAS 9 pin male -Mini D-

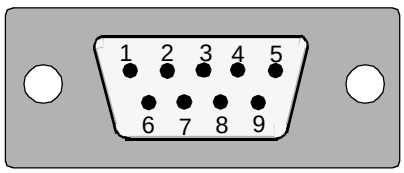


Table 4.2. RS232 Connector Pinout to MAS

Pin	
1	DCD
2	/RxD
3	/TxD
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	RI

Figure 4.3. SBS-Bus Connector to 4Ph. Modulator 15pin female-Mini D-

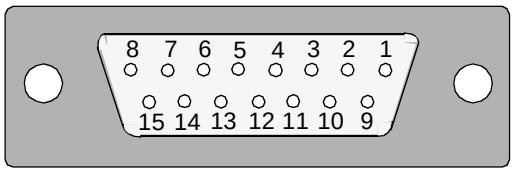


Table 4.3. SBS-Bus Connector Pinout to 4 Phase Modulator

Pin	
1	-
2	RXD+
3	/WUP
4	TXD+
5	-
6	GND
7	GND
8	GND
9	RXD-
10	-
11	TXD-
12	-
13	VRS
14	VRS
15	VRS

Figure 4.4. B-HPCU Keyboard Connector 25 pin female -Mini D-

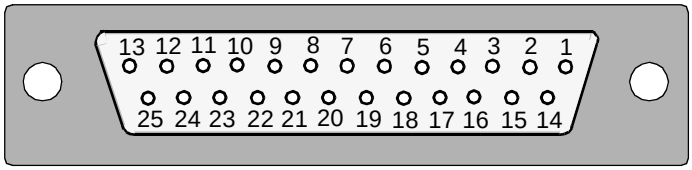


Table 4.4. B-HPCU Keyboard Connector Pinout

Pin		Pin	
1	AGND	14	-
2	-	15	DLED-
3	DLED+	16	DCLK-
4	DCLK+	17	DSTRB-
5	DSTRB+	18	TOGGLE-
6	TOGGLE+	19	FINE-
7	FINE+	20	AGND
8	AGND	21	AGND
9	AGND	22	10.9V
10	10.9V	23	10.9V
11	10.9V	24	KBC_TXD-
12	KBC_TXD+	25	KBC_RXD-
13	KBC_RXD+		

Figure 4.5. Amplifier 1H Connector 37 pin female -Mini D-

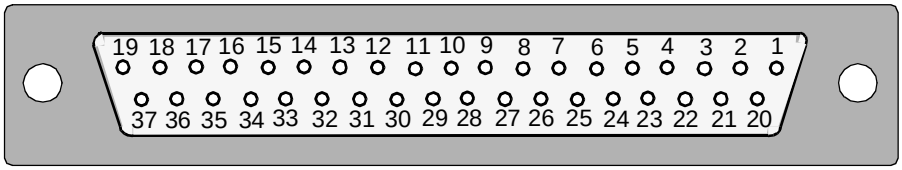


Table 4.5. Amplifier 1H Connector Pinout

Pin		Pin	
1	-	20	MAUX3
2	MHIN0	21	LIMSW0
3	-	22	LIMSW1
4	MHIN1	23	LIMSW2
5	-	24	LIMSW3
6	MHIN2	25	LIGHTPOT
7	-	26	LIGHTB1
8	MHIN3	27	AMPHID0
9	-	28	AMPHID1
10	MHOUT0	29	AMPHID2
11	SENS0	30	AMPHID3
12	MHOUT1	31	CWH
13	SENS1	32	-
14	MHOUT2	33	-
15	SENS2	34	-
16	MHOUT3	35	-
17	MAUX0	36	+12V
18	MAUX1	37	AGND
19	MAUX2		

Figure 4.6. Amplifier X Connector 37 pin Female -Mini D-

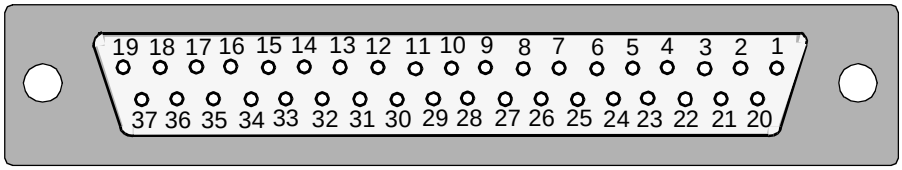


Table 4.6. Amplifier X Connector Pinout

Pin		Pin	
1	-	20	-
2	MAUX0	21	/OVDRX
3	SENS0	22	/PUWIX
4	MAUX1	23	/DUCLX
5	SENS1	24	/OVHEX
6	MAUX2	25	LIGHTB2
7	SENS2	26	-
8	MAUX3	27	AMPXID0
9	-	28	AMPXID1
10	MXOUT0	29	AMPXID2
11	-	30	AMPXID3
12	MXOUT1	31	CWX
13	-	32	HVX
14	MXOUT2	33	-
15	-	34	-
16	MXOUT3	35	-
17	-	36	+12V
18	-	37	AGND
19	-		

Figure 4.7. Amplifier Y Connector 37 pin Female -Mini D-

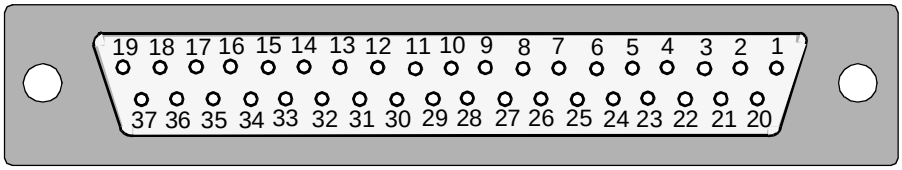


Table 4.7. Amplifier Y Connector Pinout

Pin		Pin	
1	-	20	-
2	-	21	/OVDRY
3	-	22	/PUWIY
4	-	23	/DUCLY
5	-	24	/OVHEY
6	-	25	-
7	-	26	-
8	-	27	-
9	-	28	-
10	-	29	-
11	-	30	AMPYID0
12	-	31	CWY
13	-	32	HVY
14	-	33	FORY
15	-	34	REFY
16	-	35	-
17	-	36	+12V
18	-	37	AGND
19	-		

Figure 4.8. Control Connector 34 pin Burndy RH

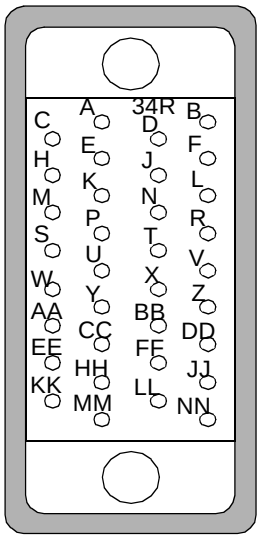


Table 4.8. Control Connector Pinout

Pin	DC Signals	Pin	HF Signals
A	HVH	B	RCP5_SH
D	*HP Relay H-	C	FORW1H
E	HVX	F	RCP6_SH
J	*HP Relay X-	H	REFL1H
K	---	L	REGSEL5SX
N	*HP Relay Y-	M	REFIN1H
P	---	R	REGSEL6SX
T	*HP Relay Z-	S	FORWX
U	---	V	GAIN H
X	*HP Rack on/off	W	REFLX
Y	---	Z	GAIN X
BB	---	AA	CWMODE
CC	---	DD	BLANK H
FF	---	JJ	BLANK X
HH	---	NN	BLANK Y
LL	---		
MM	GND (P-Supply)		
EE	+28V		
KK	PGND(+28V)		

*Only available using the B-HPCU Expansion Board.

HPCU Master (HPCM) Commands 5

System Commands 5.1

Power Up 5.1.1

H A

Performs a software reset. Returns the (4 characters) identification short string of the B-HPCU.

Reply: H A H P C M

Error Accept 5.1.2

H E

Clears an error in the B-HPCU. The B-HPCU accepts the next command only if all errors are cleared.

Reply:

no more errors:	H E	
else:	H Flex X0 X1 X2 X3 X4 S1..Sn	(next error)

X0: Error code. (00_{hex}..FF_{hex})

X1... X4: Short identifier string.

S1...Sn: Error text string terminated with a 0_{hex}.

Example:

→	H S	;any command
←	H FF_{hex} 20 H P C U O r d e r e r r o r 0_{hex}	;Error message
→	H E	;Error accept
←	H FF_{hex} 2 H P C U C h e c k s u m e r r o r 0_{hex}	;more errors
→	H E	;Error accept
←	H E	;no more errors

Change Accept (Query) *5.1.3*

H Q

With this command a system change is reported.

Reply: H Q ;if no changes have occurred

Configuration *5.1.4*

H K

Sends system configuration data (recognized at power on) to the host computer.

Reply: H K X0 X1..Xi

X0: Number of units present in the system.

X1.. Xi: Device identifier of units present.

System Order Commands *5.1.5*

Read BBIS (Bruker Board Information System)

H Z ?

Read Bruker Board Information System from serial EEPROM.

Write BBIS

H Z !

Write Bruker Board Information System to serial EEPROM.

Start program download

H Z Z

When sent twice this command loads down a new application program onto the HPCU's FLASH-EPROM. The file downloaded must be in INTEL HEX format.

Example:

→ H Z Z

← H Z Z

→ H Z Z

NOTE: After the second HZZ, the application program is deleted to allow reprogramming of the FLASH EEPROM!

← H Z Z

→ H: ;now send first record of source file

← H O K ;record is programmed error free

repeat last two points until the last record is programmed

→ H: ;now send End-Of-File record

← H F F ;end of download, B-HPCU performs reset

Version

H Z V X1

Sends information about the unit version.

X1: H = Hardware

D = First Piggy Board

L = Second Piggy Board

B = Boot software

A = Application Software

K = Kernel Software

C = Checksum Application

E = Checksum Boot EPROM

Reply: H Z V X1 X2 X3 X4 X5 X7

if X1 = B or A

X2: year

X3: year

X4: month

X5: month

X6: day

X7: day

Example:	X2	X3	X4	X5	X6	X7	
	9	3	0	2	0	8	8. Feb. 1993

if X1 = H, D or L

X2: HW-code

if X1 = C or E

X2: high byte

X3: low byte

Check System

H Z C

Checks all parts of the unit. If an error occurs an error message is returned.

Reply: H Z C ;everything is OK

H FF_{hex}.... ;error message (see error messages)

Processor Sleep

H Z D..X1

This is a dummy command. It is inserted to keep it compatible with other units, using the standard (BSMS) system commands.

Reply: H Z D $X1_{hex}$

X1: $00_{hex}/01_{hex}$ will be ignored

HPCM Application Specific Commands *5.2*

Amplifier Gain *5.2.1*

H X1 X2 D1 D2 D3 D4 Exp Flags

Loads the DAC registers for the 1H and the X amplifiers. Each amplifier can be preset with 4 gain values. The different gains can be selected by hardware RCP pulses (see hardware description).

X1: '1','2','3','4' Gain of H amplifier (for further details see hardware description)

X1: '5','6','7','8' Gain of X amplifier (for further details see hardware description)

X2: 'R' or 'W'

D1,D2..D4: Gain value 0000hex..0fffhex (D1..D2 always 00hex)

Exp.: 00_{hex}

Flags: 10_{hex}value in range, or 11_{hex} if value out of range

Reply: H X1 X2 D1 D2 D3 D4 Exp. Flags

Example: → H 5 W 0 0 3F FF 0 10_{hex}

← H 5 W 0 0 0F FF 0 11_{hex}

If the gain value is out of range it will be reduced to its limit in the reply string.

Tune (Motor) Position *5.2.2*

H X1 X2 D1 D2 D3 D4 Exp Flags

Loads the motor position (tune position) of 1H and X amplifiers.

X1: 'I', tune FH In

X1: 'O', tune FH out

X1: 'o', tune FX out

X2: 'R' or 'W'

D1,D2.D4: tune position. The range starts at 0000_{hex} The max. position value depends on amplifier type.

Exp: 00_{hex}

Flags: 10_{hex} value in range, or 11_{hex} if value out of range

Reply: H X1 X2 D1 D2 D3 D4 Exp Flags

Example: → H I W 00 00 01 FF 00 10_{hex}

← H I W 00 00 01 FF 00 10_{hex}

If the position value is out of range it will be reduced to its limit in the reply string.

Read Amplifier Type *5.2.3*

H h X1

Reads the amplifier type connected to the HPCU.

X1: 'H', read 1H/19F amplifier type

X1: 'X', read X amplifier type

Reply: H h X1 X2 S1..Sn

X1 'H' or 'X' (depending on command).

X2: amplifier type identity code.

S1...Sn: amplifier type identity string, terminated with a 0hex.

Example: → H h H

← H h H $0D_{hex}$ 6 0 0 M H z 1 H / 1 9 F, I D 0 D H 0_{hex}

or ← H h H $0E_{hex}$ u n k o w n a m p l i f i e r, I D 0 E H 0_{hex}

or ← H h H $0F_{hex}$ a m p l i f i e r m i s s i n g I D 0 E H 0_{hex}

Set CW mode bits *5.2.4*

H y X1 D1 D2 D3 D4 Exp Flags

Presets the CW mode register in the HPCU. These 3 bits are gated with the CW ALL signal by hardware. A CW(H,X,Y) bit, set high, passes through an „AND" gate then the CW ALL signal becomes „HIGH". If the CW ALL signal is „LOW", all CW mode output signals are also „LOW „.

X1: 'R' or 'W' (read or write data).

D1,D2..D3: always 00_{hex}.

D4: Bit 0, bits 4 to 7 are always 00_{hex}. All other 8 „High" / „Low"; combinations of bit 1, bit 2 and bit 3 are possible.

Table 5.1. Combinations of CW bits

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
all CW bits off	0	0	0	0	0	0	0	0
CWH on	0	0	0	0	0	0	1	0
CWX on	0	0	0	0	0	1	0	0
CWY on	0	0	0	0	1	0	0	0
all CW bits on	0	0	0	0	1	1	1	0

Exp: 00_{hex}

Flags: 00_{hex}

Example:

→ H y W 00 00 00 06_{hex} 00 00 set CWH and CWX bit (signal).

← H y W 00 00 00 06_{hex} 00 00

Set HP mode bits *5.2.5*

H i X1 D1 D2 D3 D4 Exp Flags

Sets the output level of the Expansion Board to either +12V or -12V. The default value after reset is -12V. A bit of D4 set to a logical ‘1’ sets the output level to -12V. A logical ‘0’ at D4 sets +12V.

X1: ’R’ or ’W’ (read or write data).

D1,D2..D3: always 00_{hex}.

D4: Bits 5 to 7 are always 00_{hex}. All other „High“ / „Low“ combinations of bits 0.. 4 are possible.

Table 5.2. Combinations of HPR bits

	Bit 7	Bit 6	Bit 5	Bit 4	Bits 3	Bits 2	Bits 1	Bits 0
all outputs off (-12V)	0	0	0	0	0	0	0	0
HPRelayH- on (+12V)	0	0	0	0	0	0	0	1
HPRelayX- on (+12V)	0	0	0	0	0	0	1	0
HPRelayY- on (+12V)	0	0	0	0	0	1	0	0
HPRelayZ- on (+12V)	0	0	0	0	1	0	0	0
HPReack - on (+12V)	0	0	0	1	0	0	0	0

Exp: 00_{hex}

Flags: 00_{hex}

Example: → H i W 00 00 00 10_{hex} 00 00 (switches the rack power on)

← H i W 00 00 00 10_{hex} 00 00

if the „Expansion Board“ is not present or defective, an error message is returned.

Example: → H i W 00 00 00 10_{hex} 00 00 (switches the rack power on)

← H FF_{hex} 155 HPCU Hardware error 0_{hex}

Testing the presence of the „Expansion Board“, is possible via the HPCU version command for the first piggy pack board. This information (hardware code 00_{hex}..FF_{hex}) is contained in byte X2 of the reply (see HZV.. command). X2 =00_{hex}..FE_{hex} means board present.

Example: → H Z V D

← H Z V D FF_{hex} ;expansion board not present

or ← H Z VD 01_{hex} ;expansion board present with hardware ;code 01

HPCU Slave (HPCS) Commands

6

System Commands 6.1

Power Up 6.1.1

S A

Performs a software reset. Returns the (4 characters) identification short string of the B-HPCU (Slave CPU).

Reply: S A H P C S

Error Accept 6.1.2

S E

Clears an error in the B-HPCU slave CPU. The slave CPU accepts the next command only if all errors are cleared.

Reply:

no more errors: S E

else: S FF_{hex} X0 X1 X2 X3 X4 S1..Sn (next error)

X0: Error code. (00_{hex}..FF_{hex})

X1... X4: Short identifier string.

S1...Sn: Error text string terminated with a 0_{hex}.

Example:

→	S i	;any command
←	S FF_{hex} 20 H P C U Order error 0_{hex}	;Error message
→	S E	;Error accept
←	S FF_{hex} 2 H P C U Check sum error 0_{hex}	;more errors
→	S E	;Error accept
←	S E	;no more errors

Change Accept (Query) *6.1.3*

S Q

With this command a system change is reported.

Reply: S Q ;if no changes have occurred

Configuration *6.1.4*

S K

Sends slave system configuration data (recognized at power on) to the host computer.

Reply: S K X0 X1..Xi

X0: Number of units present in the system.

X1.. Xi: Device identifier of units present.

System Order Commands *6.1.5*

Start program download

S Z Z

When sent twice this command loads down a new application program onto the PCU's slave CPU FLASH-EPROM. The file downloaded must be in INTEL HEX format.

Example:

→ S Z Z

← S Z Z

→ S Z Z

NOTE: After the second S Z Z application program is deleted to allow reprogram is deleted to allow reprogramming of the FLASH EPROM!

→ S: ;now send first record of source file

← S O K ;record is programmed error free

repeat last two points until the last record is programmed

← S Z Z

→ S: ;now send End-Of-File record

← S F F ;end of download, B-HPCU performs reset

Version

S Z V X1

Sends information about the unit version.

X1: H = Hardware

D = First Piggy Board

L = Second Piggy Board

B = Boot software

A = Application Software

K = Kernel Software

C = Checksum Application

E = Checksum Boot EPROM

Reply: H Z V X1 X2 X3 X4 X5 X7

if X1 = B or A

X2: year

X3: year

X4: month

X5: month

X6: day

X7: day

Example:	X2	X3	X4	X5	X6	X7	
	9	3	0	2	0	8	(8. Feb. 1993)

if X1 = H, D or L

X2: HW-code

if X1 = C or E

X2: high byte

X3: low byte

Check System

S Z C

Checks all parts of the unit. If an error occurs an error message is returned.

Reply: S Z C ;everything is OK

S FF_{hex}.... ;error message (see error messages)

Processor Sleep

S Z D..X1

This is a dummy command. It is inserted to keep it compatible with other units, using the standard (B-SMS) system commands.

Reply: S Z D $X1_{hex}$

X1: $00_{hex}/01_{hex}$ will be ignored

MAS Handling 7

MAS Application Specific Commands 7.1

MAS General Command 7.1.1

MAS General Command

M C S1 S2.. .. S (n-1) Sn

Sends any MAS command in BSMS format via HPCU to the MAS. Each command has the „MC“ prefix and must be terminated with a Return character. The reply also has the „MC“ prefix and is terminated with Return, Line Feed.

S1,S2..S(n-1): any known MAS command

Sn: Return $0D_{hex}$

Reply: M C S1 S2 S3 .. S(n-2) S(n-1) Sn

S1,S2..S (n-2): any known MAS acknowledge

S (n-1) Return $0D_{hex}$

Sn: Line feed $0A_{hex}$

Example: → M C T W $0D_{hex}$

← M C O K 0Dhex $0A_{hex}$

If an error occurs, it will be sent back in the BSMS format (without the „MC“ prefix and without Return, Line feed).

Example: M FF_{hex} 73 M A S _ I N V A L I D D A T A 0_{hex}

System Commands 7.2

Power Up 7.2.1

M A

This is a dummy command to keep it compatible with BSMS system commands in other units. It returns the short identification string (4 characters) of the B-MAS.

Reply: M A M A S _

Error Accept *7.2.2*

M E

Clears an error in the MAS. The MAS accepts the next command only if all errors are cleared.

Reply: no more errors: M E

else: M FF_{hex} X0 X1 X2 X3 X4 S1..Sn (next error)

X0: Error code. (00_{hex}..FF_{hex})

X1... X4: Short identifier string.

S1...Sn: Error text string terminated with a 0_{hex}.

Example:

→	M S	;any command
←	M FF_{hex} 73 M A S _ I N V A L I D D A T A 0hex	;Error message
→	M E	;Error accept
←	M FF_{hex} 1 1 5 M A S _ V A L V E E R R O R 0_{hex}	;more errors
→	M E	;Error accept
←	M E	;no more errors

Change Accept (Query) *7.2.3*

M Q

With this command a system change is reported.

Reply: M Q ;if no changes have occurred

Configuration *7.2.4*

M K

Sends system configuration data to the X32. Because of the MAS has no slave units, only an M (for MAS) is returned.

Reply: M K 01_{hex} M

4Phase Modulator 8

System Commands 8.1

Power Up 8.1.1

Q A

Performs a software reset. Returns short identification string (4 characters) of 4 Phase Modulator.

Reply: Q A 4 P H _

Error Accept 8.1.2

Q E

Clears an error in the 4 Phase Modulator. The 4 Phase Modulator accepts the next command only if all errors are cleared.

Reply: no more errors: Q E

else: Q FF_{hex} X1 X2 X3 X4 S1..Sn (next error)

Example:

→	Q S	;any command
←	Q FF_{hex} 2 4 P H _ Some Error 0_{hex}	;Error message
→	Q E	;Error accept
←	Q FF_{hex} 13 4 P H _ Check Sum Error 0_{hex}	;more errors
→	Q E	;Error accept
←	Q E	;no more errors

Change Accept (Query) 8.1.3

Q Q

With this command all system changes are reported.

Reply: Q Q ;if no changes occured

Configuration *8.1.4*

Q K

Sends system configuration data for service tool.

Reply: Q K 1 Q ;for 4 Phase Modulator only

System Order Commands *8.2*

Read BBIS (Bruker Board Information System) *8.2.1*

Q Z ?

Read Bruker Board Information System in the serial EEPROM.

X1: selects which BBIS to read (0-7). in 4PM always 0

X2: Address of first Byte to read

Reply: Q Z ! X1 X2 D1 D2 - ... - D16

Write BBIS *8.2.2*

Q Z !

Write Bruker Board Information System in the serial EEPROM.

X1: selects which BBIS to write (0-7). in 4PM always 0

X2: Address of first Byte to write

X1 - D16 16 Byte BBIS Data

Reply : Q Z ! X1 X2 D1 - D16

Start Program Download *8.2.3*

Q Z Z

When sent twice this command clears the application program on the 4 Phase Modulator's FLASH-EPROM in order to reprogram it.

Example:

→ Q Z Z

← Q Z Z

→ Q Z Z

NOTE: After the second Q Z Z, the application is deleted to allow reprogramming of the FLASH EPROM!

← Q Z Z

→ Q: ;now send first record of source file

← Q O K ;record is programmed error free

repeat last two points until the last record is programmed

→ Q: ;now send End-Of-File record

← Q F F ;end of download, 4 Phase Modulator performs a reset

Version ***8.2.4***

Q Z V X1

Sends information about the unit version.

X1: H = Hardware

D = First Piggy Board

L = Second Piggy Board

B = Boot software

A = Application Software

K = Kernel Software

C = Checksum Application

E = Checksum Boot EPROM

Reply: Q Z V X1 X2 X3 X4 X5 X7

if X1 = B or A

X2: year

X3: year

X4: month

X5: month

X6: day

X7: day

Example:	X2	X3	X4	X5	X6	X7	
	9	3	0	2	0	8	(8. Feb. 1993)

if X1 = H or D

X2: HW-code

if X1 = C or E

X2: high byte

X3: low byte

Activate Watchdog *8.2.5*

Q Z Y

Activates the watchdog. The watchdog can only be stopped by a hardware reset. This is a dummy function, as the watchdog is activated on power up in the 4 Phase Modulator.

Reply: Q Z Y

Check System *8.2.6*

Q Z C

Checks all parts of the unit. If an error occurs an error message is returned.

Reply: Q Z C ;everything is OK

Q FF_{hex}.... ;error message (see error messages)

Processor Sleep *8.2.7*

Q Z D..X1

X1: 0_{hex} ;go to sleep mode

X1: 1_{hex} ;leave sleep mode

After this command the processor is in sleep mode (no clock, no access on address- or data bus). Waking up is only possible with a reset or by pulling down the WUP line of the SBS-Bus, using the command „leave sleep mode“ Q Z D 1hex.

Reply: Q Z D X1

Application Specific Commands *8.3*

Read/Write Channel Outputs *8.3.1*

Q X1 X2 D1 D2 D3 D4 Exp Flags

Reads or writes the DAC output data of the channel selected.

X1: Channel select (see cross reference) always in ASCII

X2: ’R’ or ’W’

D1,..,D4: Output value (0 - 4095)

Reply: Q X1 X2 D1 D2 D3 D4 Exp Flags

Flags = 0, if value was ok; Flags = 1, if value was out of range

Example:

→ Q h W 0 0 1B 9C 0 0 ;load Ch 1 Amp X with 1B9C

← Q h W 0 0 0F FF 0 1 ;Ch 1 Amp X loaded with FFF

Cross Reference: all values described can be used instead of X1

Table 8.1. X1: Channel Outputs

X1 Chenille 1	X1 Channel 2	X1 Channel 3
b Ch 1 - Ampl X	B Ch 2 - Ampl X	0 Ch 3 - Ampl X
c Ch 1 - Ampl	D Ch 2 - Ampl	1 Ch 3 - Ampl
g Ch 1 - Phase	G Ch 2 - Phase	6 Ch 3 - Phase
j Ch 1 - Phase Y	J Ch 2 - Phase Y	4 Ch 3 - Phase Y
f Ch 1 - Ampl Y	F Ch 2 - Ampl Y	3 Ch 3 - Ampl Y
h Ch 1 - Ampl	H Ch 2 - Ampl	5 Ch 3 - Ampl
c Ch 1 - Phase	C Ch 2 - Phase	2 Ch 3 - Phase

Read/Write Receiver Phase *8.3.2*

Q R X1 D1 D2 D3 D4 Exp Flags

Reads or writes the value for Receiver Phase. Sine and cosine functions are derived internally from the Receiver Phase value.

X1: 'R' or 'W'

D1,..,D4: Receiver Phase output value (0 - 14399)

Reply: Q R X1 D1 D2 D3 D4 Exp Flags

Flags = 0, if value was ok; Flags = 1, if value was out of range

Example:

→ Q R W 0 0 2B 48 0 0 ;load receiver phase with 2B48 (11080)

← Q R W 0 0 2B 48 0 0 ;receiver phase loaded with 2B48 (11080)

Q U X1 D1..D50

Reads or writes the complete parameter set for all DAC outputs and receiver phase. Each parameter needs two bytes (word format). In this command the parameter is not checked against under- or overflows

X1: 'R' or 'W'

D1,..,D50: complete parameter set

Reply: Q U X1 D1..D50

Parameter set order:

D1, D2	Ch 1 - Ampl X
D3, D4	Ch 1 - Ampl
D5, D6	Ch 1 - Phase
D7, D8	Ch 1 - Phase Y
D9, D10	Ch 1 - Ampl Y
D11, D12	Ch 1 - Ampl
D13, D14	Ch 1 - Phase
D15, D16	not used
D17, D18	Ch 2 - Ampl X
D19, D20	Ch 2 - Ampl
D21, D22	Ch 2 - Phase
D23, D24	Ch 2 - Phase Y
D25, D26	Ch 2 - Ampl Y
D27, D28	Ch 2 - Ampl
D29, D30	Ch 2 - Phase
D31, D32	not used
D33, D34	Ch 3 - Ampl X
D35, D36	Ch 3 - Ampl
D37, D38	Ch 3 - Phase
D39, D40	Ch 3 - Phase Y
D41, D42	Ch 3 - Ampl Y
D43, D44	Ch 3 - Ampl
D45, D46	Ch 3 - Phase
D47, D48	not used
D49, D50	Receiver Phase

Test Commands 8.4

Test Mode 8.4.1

Q T T X1

Enables or disables test mode. The test commands do not work when test mode is off!

X1: On/Off

Reply: Q T T X1

DAC Sweeper 8.4.2

Q T S X1

Sweeps selected DAC output from minimum to maximum output value (triangle function).

X1: Number of selected DAC (see DAC numbering scheme).

Reply: Q T S X1

DAC Switcher 8.4.3

Q T W X1

Switches multiplexer output selected. Function sequentially closes and opens all the analog switches of the selected group.

X1: Number of the analog switch selected (see group numbering scheme).

Reply: Q T W X1

DAC Stepper 8.4.4

Q T V X1

Function sequentially sets each bit of the DAC selected to high.

X1: Number of DAC selected (see DAC numbering scheme).

Reply: Q T V X1

TuneOffset 8.4.5

Q T O X1

Brings output of the DAC selected to 0 volts, to tune the offset of output circuit.

X1: Number of selected DAC (see DAC numbering scheme).

Reply: Q T O X1

Set Output ***8.4.6***

Q T X X1 D1 D2 D3 D4 Exp Flags

Loads DAC selected with value and switches corresponding analog switch. This enables output of test values to the control outputs.

X1: Number of DAC selected (see DAC numbering scheme).

D1.. D4: DAC output value (0 to 4095)

Reply: Q T X X1 D1 D2 D3 D4 Exp Flags

Flags = 0, if value was ok; Flags = 1, if value was out of range

Table 8.2. DAC numbering scheme

X1 Channel 1	X1 Channel 2	X1 Channel 3
0 Ch 1 - Ampl X	8 Ch 2 - Ampl X	16 Ch 3 - Ampl X
1 Ch 1 - Ampl	9 Ch 2 - Ampl	17 Ch 3 - Ampl
2 Ch 1 - Phase	10 Ch 2 - Phase	18 Ch 3 - Phase
3 Ch 1 - Phase Y	11 Ch 2 - Phase Y	19 Ch 3 - Phase Y
4 Ch 1 - Ampl Y	12 Ch 2 - Ampl Y	20 Ch 3 - Ampl Y
5 Ch 1 - Ampl	13 Ch 2 - Ampl	21 Ch 3 - Ampl
6 Ch 1 - Phase	14 Ch 2 - Phase	22 Ch 3 - Phase
7 Sin Output	15 Cos Output	23 not used Output

Table 8.3. Group numbering scheme

X1 = 0 Channel 1 x	X1 = 2 Channel 2 x	X1 = 4 Channel 3 x
X1 = 1 Channel 1 y	X1 = 3 Channel 2 y	X1 = 5 Channel 3 y

Error codes and xmessages

9

On all units, the HPCU, MAS and 4 PH Modulator, the error codes between 1 and 69 have the same meaning. These errors are system errors. The error codes from 70 until 255 indicate application specific errors.

HPCU Master error list in application mode 9.1

In this mode the CPU executes the application program located in the FLASH EEPROM. The error messages have the prefix „HPCU“ to show that all the HPCU functions are available.

Error 2	HPCU Check sum error
Error 15	HPCU BBIS not responding
Error 20	HPCU Order error
Error 134	HPCU Missing pulses
Error 135	HPCUCPU link timeout
Error 137	HPCU Timeout error
Error 138	HPCU Data transmission error on SBS bus
Error 139	HPCU Checksum error on SBS bus
Error 140	HPCU MAS Timeout error
Error 144	HPCU Function not implemented
Error 145	HPCU Motor power is off
Error 146	HPCU Motor still running
Error 152	HPCUCPU link syntax error
Error 153	HPCU E2PROM timeout error
Error 154	HPCU E2PROM write error
Error 156	HPCU Motor power below 20V
Error 157	HPCU not in hardware test mode
Error 158	HPCU Autotune aborted, motor power below 20V
Error 159	HPCU SBS timeout
Error 160	HPCU MAS not responding

HPCU Master error list in boot mode 9.2

In this mode the CPU executes the boot program located in the EPROM. There is either no application software available or a download has been started. The HPCU can only perform the master commands. So all error messages have the prefix HPCM.

Error 2	HPCM Check sum error
Error 11	HPCM ROM error
Error 15	HPCM BBIS not responding
Error 20	HPCM Order error
Error 3	HPCM Wrong data count
Error 32	HPCM Wrong address
Error 33	HPCM Wrong rec type
Error 34	HPCM Wrong check sum
Error 36	HPCM Programmer fail
Error 37	HPCM Erase fail
Error 38	HPCM Wrong transmission check sum
Error 70	HPCM CPU link timeout
Error 136	HPCM CPU link syntax error

HPCU Slave error list 9.3

Error 2	HPCS Check sum error
Error 15	HPCS BBIS not responding
Error 20	HPCS Order error
Error 31	HPCS Wrong data count
Error 32	HPCS Wrong address
Error 33	HPCS Wrong rec type
Error 34	HPCS Wrong check sum
Error 36	HPCS Programmer fail
Error 37	HPCS Erase fail
Error 38	HPCS Wrong transmission check sum
Error 70	HPCS CPU link syntax error

Quad Phase modulator error list 9.4

Error 2	4PH_ Check sum error

Error 11	4PH_ ROM error
Error 15	4PH_ BBIS not responding
Error 20	4PH_ Order error
Error 31	4PH_ Wrong data count
Error 32	4PH_ Wrong address
Error 33	4PH_ Wrong rec type
Error 34	4PH_ Wrong check sum
Error 36	4PH_ Programmer fail
Error 37	4PH_ Erase fail
Error 38	4PH_ Wrong transmission check sum

MAS error list *9.5*

Error 20	MAS_ INVALID COMMAND
Error 73	MAS_ INVALID DATA
Error 74	MAS_ TOO FEW PARAMETERS
Error 75	MAS_ TOO MANY PARAMETERS
Error 76	MAS_ DATA OUT OF RANGE
Error 77	MAS_ SPIN INPUT DATA TOO HIGH
Error 78	MAS_ Kp OUT OF RANGE
Error 79	MAS_ Ki OUT OF RANGE
Error 80	MAS_ Kd OUT OF RANGE
Error 81	MAS_ IT VALUE TOO LOW
Error 82	MAS_ CHECKSUM ERROR
Error 83	MAS_ INVALID PASSWORD
Error 91	MAS_ PRESSURE MEASUREMENT ERROR
Error 92	MAS_ MAIN PRESSURE NOT DETECTED
Error 93	MAS_ BEARING SENSE PRESSURE LOSS REDUCING DRIVE
Error 94	MAS_ PRESSURE MEASUREMENT SATURATION ERROR
Error 95	MAS_ PRESSURE MEASUREMENT CHANELL ERROR
Error 97	MAS_ PRESSURE REGULATION ERROR
Error 98	MAS_ ZERO DRIVE PRESSURE NOT FOUND
Error 99	MAS_ ZERO BEARING PRESSURE NOT FOUND
Error 102	MAS_ PG1 FAILED
Error 103	MAS_ PG2 FAILED
Error 104	MAS_ NO SPIN DETECTED, ABORT REGULATION
Error 105	MAS_ SPINNING PROBE NOT DETECTED

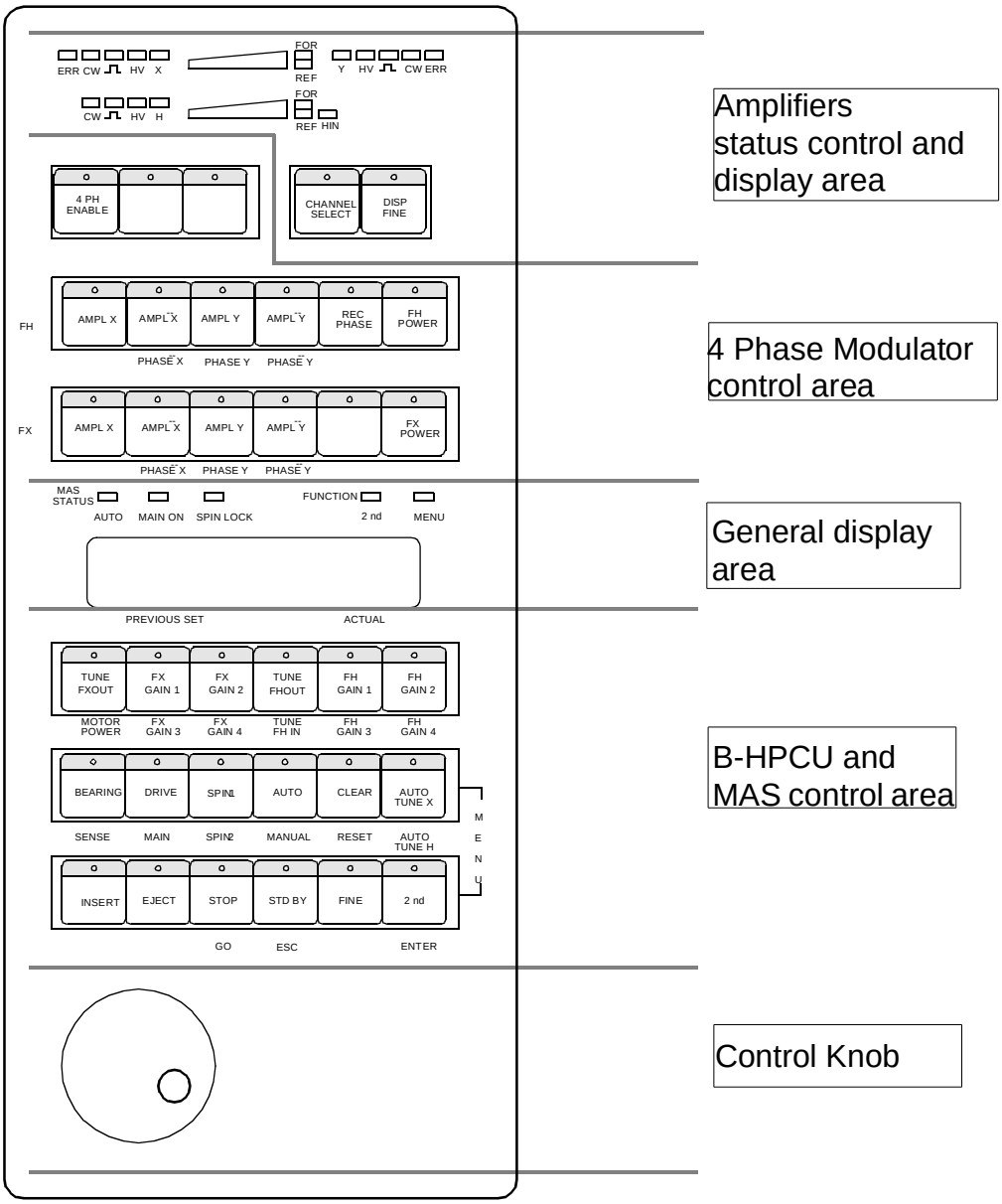
Error 106	MAS_ SPIN REGULATION ERROR
Error 107	MAS_ NO INSERT OR EJECT WHILE ROTOR SPINS
Error 108	MAS_ DRIVE PRESSURE MAXIMUM, ABORT REGULATION
Error 109	MAS_ LOSS OF OPERATIONAL SPIN SPEED
Error 111	MAS_ FATAL SYSTEM ERROR
Error 112	MAS_ WATCHDOG RESET
Error 113	MAS_ RAMTEST FAILED
Error 114	MAS_ COMMUNICATION ERROR
Error 115	MAS_ VALVE ERROR
Error 116	MAS_ CALIBRATION DATE INCORRECT
Error 117	MAS_ A/D CONVERTER TIMEOUT
Error 118	MAS_ BOARD IDENTIFIER MISSING
Error 119	MAS_ VALVE CONNECTOR NOT FOUND
Error 120	MAS_ NO CONTACT TO LCD DISPLAY
Error 121	MAS_ TIMEOUT IN PROGRAM
Error 122	MAS_ SPIN RATE MEASUREMENT ERROR

How to use the HPCU keyboard

10

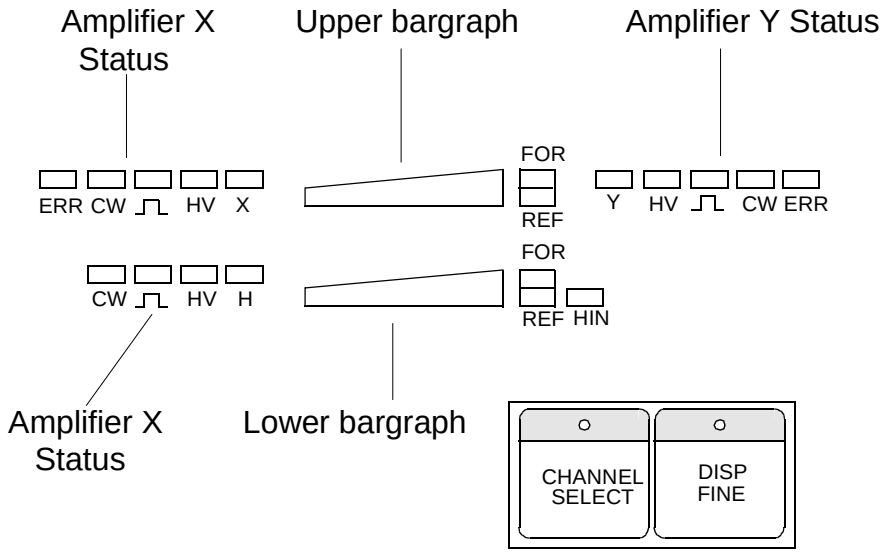
The keyboard can be used to control the B-HPCU, the MAS and the 4 Phase Modulator. The user layout is divided into 5 logical parts. See following diagram.

Figure 10.1. HPCU Keyboard



Amplifier status control and display area *10.1*

Figure 10.2. HPCU Keyboard Display



Each amplifier, X, Y and 1H has its own status LED's. They show ERROR (with the exception of the H amplifier), CW mode, Blank pulse and the presence of the High Voltage for the tube amplifier. The error LED's light at over-temperature, pulse width too great or wrong pulse duty cycle.

The LEDs X, Y, and H indicate the amplifier whose power is shown at the upper and/or lower bargraph. The LEDs FOR, REF and HIN at the right of the bargraphs, indicate the kind of power shown on the corresponding bargraph.

The CHANNEL SELECT switch can be used to select the power type shown on the bargraph. There are 4 possibilities. Pressing the CHANNEL SELECT switch moves through the 4 options in sequence.

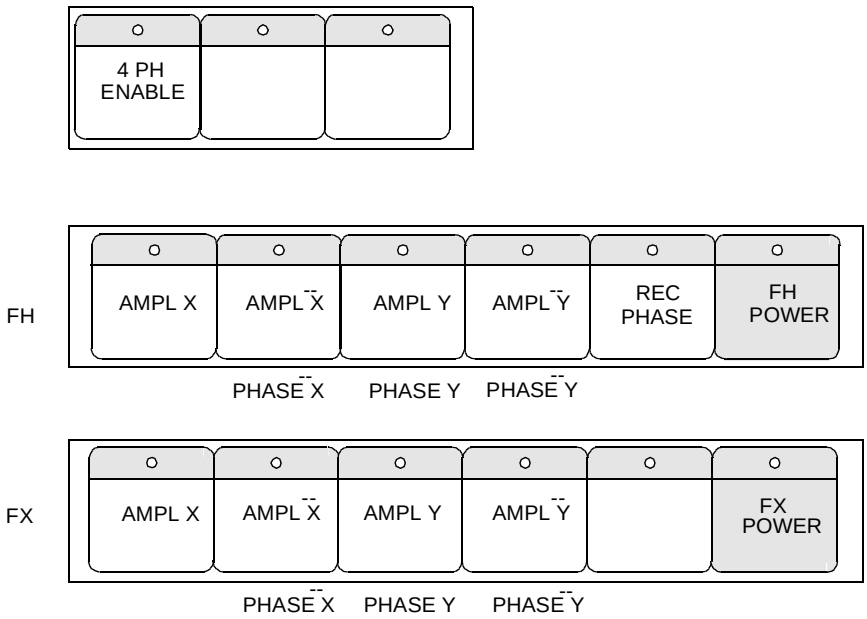
Table 10.1. Amplifier Channel Select

Pos.	upper bargraph	lower bargraph
1	X forward	H forward
2	X reflected	H reflected (out)
3	H forward	H reflected (in)
4	Y forward	Y reflected

The switch DISP FINE toggles between a coarse and fine resolution of the bargraph LEDs. In fine mode the resolution of the first bargraph LED is amplified to all (ten) LEDs.

Phase Modulator control area *10.2*

Figure 10.3. 4 Phase Modulator Control Area



Switches FH POWER and FX POWER (greyed) are not used. They are reserved for future use.

The switch 4PH ENABLE is used to enable or disable the access to the 4 Phase Modulator by the keyboard. The LED on this toggle switch lights when access is enabled. It is disabled at power on of the B-HPCU by default. It will also be disabled about 30 seconds after the last access to the 4 Phase Modulator automatically, to prevent unintended modification of parameters.

All other switches in this area (described below) each control a DC voltage. A separate 12 bit DAC is used to generate each voltage. So the range for each switches is the same: from 0..4095. The actual and the previous value are shown in the general display area.

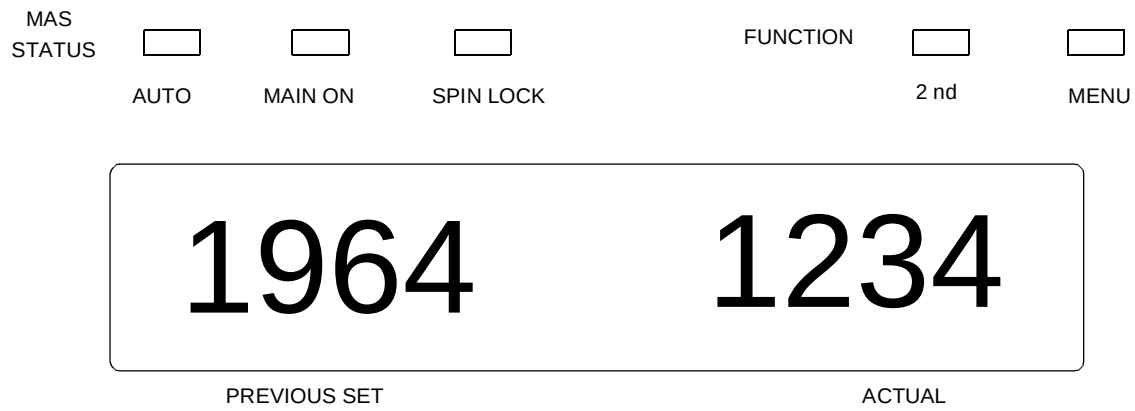
The switches AMPL X, AMPL X^{-}, AMPL Y, and AMPL Y^{-} of channel FH and FX can be used to perform a +/-10% amplitude adjustment of the transmitter outputs.

The switches PHASE X^{-}, PHASE Y, and PHASE Y^{-} of channel FH and FX can be used to perform a +/-10% phase position alteration of the transmitter outputs. There is no switch to modify PHASE X, because it is always 0 (used for reference).

Switch REC PHASE modifies the phase of the receiver.

General display area 10.3

Figure 10.4. General display area



MAS Status 10.3.1

The AUTO LED lights when the MAS is in automatic mode. It is off when MAS is in manual mode. (At present the LED indicates only if the MAS status is changed by an X32 command via the B-HPCU).

MAIN ON LED: Function not yet implemented.

SPIN LOCK LED: Function not yet implemented.

Function LEDs 10.3.2

To enter a second function of a switch, the 2nd switch must be pressed first, then the desired switch can be pressed. The 2nd LED in the general display area will light. It is off when the first function of a switch is in use or if STD BY is pressed.

16 Character ASCII display 10.3.3

This display shows the previous value set on the left side and the actual value on the right side. An altered value can be changed to its previous set value by pressing the actual switch again.

The display shows error messages and status information. The complete menu handling is performed on this display with the help of the Control Knob, ENTER and ESC switches.

B-HPCU and MAS control area 10.4

B-HPCU controlling 10.4.1

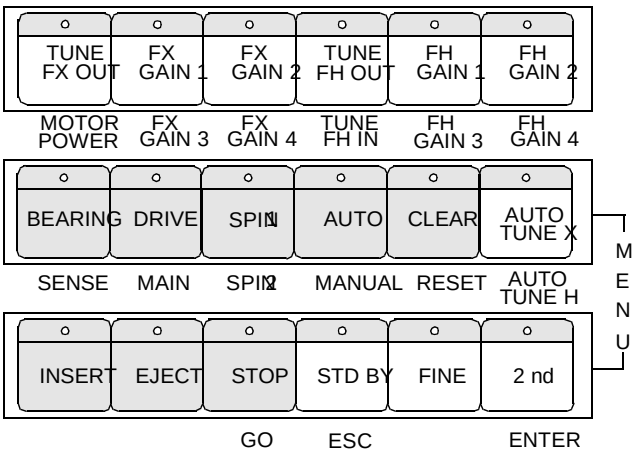
The non greyed switches can be used to control the B-HPCU.

The second function of the first switch in row 1 is MOTOR POWER on/off. This toggles the power to the stepper motors, used in the amplifiers, on and off. It is useful to switch the power off, after tuning an amplifier, to prevent unintended motor movement do to keyboard handling errors. There is no LED to indicate the status as to whether the motor power is on or off. If someone tries to tune an amplifier while motor power is off, an error message

„ERROR 145 HPCU Motor power is off. press 'STD BY'"

is shown on the display.

Figure 10.5. Display Switches



Switches TUNE FXOUT, TUNE FHOUT and its second function TUNE FHIN are used to tune a 1H or X high power amplifier. The control range (depends on amplifier type) is between 0..200 and 0..4096. The forward and reflected power will be shown on the bargraph LEDs.

Automatic amplifier tuning can be done by pressing switch AUTO TUNE X / AUTO TUNE H. It is a toggle switch. This means, pressing the switch one more time, while the auto tune procedure is running, aborts the tuning. During auto tune the LED of this switch flickers. The LED goes out when tuning is finished.

The switches FX GAIN1..FX GAIN4 and FH GAIN1..FH GAIN4 control the gain of the 1H or X high power amplifier. The keyboard directly controls two 12 bit DACs, one for FH and one for FX. These voltages are fed to the high voltage power supply. Only one of four gains controls the amplifier (selected by RCP signals). The other three gains are preset values. For more detail see hardware description.

MAS control 10.4.2

The MAS functions are yet implemented

Menu 11

In menu mode the switch 2nd becomes the ENTER function, and STD BY becomes the ESC function.

To get the menu press the AUTO TUNE X and 2nd switches at the same time. The main menu has the following structure:

1. Keyboard
2. MAS
3. Service
4. Bargraph cal.
5. Amplifier cal.
6. 4-Phase Mod.

To work with the menu:

use the code wheel, to go up and down

press enter to select (to step into) a menu point

or press ESC to leave a menu point.

Keyboard *11.1*

The keyboard menu has the following structure:

1. Keyboard
   - 1.1 Lock Keyboard (lock with ENTER, unlock with ESC)
   - 1.2 Brightness (brightness of all LED's and display, 0 = dark, 6= bright)
   - 1.3 Display test (lamp test of LED'S and display dots)
   - 1.4 App SW-Date (shows the keyboard application software date (format dd/mm/yy)
   - 1.5 Boot SW -Date (shows the keyboard boot software date, format dd/mm/yy)
   - 1.6 HW-Ver (shows the hardware version of the controller board)
   - 1.7 Disp. HW-Ver (shows the hardware version of the display board)

MAS *11.2*

The MAS menu section is not yet implemented.

Service *11.3*

3. Service

3.1 Sec.-Code

3.2 Save Config (not yet implemented)

To get into the menu points 3.2 and 5 and 6, a security code must be entered. These menu points are protected to prevent unintended HPCU parameter exchange (may cause hardware defects) by unauthorised people. This should only be done by a service engineer.

Bargraph calibration *11.4*

In this menu a full scale adjustment of all seven bargraph signals can be done.

4.Bargraph cal.

4.1 For X (full scale of forward X signal path)

4.2 Ref. X (full scale of reflected X signal path)

4.3 For Y (full scale of forward Y signal path)

4.4 Ref. Y (full scale of reflected Y signal path)

4.5 For H (full scale of forward H signal path)

4.6 Ref. H (full scale of reflected H signal path)

4.7 Ref. H (full scale of reflected Hin signal path)

This is done by:

- selecting the desired signal path (with the code wheel and enter switch)
- giving pulses with desired (max.) amplitude to the corresponding amplifier
- turning the code wheel left or right (range of 0..99) until 10 LED are lit
- pressing enter to store the configuration in the B-HPCU

Amplifier calibration *11.5*

In this menu the position calibration of cavity type amplifiers can be done. It should only done by authorised service engineers.

5. Amplify cal.

5.1 Zero pos. (electrical zero position adjustment)

5.2 Max pos. (electrical maximal position adjustment)

4 Phase Modulator *11.6*

This menu point is used to control an optional 4 Phase Modulator (Y Channel).

6. 4-Phase Mod.
    - 6.1 FY AMPL X
    - 6.2 FY AMPL /X
    - 6.3 FY PHASE /X
    - 6.4 FY AMPL Y
    - 6.5 FY PHASE Y
    - 6.6 FY AMPL /Y
    - 6.7 FY PHASE /Y
    - 6.2 FY POWER (not yet implemented)

The menu points AMPL X, AMPL X^{-}, AMPL Y, and AMPL Y^{-} of channel FY can be used to perform a +/-10% amplitude adjustment of the transmitter outputs.

The menu points PHASE X^{-}, PHASE Y, and PHASE Y^{-} of this can be used to perform a +/-10% phase position adjustment of the transmitter outputs. There is no switch to modify PHASE X, because it is always 0 (used for reference).

Appendix 12

PAL Listings 12.1

Master CPU Decoder 12.1.1

The HPCU has 5 PALs: two 18p8 and three 22V10.All PALs work on PCB hardware versions A and B

Master CPU Decoder 12.1.2

```
*IDENTIFICATION
;*********************************************************************
 Date:              17.07.92
 Company:           BRUKER ELEKTRONIK GmbH
                    D-7512 Rheinstetten 4/Karlsruhe
                    Akazienweg 2
 Designer:          Helmut Knoerr
 PAL:               HPC0AA04-KE
 Part name:         HPCU Board master CPU decoder
 Part number:       H6513
 Revision:          AA
 Assembly:          B-HPCU
 Part name:         B-HPCU Motor Control Unit Kpl.
 Part number:       H5149, (H6509)
 Revisions:         A+B
 Pal location:      IC U40
 File name:         HPCMASTA.DCB
;*********************************************************************
*X-NAMES
 WR       ; LOW ACTIVE CPU WRITE SIGNAL
 RD       ; LOW ACTIVE CPU READ SIGNAL
 P17      ; HIGH ACTIVE MODE BIT FOR DOWN LOAD
 A[15..6] ; HIGH ACTIVE CPU ADDRESSES
 PSEN     ; LOW => CODE MEMORY, HIGH => DATA MEMORY
*Y-NAMES
 X32ENABLE   ; LOW ACTIVE CHIP SELECT FOR X32 UART
```

```
 IOENABLE    ; LOW ACTIVE CHIP SELECT FOR OTHER PALS IN I/O AREA
 CSRAM       ; LOW ACTIVE CHIP SELECT FOR RAM
 CSROM       ; LOW ACTIVE CHIP SELECT FOR EPROM
 CSFLASH     ; LOW ACTIVE CHIP SELECT FOR FLASH EPROM
 DIR         ; LOW ACTIVE, ENABLES DATAFLOW FROM EXT. TO INT. DATABUS
 RDFLASH     ; LOW ACTIVE, READ SIGNAL EITHER /RD OR /PSEN FOR EPROM
 WRFLASH     ; LOW ACTIVE, READ SIGNAL EITHER /RD OR /PSEN FOR RAM
;_____________________________________________________________________
*FUNKTION-TABLE
$(A[15..8]),A7,A6,RD,WR,PSEN,P17: DIR ;
    80H    ,- ,- ,0 ,1 ,  1 , 0 :   0 ; NORMAL MODE (ALL HW)
    80H    ,- ,- ,1 ,0 ,  1 , 0 :   1 ; NORMAL MODE (ALL HW)
    FFH    ,1 ,1 ,0 ,1 ,  1 , - :   0 ; BOOT MODE (ONLY X32 RS232)
    FFH    ,1 ,1 ,1 ,0 ,  1 , - :   1 ; BOOT MODE (ONLY X32 RS232)
    REST                        :   1 ;
$(A[15..8]),A7,A6 , RD , WR , PSEN , P17 : X32ENABLE;
    FFH    ,1 , 1 ,  - , -  ,  1   ,  -  :   0 ; ALL TIME ACCESSABLE
    FFH    ,1 , 1 ,  - , -  ,  1   ,  -  :   0 ; ALL TIME ACCESSABLE
    REST                                :   1 ;
(A[15..8]) , RD , WR , PSEN , P17 :  IOENABLE;
    80H    ,  - , -  ,  1   ,  0  :   0 ; NORMAL MODE
    80H    ,  - , -  ,  1   ,  0  :   0 ; NORMAL MODE
    REST                          :   1 ;
$ (A[15..8]),(A[7..6]) , P17 , PSEN : CSROM,CSFLASH,CSRAM;
  00H..0FEH ,  0H..3H ,  1  ,  -    :   0  ,   0   ,   1  ; BOOT MODE
       0FFH ,  0H..2H ,  1  ,  -    :   0  ,   0   ,   1  ; BOOT MODE
       0FFH ,  3H     ,  1  ,  -    :   0  ,   1   ,   1  ; BOOT MODE
  00H..07FH ,  0H..3H ,  0  ,  -    :   1  ,   0   ,   0  ; NORM MODE
  80H..0FEH ,  0H..3H ,  0  ,  -    :   1  ,   0   ,   1  ; NORM MODE
       0FFH ,  0H..2H ,  0  ,  -    :   1  ,   0   ,   1  ; NORM MODE
       0FFH ,  3H     ,  0  ,  -    :   1  ,   1   ,   1  ; NORM MODE
   REST                             :   1  ,   1   ,   1  ;
$ (A[15..8]) , P17 , RD , WR ,PSEN : RDFLASH , WRFLASH;
    00H..0FFH,  1  , 1  ,  0 ,  1  :   1    ,    0    ; FLASH WRITE
    00H..0FFH,  1  , 0  ,  1 ,  1  :   0    ,    1    ; FLASH READE
    00H..0FFH,  0  , 1  ,  1 ,  0  :   0    ,    1    ; FLASH FETCH
    REST                           :   1    ,    1    ;
;_____________________________________________________________________
*RUN-CONTROL
 LISTING  = SYMBOLTABLE ;
 OPTIMIZE = P-TERMS ;
*END
```

```
*IDENTIFICATION
 Date:               17.07.92
 Company:            BRUKER ELEKTRONIK GmbH
                     D-7512 Rheinstetten 4/Karlsruhe
                     Akazienweg 2
 Designer:           Helmut Knoerr
 PAL:                HPC0AA04-KE
 Part name:          HPCU Board master CPU decoder
 Part number:        H6513
 Revision:           AA
 Assembly:           B-HPCU
 Part name:          B-HPCU Motor Control Unit Kpl.
 Part number:        H5149, (H6509)
 Revisions:          A+B
 Pal location:       IC U40
 File name:          HPCMASTA.DDV
;*********************************************************************
@DEVICE = HPCMASTA;
*PLD
 TYPE = AMPAL22V10;
 CHECKSUM = COMPUTE;
*PINS
 P17      = 1,
 A8       = 2,
 A9       = 3,
 A10      = 4,
 A11      = 5,
 A12      = 6,
 A13      = 7,
 A14      = 8,
 A15      = 9,
 WR       = 10,
 RD       = 11,
 PSEN     = 13,
 A6       = 14,
 A7       = 15,

 WRFLASH  = 16,
 RDFLASH  = 17,
 CSRAM    = 18,
 CSROM    = 19,
 CSFLASH  = 20,
```

```
 DIR       = 21,
 IOENABLE  = 22,
 X32ENABLE = 23;

*RUN-CONTROL
 LISTING = EQUATIONS,PINOUT;
 PROGFORMAT = JEDEC;
 @ENDDEVICE = HPCMASTA;
*END
```

Slave CPU decoder *12.1.3*

```
*IDENTIFICATION
;*************************************************************************
 Date:               92.07.17
 Company:            BRUKER ELEKTRONIK GmbH
                     D-7512 Rheinstetten 4/Karlsruhe
                     Akazienweg 2
 Designer:           Helmut Knoerr
 PAL:                HPC0AA05-KE
 Part name:          HPCU Board slave CPU decoder
 Part number:        H6512
 Revision:           AA
 Assembly:           B-HPCU
 Part name:          B-HPCU Motor Control Unit Kpl.
 Part number:        H5149, (H6509)
 Revisions:          A+B
 Pal location:       IC U46
 File name:          HPCSLAVA.DCB
;*************************************************************************
*X-NAMES
 WR       ; LOW ACTIVE CPU WRITE SIGNAL
 RD       ; LOW ACTIVE CPU READ SIGNAL
 P14      ; HIGH ACTIVE MODE BIT FOR DOWN LOAD
 A[15..8] ; HIGH ACTIVE CPU ADDRESSES
 PSEN     ; LOW => CODE MEMORY, HIGH => DATA MEMORY
*Y-NAMES
 CSROM      ; LOW ACTIVE CHIP SELECT FOR EPROM
 CSFLASH    ; LOW ACTIVE CHIP SELECT FOR FLASH EPROM
 RDFLASH    ; LOW ACTIVE, READ SIGNAL EITHER /RD OR /PSEN FOR EPROM
 WRFLASH    ; LOW ACTIVE, READ SIGNAL EITHER /RD OR /PSEN FOR RAM
;_________________________________________________________________
```

```
*FUNKTION-TABLE
 $(A[15..8]), P14, PSEN: CSROM, CSFLASH;
   00H..0FFH,  1 ,  -  :   0  ,   0    ; BOOT MODE
   00H..0FFH,  0 ,  -  :   1  ,   0    ; NORMAL MODE
   REST               :   1  ,   1    ;
  $(A[15..8]), P14, RD, WR,PSEN: RDFLASH, WRFLASH;
    00H..0FFH,  1 , 1 , 0 ,  1 :   1    ,    0   ; FLASH WRITE
    00H..0FFH,  1 , 0 , 1 ,  1 :   0    ,    1   ; FLASH READE
    00H..0FFH,  0 , 1 , 1 ,  0 :   0    ,    1   ; FLASH FETCH
    REST                       :   1    ,    1   ;
;_____________________________________________________________
*RUN-CONTROL
 LISTING  = SYMBOLTABLE;
 OPTIMIZE = P-TERMS;
 *END



*IDENTIFICATION
;*********************************************************************
 Date:               92.07.17
 Company:            BRUKER ELEKTRONIK GmbH
                     D-7512 Rheinstetten 4/Karlsruhe
                     Akazienweg 2
 Designer:           Helmut Knoerr
 PAL:                HPC0AA05-KE
 Part name:          HPCU Board slave CPU decoder
 Part number:        H6512
 Revision:           AA
 Assembly:           B-HPCU
 Part name:          B-HPCU Motor Control Unit Kpl.
 Part number:        H5149, (H6509)
 Revisions:          A+B
 Pal location:       IC U46
 File name:          HPCSLAVA.DDV
;*********************************************************************
@DEVICE = HPCSLAVA;
*PLD
 TYPE = AMPAL18P8;
 CHECKSUM = COMPUTE;
*PINS
 P14    = 7,
```

```
 A15    = 1,
 A14    = 2,
 A13    = 3,
 A12    = 4,
 A11    = 5,
 A10    = 6,
 A9     = 18,
 A8     = 19,
 WR     = 8,
 RD     = 9,
 PSEN   = 11,
 WRFLASH = 17,
 RDFLASH  = 16 ,
 CSROM    = 12 ,
 CSFLASH  = 15 ;
*RUN-CONTROL
 LISTING  = EQUATIONS,PINOUT ;
 PROGFORMAT = JEDEC ;
@ENDDEVICE = HPCSLAVA;
*END
```

I/O decoder *12.1.4*

```
*IDENTIFICATION
;*********************************************************************
 Date:              92.07.17
 Company:           BRUKER ELEKTRONIK GmbH
                    D-7512 Rheinstetten 4/Karlsruhe
                    Akazienweg 2
 Designer:          Helmut Knoerr
 PAL :              HPC0AA04-KE
 Part name:         HPCU Board I/O decoder
 Part number:       H6513
 Revision:          AA
 Assembly:          B-HPCU
 Part name:         B-HPCU Motor Control Unit Kpl.
 Part number:       H5149, (H6509)
 Revisions:         A+B
 Pal location:      IC U41
 File name:         HPCIOA.DCB
;*********************************************************************
*X-NAMES
```

```
 RD            ; CPU READ SIGNAL
 WR            ; CPU WRITE SIGNAL
 IOENABLE      ; GENERAL BOARD SELECT
 X32ENABLE     ; ENABLE INPUT FOR X32 SIO
 A[7..0]       ; CPU ADDRESSES
 *Y-NAMES
 CSSIO0         ; LOW ACTIVE CHIPSELECT SIGNAL FOR SIO 1
 CSSIO1         ; LOW ACTIVE CHIPSELECT SIGNAL FOR SIO 2
 CSSIO2         ; LOW ACTIVE CHIPSELECT SIGNAL FOR SIO 3
 CSAMPID        ; LOW ACTIVE CHIPSELECT SIGNAL AMPLIFIER ID
 CSSENS         ; LOW ACTIVE CHIPSELECT SIGNAL LIGHT BARRIERS
 F0             ; LOW ACTIVE SLOT ENABLE SIGNAL
 CSXOUT         ; HIGH ACTIVE CHIPSELECT SIGNAL X OUT MOTOR
 CSAUX          ; HIGH ACTIVE CHIPSELECT SIGNAL AUX MOTOR
 CSHOUT         ; HIGH ACTIVE CHIPSELECT SIGNAL H OUT MOTOR

 CSHIN          ; HIGH ACTIVE CHIPSELECT SIGNAL H IN MOTOR
;_____________________________________________________________
*FUNKTION-TABLE
 $(A[7..0]),IOENABLE, X32ENABLE: CSSIO0, CSSIO1, CSSIO2;
   10H..17H,   0    ,     -    :    1  ,    1  ,   0   ;8010H..8017H
   18H..1FH,   0    ,     -    :    1  ,    0  ,   1   ;8018H..801FH
   F8H..FFH,   -    ,     0    :    0  ,    1  ,   1   ;FFF8H..FFFFH
   REST                        :    1  ,    1  ,   1   ; ---
 $(A[7..0]), IOENABLE , RD , WR : CSAMPID , CSSENS;
   09H     ,  0       ,  0 ,  1 :    0    ,    1  ;  8009H
   0AH     ,  0       ,  0 ,  1 :    1    ,    0  ;  800AH
   REST                          :    1    ,    1  ; ---
 $(A[7..0]), IOENABLE , RD , WR : CSXOUT, CSAUX,CSHOUT,CSHIN;
   0BH     ,  0       ,  1 ,  0 :    1  ,   0  ,   0  ,  0  ; 800BH
   0CH     ,  0       ,  1 ,  0 :    0  ,   1  ,   0  ,  0  ; 800CH
   0DH     ,  0       ,  1 ,  0 :    0  ,   0  ,   1  ,  0  ; 800DH
   0EH     ,  0       ,  1 ,  0 :    0  ,   0  ,   0  ,  1  ; 800EH
   REST                          :    0  ,   0  ,   0  ,  0  ;  ---
 $(A[7..0]), IOENABLE , RD , WR : F0 ;
   20H..FFH,  0       ,  - ,  - :  0 ;  8020H..80FFH
   REST                          :  1 ; ---
;_____________________________________________________________
*RUN-CONTROL
 LISTING  = SYMBOLTABLE ;
 OPTIMIZE = P-TERMS ;
*END
```

```
 *IDENTIFICATION
;*********************************************************************
 Date:               92.07.17
 Company:            BRUKER ELEKTRONIK GmbH
                     D-7512 Rheinstetten 4/Karlsruhe
                     Akazienweg 2
 Designer:           Helmut Knoerr
 PAL :               HPC0AA04-KE
 Part name:          HPCU Board I/O decoder
 Part number:        H6513
 Revision:           AA
 Assembly:           B-HPCU
 Part name:          B-HPCU Motor Control Unit Kpl.
 Part number:        H5149, (H6509)
 Revisions:          A+B
 Pal location:       IC U41
 File name:          HPCIOA.DDV
;*********************************************************************
@DEVICE = HPCIOA;
*PLD
 TYPE = AMPAL22V10;
 CHECKSUM = COMPUTE;
*PINS
 A0       = 1 ,
 A1       = 2 ,
 A2       = 3 ,
 A3       = 4 ,
 A4       = 5 ,
 A5       = 6 ,
 A6       = 7 ,
 A7       = 8 ,
 WR       =  9,
 RD       = 10,
 IOENABLE = 11,
 X32ENABLE= 13,
 CSSIO0    = 16,
 CSSIO1    = 15,
 CSSIO2    = 14,
 CSAMPID   = 19,
 CSSENS    = 18,
 F0        = 17,
 CSXOUT    = 20,
```

```
 CSAUX     = 21,
 CSHOUT    = 22,
 CSHIN     = 23;
*RUN-CONTROL
 LISTING  = EQUATIONS,PINOUT ;
 PROGFORMAT = JEDEC ;
 TESTVECTORS = GENERATE;
 @ENDDEVICE = HPCIOA;
*END
```

DAC register output multiplexer *12.1.5*

```
*IDENTIFICATION
;*********************************************************************
 Date:               91.12.17
 Company:            BRUKER ELEKTRONIK GmbH
                     D-7512 Rheinstetten 4/Karlsruhe
                     Akazienweg 2
 Designer:           Helmut Knoerr
 PAL :               HPC0AA01-KE
 Part name:          HPCU Board DAC register output multiplexer
 Part number:        H6510
 Revision:           AA
 Assembly:           B-HPCU
 Part name:          B-HPCU Motor Control Unit Kpl.
 Part number:        H5149, (H6509)
 Revisions:          A+B
 Pal location:       IC U35
 File name:          HPCGMUX.DDV
;*********************************************************************
*X-NAMES
 P42           ; LOW ACTIVE ENABLE INPUT
 P43           ; INPUT SELECT 0 MEANS CONTROLLED BY P40 and P41
 P4[1..0]      ; SELECT SIGNALS (ADDRESSES)
 RCPH[1..0]    ; REAL TIME CLOCK PULSE FOR H REGISTERS  (ADDRESSES)
 RCPX[1..0]    ; REAL TIME CLOCK PULSE FOR X REGISTERS  (ADDRESSES)
*Y-NAMES
 ENH[3..0]     ; LOW ACTIVE ENABLE INPUTS FOR REGISTERS
 ENX[3..0]     ; LOW ACTIVE ENABLE INPUTS FOR REGISTERS
 ;_____________________________________________________________________
```

```
*FUNKTION-TABLE
 $(P4[1..0]), (RCPH[1..0]),  P43     ,   P42   : (ENH[3..0]);
   00B      ,  0H..03H    ,   0      ,    0    : 1110B;
   0H..03H  ,     00B     ,   1      ,    0    : 1110B;
   01B      ,  0H..03H    ,   0      ,    0    : 1101B;
   0H..03H  ,     01B     ,   1      ,    0    : 1101B;
   10B      ,  0H..03H    ,   0      ,    0    : 1011B;
   0H..03H  ,     10B     ,   1      ,    0    : 1011B;
   11B      ,  0H..03H    ,   0      ,    0    : 0111B;
   0H..03H  ,     11B     ,   1      ,    0    : 0111B;
   REST                                       : 1111B;
 $(P4[1..0]), (RCPX[1..0]),  P43     ,   P42   : (ENX[3..0]);
   00B      ,  0H..03H    ,   0      ,    0    : 1110B;
   0H..03H  ,     00B     ,   1      ,    0    : 1110B;
   01B      ,  0H..03H    ,   0      ,    0    : 1101B;
   0H..03H  ,     01B     ,   1      ,    0    : 1101B;
   10B      ,  0H..03H    ,   0      ,    0    : 1011B;
   0H..03H  ,     10B     ,   1      ,    0    : 1011B;
   11B      ,  0H..03H    ,   0      ,    0    : 0111B;
   0H..03H  ,     11B     ,   1      ,    0    : 0111B;
   REST                                       : 1111B;
 ;_____________________________________________________________________
*RUN-CONTROL
 LISTING  = SYMBOLTABLE ;
 OPTIMIZE = P-TERMS ;
*END



*IDENTIFICATION
;*********************************************************************
 Date:               92.12.17
 Company:            BRUKER ELEKTRONIK GmbH
                     D-7512 Rheinstetten 4/Karlsruhe
                     Akazienweg 2
 Designer:           Helmut Knoerr
 PAL :               HPC0AA01-KE
 Part name:          HPCU Board DAC register output multiplexer
 Part number:        H6510
 Revision:           AA
 Assembly:           B-HPCU
 Part name:          B-HPCU Motor Control Unit Kpl.
```

```
 Part number:       H5149, (H6509)
 Revisions:         A+B
 Pal location:      IC U35
 File name:         HPCGMUX.DDV
;*********************************************************************

@DEVICE = HPCGMUXA;
*PLD
 TYPE = AMPAL18P8;
 CHECKSUM = COMPUTE;
*PINS
 P40    = 3 ,
 P41    = 4 ,
 P42    = 1 ,
 P43    = 2 ,
 RCPH0  = 7 ,
 RCPH1  = 8 ,
 RCPX0  = 9 ,
 RCPX1  = 11,
 ENH0     = 12 ,
 ENH1     = 13 ,
 ENH2     = 14 ,
 ENH3     = 15 ,
 ENX0     = 16 ,
 ENX1     = 17 ,
 ENX2     = 18 ,
 ENX3     = 19 ;
*RUN-CONTROL
 LISTING  = EQUATIONS,PINOUT ;
 PROGFORMAT = JEDEC ;
 TESTVECTORS = GENERATE;
 @ENDDEVICE = HPCGMUXA;
*END
```

DAC register load decoder *12.1.6*

```
*IDENTIFICATION
;*********************************************************************
 Date:              92.12.17
 Company:           BRUKER ELEKTRONIK GmbH
                    D-7512 Rheinstetten 4/Karlsruhe
                    Akazienweg 2
```

```
 Designer:           Helmut Knoerr
 PAL :               HPC0AA02-KE
 Part name:          HPCU Board DAC register load decoder
 Part number:        H6515
 Revision:           AA
 Assembly:           B-HPCU
 Part name:          B-HPCU Motor Control Unit Kpl.
 Part number:        H5149, (H6509)
 Revisions:          A+B
 Pal location:       IC U39
 File name:          HPCGMUX.DCB
;*********************************************************************
*X-NAMES
 RD            ; CPU READ SIGNAL
 WR            ; CPU WRITE SIGNAL
 IOENABLE      ; GENERAL BOARD SELECT
 A[7..0]       ; CPU ADDRESSES

*Y-NAMES
 CLKH[3..0]     ; HIGH ACTIVE CLOCKS FOR DATA REGISTERS
 CLKX[3..0]     ; HIGH ACTIVE CLOCKS FOR DATA REGISTERS
 CLK_MODE       ; HIGH ACTIVE CLOCK FOR MODE REGISTER
;_____________________________________________________________________
*FUNKTION-TABLE
 $(A[7..0]), IOENABLE , RD , WR : (CLKH[3..0]);
   00H     ,  0       ,  1 ,  0 :    0001B    ;  8000H
   01H     ,  0       ,  1 ,  0 :    0010B    ;  8001H
   02H     ,  0       ,  1 ,  0 :    0100B    ;  8002H
   03H     ,  0       ,  1 ,  0 :    1000B    ;  8003H
   REST                          :    0000B    ;

 $(A[4..0]), IOENABLE , RD , WR : (CLKX[3..0]);
   04H     ,  0       ,  1 ,  0 :    0001B    ;  8004H
   05H     ,  0       ,  1 ,  0 :    0010B    ;  8005H
   06H     ,  0       ,  1 ,  0 :    0100B    ;  8006H
   07H     ,  0       ,  1 ,  0 :    1000B    ;  8007H
   REST                          :    0000B    ;
 $(A[4..0]), IOENABLE , RD , WR : CLK_MODE    ;
   08H     ,  0       ,  1 ,  0 :    1        ;  8008H
   REST                          :    0        ;
;_____________________________________________________________________
*RUN-CONTROL
```

```
 LISTING  = SYMBOLTABLE ;
 OPTIMIZE = P-TERMS ;
*END



*IDENTIFICATION
;*********************************************************************
 Date:               92.12.17
 Company:            BRUKER ELEKTRONIK GmbH
                     D-7512 Rheinstetten 4/Karlsruhe
                     Akazienweg 2
 Designer:           Helmut Knoerr
 PAL :               HPC0AA02-KE
 Part name:          HPCU Board DAC register load decoder
 Part number:        H6515
 Revision:           AA
 Assembly:           B-HPCU
 Part name:          B-HPCU Motor Control Unit Kpl.
 Part number:        H5149, (H6509)
 Revisions:          A+B
 Pal location:       IC U39
 File name:          HPCGMUX.DDV
;*********************************************************************
@DEVICE = HPCGLOGA;
*PLD
 TYPE = AMPAL22V10;
 CHECKSUM = COMPUTE;
*PINS
 A0       = 1 ,
 A1       = 2 ,
 A2       = 3 ,
 A3       = 4 ,
 A4       = 5 ,
 A5       = 6 ,
 A6       = 7 ,
 A7       = 8 ,
 WR       = 9,
 RD       = 10,
 IOENABLE = 11,
 CLKH0    = 14 ,
 CLKH1    = 15 ,
```

```
 CLKH2    = 16 ,
 CLKH3    = 17 ,
 CLKX0    = 19 ,
 CLKX1    = 20 ,
 CLKX2    = 21 ,
 CLKX3    = 22 ,
 CLK_MODE = 23 ;
*RUN-CONTROL
 LISTING  = EQUATIONS,PINOUT ;
 PROGFORMAT = JEDEC ;
 TESTVECTORS = GENERATE;
@ENDDEVICE = HPCGLOGA;
*END
```

Schematics *12.2*

Documentation overview *12.2.1*

B-HPCU MAIN OVERVIEW	Revised: December 14, 1992
File MCUCON_B.SHE	Sheet 1 of 24
H3S100253´	Revision: B
B-HPCU Bitstream Overview	Revised: December 14, 1992
File MCBIT_B.SHE	Sheet 2 of 24
H3S100263	Revision: B
B-HPCU Bitstream CPU	Revised: December 14, 1992
File MCBICP_B.SHE	Sheet 3 of 24
H3S100273	Revision: B
B-HPCU Pulse Detection Overview	Revised: December 14, 1992
File MCBIPU_B.SHE	Sheet 4 of 24
H3S100293	Revision: B
B-HPCU Y Pulse Detection	Revised: December 14, 1992
File MCBIPY_A.SHE	Sheet 5 of 24
H3S100323	Revision: B
B-HPCU H Pulse Detection	Revised: December 14, 1992
File MCBIPH_B.SHE	Sheet 6 of 24
H3S100313	Revision: B
B-HPCU X Pulse Detection	Revised: December 14, 1992
File MCBIPX_B.SHE	Sheet 7 of 24
H3S100303	Revision: B
B-HPCU Keyboard I/O	Revised: December 14, 1992
File MCBIIO_B.SHE	Sheet 8 of 24
H3S100293	Revision: B

B-HPCU Motor / Interface Overview	Revised: December 14, 1992
File MCMOIN_B.SHE	Sheet 9 of 24
H3S100333	Revision: B
B-HPCU GAIN CONTROL Overview	Revised: December 14, 1992
File MCGAIN_B.SHE	Sheet 10 of 24
H3S100433	Revision: B
B-HPCU Registers Gain H	Revised: December 14, 1992
File MCGREH_B.SHE	Sheet 11 of 24
H3S100453	Revision: B
B-HPCU Registers Gain X	Revised: December 14, 1992
File MCGREX_B.SHE	Sheet 12 of 24
H3S100463	Revision: B
B-HPCU Digital To Analog Part	Revised: December 14, 1992
File MCGAOU_B.SHE	Sheet 13 of 24
H3S100473	Revision: B
B-HPCU GAIN Logic Control	Revised: December 14, 1992
File MCGLOG_B.SHE	Sheet 14 of 24
H3S100443	Revision: B
B-HPCU Main CPU	Revised: December 14, 1992
File MCMACP_B.SHE	Sheet 15 of 24
H3S100343	Revision: B
B-HPCU RS232 / SBS Interfaces Overview	Revised: December 14, 1992
File MCINTF_B.SHE	Sheet 16 of 24
H3S100393	Revision: B
B-HPCU RS232 To MAS	Revised: December 14, 1992
File MCINMA_B.SHE	Sheet 17 of 24
H3S100413	Revision: B

B-HPCU SBS Bus	Revised: December 14, 1992
File MCINSB_B.SHE	Sheet 18 of 24
H3S100423	Revision: B
B-HPCU RS232 To X32	Revised: December 14, 1992
File MCINX2_B.SHE	Sheet 19 of 24
H3S100403	Revision: B
B-HPCU Motor Control Overview	Revised: December 14, 1992
File MCMOCO_B.SHE	Sheet 20 of 24
H3S100353	Revision: B
B-HPCU 1H Amplifier Motor Control	Revised: December 14, 1992
File MCMO1H_B.SHE	Sheet 21 of 24
H3S100363	Revision: B
B-HPCU Amplifier / Control Connector	Revised: December 17, 1992
File MCMOIO_B.SHE	Sheet 22 of 24
H3S100383	Revision: B
B-HPCU X Amplifier Motor Control	Revised: December 14, 1992
File MCMOX_B.SHE	Sheet 23 of 24
H3S100373	Revision: B
B-HPCU Expansion Slot	Revised: December 14, 1992
File MCEXCO_B.SHE	Sheet 24 of 24
H3S100483	Revision: B

Figure 12.1. B-HPCU MAIN OVERVIEW

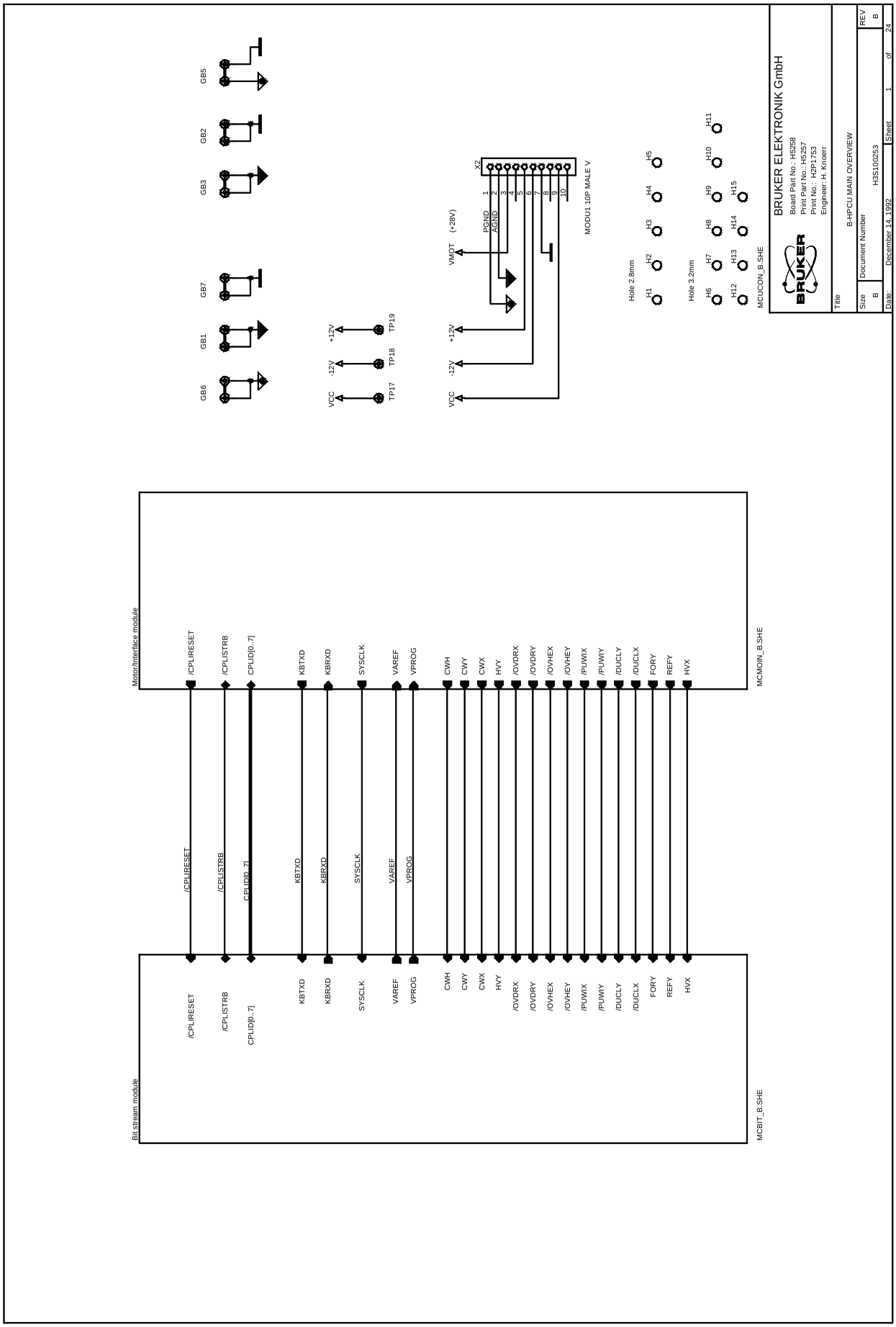


Figure 12.2. B-HPCU Bits Tream Overview

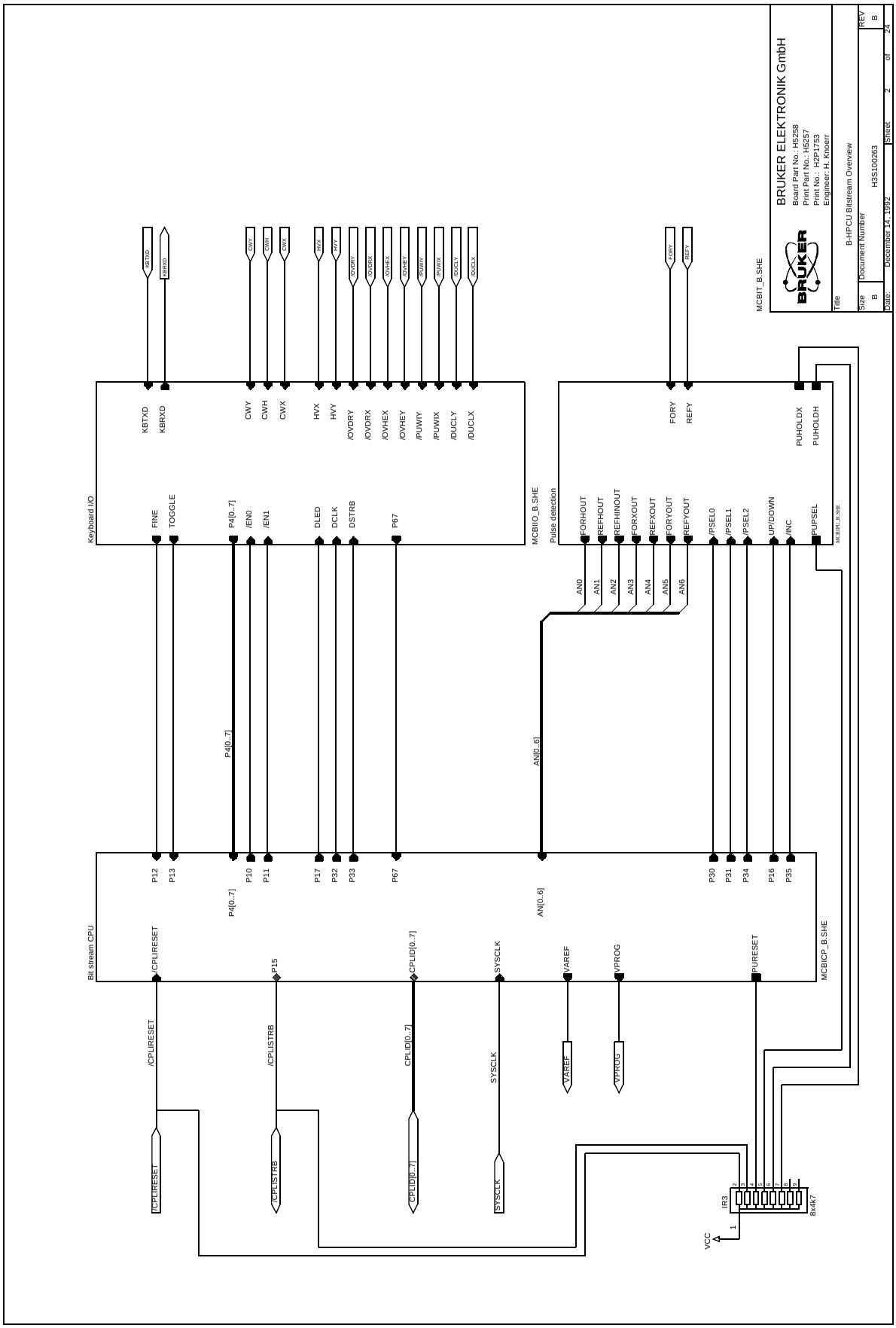


Figure 12.3. B-HPCU Bitstream CPU

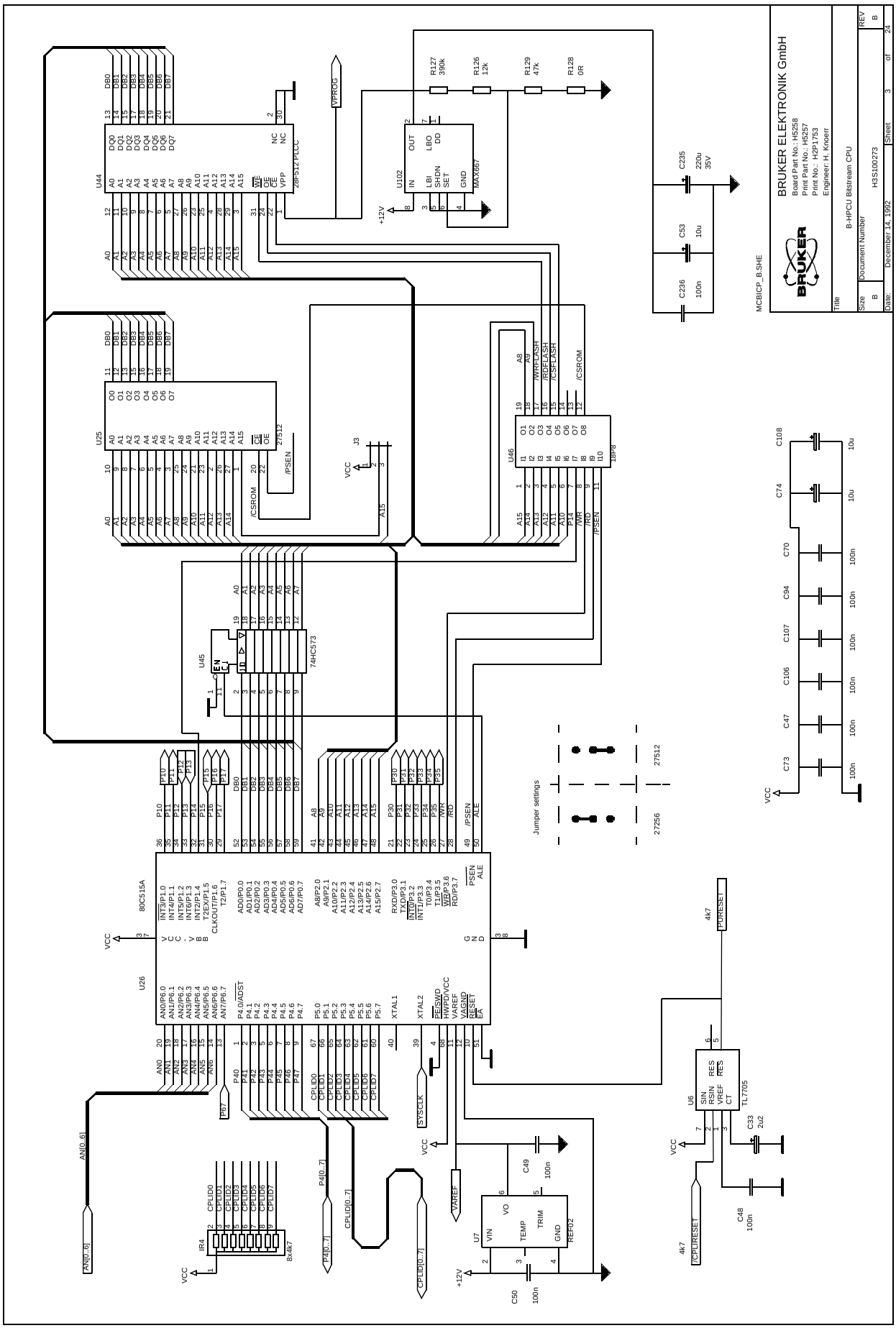


Figure 12.4. B-HPCU Pulse Detection Overview

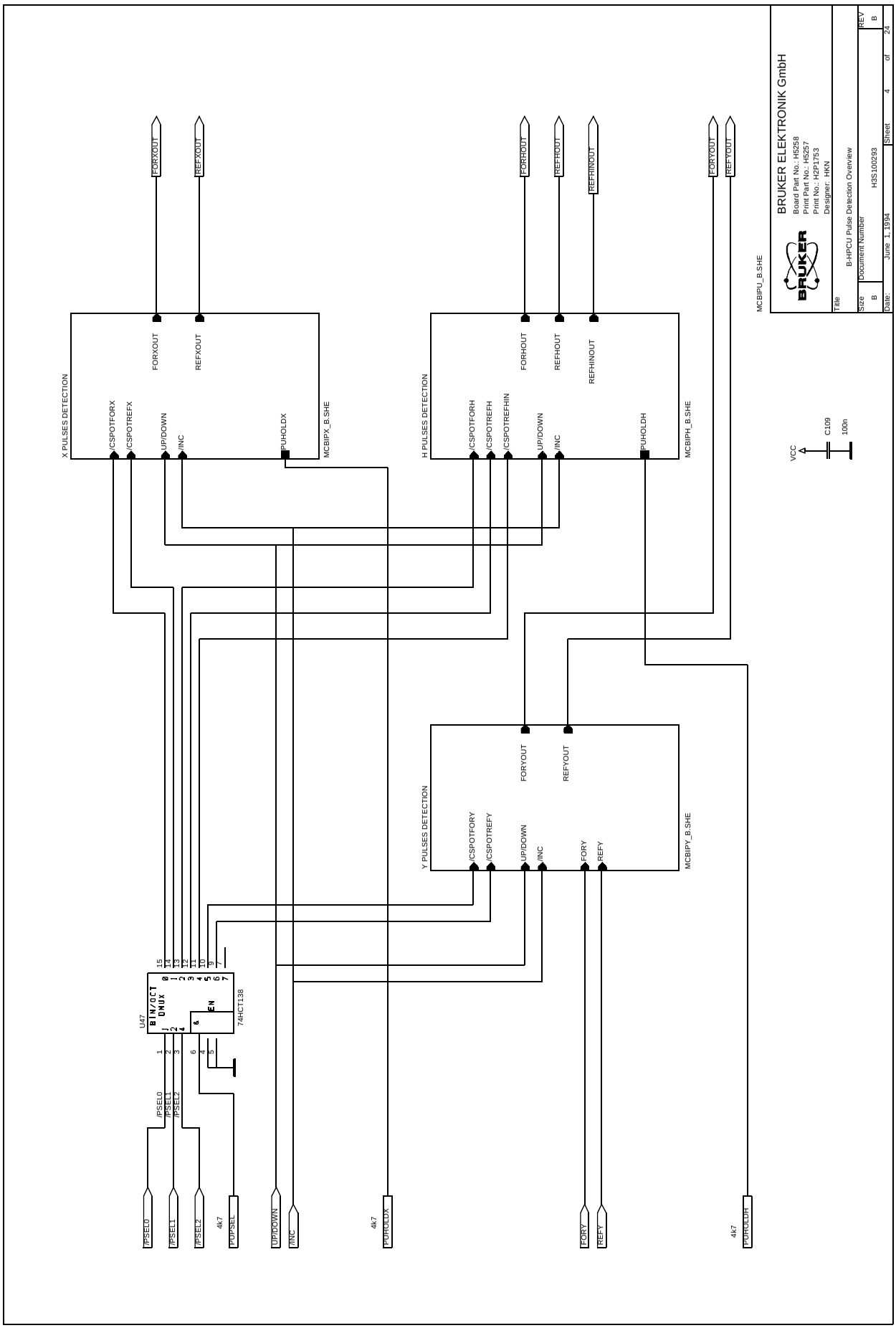


Figure 12.5. Pulse Detection

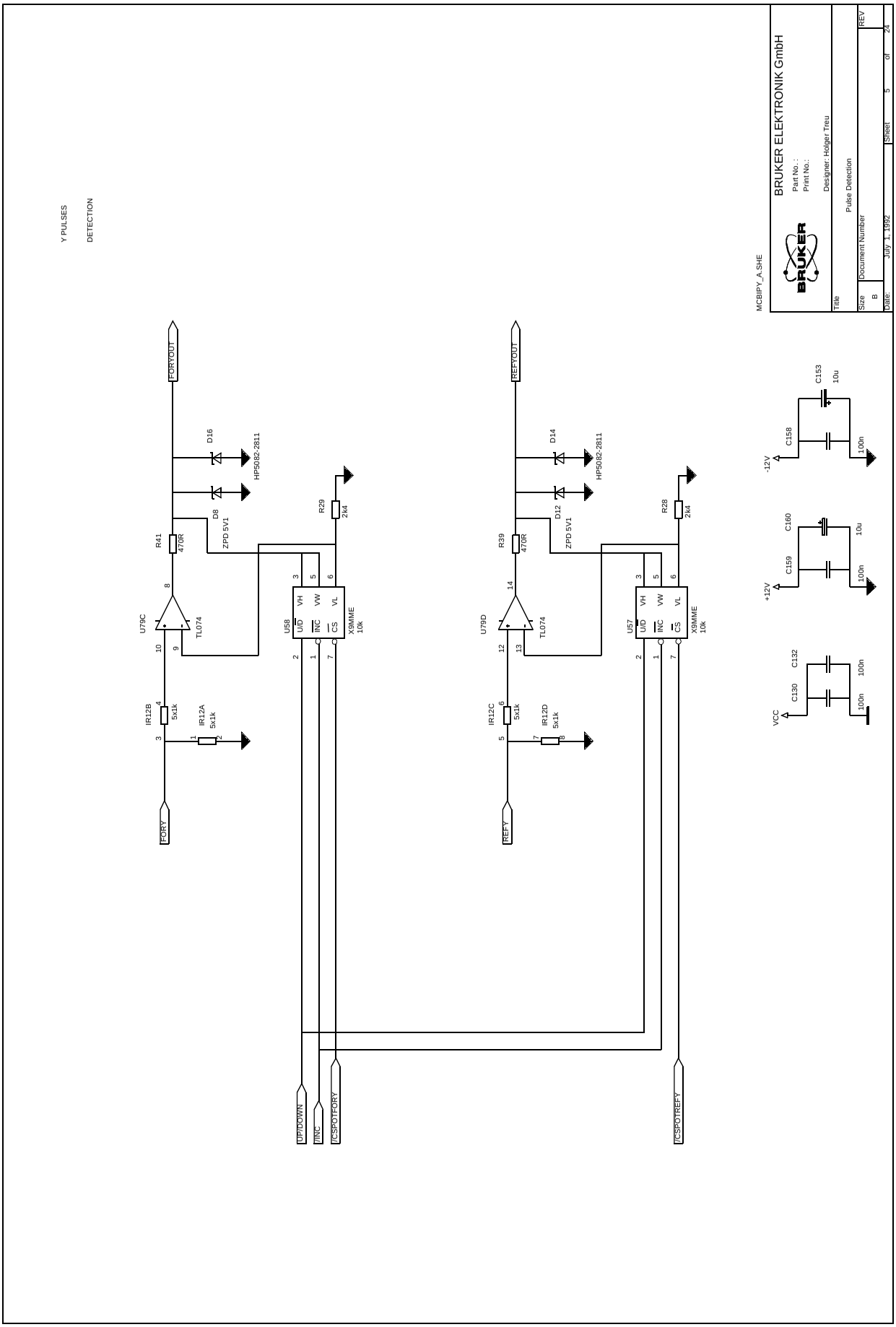
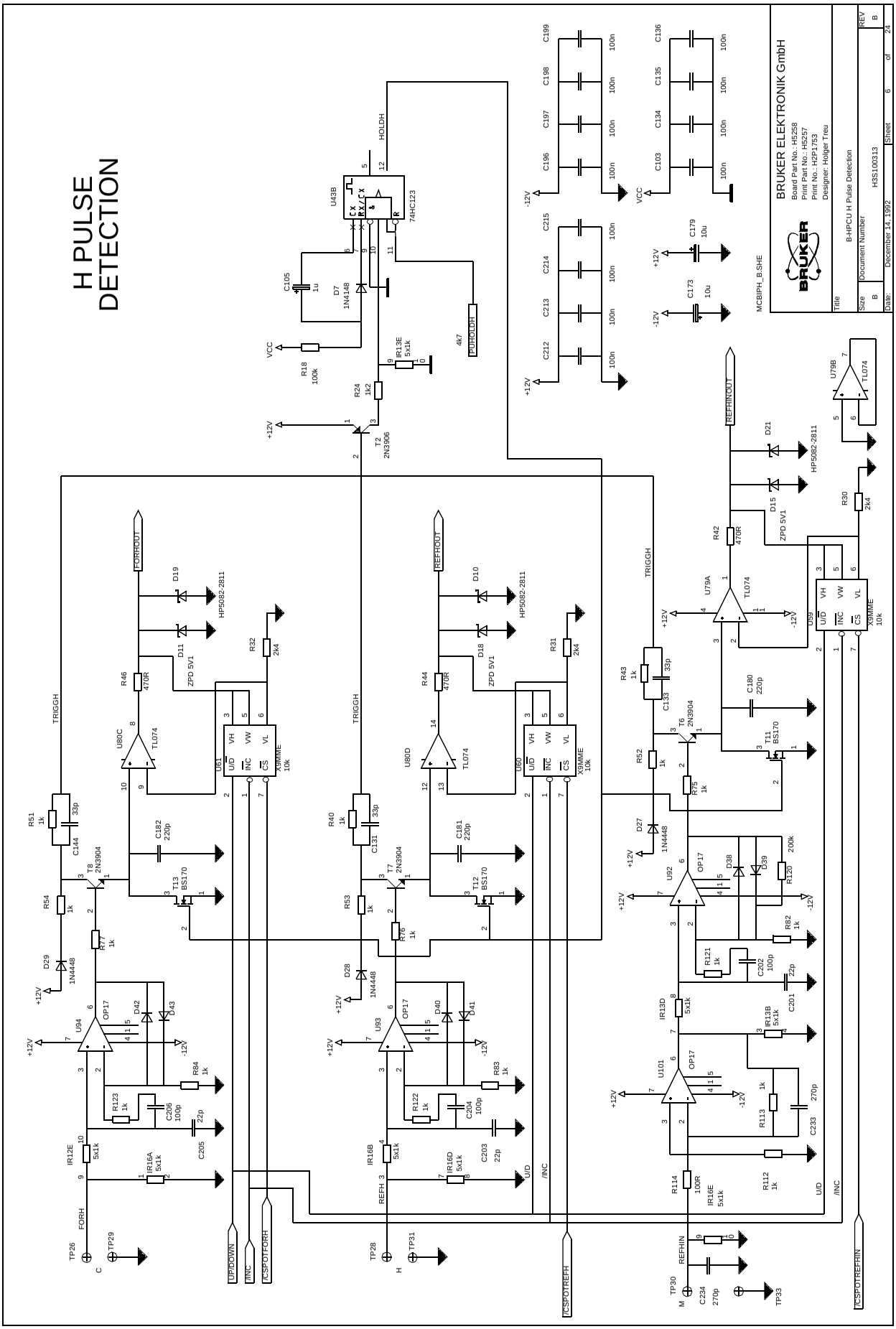


Figure 12.6. B-HPCU H Pulse Detection



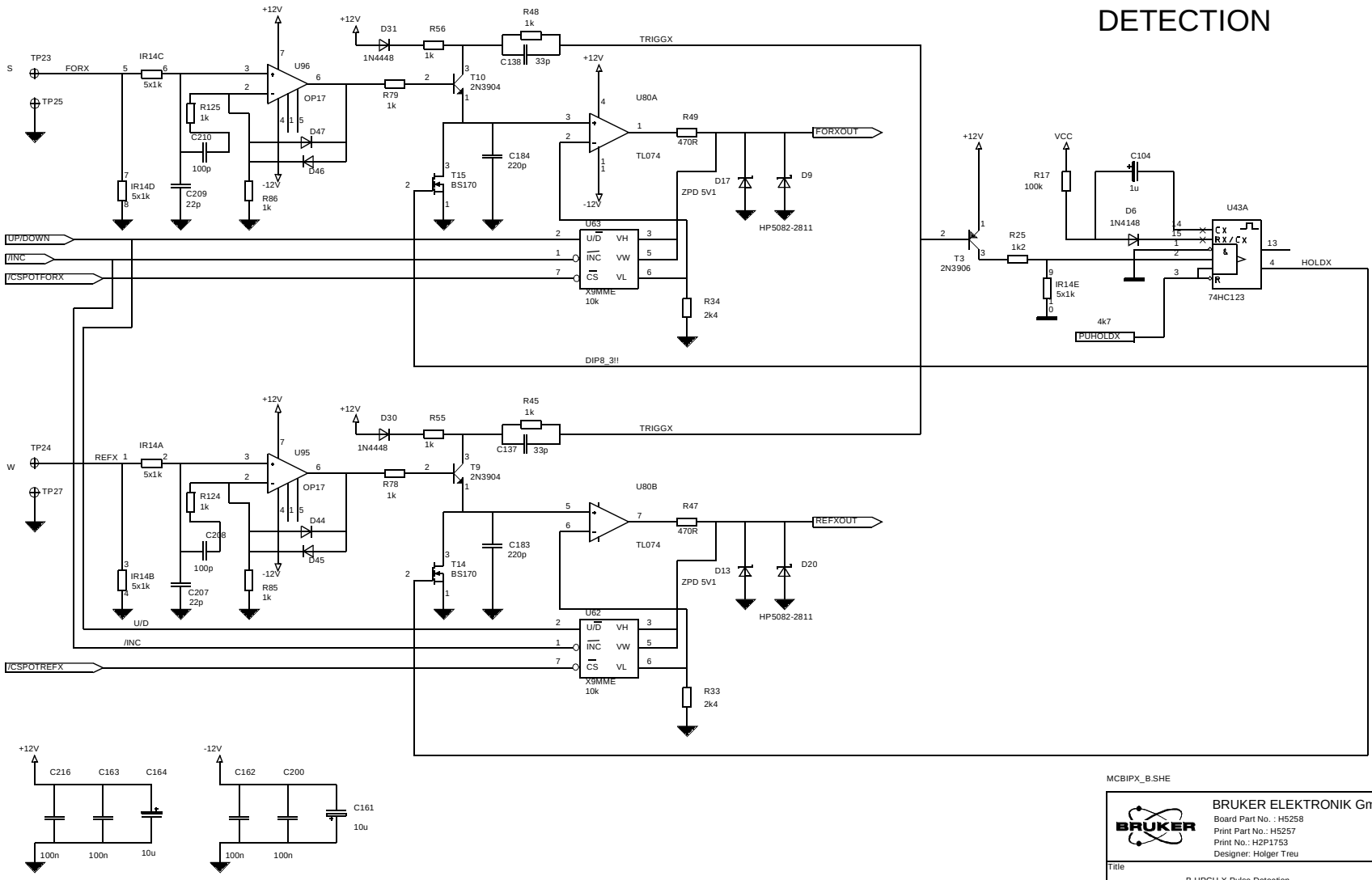


Figure 12.7. B-HPCU X Pulse Detection

Figure 12.8. B-HPCU Keyboard I/O

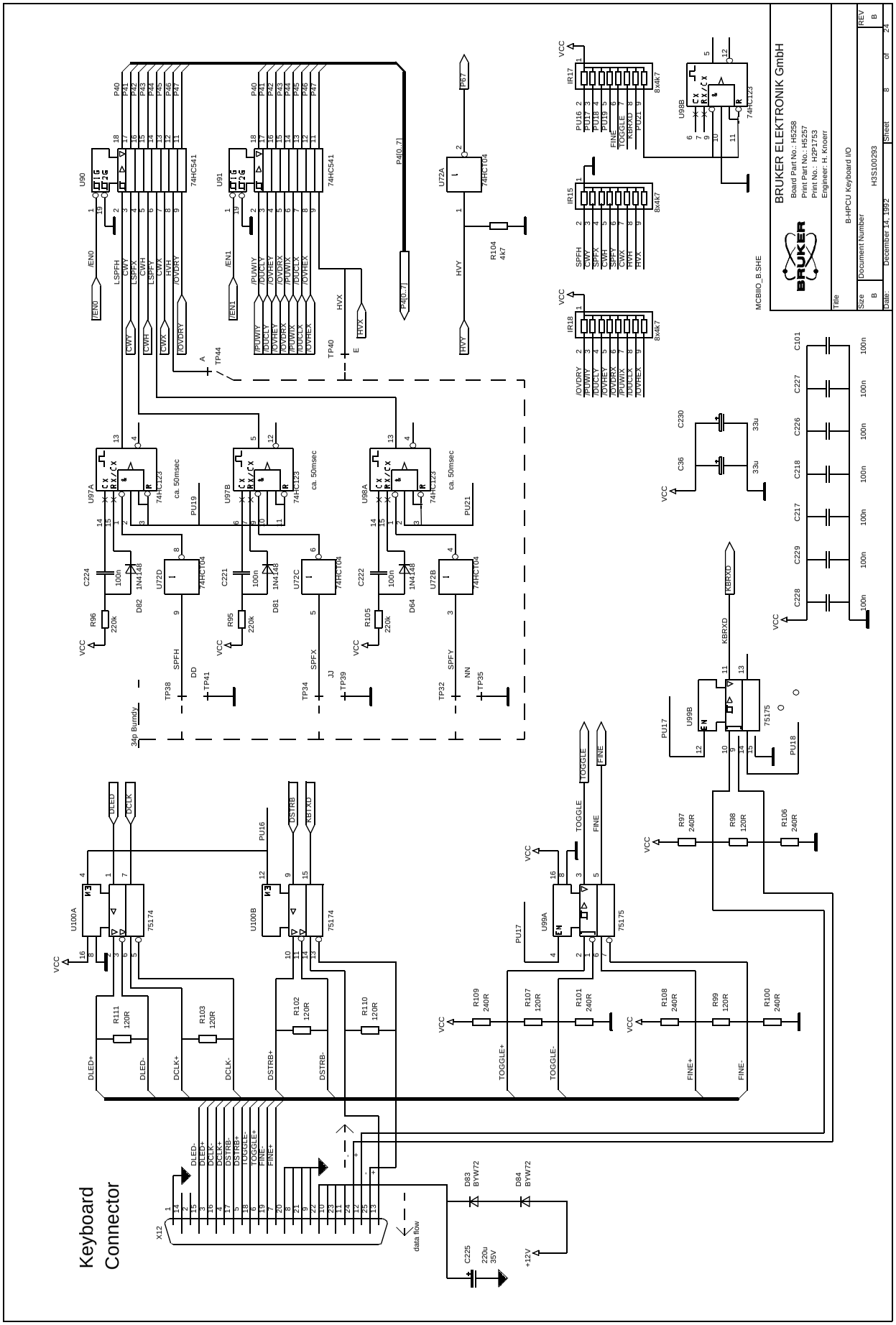


Figure 12.9. B-HPCU Moter Interface Overview

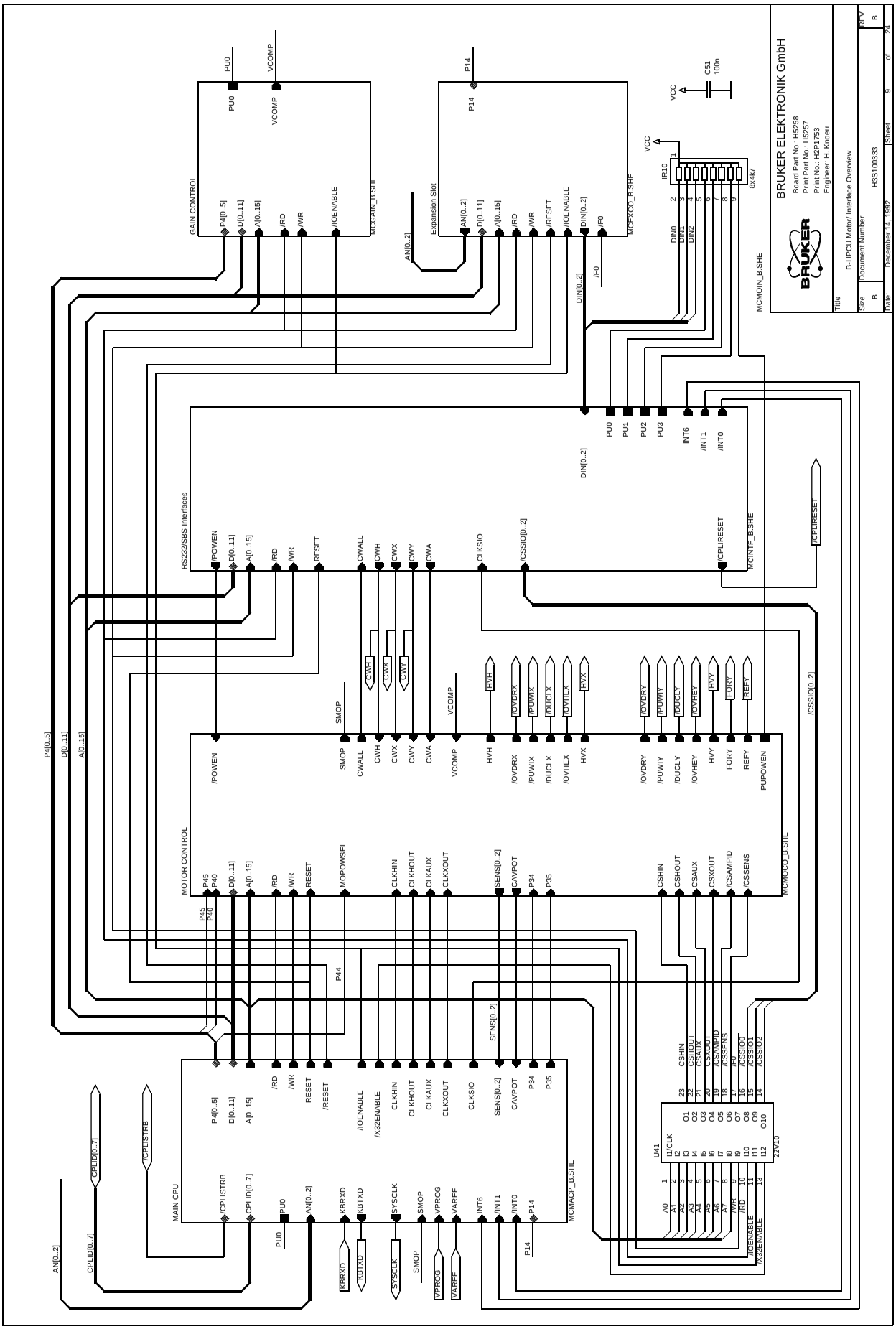


Figure 12.10.B-HPCU GAIN CONTROL Overview

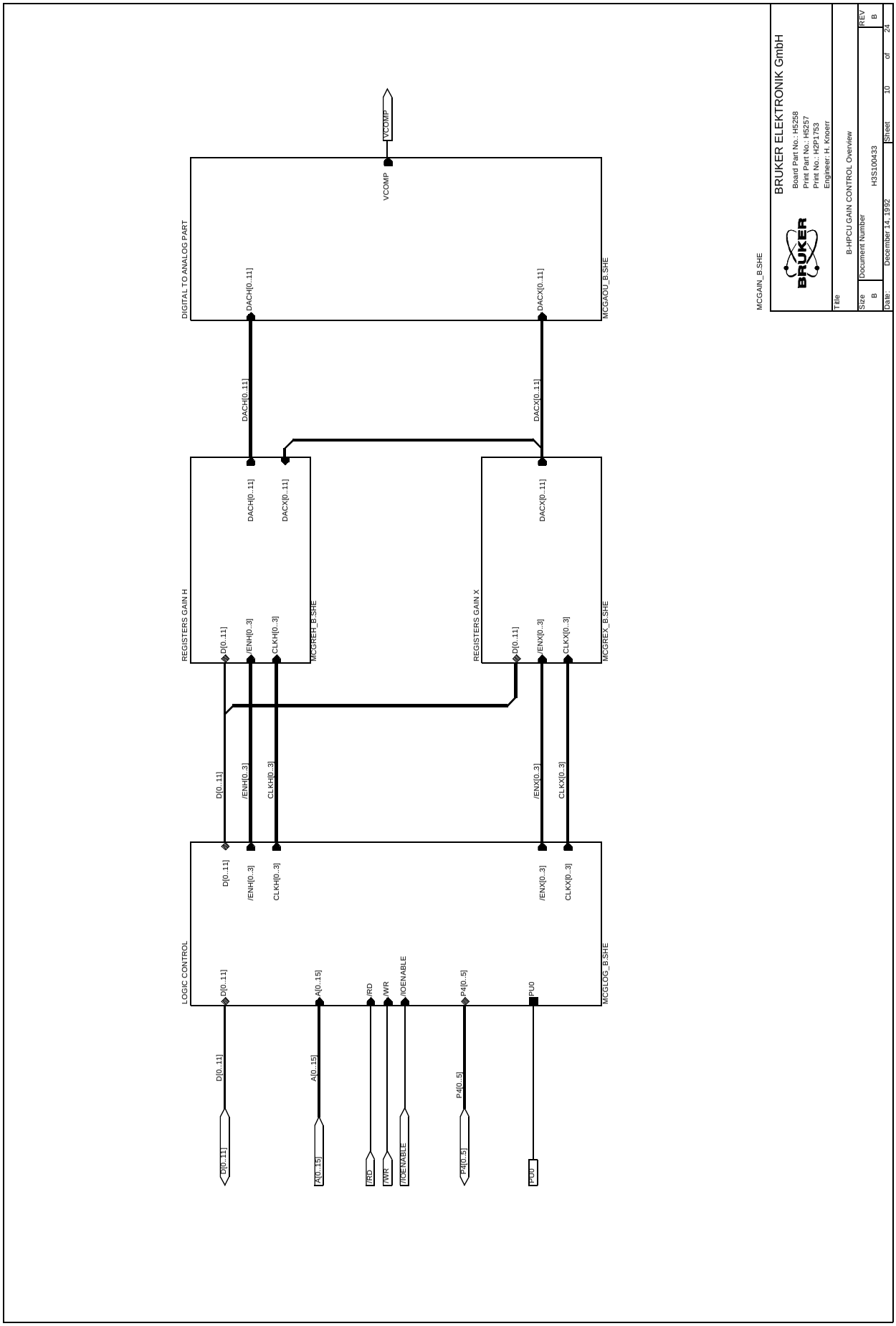


Figure 12.11.B-HPCU Registers Gain H

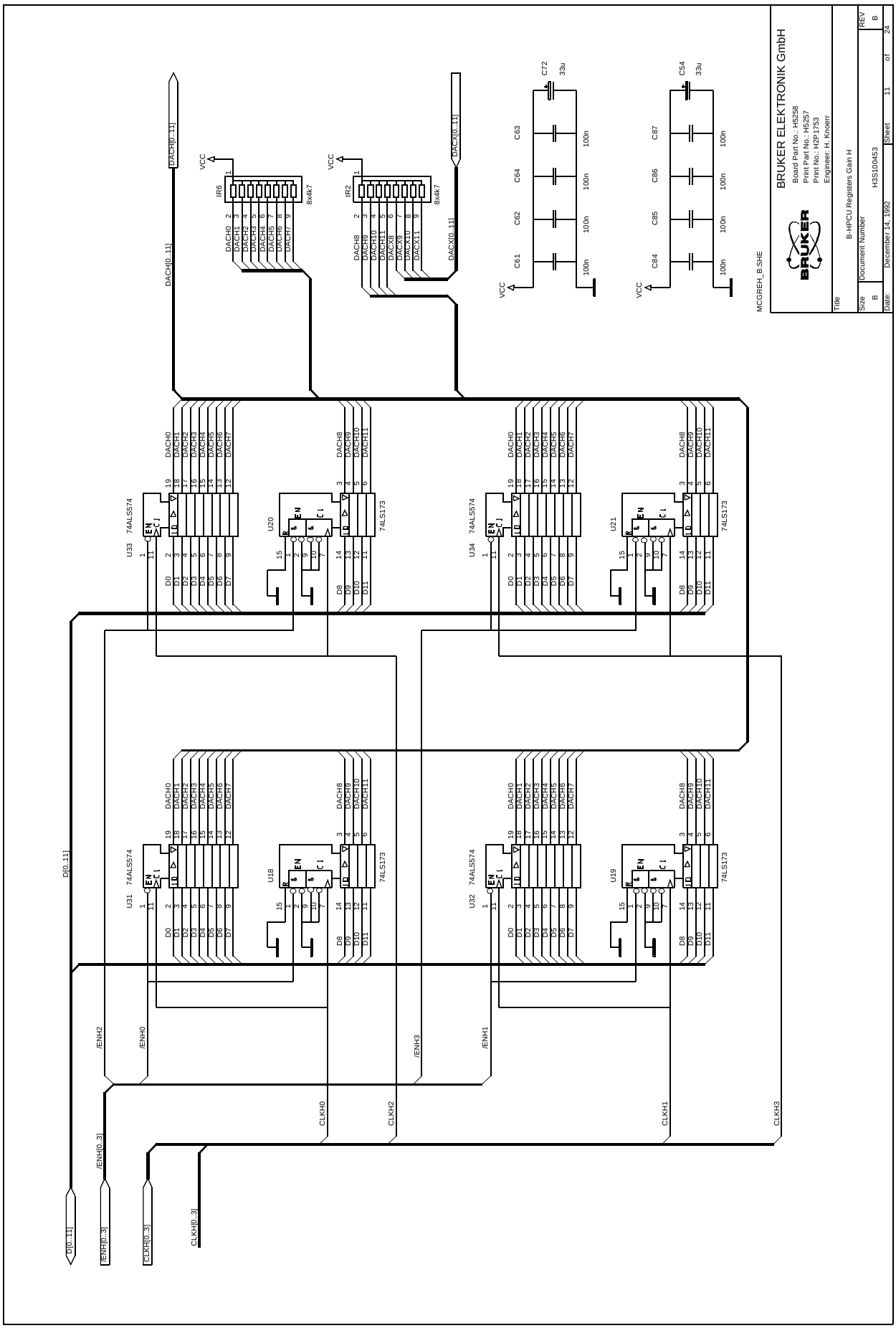


Figure 12.12.B-HPCU Registers Gain X

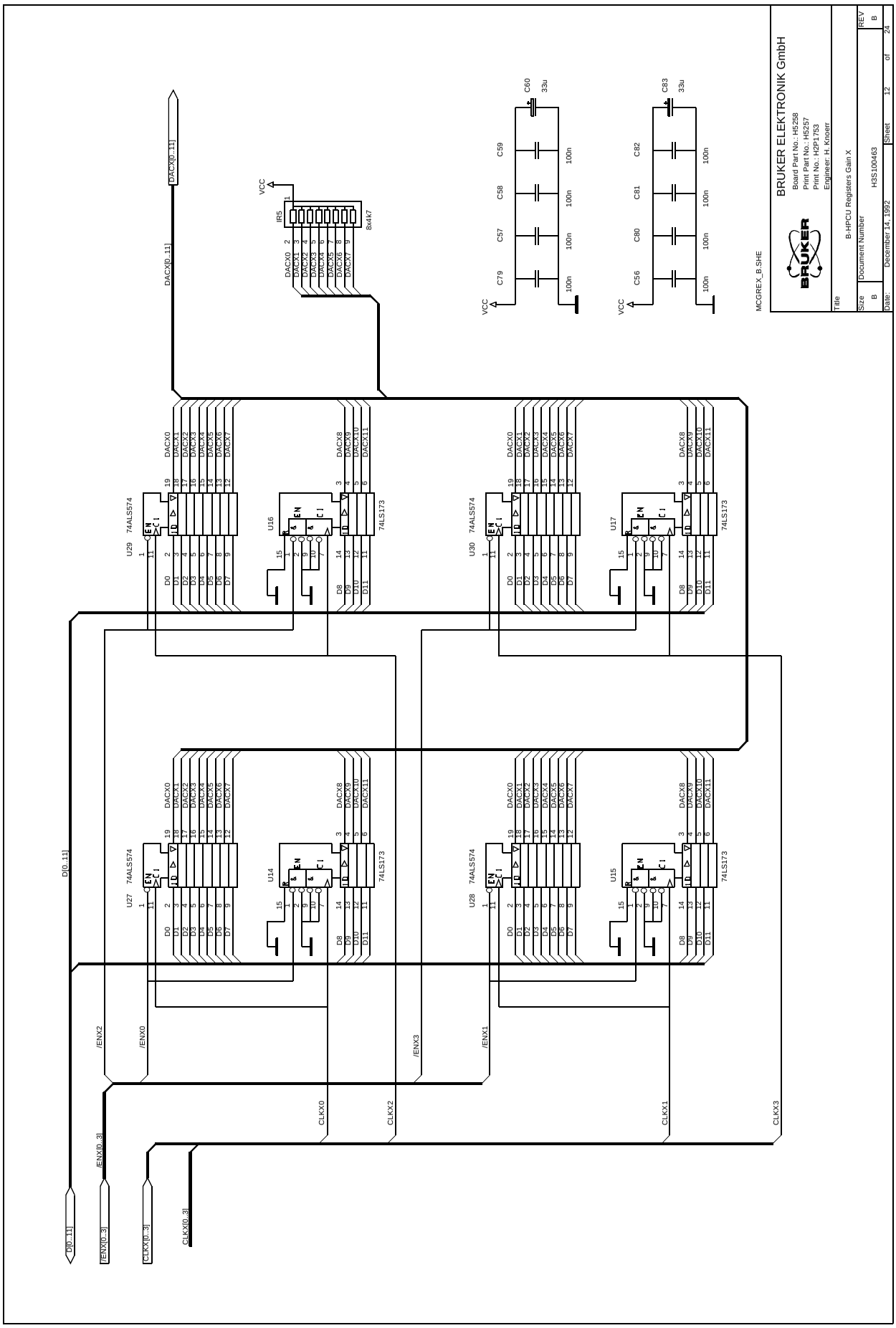


Figure 12.13.B-HPCU Digital To Analog Part

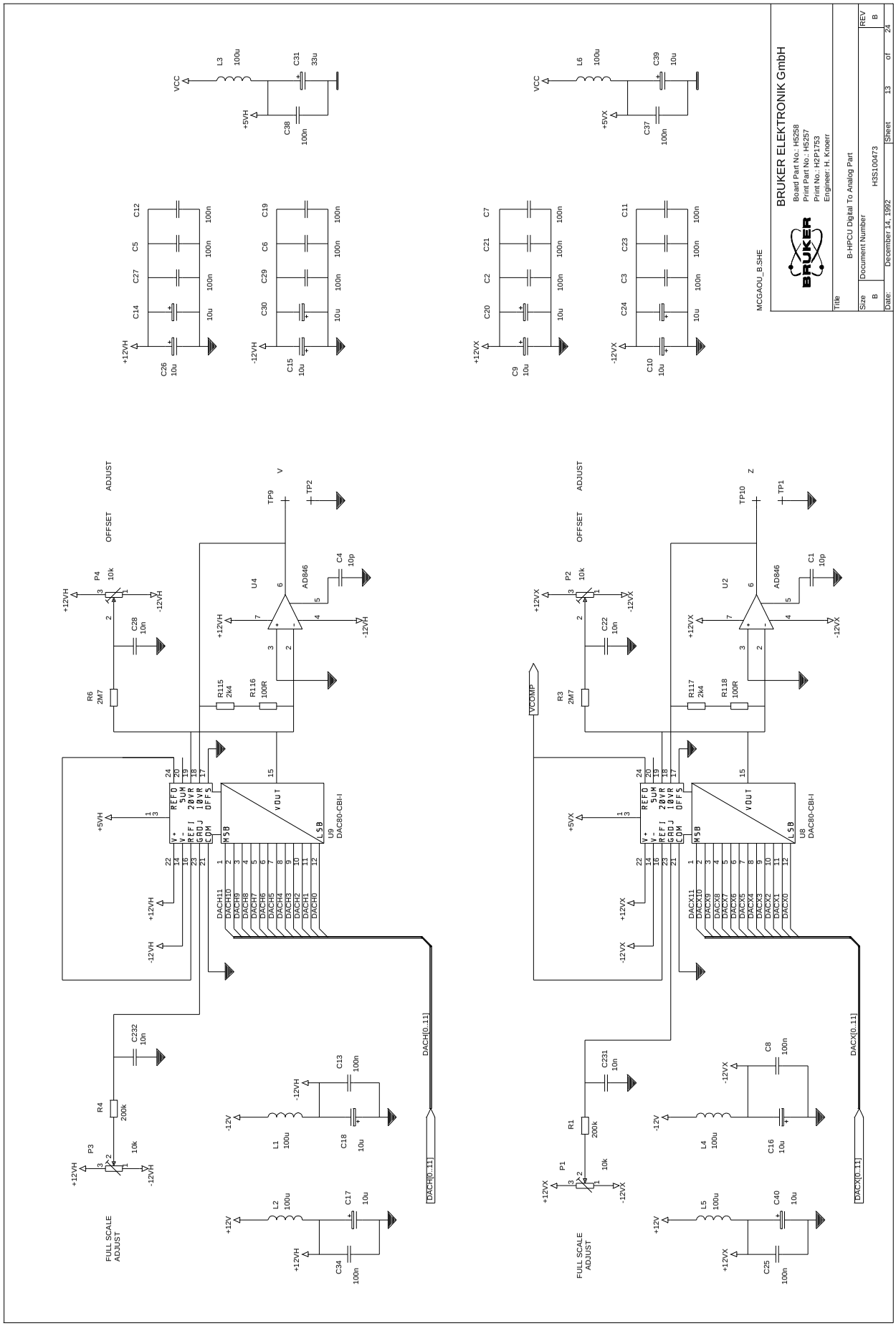
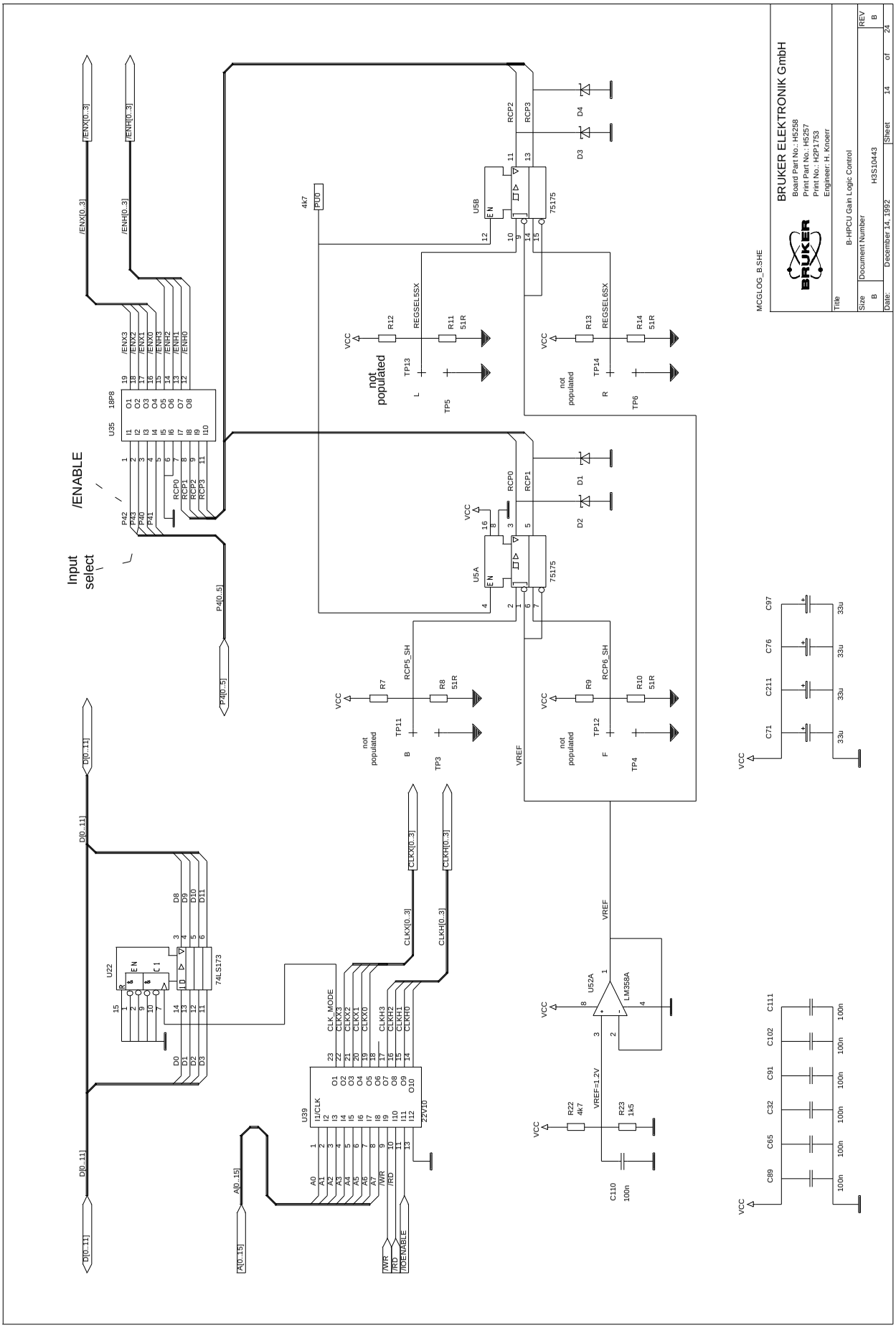


Figure 12.14.B-HPCU Gain Logic Control



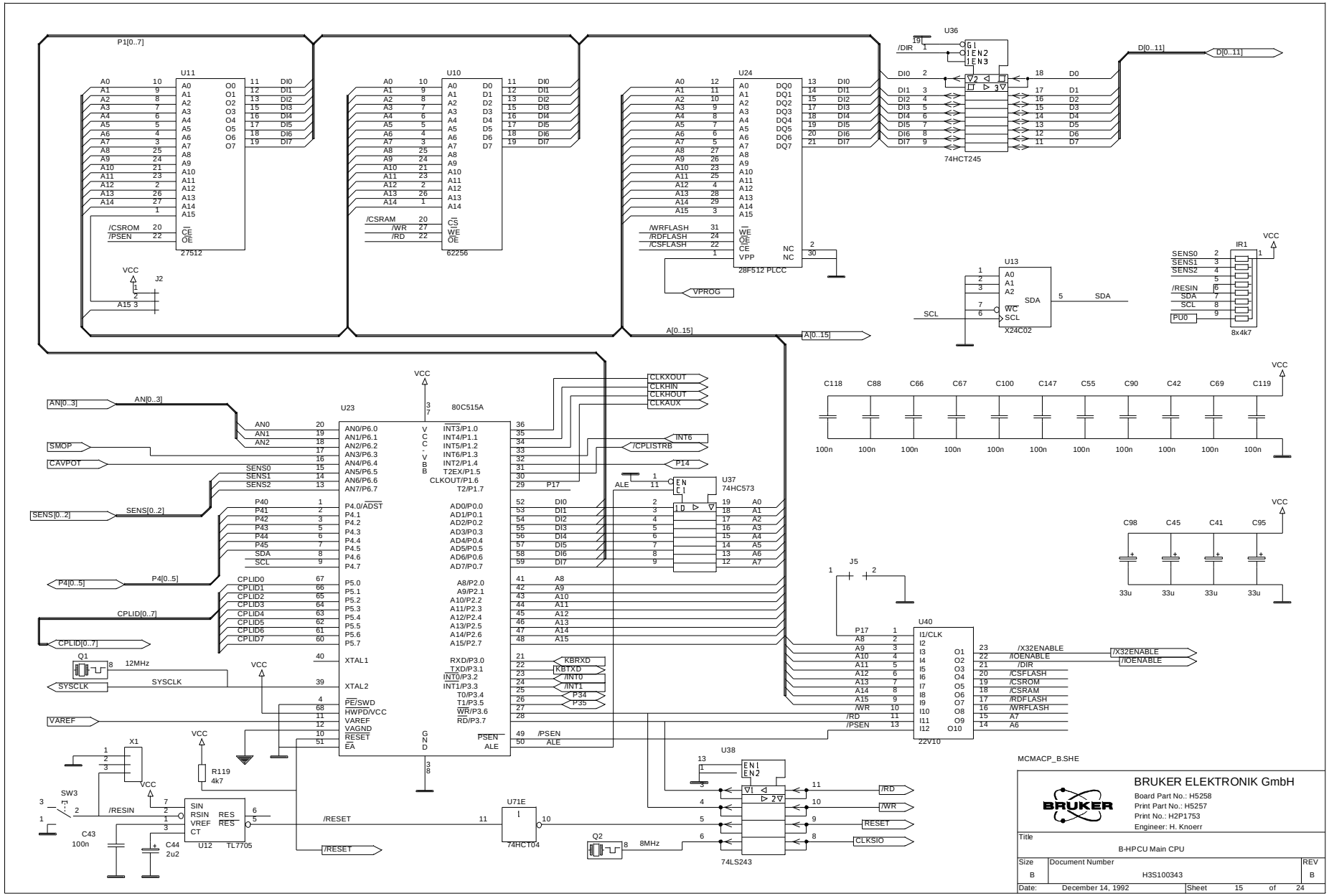


Figure 12.15.B-HPCU Main CPU

Figure 12.16.B-HPCU RS232/SBS Interface Overview

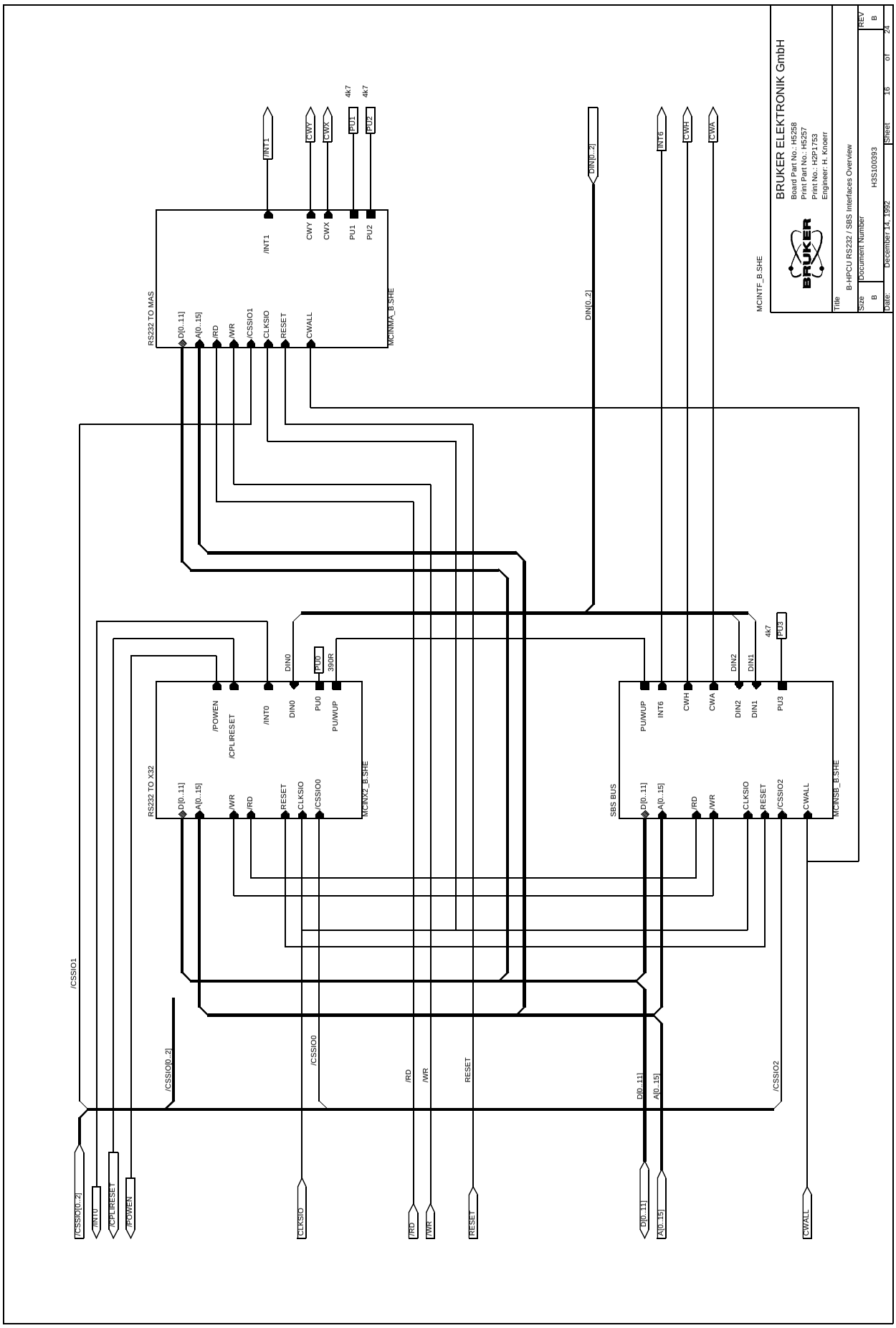


Figure 12.17.B-HPCU RS232 To MAS

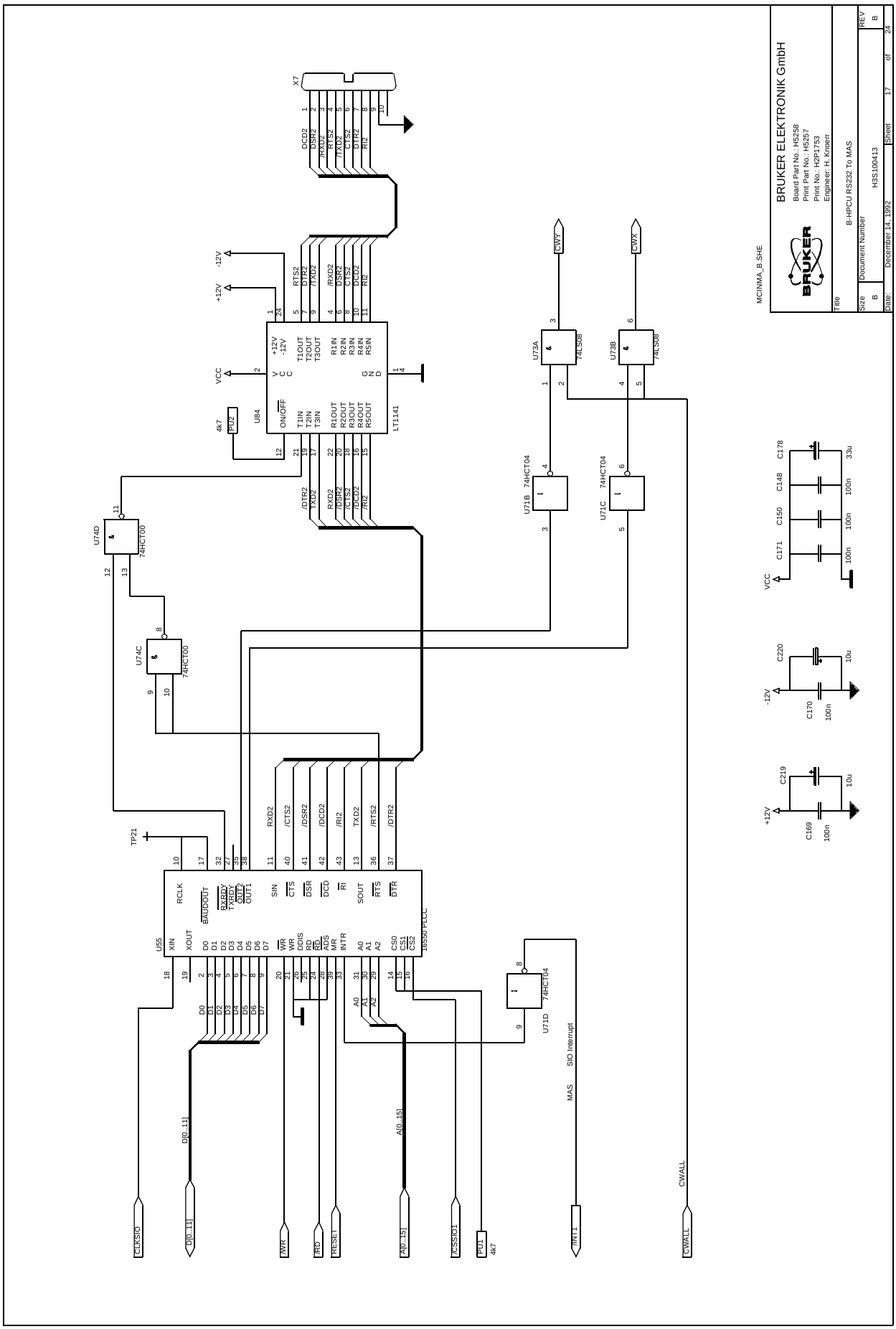


Figure 12.18.B-HPCU SBS Bus

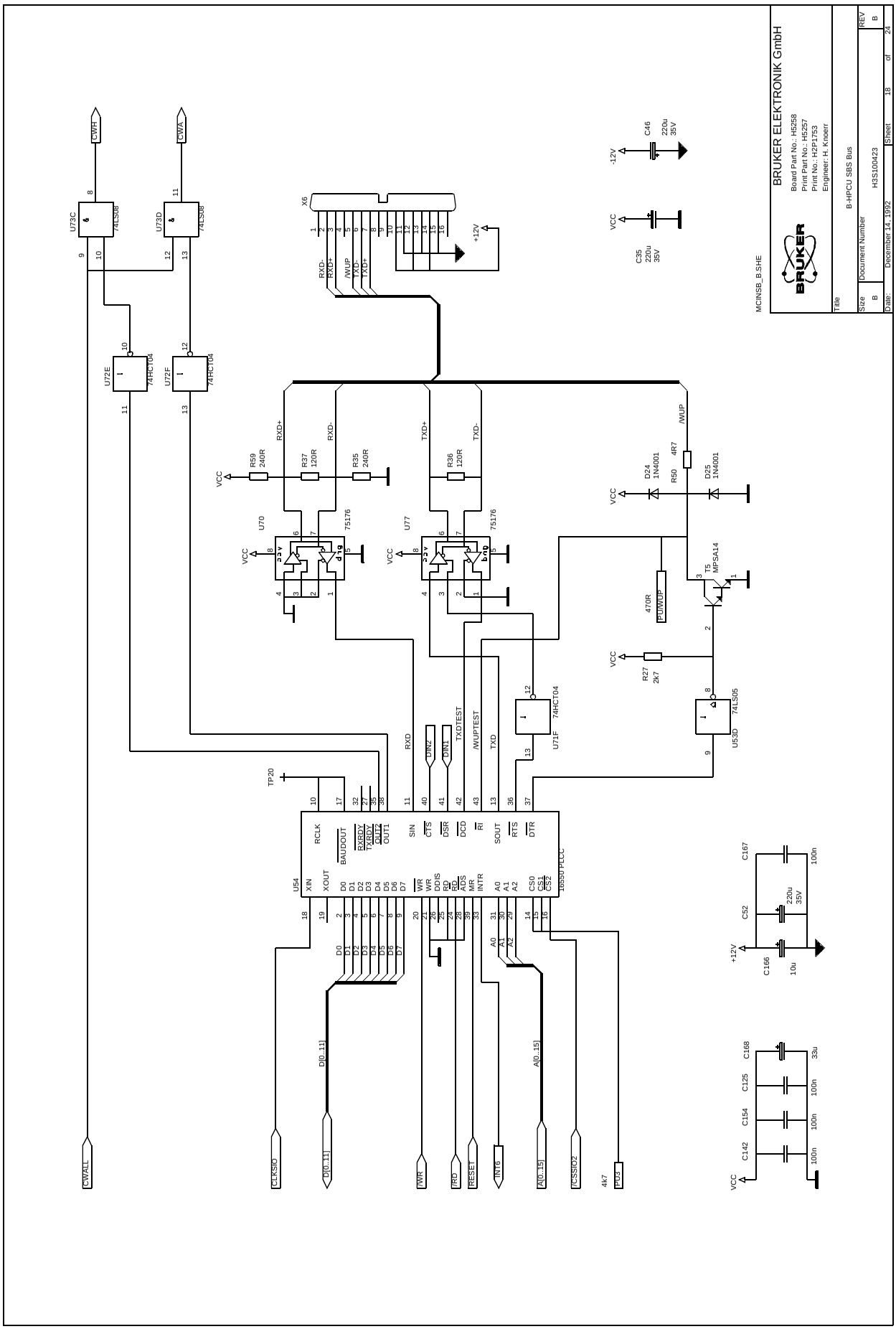


Figure 12.19.B-HPCU RS232 To X32

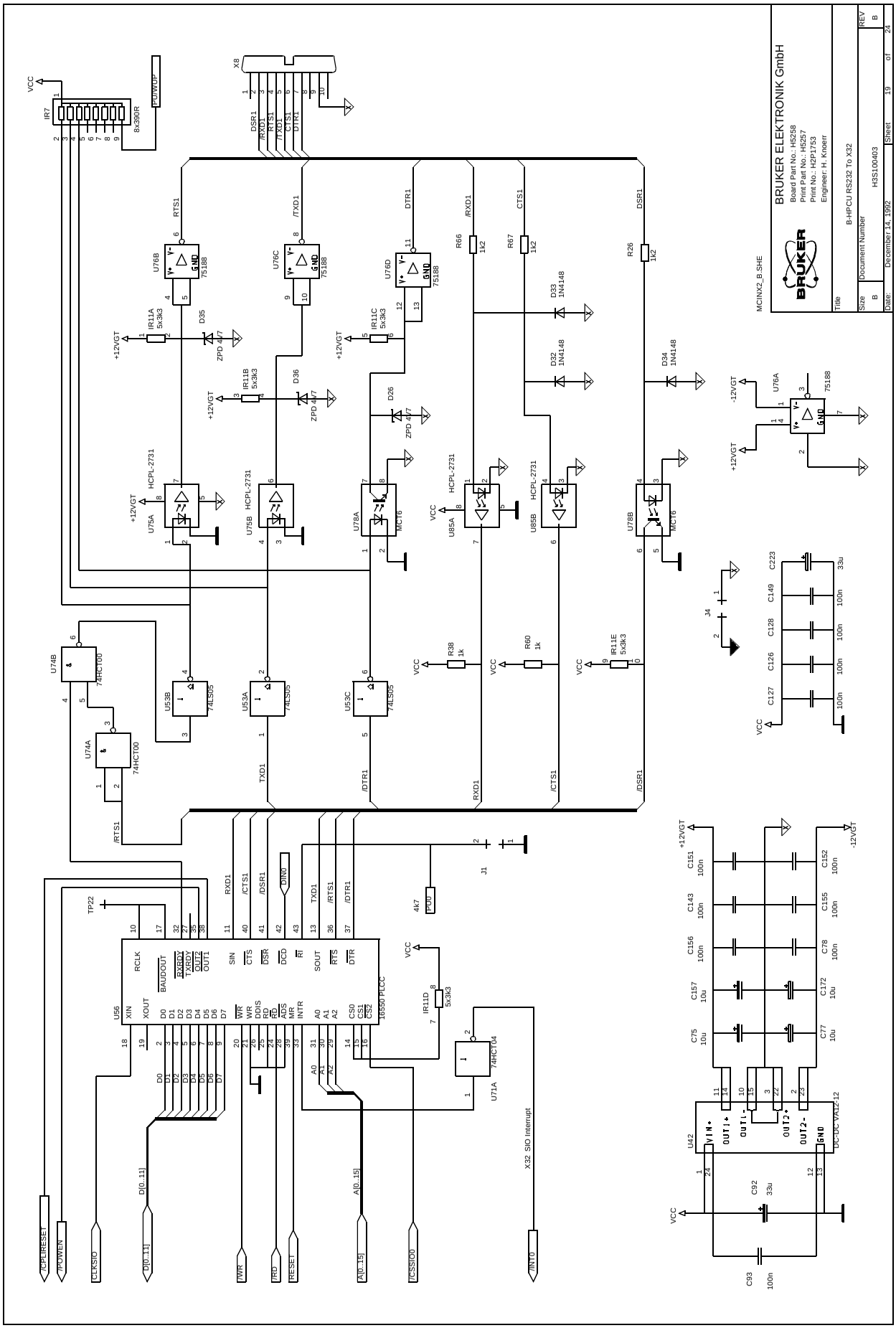


Figure 12.20.B-HPCU Motor Control Overview

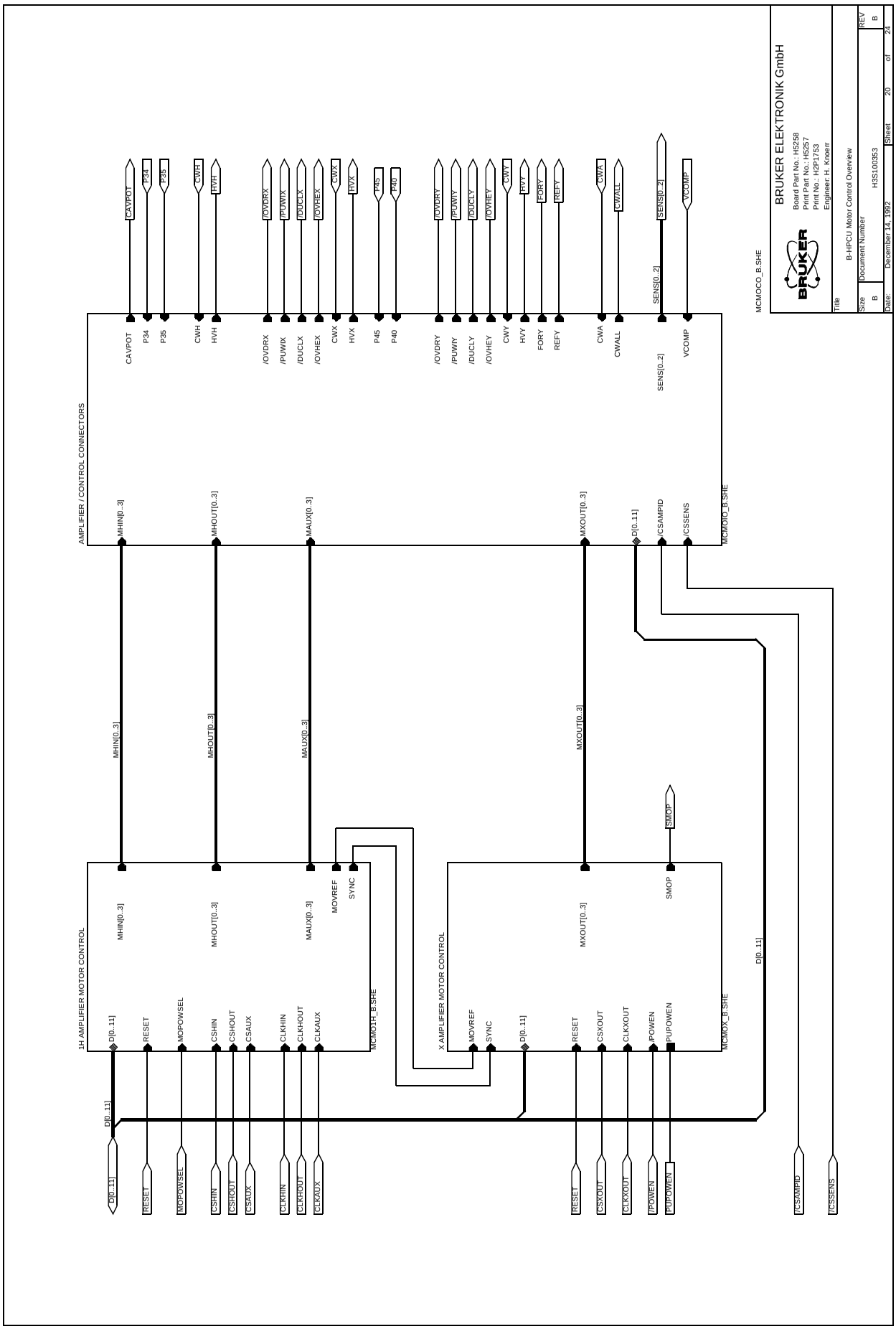
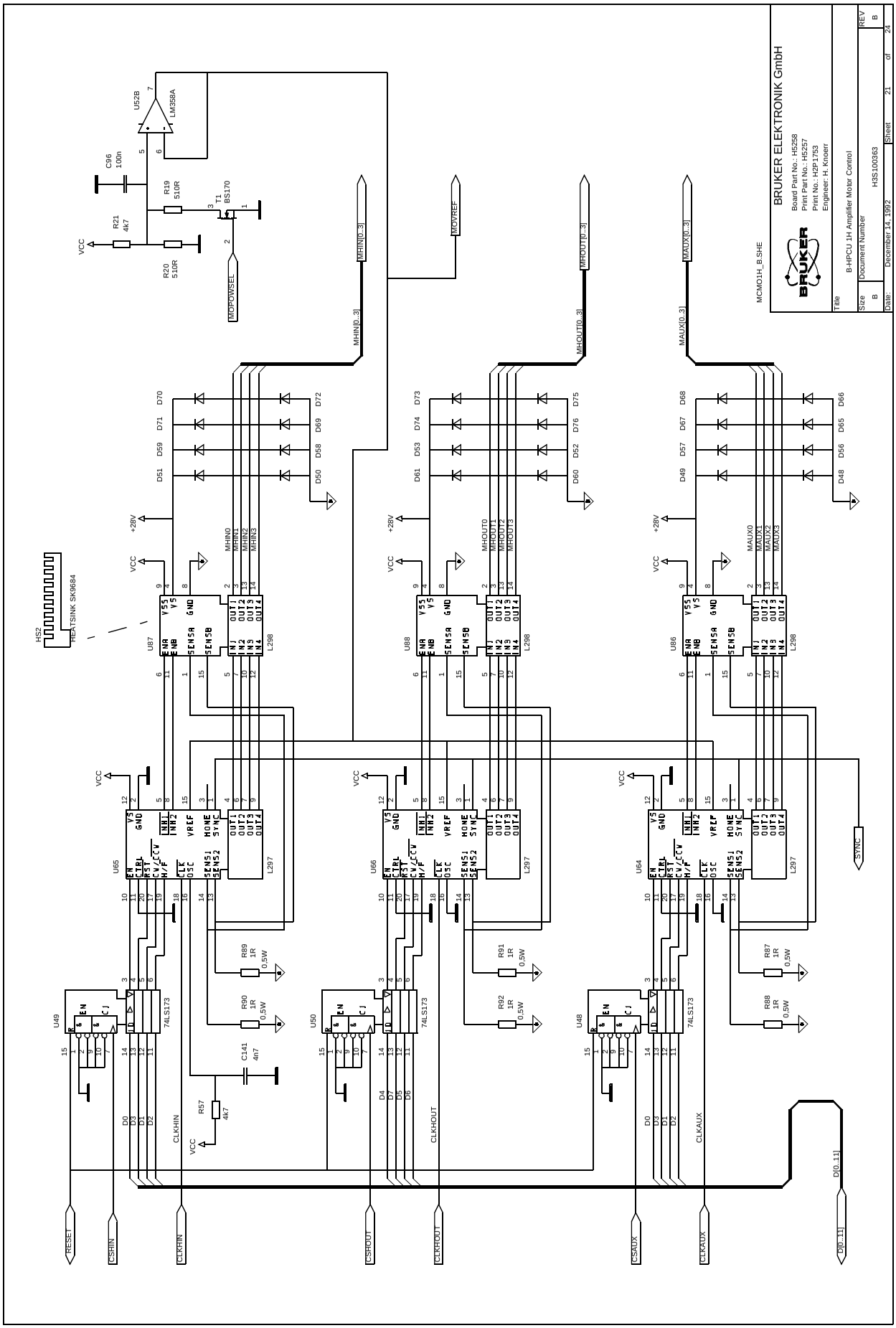


Figure 12.21.B-HPCU 1H Amplifier Motor Control



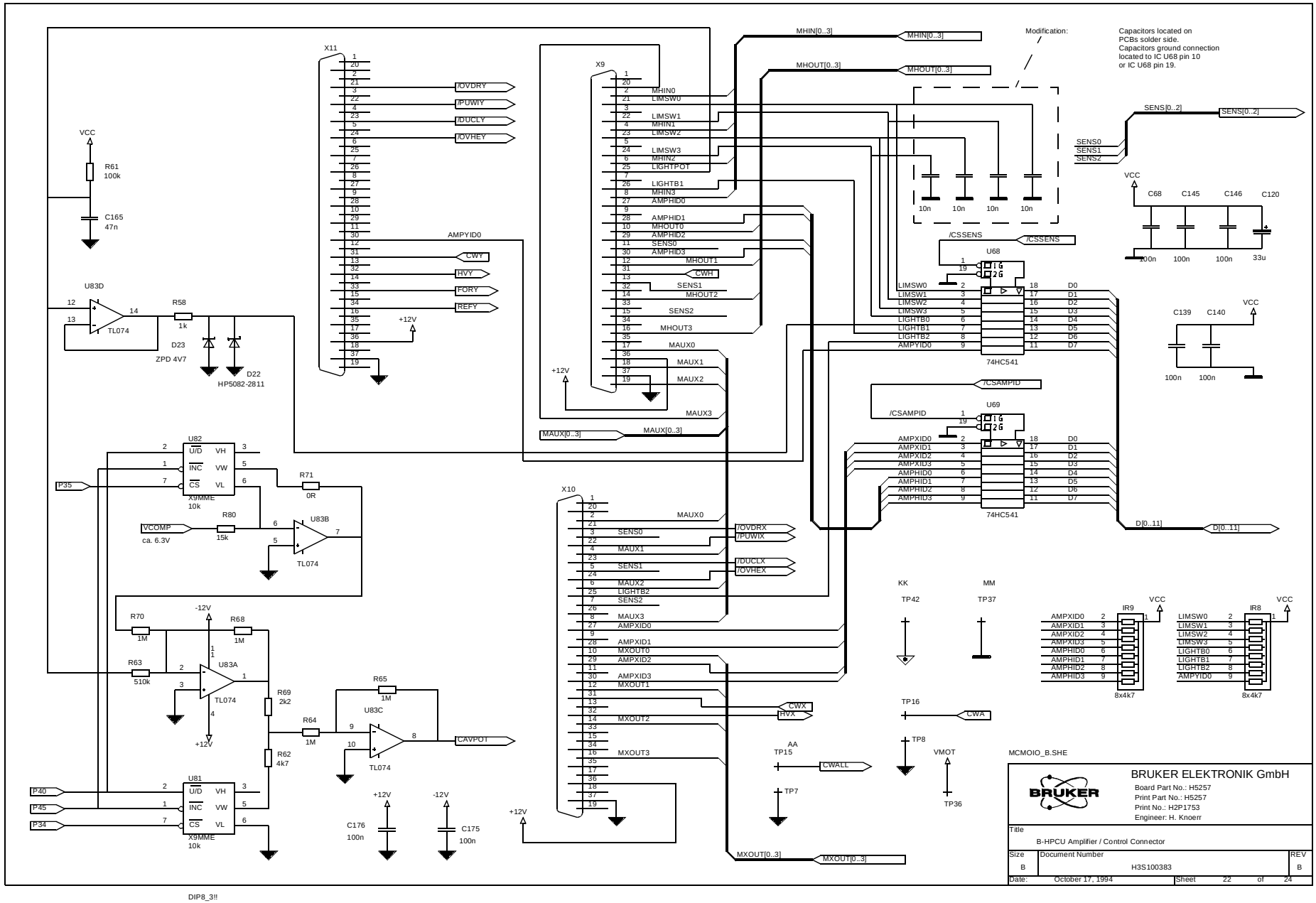


Figure 12.22.B-HPCU Amplifier / Control Connector

Figure 12.23.B-HPCU X Amplifier Motor Control

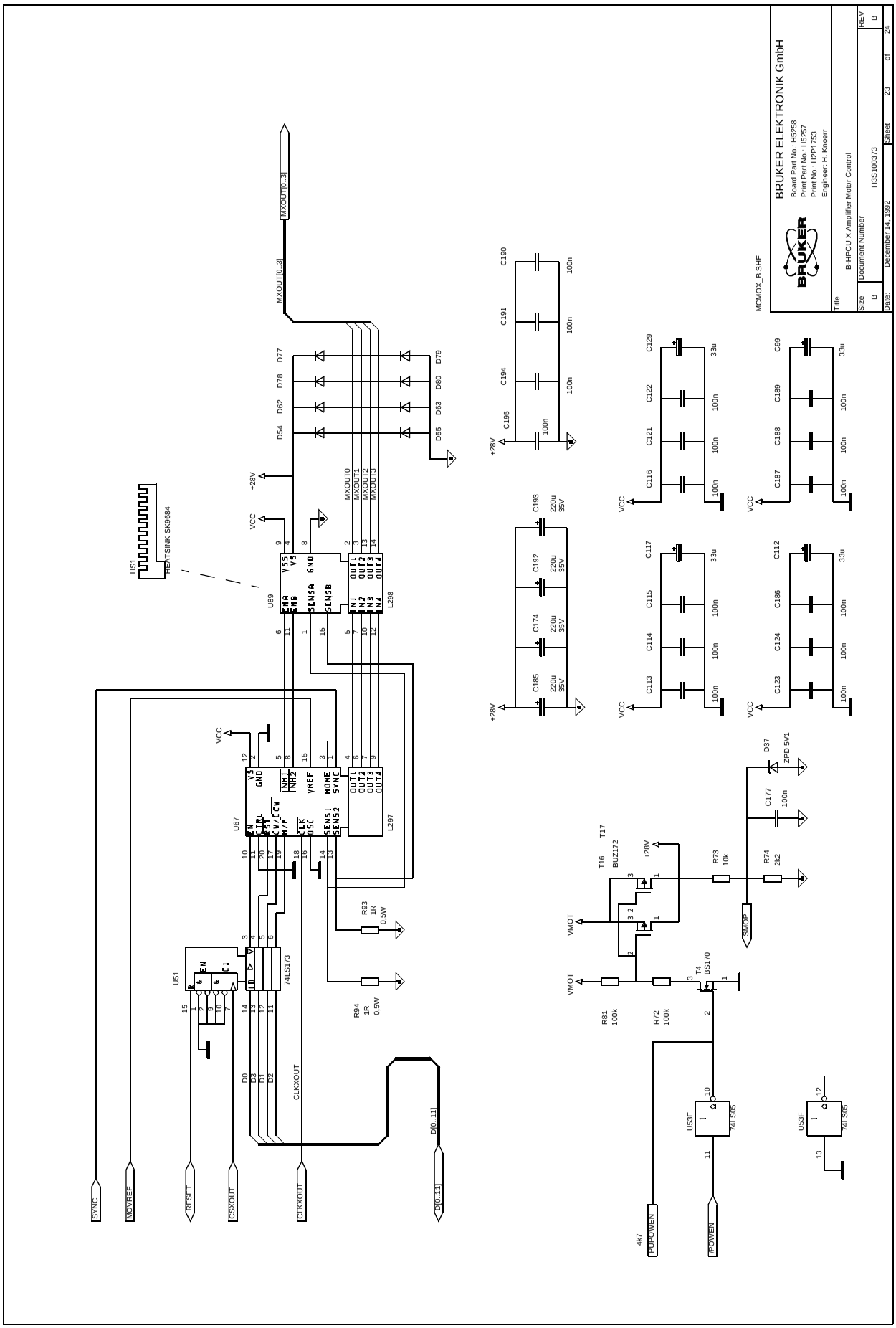
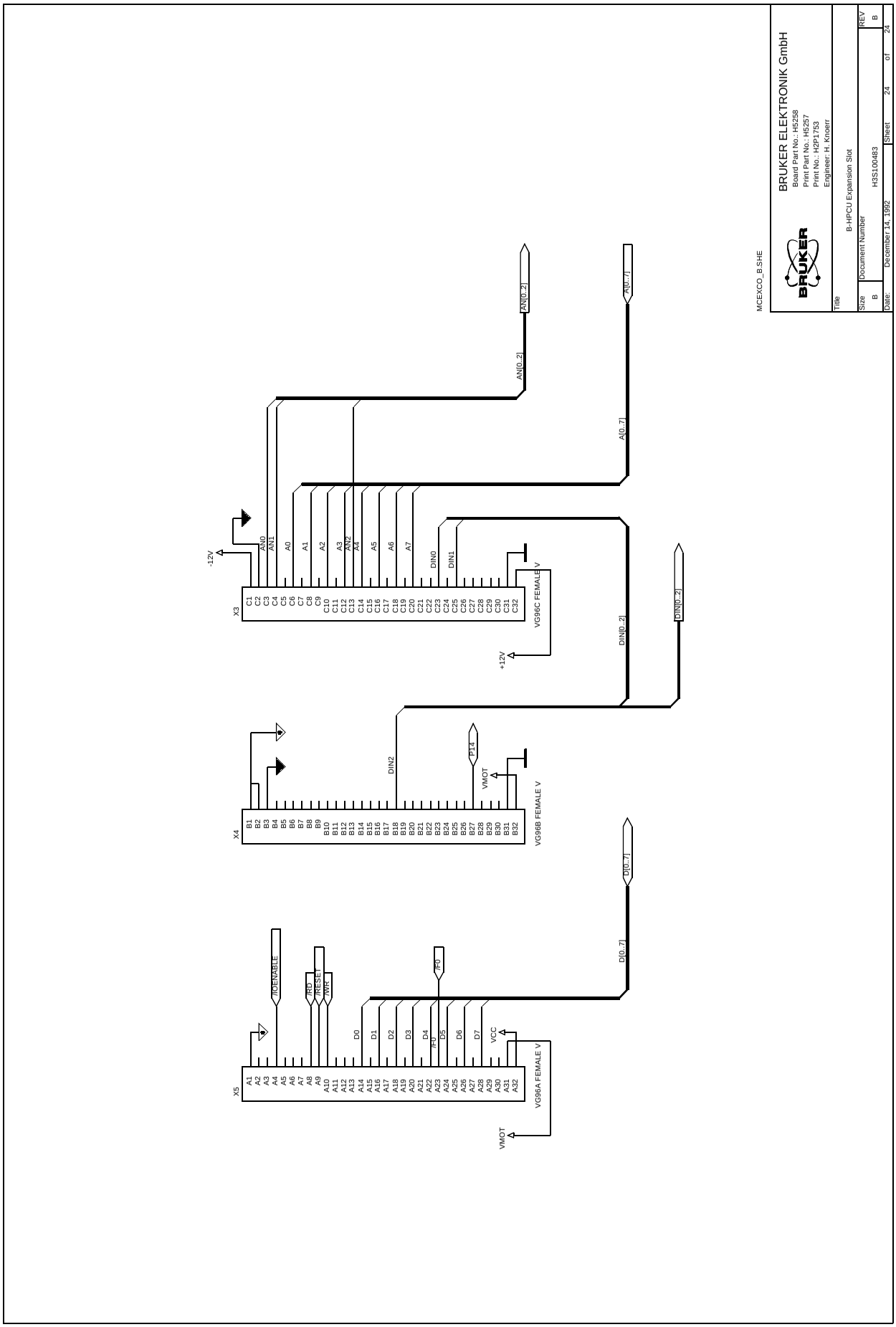


Figure 12.24.B-HPCU Expansion Slot



Figures

Figures

Tables

Tables