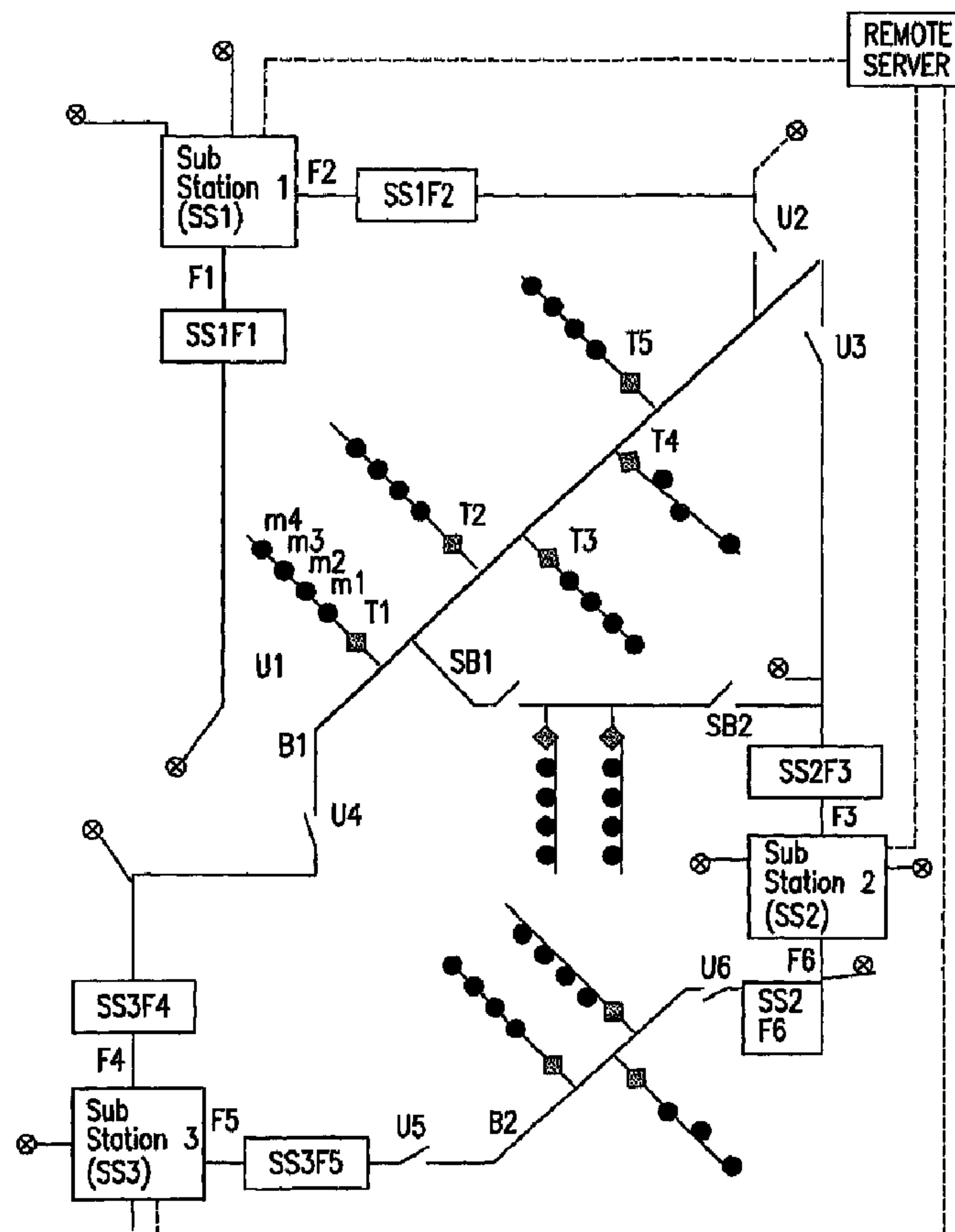




(86) Date de dépôt PCT/PCT Filing Date: 2006/11/22
(87) Date publication PCT/PCT Publication Date: 2007/05/31
(85) Entrée phase nationale/National Entry: 2008/05/22
(86) N° demande PCT/PCT Application No.: US 2006/045457
(87) N° publication PCT/PCT Publication No.: 2007/062232
(30) Priorités/Priorities: 2005/11/23 (US60/739,375);
2006/06/15 (US60/813,901)

(51) Cl.Int./Int.Cl. *G06F 11/00* (2006.01)
(71) Demandeur/Applicant:
QUADLOGIC CONTROLS CORPORATION, US
(72) Inventeurs/Inventors:
SWARZTRAUBER, SAYRE A., US;
MALIK, SIDDHARTH, US
(74) Agent: OSLER, HOSKIN & HARCOURT LLP

(54) Titre : SYSTEMES ET PROCEDES D'ELECTROMETRIE
(54) Title: SYSTEMS AND METHODS FOR ELECTRICITY METERING



(57) Abrégé/Abstract:

In one aspect, the invention comprises a system comprising: a master data clock source; one or more transponders; and a plurality of remote power line transceivers; wherein all of said plurality of transceivers are connected to a common alternating current power



(57) **Abrégé(suite)/Abstract(continued):**

distribution grid; and wherein each of said plurality of transceivers has a location is operable to monitor a voltage waveform of a power line prevailing at said location. In another aspect, the invention comprises a system comprising: transponders and remote power line transceivers each connected to a common alternating current power distribution grid each operable to monitor the voltage waveform of the power line prevailing at its own location, and generate selectable frequencies from said local power line waveform of a frequency of p/q times the frequency of said power line where p and q are positive integers greater than or equal to 1.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
31 May 2007 (31.05.2007)

PCT

(10) International Publication Number
WO 2007/062232 A2

(51) International Patent Classification:
G06F 11/00 (2006.01)

(21) International Application Number:
PCT/US2006/045457

(22) International Filing Date:
22 November 2006 (22.11.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/739,375 23 November 2005 (23.11.2005) US
60/813,901 15 June 2006 (15.06.2006) US

(71) Applicant (for all designated States except US): **QUAD-LOGIC CONTROLS CORPORATION** [US/US]; 520 Eighth Avenue, New York, NY 10019 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **SWARZ-TRAUBER, Sayre, A.** [US/US]; 8 Peter Cooper Road, New York, NY 10010 (US). **MALIK, Siddharth** [IN/US]; 316 East 83rd St., Apt. 3D, New York, NY 10028 (US).

(74) Agents: **UNDERWOOD, Steven, D.** et al.; Morgan Lewis & Bockius LLP, 101 Park Ave., New York, NY 10178 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

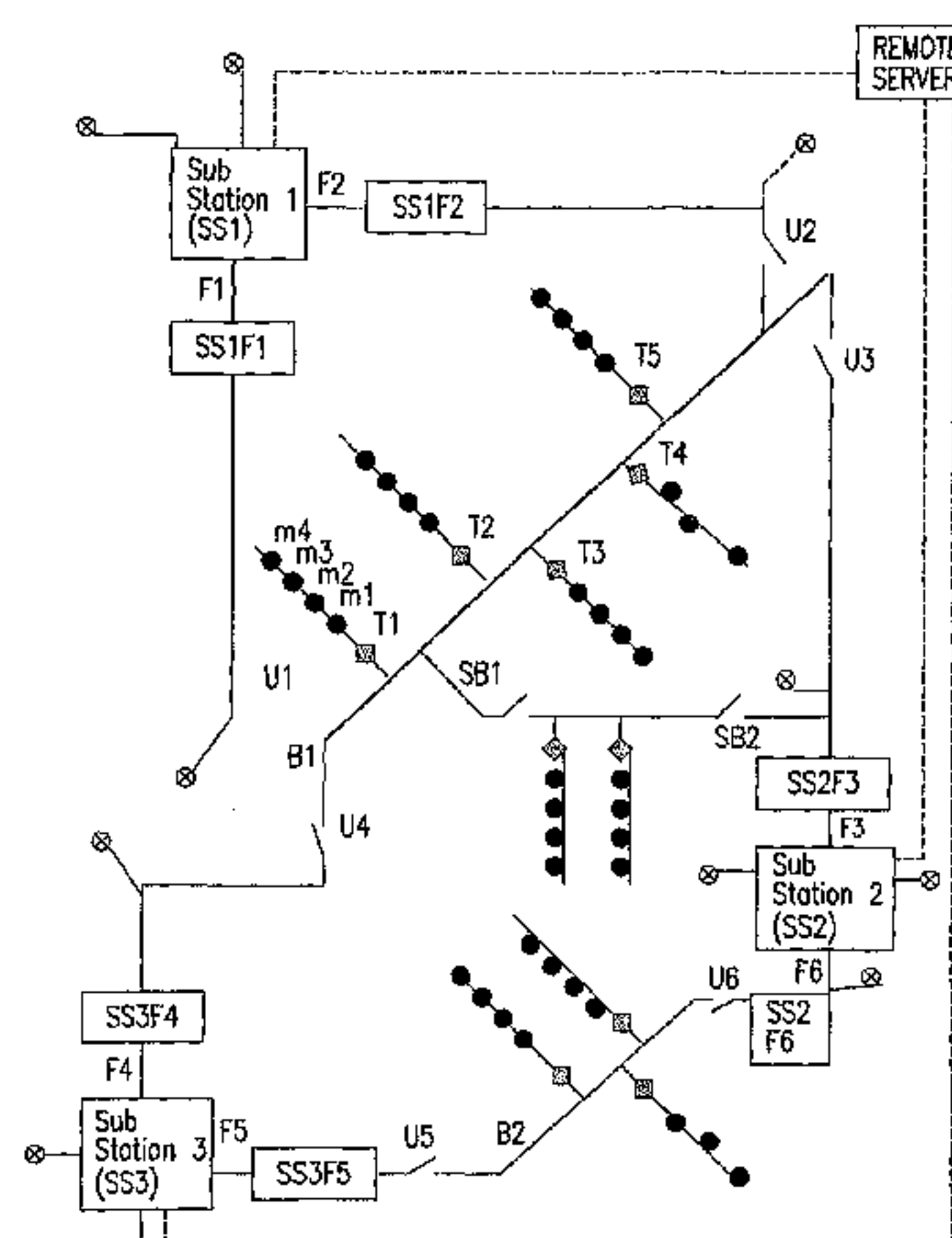
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEMS AND METHODS FOR ELECTRICITY METERING



(57) Abstract: In one aspect, the invention comprises a system comprising: a master data clock source; one or more transponders; and a plurality of remote power line transceivers; wherein all of said plurality of transceivers are connected to a common alternating current power distribution grid; and wherein each of said plurality of transceivers has a location is operable to monitor a voltage waveform of a power line prevailing at said location. In another aspect, the invention comprises a system comprising: transponders and remote power line transceivers each connected to a common alternating current power distribution grid each operable to monitor the voltage waveform of the power line prevailing at its own location, and generate selectable frequencies from said local power line waveform of a frequency of p/q times the frequency of said power line where p and q are positive integers greater than or equal to 1.

WO 2007/062232 A2

SYSTEMS AND METHODS FOR ELECTRICITY METERING

Cross-reference to Related Applications

This application claims the benefit of U.S. Provisional Patent Application No. 60/739,375, filed November 23, 2005, and U.S. Provisional Application No. 60/813,901, filed June 15, 2006. The entire contents of each of those applications are incorporated herein by reference.

Background

There are existing automated meter reading (AMR) power line carrier (PLC) systems that provide for PLC communication between a data concentrator at a substation and a meter installed down the power line in the low voltage service territory. However, most current systems have shortcomings, including single point access, limited capacity, low data rates, additional equipment to bypass the distribution transformer and, above all, lack of scalability. Very low data rates are required in order to provide overall communication reliability, which translates directly into a scalability limitation. For example, prior art systems have utilized transmit and receive frequencies as low as in the audio range in order to pass through distribution transformers. Some of these frequencies are integral multiples of the line frequency ($n \times f_{line}$, where n does not exceed 100), and others are simple fractions of the line frequency ($f_{line}/(2n)$, where $n > 1$). The prior art employing the latter technique allows an energy consumption signal to be superimposed on the power signal at a frequency lower than that of the power signal itself. This places a limitation on the data rates that the system can deliver. The limitation on scalability is primarily caused by the limited number of meters that be communicated with at one time and the manual programming required when changes are made to the service territory. Overall, the shortcomings of current systems include lack of reliability, flexibility, and scalability.

PLC systems make it possible to analyze network disturbances using electrical connectivity. Using PLC systems, the supply of electricity can be much more directly verified, as compared to systems that depend on wireless coverage. Various prior art PLC have used polling mechanisms to detect outages, while others have kept the meter and data collector continuously in communication. Also, there are prior art systems that report an outage event by a battery-backed up system that senses loss of power and

activates a modem that relays the power loss information. One disadvantage of such systems is that when many meters simultaneously lose power, the concurrent "last gasp" messages can create considerable collisions and noise.

5 SCADA-like systems use transceivers at substations and various infrastructure points (e.g., distribution transformers and substation feeders) to check the status of the power transmission network. These transceivers constantly monitor the operation of such instruments and relay information when a fault is encountered.

What are needed are AMR systems that require minimal manual intervention and are scalable as the number of installed meters increases, either due to mandatory
10 procedures in place or due to high energy costs and the need to eliminate unmetered services. As utilities strive to reduce operating costs, a system that is economically scalable and overcomes some or all of the above-mentioned problems is highly desirable. The scalability issue also implies that an automated system that the utility can install across the entire service territory (including multiple generating stations) or a subsection
15 thereof (including multiple substations), which provides a single-point control which provides data and status of installed meters, is needed. In addition, any technological progress that lowers the cost per metering point for a large system (e.g., more than 500 meters) by eliminating any additional equipment required at each transformer for PLC signaling is always welcomed by utilities.

20 It is a goal of this invention to present a two-way PLC AMR system that avoids the above-mentioned shortcomings of the prior art systems.

The current invention, in at least one embodiment, comprises a two-way communication system for reading interval metering data over medium tension distribution lines (4-33 kV), traversing distribution transformers to the metering devices
25 on low tension lines (120-600 volts), without requiring any special equipment at the distribution transformers, while maintaining a reliable and cost effective AMR solution.

The use of power lines for signaling, meter reading, load control, and other communication purposes has been well documented (see, for example, U.S. Pat. No. 6,947,854, to Swarztrauber, incorporated herein by reference). In a network installation
30 with a population of more than one meter, and a transponder accessing this population, the technology described by Swarztrauber presented a PLC communication system that included programming the meter to a specific channel (one of 16 in each of two bands

that cover 15-35 kHz). The transponder could remotely program the channel of each meter by utilizing a "base channel" that all meters could recognize, to direct each meter to its proper "resting" channel, isolated from the other channels by a sufficient frequency difference to allow simultaneous communications of each transponder to each meter.

5 However, as the system size grows, following the above procedure, each transponder requires at least two unique frequencies to avoid interference from other installed devices using RF communication over power lines. In addition, the system maintains a cross reference list at the transponder, listing the meters for which the transponder is responsible. In an environment with multiple transponders and multiple
10 polyphase devices, cross coupling of PLC signals can result in degradation of the overall throughput.

Summary

It is accordingly a general object of this invention to provide an improved method to manage the above-referenced scalability issues and to provide a considerable
15 improvement over the existing PLC methodology.

It is also an object of this invention to provide an improvement over the existing methods for performing PLC in a high line noise environment that results in a high signal to noise ratio (SNR) and to eliminate the need for two unique frequencies to avoid interference from devices using PLC communication or random noise.

20 It is another object of this invention to provide a device for receiving messages via powerline carrier using a microprocessor to decode either Frequency Shift Keying (FSK) or Phase Shift Keying (PSK) signals using a Fast Fourier Transform (FFT) algorithm.

It is another object of this invention to provide a method for obtaining reliable data and event information received from PLC communication with meters, making
25 logical deductions, performing statistical analysis about the service territory and providing an added service to the utility. This may include, but is not limited to, a comprehensive meter-territory map that the system dynamically and automatically updates as changes occur in the meter territory. The dynamic solution is uniquely determined by the ability of meters to decode PLC signals from multiple scan
30 transponders (STs) simultaneously.

Thus, in at least one embodiment, the invention provides an improvement over the prior-art technology to use FFT as the basis for simultaneous decoding of a plurality of transponder communications. For background purposes, the following are incorporated herein by reference in their entirety: U.S. Pat. App. No. 11/198,795, filed August 4,
5 2005, to Swarztrauber et al., and U.S. Pat. No. 6,947,854, discussed above.

- The above objects and features will be best understood from the detailed description below of certain embodiments, selected for purposes of illustration, along with the drawings. Those skilled in the art will realize that several implementations variations are possible without deviating from the scope of this invention.

10 A typical installation includes more than one ST located at each of remotely located substations feeding a section of utility service territory via medium tension lines terminating at distribution transformers from which low voltage lines emanate. Whereas meters are generally installed at customer premises, utilities may install a meter at the output of every distribution transformer, hence increasing the meter population in the
15 service territory. More than one meter typically is located in the low voltage service territory and communicates with its ST. All the STs in a system preferably are connected to a remote server that has a high speed data link in a LAN or WAN configuration and constantly communicates with all the STs. The remote server may itself operate on a clock that is derived from the utility line frequency. This can be implemented by using
20 RTC circuits that use the 60 Hz line frequency as a reference (such as Intersil CDP68HC68T1, a multifunctional CMOS real time clock). With a setup as above, all the STs are synchronously connected and operate using a network protocol (such as Network Time Protocol) so that they all share the same master clock dictated by the server thereby maintaining synchronicity by locking every ST to a common time source.

25 In one aspect, the current invention enables individual meters to receive, demodulate, and interpret simultaneous communications from all of the Transponders on all bands, communicating on different frequencies at once, eliminating the need for a "base channel" and for programming of a "resting channel." Each meter can listen to all of the STs and respond to the one that requests data from it. Moreover, each meter can
30 communicate information regarding the signal strength of each Transponder that it can hear to the one transponder that is requesting data. This enables moving meters to the "best" transponder for each meter.

The present invention, in at least one aspect, utilizes the installed PLC AMR infrastructure to provide an Event Management System (EMS) that provides a more extensive, practical, and efficient means for reporting events and tracking faults. The invention, in this aspect, thus helps utilities and metering entities to: (1) reduce the number of dispatches made in error based on verification algorithms; (2) automate the integration of an AMR infrastructure to provide a dynamically updated network map; (3) integrate power quality information; (4) use algorithms and back-end processing to proactively verify status of several parts of network; (5) include load profile information for energy forecasting; (6) perform preventive maintenance; (7) indicate status change of network switches, feeder changers, and reclosers; and (8) report such changes to a utility's central control center. For example, collecting network information about power quality may provide information on parts of a network territory with transients. One embodiment provides a Dynamic Mapping Mode of PLC AMR system operation that selects meters (either randomly or based on strategically predetermined criteria) and initiates probing.

In one aspect, the invention comprises a system comprising: a master data clock source; one or more transponders; and a plurality of remote power line transceivers; wherein all of said plurality of transceivers are connected to a common alternating current power distribution grid; and wherein each of said plurality of transceivers has a location is operable to monitor a voltage waveform of a power line prevailing at said location.

In various embodiments: (1) the system is operable to generate a local data clock from said local power line waveform of a frequency of p/q times the frequency of said power line where p and q are positive integers greater than or equal to 1; (2) the master data clock source operable to transmit information regarding the phase and frequency its own local clock to said transponders; the local data clock of the master data clock source being called the master data clock; (3) said transponders and said remote transceivers each operable to inject and receive signals on the power line; (4) said transponder is operable to (a) reconstruct the master data clock from the phase and frequency information received from the master data clock source and its own local data clock; and (b) utilize the reconstructed master data clock to align data bits injected onto the power line; (5) said remote power line transceiver is operable to: (a) receive signals from at least one, but not necessarily all, of the transponders; and/or (b) measure the difference in phase of the local data clock and the master clock by monitoring the signals transmitted

from any one or more the transponders; (6) said master data clock source is also a transponder; (7) said remote power line transceiver is capable of storing said phase difference between its local clock and the master data clock so as to be able to create a copy of the master data clock from its own local clock without having to continually
5 reconstruct the data clock by monitoring the received signals; (8) said remote transceivers are also electricity meters; (9) said remote transceivers can receive and interpret signals from more than one transponder simultaneously; (10) each of said transponders is operable to request and receive data to the said meter(s) via power line communications and transmit said data to a remotely located computer; (11) said remote power line
10 transceiver is operable to: (a) measure the phase of the one or more voltage waveforms present at its location; and/or (b) report the phase information of said waveforms to the transponder; and (12) the waveform phase information is correlated with the metering information to allow the voltages, currents and power quantities to be added using vector additions to aggregate such quantities at key points in the power distribution grid.

15 In another aspect, the invention comprises a system comprising: one or more transponders and a plurality of remote power line transceivers each connected to a common alternating current power distribution grid each operable to monitor the voltage waveform of the power line prevailing at its own location, and generate selectable frequencies from said local power line waveform of a frequency of p/q times the
20 frequency of said power line where p and q are positive integers greater than or equal to 1.

In various embodiments: (1) said transponders and said remote transceivers are each operable to inject and receive signals on the power line; (2) said signals each have a frequency of p/q times the line frequency where p and q are selectable from the set of
25 whole integers; (3) said transponders and said remote transceivers alternate among different frequencies by changing the factor p or invert the phase of a fixed frequency so as to effect FSK or PSK modulation; (4) the frames of the data bits are uniform across the population of transponders and remote transceivers and correspond to the period and phase of the master data clock; (5) binary FSK modulation is used by selecting two values
30 of p , p_1 and p_2 for the frequencies of the ones and zeros; (6) the receiver of either a transponder or a remote transceiver: (a) utilizes FFT or DFT algorithms calculated successively over the sequential data bit frames; and/or (b) demodulates the data bit at

during each data frame by comparing the amplitudes of the signals corresponding to p1 and p2 over the course of each data bit frame.

In another aspect, the invention comprises an apparatus to implement a PLL comprising a input signal source, a VCO, a microprocessor, a DAC, an ADC wherein the VCO is used to drive the clock of the microprocessor; the microprocessor controls the sampling time of the ADC at times determined by its system clock; the ADC monitors the input signal source; the microprocessor reads the ADC; the microprocessor performs some filtering calculations on the signal from the ADC; the microprocessor controls the output of the DAC based upon the said calculations; and the DAC controls the input of the VCO so as to close a PLL around all of the aforementioned elements.

In various embodiments: (1) the input signal is a conditioned copy of the waveform of the A/C power line; and (2) the DAC is a pulse width modulator followed by a low pass filter.

In another aspect, a remotely located computer is operable to identify changes in operation or connectivity of electricity distribution network components. In various embodiments: (1) said components comprise one or more of: meters, transformers, transponders, switches, and feeders; (2) said remotely located computer is operable to distinguish meter changes from transformer changes; (3) said changes comprise outages; (4) said remotely located computer is operable to calculate current output at each of a plurality of transformers; and (5) said remotely located computer is operable to calculate current output at each of said plurality of transformers based on a vector sum of signals on each phase.

Brief Description of the Drawings

FIG. 1 is a diagram of an exemplary installation of a preferred system; substations are indicated as SS_p ; distribution transformers as T_p ; switches as U_p ; feeders as F_q ; and meters as m_p . Transponders are indicated as $ST_p F_{q_i}$, where p and q are integers and $i = 1, 2, \text{ or } 3$ for a 3-phase feeder line.

FIG. 2 is a block diagram of a preferred automatic tuning module.

FIG. 3 depicts a preferred substation installation, indicating equipment on each phase of the feeder in a substation.

FIG 4 depicts preferred FIR specifications for 10-25 kHz.

FIG 5 depicts preferred FIR specifications for 25-50 kHz.

FIG. 6 depicts preferred FIR specifications for 70-90 kHz.

FIG. 7 illustrates line noise spectra for 10-100 kHz.

FIG. 8 illustrates injecting PLC signals at half-odd harmonics of 60 Hz.

FIG. 9 depicts the 12 possibilities in which an FFT frame received by the meter
5 can be out of phase with an ST FFT frame. Dotted lines correspond to a 30 degree rotation to account for a delta transformer in the signal path between the ST and the meter.

FIG. 10 depicts server determined time = 0 reference and 30 Hz cycles of various meters.

10 FIG. 11 illustrates SNR degradation effects of FSK decoding by meter when the data frames are aligned and not aligned.

FIG. 12 depicts distribution of SNR as meter M1 tries to align its data frames to incoming ST's data frames.

15 FIG. 13(a) depicts zeros of a Sinc function; 13(b) depicts overlapping zeros of multiple Sinc functions when meter data frames are aligned with ST data frames.

FIG. 14 is a block diagram of a preferred analog front-end for metering.

FIG. 15 depicts preferred FIR specs for decimating metering data.

FIG. 16 depicts FFT frames for voltage indicating the harmonics

FIG. 17 depicts an exemplary directory structure of a system map.

20 FIG. 18 is a flowchart of an example of logical analysis on received PLC data.

FIG. 19 is a block diagram of a preferred D meter (this is one of at least two versions of a D meter).

FIG. 20 depicts schematics for a preferred board for implementing the FFT embodiments.

25 FIG. 21 has preferred schematics for a power board.

FIG. 22 has preferred schematics for an I/O extension board.

FIG. 23 has preferred schematics for a CPU board (PCB 202).

FIG. 24 has preferred schematics for metering, power supply, and PLC transmit and receive circuitry for a residential meter (PCB 240).

FIG. 25 has preferred schematics for a display board (PCB 220).

FIG. 26 illustrates a microprocessor being part of a phase locked loop.

5

Detailed Discussion of Certain Embodiments

One preferred method enabling simultaneous multiple meter-ST communication is discussed with respect to an SCH161 implementation of the device. See FIG. 20.

Table 1: Key to FIG. 20

FIG. 20A	Hierarchical interconnections
FIG. 20B	SDRAM memory
FIG. 20C	MCF5271 CPU
FIG. 20D	Debug
FIG. 20E	Ethernet interface
FIG. 20F	Maxim chip
FIG. 20G	Flash memory
FIG. 20H	Power supply unit
FIG. 20I	Reset configuration and clocking circuitry
FIG. 20J	Serial I/O interfaces
FIG. 20K	Meter-V
FIG. 20L	PLC

Notes regarding FIG. 20: (1) all decoupling caps less than or equal to 0.1 μ F are COG SMD unless otherwise stated; (2) all decoupling caps greater than 0.1 μ F are X7R SMD 0805 unless otherwise stated; (3) all connectors are denoted Jx; (4) all jumpers are denoted JPx; (5) all switches are denoted SWx; and (6) all test points are denoted TPx.

It is more convenient to use a 24.576 MHz crystal for deriving the PLC frequencies. Specific to at least one embodiment are:

15

- A Phase Locked Loop (PLL) to lock the sampling of the signal streams to a multiple of the incoming AC line (synchronous sampling to the power line frequency).
- A Voltage Controlled Oscillator (VCO) at 90-100MHz controlled by digital signal processor (DSP) via two Pulse Width Modulators (PWMs)

modules directly driving the system clock, hence making the DSP coherent with the PLL. See FIG. 26.

- A synchronous phase detector that responds only to the fundamental of the incoming line frequency wave and not to its harmonics.
- An option for performing Frequency Shift Keying (FSK) and Phase Shift Keying (PSK) modulation schemes

FAST FOURIER TRANSFORM (FFT) USED IN CERTAIN EMBODIMENTS

In at least one aspect of the invention:

(1) The transponders use frequencies which are multiples of 60 Hz in the range of 15-35 kHz. For FSK, the Transponders preferably use two adjacent frequencies, for PSK, they preferably use just one frequency. The STs must have accurate system clocks from which they generate the carrier frequencies - especially in the case of PSK. By sharing one common clock with 1 ppm accuracy using a device such as the Maxim DS4000 TCXO, these conditions are easily met.

(2) A bank of transponders derives a data clock by synchronizing to a particular phase (e.g., the "A" phase of a trunk line with phases A, B, and C). All STs (even the ones in different banks) can utilize the same data clock to separate the bits of the FSK or PSK transmission.

(3) The Meters receive the data, pass it through an anti-aliasing filter and sample it:

(a) A MAX1308 ADC is controlled by an MCF5271 microprocessor to sample data at a rate of 60×2048 or 122880 Hz. (Other channels of the MAX1308 or MAX1320 are used for reading voltage and current for accumulating the metering data that will be transmitted to the Transponders. The metering data is sampled simultaneously with the powerline communications data).

(b) The MAX1308 uses two JK flip flops to control the DMA channel of the MCF5271 to put the sample data directly into the memory of the Coldfire.

(c) The Coldfire receives two frames of data (1/60 of a second, each containing 2048 points) and uses one frame for the real part of 2048 complex points and

the second frame for the imaginary part of 2048 points. The data frames must be synchronized to the 60 Hz line as well.

Because many meters are not on the A phase of the 60 Hz line, they must hunt for the correct clock frequency. One exemplary method of hunting for valid preambles
5 comprises dividing the 60 Hz line into 8 phases and trying each of the 8 phases until the correct phase is found. In one embodiment of the present invention, this method is only employed once by the meter until it determines the correct phase of the 60 Hz line, because once connected the meter will never change phase. The present invention, in at least one embodiment, divides the line frequency into more than 12 parts, to allow for a
10 minimum of 30 degree resolution in the line frequency. This allows for the possible phase shifts that may occur in distribution transformers.

(d) The ColdFire then does a 2048 point complex FFT (which takes about 9.8 msec every 33 msec for about 30% of the CPU computing bandwidth). The complex 2048 fft is then decomposed into two real-2048 bit ffts by well known methods of adding
15 and subtracting positive and negative mirror frequencies for the real and imaginary parts, respectively. Thus, every FFT yields two bits of data every 33 msec.

(e) The Coldfire then analyzes the data looking for valid preambles from as many Transponders as it can see. The preamble is a 32 bit number that is known and shared between the Transponders and the meters. It is a code that defines the beginning
20 of the message. The FSK analysis preferably is performed by comparing the amplitudes of the adjacent bins.

(f) To use PSK requires another step. The preferred algorithm is to collect the complex phase information from the single bins into a buffer that is sufficiently large to hold an entire preamble (e.g., a 32 bit preamble). The crystal clock of the meter has an
25 accuracy of 30 ppm. Therefore, over a 32 bit preamble the phase error is 180 degrees. This requires a first order linear correction factor. While scanning for 32 bit preambles, the algorithm checks for phase inversions in adjacent bits. But there is a phase rotation that must be corrected, and an unknown starting phase. The system preferably tries to find the rotation correction factor that is due to the error of its own crystal factor by trial
30 and error, rescanning frames of 32 bits against 32 possible rotation correction factors that will get the correction factor to within 1 ppm, an acceptable error. Once the error is found, the drift is very slow and the meter can keep a record of the error of its own crystal

relative to the known good frequency of the bank of transponders. To get the constant error, the PSK algorithm subtracts a constant phase from each point in the 32 bit preamble window. If no preambles are found in the 32 bit window, the algorithm waits for the next two bits from the FFT, eliminates the oldest two bits and brings in the newest two bits and
5 repeats the scan to determine the phase and frequency error between the Transponder and the meter itself. After the successful determination of the error frequency, later scanning for frames needs to look only in a small window of rotation correction factors around the known error. This allows for continuous monitoring of the frequency error with less processing power. A similar technique of locking to the 60 Hz line using phase error
10 information is disclosed in baudpll.c (included in the Appendix below).

TRAVERSING CARRIER FREQUENCIES THROUGH DISTRIBUTION TRANSFORMERS

As discussed, the prior art suffers from a disadvantage of not being able to pass high frequency signals (starting in the kHz range) through existing distribution
15 transformers without using any additional equipment at the transformer. In other approaches, the transformer is bypassed using expensive additional equipment, thereby increasing overall system cost.

One embodiment comprises an arrangement for making the PLC signal go through the Distribution Transformers (DTs). It is well-established that the magnetic
20 field in the DTs and noise on the line present far from ideal conditions for the PLC signal to propagate to the meters. Solving this problem preferably involves, in one embodiment, a two step process:

1. Signal Coupling: a strategically designed coupler couples the radio-frequency signal to either underground or overhead Medium Tension (MT)
25 electrical distribution cables.
2. Coupler Tuning: the signal coupler is automatically tuned to the highest efficiency to maximize the Signal to Noise Ratio (SNR) as the current on the MT line varies.

Preferably, the coupler introduces a small inductance in the MT line, which then is
30 tuned for a given carrier frequency by a bank of capacitors, thus providing a high SNR for communication. The signal tuning preferably utilizes a tank circuit that automatically maximizes the impedance match of PLC signals on the line by mounting a coupler at the

point where the trunk begins. No additional installation is required near the transformer. This has the effect of maximizing the signal on the line as the low impedance of the trunk line provides a return path for the current. The coupler, which preferably comprises a ferrite core with calculated wire turns wound on it, provides a fixed inductance for the PLC signal. The capacitance for the tank circuit is provided by a Capacitor Relay bank (CRB). An Automatic Tuning Module (ATM) comprises circuitry to control the capacitors and relays in the CRB.

A simplified diagram of the ATM is given in FIG. 2, where CV is Communication Voltage and CN is Communication Neutral.

To determine the data for tuning performance, the ATM calculates the ratio of PEAK1/PEAK2 for all possible values ($2^{10} = 1024$ in this embodiment with CRB with 10 capacitors) of the capacitor combinations given a fixed inductance (taking into account inductance drifts due to temperature, etc.) and stores the variables or settings of the best ratio achieved. All further determinations are done relative to this ratio. A typical operation involves the following steps: choose capacitance value, send signal to relay, wait for relay operation, wait for relay settling, calculate the ratio, compare with other ratios and send signal to disconnect relay and wait for relay operation to settle, store the result in memory, and repeat the process with other capacitance values.

In an alternate embodiment, various improvements can be made to the above process. As an example, another embodiment combines ATM and CRB units into a single Automatic Tuning Unit (ATU). The improvements include, but are not limited to:

1. Finer tuning resolution by increased windings on the coupler up to 24 turns and increased capacitance choices up to 4096.
2. Replacing the continuous tone provided by the ST by an on-board signal generator.
3. Calculating PEAK1/PEAK2 ratio (P1/P2 Ratio) as a complex number, thereby detecting both amplitude and phase for the ratio. This improvement provides a better sense of the choice of inductance and capacitance for the resonant circuit, thereby reducing randomness in choosing capacitance values. By determining the phase in addition to the amplitude, lead/lag behavior and consequently an optimal choice of L and C is determined much faster. This in turn results in minimizing relay operation and increasing relay life.

4. Providing ATU Transmit power levels compatible with up to 20W of PLC transmit power in a frequency band from 10-110 kHz.

5. Ability to tune the coupler to an impedance of at least 120 Ohms at resonance.

5 Tuning Operation

Referring to FIG. 2, during the tuning operation, Relays M, 1, and 2 are closed, whereas Relay R is open. As a result, the 50 Ohm resistor is selected in the series path of transponder and coupler. This is done to avoid damage to the ST transmitter so that if for some reason the impedance of the coupler is infinitely small, the signal still sees a load of at least 50 Ohm to perform the tuning. Relay M selects the coupler and the tuning process is initiated. Preferred steps comprise:

1. ST indicates to ATM/CRB that tuning can be initiated.
2. ATM/CRB initiates a request for ST to send out continuous tones of communication signal.
- 15 3. The ratio PEAK1/PEAK2 is calculated. This ratio corresponds to a DC voltage sensed by the ATM.
4. Responding to this voltage level, ATM calculates the optimum value of capacitance required for resonance and sends a signal to CRB.
5. The appropriate capacitance is selected in CRB, achieved by opening and closing of relays.
- 20 6. The ratio is calculated again with the new capacitance.
7. The process repeats for multiple values of capacitance, and when the ratio is as high as possible, the settings of capacitance and inductance are stored.
- 25 8. This information is conveyed to ST, and concludes the tuning process.

Normal Communication Operation

After tuning, the normal PLC communication operation proceeds: Relays M and R are closed, and Relays 1 and 2 are open.

All of the improvements mentioned above result in improved tuning efficiency and accuracy while maximizing system life by reducing unwanted relay operations.

Such a coupling set up is further discussed in connection with FIG. 2 and in U.S. Pat. App. No. 11/198,795, mentioned above.

5 Owing to cross-links provided by polyphase devices, the PLC signal injected on a particular phase of a feeder in a substation can couple with other phases of either the same or different feeders of other substations. It becomes important to ensure the appropriate return PLC signal path. To this end, a Bypass Capacitor preferably is installed on each phase across the neutral on the main medium tension bus in the substation as shown in
10 FIG 3. This installation not only ensures that the return path of the PLC signal is the same feeder, but also that the majority of injected PLC signal flows towards the load.

USING FFT FOR PERFORMING PLC COMMUNICATION

There are three distinct bands that embodiments of the current invention may use for PLC communication: (1) 10-25 kHz for communication through distribution
15 transformers; (2) 25-50 kHz for low voltage communication; and (3) 70k-95 kHz for performing Medium Tension (MT) coupler-to-coupler communication in cases when a plurality of couplers are installed on the same medium tension power line.

A unique feature of these embodiments is that the transponders use communication frequencies in the kHz range that are rational multiples of the line
20 frequency (that is, of the form $(p/q) \times f_{line}$, where p and q are positive integers). The PLC signal is sampled at about 240 kHz ($2^{12} \times 60$). Depending upon the selection of one of the above frequency bands of operation, the appropriate Finite Impulse Response (FIR) filter is applied to decimate the data. The FIR specifications are given in FIGS. 4 and 5.

Those skilled in the art will recognize the need to make modifications to the
25 current implementation discussed to incorporate the use of 70-90 kHz frequency band owing to the front-end anti-aliasing filter specifications in this embodiment. Embodiments of the current invention use this frequency range to enable communication between multiple scan transponders on medium tension lines for long distances. The FIR specifications are given in FIG. 6.

30 Depending upon the selection of the appropriate FIR filter, the decimation is done to either 120 kHz ($2^{11} \times 60$) or 60 kHz ($2^{11} \times 30$), in the case of communicating through

transformers. A 2048 point FFT is then performed on the decimated data. The data rate is thus determined to be either 60 baud or 30 baud depending on the choice of FIR filters. Every FFT yields two bits approximately every 66 msec when traversing through distribution transformers.

5 This unique ability of both transponders and meters to perform FFT allows the meters to receive, demodulate, and interpret simultaneous communications from all of the transponders on all of the bands at once, eliminating the need for base and resting channels. Each meter can thus listen to all of the transponders and respond to the one that requests data from it. In addition, each meter can communicate information regarding the
10 signal strength of each transponder that it can hear to the one to which it responds for data requests.

PLC COMMUNICATION IN LINE NOISE ENVIRONMENT

A distinction is made between PLC communication over medium tension (4-35 kV) and low voltage (LV) (< 600V) lines as both power transmitting mediums present a
15 different environment to PLC signals. Whereas medium tension presents its own challenges, it is a quieter environment for PLC communications than LV presenting well-characterized corona discharge noise. Embodiments of this invention overcome the historical challenge of performing PLC communication in a high line-noise environment.

Shown in FIG. 7 is a snapshot of averaged low voltage noise spectrum in 60 Hz
20 power lines from 0-100 kHz. Whereas the noise levels are sufficiently low at the higher end of the frequency range, at 10-25kHz the noise rises faster than the signal. At least one embodiment of the invention comprises a method to solve this problem by injecting PLC signals at half odd harmonics of line frequency. This is shown in FIG. 8.

When traversing through transformers, since FFT is done every 30 Hz and the
25 harmonics are separated by 60 Hz, the data bits reside in the bin corresponding to the 201.5th and 202.5th harmonic of 60 Hz as shown in FIG. 8. When using an FSK scheme, the preferred algorithm considers these two bins of frequencies and compares the amplitude of the signal in the two to determine 1 or 0. This FSK scheme uses two frequencies and yields a data rate of 30 baud. It will be apparent to the skilled in the art
30 that other schemes such as QFSK can be implemented to yield 60 baud.

The significant advantage of communicating at these frequencies is that it results in improvements in SNR of more than 40 dB. Similar results are obtained across other frequency ranges where the noise floor is ~80 dB below the harmonics.

5 REMOVING PHASE AMBIGUITY IN PLC COMMUNICATION IN A POLYPHASE ENVIRONMENT

The transponders communicate by allocating time windows for each meter. In most applications, the time window is one line-cycle wide. However, as mentioned, when communicating through distribution transformers, the time slot can be two line-cycles wide, as shown in FIG 10. Given the ability of simultaneous communication of multiple meters and transponders, each meter performs a shift in its internal clock to align its data frames with the incoming data frames from the transponders. This preferably is achieved by:

1. Establishing a $t = 0$ reference: In order to establish data-frame alignment between an ST and the meter, a zero time reference for communication is required. This is provided by the remote server that is itself locked to a particular phase (say, the A phase). This can be implemented by using Real Time Clock (RTC) circuits that use the 60 Hz line frequency as the time reference (such as Intersil CDP68HC68T1, a multifunctional CMOS real time clock). This time reference is communicated from the server to all the STs via a high speed network.

2. Aligning meter data frames with multiple transponders: When a meter is powered up, it listens to multiple STs in the territory. However, the meters are themselves on different phases, and each data frame received by the meter can undergo various phase variations due to the line topology. The probability of error increases as frames are more and more misaligned, reducing the overall SNR and the ability to differentiate between 1 and 0. As the meter tries to align its data frames with various STs that it can listen to, it shifts its data frames and calculates the SNR for every possible combination (24 for 30 Hz data frames, and 12 for 60 Hz data frames). Further, it locks to the ST which results in maximum SNR.

When traversing through transformers, both STs and meters perform FFT on the PLC and data signals every 30 Hz in the 10-25 kHz range. Because the PLLs

implemented in both the ST and the meter are locked to the line, the data frames are synchronized to the 60 Hz line as well. However the data frames can shift in *phase* due to:

- 5 (a) Various transformer configurations that can exist in the path between the ST and the meter (delta-Wye, etc.).
- (b) Shifts in phase due to the fact that STs are locked on a particular phase, whereas single and polyphase meters can be powered up by other phases.

The SNR ratio is maximized when the meter data frame and ST data frames are most closely aligned. From a meter's standpoint, this requires receiving PLC signals from all possible STs that it can "hear," decoding the signal, checking the SNR ratio by aligning data frames and then responding to the ST yielding maximum SNR. FIG 9 shows the 12 possible ways in which the data frames can be off in phase. In addition, because the data frames are available every 30 Hz on a 60 Hz line, there are two possibilities corresponding to the 2 possible phases obtained by dividing 60Hz by 2. Hence there are 24 ways that meter data frames can be misaligned with ST data frames.

The significant advantage offered by locking the data frames to the line frequency is explained thus: there are whole number of carrier cycles in each data frame. Keeping this in mind, and recalling that the Fourier transform of a rectangular function yields a Sinc function (see FIG. 13(a)), when several meters are communicating simultaneously, with each having shifted its internal data clock cycle to align with incoming data frames, the nulls of the Sinc function overlap and no smearing of data bits and no SNR degradation occurs (see FIG. 13(b)). This yields a very high SNR even when multiple meters communicate simultaneously. This is precisely the reason that the system goes through a communication alignment mode wherein: (1) in one embodiment, a remote server may assign the global clock (which maybe derived from the line frequency) to all STs; (2) meters receive data simultaneously from multiple STs; (3) meters determine the shift in their data clocks to align data frames with multiple STs; and (4) meters lock to the ST that results in highest SNR.

FFT preferably is performed every 30 Hz or 2 cycles of line frequency of 60 Hz in the 10-25 kHz frequency band. In each frame of the ST, there are an odd integral number of cycles of the carrier frequency. The preferred modulation scheme being Frequency Shift Keying (FSK), if there are n cycles for transmitting bit 1, bit 0 is transmitted using

$n+2$ cycles of the carrier frequency. It becomes important for the meter to recognize its own 2 cycles of 60 Hz in order to be able to decode its data bits which are available every $1/30^{\text{th}}$ of a second (FIG. 10). In FIG. 11, two STs (ST1 and ST2) that transmit bits are intercepted by a meter (M1) for the case when data frames are aligned (Case I) and when they are misaligned by different degrees (Cases II and III). Both STs use different frequencies to communicate. FSK is used to decode the signal for bits.

In cases II and III, M1 decodes signals with misaligned data frames; hence, there is energy that spills over in the adjacent (half-odd separated) frequencies. If the signal level that falls in the "adjacent" frequency bin is less than the noise floor, the signal can be decoded correctly. However, if the spill-over is more than the noise floor (as with Case III), the ability to distinguish between 1 and 0 decreases, and hence the overall SNR drops, resulting in an error in decoding. Thus,

a. If the frames are misaligned, smearing of data bits occurs and the SNR degrades.

b. In the event that the frequency changes and there are misaligned data frames, there is a substantial amount of energy that spills over in the adjacent FFT bins, interfering with the other STs in the system that communicate using frequencies in those specific bins.

In FIG. 11, $\text{SNR}_1 > \text{SNR}_2 > \text{SNR}_3$. The SNR distribution is expected to look like a modified normal distribution, with one of the STs with which the meter data frames are aligned resulting in the max SNR. The meter then locks to this ST for further communication (FIG. 12). The meter locks until a significant change in SNR ratio is encountered by the meter, in which case the process repeats.

The above technique provides a substantial improvement over the existing art of performing PLC through distribution transformers without bypassing these transformers while maintaining robust and reliable communication resulting in high throughput.

ANALOG SIGNAL CHAIN IN PREFERRED EMBODIMENT

This section discusses the PCB 202 block diagram (see FIG. 14) and the blocks can be cross referenced with the schematic shown in FIG.23. Each metering and communication channel preferably comprises front-end analog circuitry followed by the signal processing.

The current embodiment uses an anti-aliasing filter with fixed gain which provides first-order temperature tracking, hence eliminating the need to recalibrate meters when temperature drifts are encountered. The analog front-end for voltage (current) channels preferably comprises voltage (current) sensing elements and a programmable attenuator followed by an anti-aliasing filter. The attenuator reduces the incoming signal level so that no clipping occurs after the anti-aliasing filter. The constant gain anti-aliasing filter restores the signal to full value at the input of the ADC. For metering, the anti-aliasing filter cuts off frequencies above 5 kHz. The inputs are then fed into the ADC, which is part of the DSP.

Whereas it is a common practice in current art to include a Programmable Gain Amplifier (PGA) followed by a low gain anti-aliasing filter, the advantage offered by the use in the described embodiment of a programmable attenuator followed by a large fixed-gain filter will be apparent to those skilled in the art. In addition, the implementation of both the anti-aliasing filters on a single chip is exactly identical using the same Quad Op Amps along with 25 ppm resistors and NPO/COG capacitors. This provides a means for both V and I channels to track temperature drifts up to first order without recalibrating the meter.

In contrast, using a PGA along with a low gain filter will not permit tracking of the phase shift in the V and I signals introduced due to temperature. This is due to the fact that the phase shift introduced by PGA is a function of the gain.

This unique implementation that includes pairing the anti-aliasing filters ensures that the phase drifts encountered in both voltage and current channels are exactly identical and hence accuracy of the power calculation (given by the product of V and I) is not compromised.

DIGITAL SIGNAL CHAIN IN PREFERRED EMBODIMENTS

At least one embodiment preferably uses a Phase Lock Loop (PLL) to lock the sampling of the signal streams to a multiple of the incoming AC line frequency. In an embodiment discussed above, the sampling is at a rate asynchronous to the power line. In the meter circuit represented by FIG. 23 (also referred to herein as the "D meter") there is a Voltage Controlled Oscillator (VCO) at 90-100 MHz which is controlled by the Digital Signal Processing (DSP) engine via two Pulse Width Modulators (PWMs). The VCO directly drives the system clock of the DSP chip (disabling the internal PLL), so the DSP

becomes an integral part of the PLL. Locking the system clock of the DSP to the power line facilitates the alignment of the sampling to the waveform of the power line. The phase detector should function so as to respond only to the fundamental of the incoming 60 Hz wave and not to its harmonics. FIG. 19 is a block diagram of this preferred DSP implementation.

A DSP BIOS or voluntary context switching code provides three stacks, each for background, PLC communications and serial communications. The small micro communicates with the DSP using an I²C driver. The MSP430F2002 integrated circuit measures the power supplies, tamper port, temperature and battery voltage. The tasks of the MSP430F2002 include:

- i. maintain an RTC;
- ii. measure the battery voltage;
- iii. measure the temperature;
- iv. measure the +U power supply;
- 15 v. reset the DSP on brown out;
- vi. provide an additional watchdog circuit; and
- vii. provide a 1-second reference to go into the DSP for a time reference to measure against the system clock from the VCO.

IMPLEMENTATION OF METERING IN PREFERRED EMBODIMENT

Each data stream has an associated circuit to effect analog amplification and anti-aliasing.

Each of the analog front end sections has a programmable attenuator that is controlled by the higher level code. The data stream is sampled at 60 kHz ($2^{10} * 60$) and then a FIR filter is applied to decimate the data stream to ~15 kHz ($2^8 * 60$). The filter specifications are shown in FIG. 15.

Since only the data up to 3 kHz is of interest, a 3-12 kHz rolloff on the decimating FIR is used with ~15 kHz sample rate. The frequencies from 0-3 or 12-15 kHz are mapped into 0-3 kHz. A real FFT is performed to yield 2 streams of data which can be further decomposed into 4 streams of data: Real and Imaginary Voltage and Real and Imaginary Current. This is achieved by adding and subtracting positive and negative

mirror frequencies for the real and imaginary parts, respectively. Since the aliased signal in the 12-15 kHz range falls below 80 dB, the accuracy is achieved using the above discussed FIR filter. Alternatively, a 256-point complex FFT can be performed on every phase of the decimated data stream. This yields 2 pairs of data streams – a real part,
 5 which is the voltage, and an imaginary part, which is the current. This approach requires a 256 complex FFT every 16.667 milliseconds.

Performing either FFT results in the following voltage and current, where the notation $V_{m,n}$ denotes the m^{th} harmonic of the n^{th} cycle number. For example, V_{11} and I_{11} correspond to the fundamental of the first cycle and V_{21} and I_{21} to the first harmonic of
 10 the first cycle, etc., as shown in FIG. 16.

The real and imaginary parts of the harmonic content of any k^{th} cycle are given by:

$$V_{mk} = \text{Re}(V_{mk}) + i \text{Im}(V_{mk}); m = 1 \dots M$$

$$I_{mk} = \text{Re}(I_{mk}) + i \text{Im}(I_{mk}); k = 1 \dots n$$

The imaginary part of voltage is the measure of lack of synchronization between
 15 the PLL and the line frequency. In order to calculate metering quantities, the calculations are done in the time-domain. In the time-domain, the FFT capability offers the flexibility to calculate metering quantities using only the fundamental or including the harmonics. Using the complex form of voltage and current obtained from FFT, the metering quantities are calculated as:

$$P = V_{mk} * I_{mk}^*$$

$$W = \text{Re}(P) = \text{Re}(V_{mk}) * \text{Re}(I_{mk}) + \text{Im}(I_{mk}) * \text{Im}(V_{mk})$$

$$Var = \text{Im}(P) = \text{Re}(I_{mk}) * \text{Im}(V_{mk}) - \text{Re}(V_{mk}) * \text{Im}(I_{mk})$$

$$PowerFactor = W / P$$

In the above formulas, when the harmonics are included
 (V_{mk} & $I_{mk}; m = 1 \dots M, k = 1 \dots n$), all metering quantities include the effects of harmonics. On the other hand, when only the fundamental is used (V_{1k} & I_{1k}), all
 25 calculated quantities represent only the 60Hz contribution. As an example, we show the calculations when only the fundamental is used to perform calculations. Only V_1 and I_1 are used from all FFT data frames. The following quantities are calculated for a given set of N frames and a line frequency of f_{line} :

$$kWh = \sum_{i=1}^N [\text{Re}(V_{li}) * \text{Re}(I_{li}) + \text{Im}(V_{li}) * \text{Im}(I_{li})] * \Delta t_i * 10^{-3}$$

$$kVAR = \sum_{i=1}^N [\text{Re}(I_{li}) * \text{Im}(V_{li}) - \text{Re}(V_{lk}) * \text{Im}(I_{lk})] * \Delta t_i * 10^{-3}$$

$$kVAh = \sum_{i=1}^N |V_{li}| * |I_{li}| * \Delta t_i * 10^{-3}$$

$$V^2h = \sum_{i=1}^N |V_{li}|^2 * \Delta t_i$$

$$I^2h = \sum_{i=1}^N |I_{li}|^2 * \Delta t_i$$

$$\Delta t = \frac{1}{f_{line}}$$

The displacement power factor is given by:

$$\cos(\theta) = \frac{|W|}{|VA|}; \text{ where } W \text{ and } VA \text{ only includes the fundamentals and}$$

$$5 \quad VA_1 = V_1 RMS * I_1 RMS; \text{ where}$$

$$V_1 RMS = \sqrt{\sum_{n=1}^N |V_{1,n}|^2} \text{ \& } I_1 RMS = \sqrt{\sum_{n=1}^N |I_{1,n}|^2}; \text{ for } N \text{ cycles}$$

The THD is the measurement of the harmonic distortion present and is defined as the ratio of the sum of the powers of all harmonic components to the power of the fundamental. For the n^{th} cycle, this is evaluated as:

$$10 \quad VTHD_n = \frac{\sqrt{\sum_{m=2}^M V_{mn}^2}}{V_{1n}} \text{ \& } ITHD_n = \frac{\sqrt{\sum_{m=2}^M I_{mn}^2}}{I_{1n}}; V_{m,n} (I_{mn}) \text{ is the } m^{\text{th}} \text{ harmonic}$$

from the n^{th} cycle obtained from the FFT, where

$$V_{m,n}^2 = \text{Re}(V_{m,n})^2 + \text{Im}(V_{m,n})^2 \text{ \& } I_{m,n}^2 = \text{Re}(I_{m,n})^2 + \text{Im}(I_{m,n})^2.$$

This provides the flexibility to either include or exclude the harmonics when calculating metering quantities.

15 LOGICAL DEDUCTIONS FROM DATA RECEIVED FROM PLC COMMUNICATION

Embodiments of the current invention permit demodulation of messages from multiple scan transponders and meters simultaneously, thus providing a significant improvement in communications. Once a network of STs is established along with preferred meters in the service territory and the appropriate tuning and coupling
5 installations are made, the system preferably operates in three distinct modes:

1. Communication Alignment Mode- Prior to collecting data from the meters, the STs sends out a periodic burst of signal stream of alternating 1 and 0 bits for ~ 5 minutes. All the meters in the service territory are programmed to receive this burst mode. The meters align their data clocks and choose the best ST
10 with which to communicate for other modes of system operation.

2. Data Collection Mode- Once the data clocks are aligned with incoming FFT frames, each of the STs in the network communicates with the meters in its latest Cross Reference list and collects data stored in the memory of the meters using PLC, either on a demand or a scheduled basis.

15 3. Dynamic Mapping Mode- The entire ST network preferably cooperates to detect changes in the service territory. These may include, but are not limited to:

- a. Isolated hardware failure
 - i. Meter hardware failure
 - 20 ii. Transformer Fuse failure
- b. Power failure
 - i. Distribution transformer failure
 - ii. Feeder Failure
- c. Switching of Feeders
 - 25 i. Feeder Faults
 - ii. System wide load balancing
- d. Addition and updating of meters

It is a common utility practice to switch feeder trunks (for example, to take a feeder out of service for maintenance, to switch feeders due to feeder faults, or to balance
30 loads in the system). Under any one of these events, the scan transponder loses communication with the meters since they no longer can be contacted by the ST. The manual update of a cross-reference type of list, as performed by certain prior art, presents

a significant concern with respect to scalability of the system. The current invention, in at least aspect, addresses this issue as follows:

Consider a typical utility setup as shown in FIG. 1: B1 and B2 are two feeder branches connected to 9 distribution transformers (T_i ; $i = 1$ to 9). These transformers feed a heterogeneous (single-phase and poly-phase) population of meters connected on the secondary low voltage side. Branch B1 can be fed from feeders emanating from any one of the three substations by the use of switches U1, U2, U3, and U4. Similarly, meters connected to B2 can be fed either from substation 2 or substation 3 by using switches U5 and U6. In addition, there is a Sub-branch that can be fed from any of the substations by using Sub Branch switches SB1 and SB2.

Remote Server Directory

The remote server to which the system of STs is connected maintains a directory (for example, Lightweight Directory Access Protocol or LDAP) which is essentially a hierarchical framework of objects with each object representing a shared entity. Once the system configuration is fed into the directory, the algorithm constantly updates this map as changes are made in the territory. This involves communicating with the meters and automatically mapping the system configuration by including information on primary and alternate paths to every meter. See FIG. 17.

The directory thus contains information regarding various abstraction levels in the network- feeder level, phase level, distribution transformer level, and meter level. The server runs a program that monitors the communication performance of the various STs deriving their master clocks from it. Every transformer is assigned a primary meter (typically the first-connected meter, m_1) with which the STs constantly communicate in order to detect outage and other changes in the service territory.

For example, SS1 feeds B1 by switch U1. In this case, the directory comprises the following information for meter m_1 connected to T1 in a look-up table:

Table 2

m1 Path	Substation	Feeder	Switch	Transformer	Scan Transponder
Primary	1	F1	U1	T1	ST1F1 ₁
Alternate 1	1	F2	U2	T1	ST1F2 ₁
Alternate 2	2	F1	U3	T1	ST2F1 ₁
Alternate 3	3	F2	U4	T1	ST3F2 ₁

The scan transponders preferably are named such that the first number is indicative of the corresponding substation and the number following F is indicative of the feeder number emanating from that substation, and the subscript indicates the phase on which it is installed. FIG. 17 depicts an exemplary directory structure implemented in the server, which can be configured for various event information. These events may include: (1) basic consumption data; (2) outage data; (3) power quality information; (4) status verification flags of several parts of a network; (5) load profile information for certain meters; (6) preventive maintenance flags for part of network infrastructure; and (7) status change flags of network elements such as switches, feeder changers, and reclosers. A preferred algorithm to raise status change flags of several network elements and for localizing outages is discussed below.

To explain the algorithm, we assume a population of m transformers and n meters per transformer. Running index i goes from 1 to m , and index k goes from 1 to n .

After a typical data collection operation period, the server preferably creates a list of meters that failed communication with their respective STs and hence failed to report consumption data. *LIST* is a preferred data structure listing meters that failed communication. Referring to FIG 18, preferably,

1. During the data collection mode, the STs communicate with the meters in their cross reference list and collect energy consumption interval data.
2. All the meters that fail to communicate with the STs are grouped into a data structure called *LIST*. This data structure is stored in the server.
3. The server determines the alternate paths by which the meters can be accessed by using the look-up table (Table 2) in its memory.
4. The alternate paths for all the meters are traced.
5. Logical conclusions are made, outage is localized, flags are set and reporting is provided to the utility by sending a command to the utility control center.
6. The service map in LDAP and the cross reference list of STs are updated to access meters.
7. The above steps continue to take place after every data collection period is completed.

By implementing the above process steps, not only is the system map dynamically updated, the utility also gets immediate notifications of changes made in a service territory (outages, feeder switching, etc.). In addition, if the utility decides to discontinue power to some customers (typically due to sustained failure of payment), the
5 corresponding meters fail to communicate. This change, once noticed by the EMS, can be verified with the utility by interfacing the remote server with a utility Customer Information System (CIS). This eliminates manual updating of the meter cross reference list for STs, thus making the system scalable for both utility and submetering installations.

10 As discussed, one unique feature of certain embodiments of this invention is the synchronization of all transponder data clocks to a global data clock, which may be derived from a remote server that may derive its own clock from one of the phases of the line frequency. Further, when the slave devices (typically meters) perform FFT on data frames, they preferably shift their own data clocks to align their FFT frames with the
15 incoming data bits (see FIG 9). Once this shifting is completed, each meter has knowledge of the absolute phase (absolute phase with 0 degrees referred to as "phase A," absolute phase with 120 degrees lead referred to as "Phase B," and absolute phase with 240 degree lead referred to as "phase C"). This unique capability enables determination by the remote server of absolute phase across the entire system territory.

20 Prior art systems do not allow for such a determination of absolute phase for a meter. The meters in some systems contain some information regarding phases, but only of relative phases, since the meter "sees" three phases 120 degrees apart. This lack of information regarding in phase continuity is also why it becomes difficult to exactly determine the absolute phase that feeds a wall socket, in a room with multiple sockets, on
25 a given floor with multiple rooms, in a multi floor building being fed from three utility phases.

Embodiments of the current invention provide the continuity of phase information throughout the territory, from the remote server to transponders installed in substations down to meters installed in the low voltage territory. This capability enables
30 identification of the absolute phase by which each single phase meter is powered up in the service territory.

Given the above capability, embodiments of the current invention enable reconstructing the load of a distribution transformer by phase, without actually installing a three phase meter at the transformer's secondary output. For a typical utility installation consisting of multiple transformers, this reduces system costs while providing value
5 added service. By performing a vector sum of the currents on the three phases, the total load on the distribution transformer can be accurately determined at the substation.

Submetering involves the allocation of energy costs within a multi-tenant property according to the energy consumption by individual tenants. The meters measure electricity consumed by individual tenants and communicate the consumption data to a
10 Scan Transponder, preferably installed at an entry point to the property, using the power lines in the property. This data then may be accessed from the transponder by a host of communication infrastructures (e.g., wireless, phone line, GPRS, etc.). In a preferred submetering installation, all the components for medium tension installation are eliminated, since both the STs and meters are installed on the low voltage line.

15 In a submetering environment with multiple electrical services feeding a large building, multiple STs are installed, one for each service. However, due to cross coupling of PLC signals (via the neutral which is common to all services or via phase to phase loads), the assignment of specific meters to each ST can be a tedious process. This invention allows the STs installed on different services to be connected to a remote server
20 that can dynamically assign a meter cross reference list for every transponder as the communication environment changes.

A preferred submetering control module comprises a Power Board (see FIG. 21 for schematic) that also has the PLC transmit and receive circuitry on it. The Power Board provides power to the CPU board. The control module also comprises an I/O
25 extension board (see FIG. 22 for schematic), which has several I/O extension options that enable communication from metering modules to the CPU board.

A preferred control module also comprises a CPU Board (see FIG. 23 for schematic), which has a Digital Signal Processing (DSP) processor.

For residential applications where limited data is expected (typically energy
30 consumption only), another embodiment may include a low-cost meter with reduced resources compared to that presented in FIG 23. This meter circuit is PCB 240, presented in FIG 24.

Each residential meter preferably also has a 9-digit display board (PCB 220; see FIG. 25 for schematic).

Although FFT has been described herein in various contexts, those skilled in the art will recognize that discrete Fourier transform (DFT) could also be used in each case.

- 5 The various embodiments described above are provided as an illustration only and do not limit the invention. The skilled in the art will recognize the various modifications that can be made to the embodiments discussed, without departing from the scope of the invention, which is set in the claims below.

APPENDIX

FFT UPDATE

```

*****
5  *****

        .text
        .align          2
        .globl          fft_executel3
        .type            fft_executel3, @function

10  fft_executel3:
        /* Move stack to sram, save registers there */
        move.l sram_stack, %a0
        movm.l #0xfcfc, (%a0)    /* saves d2-d7, a2-a5, fp, sp */

15        move.l 4(%sp), %a5
        move.l 8(%sp), %a4
        move.l bit_reverse_index, %a1
        move.l 12(%sp), %a3
        move.l 16(%sp), %a2

20        move.l %a0, %sp

        /* put copy of arguments on new stack */
        move.l %a5, 72(%sp)    /* inr */
25        move.l %a4, 76(%sp)    /* ini */
        move.l %a3, 80(%sp)    /* outr */
        move.l %a2, 84(%sp)    /* outi */

30        /***** decimate *****/
        clr.l %d0

1:
        move.w (%a1)+, %a0
        move.w (%a5, %a0.1*2), %a6
35        move.l %a6, (%a3)+

        move.w (%a4, %a0.1*2), %a6
        move.l %a6, (%a2)+

```

```

    addq.l #1,%d0
    cmp.l #2047,%d0
    jble 1b

5
    /***** fft outer loop *****/
    moveq.l #1,%d0
    move.l %d0,60(%sp)    /* nab */
10    move.l #1024,%d0
    move.l %d0,56(%sp)    /* ngrps */
    clr.l 52(%sp)         /* stage */

    2:
15    move.l 80(%sp),%a5    /* waccpxr */
    move.l 84(%sp),%a4    /* waccpxi */
    move.l 60(%sp),%d0    /* nab */
    lsl.l #2,%d0
    lea (%a5,%d0.1),%a2    /* waccpyr */
20    lea (%a4,%d0.1),%a0    /* waccpyi */
    clr.l 48(%sp)
    move.l 56(%sp),%d0
    cmp.l 48(%sp),%d0
    jble 6f
25    move.l 60(%sp),%d0
    lsl.l #2,%d0
    move.l %d0,%a6

    3:
30    move.l twiddle_real,%a3
    move.l twiddle_imag,%a1
    clr.l %d6
    cmp.l 60(%sp),%d6
    jbge 5f
35    move.l 56(%sp),%d7
    lsl.l #2,%d7

    4:
    move.l (%a5),%d0

```

```

    move.l (%a4),%d5
    move.l (%a2),%d3
    move.l (%a0),%d4
    move.l (%a3),%d2
5      move.l (%a1),%d1

    mov.l %d0, %ACC0
    mov.l %d5, %ACC1
    mov.l %d0, %ACC2
10     mov.l %d5, %ACC3
    mac.l %d3, %d2, %ACC0
    msac.l %d4, %d1, %ACC0
    mov.l %ACC0, %d0

15     move.l %d0, (%a5)+

    mac.l %d3, %d1, %ACC1
    mac.l %d4, %d2, %ACC1
    mov.l %ACC1, %d0
20

    move.l %d0, (%a4)+

    msac.l %d3, %d2, %ACC2
    mac.l %d4, %d1, %ACC2
25     mov.l %ACC2, %d0

    move.l %d0, (%a2)+

    msac.l %d3, %d1, %ACC3
30     msac.l %d4, %d2, %ACC3
    mov.l %ACC3, %d0
    move.l %d0, (%a0)+
    addq.l #1,%d6
    add.l %d7,%a3
35     add.l %d7,%a1
    cmp.l 60(%sp),%d6
    jblt 4b

5:     moveq.l #1,%d0

```

```

        add.l %d0, 48(%sp)
        add.l %a6, %a5
        add.l %a6, %a4
        add.l %a6, %a2
5       add.l %a6, %a0
        move.l 56(%sp), %d0
        cmp.l 48(%sp), %d0
        jbgt 3b

10      6:
        moveq.l #1, %d0
        add.l %d0, 52(%sp)
        move.l 60(%sp), %d0
        add.l %d0, %d0
15      move.l %d0, 60(%sp)
        move.l 56(%sp), %d0
        asr.l #1, %d0
        move.l %d0, 56(%sp)
        moveq.l #10, %d0
20      cmp.l 52(%sp), %d0
        jbge 2b
        movm.l (%sp), #0xfcfc    /* restores d2-d7, a2-a5, sp */

        rts
25

```

ABS.DEF

```

30      /*
        version 2.00 for slash 2 9/8/2000 sas
        */
        #define abs(x) (x<0?-x:x)

```

35

ALIGNPH.C

```

40      /* align the phases of the meter by trying 18 possibilities from table
        */
        /* normally 6 communication phases on each of 3 relay settings */
        /*
        version 2 9/28/1994 sas
        1)    include tryplc def
45      2)    move required receptions and xmit relay array to non vol ram

```

```

3)      change method of calculating best parms after REQUIRED RECEPTIONS
met
      a)      first isolate best phase, then best delay
      */
5
/*
version 1 created for tmx5 12/28/1999 sas
      1)      eliminate searches for delays, because all TMX-5
              automatically search delays on receive
10      2)      add more xmit relays, NUM_XMIT_IDXS per NUM_RCV_IDXS
      3)      align a meter completely before going on. Write meter_align_stru
to flash
              when done.

15 version 1.02 11/19/2001 sas
      1)      don't defer a slot in alignph on con-comm – return non comm and
let do_scan() defer it

version 1.03 2003-06-18
20      1)      clean up looping in setpoll
      2)      fix bug where master edgewart time was set to 20 seconds at end
of setpoll
      3)      change order of search on setpoll - channel 2 first then channel
1
25      4)      search phases first, then channels
      */
      #include "clklink.def"
      #include "mtrlink.def"
      #include "plc.def"
30
      #include "serlink.def"

      #include "copymem.h"
      #include "plc.h"
35      #include "scan.h"
      #include "log.h"
      #include "flash_io.h"

      const cmdstru8 init_setpoll_cmd = {0xa, 0,0,0,0,sizeof
40      (mastcmdastru) /sizeof (long) ,0,0} ;
      const cmdstru8 init_alignph_cmd = {0x0, 0,0,0,0,sizeof
(mastcmd0stru) /sizeof (long) ,0,0} ;

      #pragma region ("data=scanRamInit2")
45      void (*alignph_default_scan_statep) (void) =alignph_default_scan_state;
      void (*setpoll_default_scan_statep) (void) =setpoll_default_scan_state;
      #pragma region ("data=scanRamInit3")
      void (*alignph_init_statestrp) (void) =alignph_init_statestr;
      void (*setpoll_init_statestrp) (void) =setpoll_init_statestr;
50      #pragma region ("data=scanWarmStart")
      void (*alignph_warm_startp) (void) =alignph_warm_start;
      void (*setpoll_warm_startp) (void) =setpoll_warm_start;

      #pragma region ("data=data")
55
      SCANFN_RET init_setpoll (SCANFN_ARGS)
      {
              return (STATE_INITIALIZED) ;
60      }

```

```

#define SHORT_EDGEWAIT (80)

int chan_switch_table [4] [2] = {
    {-1,-1}, // none -- quit
5    {1,0},      // low channels 1 and 2
    {2,3},      // high channels 3 and 4
    {4,0}       // xlow channels 1 and 5
};

10 int setpoll_retries, setpoll_rcv_idx, setpoll_changroup, setpoll_chan ;
SCANFN_RET setpoll (scanfnargs *sfap)
{
    messu mess_u ;
    messghdr messgHdr ;
15    setpollmessg *setpoll_messgp ;
    setpollstru set_poll ;
    reg8stru reg8 ;
    unsigned int rcv_idx, good_resp;
    int i, j, k, l, n, cg, flag;
20    SCANFN_RET returned ;
    int try_cg, default_cg, num_cg ;
    int masterEdgewaitTime_save = parmstru.channel.masterEdgewaitTime;
    comparmstru temp_comm_parm_stru ;

25    returned = NO_COMM ;

    temp_comm_parm_stru = *sfap->comm_parm_strup ;

    /* get the last time the poll/cg was set */
30    mess_u = scandatap [sfap->scan_slot+1] {sfap->meter_state} ;

    default_cg = scan_stru.pters.xref_strup->reg8.xref_5.changroup ;
    num_cg = 1 ;
    if (default_cg != 0)
35    {
        ++num_cg;
    }
    if (mess_u.messp)
    {
40        /* additional changroup to try if it is not 0 and not the
        default cg */
        try_cg = mess_u.setpolleventp->_setpoll_messg.reg8.setpoll_
        5.changroup;
        if (try_cg != 0 && try_cg != default_cg)
45        {
            ++num_cg;
        }
    }
    else
50    {
        try_cg = default_cg ;
    }

    /* set up a record for the setpoll event */
55

    messgHdr = getMessgHdr (SCAN_DATA, messglen (setpollmessg));
    messgHdr.time rtc = sfap->sp->serno;

60    mess_u = putNextDailyMessgHdr (&messgHdr) ;

```

```

if (mess_u.messp)
{
    setpoll_messgp = &mess_u.setpolleventp->settpoll_messg;
    reg8.ulong=0
5    reg8.setpoll_5.changroup = default+cg ;
    reg8.setpoll_5.fnIndex = SETPOLL_STATE ;

    /* put only the first 16 bits of the reg8 save rest for later */
    if (0==putWord(&setpoll_messgp->reg8.ushorta[0],reg8.ushorta
10    [0]))
    {
        storePointers(mess_u,DAILY_EVENTS,ONLY_ONE_EVENT,NO_WAIT);

        /* set up the command */
15        parmstru.parm.prot.cmd.command = init_setpoll_cmd.command ;
        parmstru.parm.prot.cmd.size = init_setpoll_cmd.size ;
        /* set the channel group in the meter to the NEW channel
        group */
        parmstru.parm.prot.cmd.reg = default_cg ;
20        /* re-lock the channel number except channel 1 on xlow */
        if (freq_range(parmstru.parm.comm_parm_stru.rcvRelay)
        ==PLC_PH_XLOW_FREQ && parmstru.channelNbr==0)
        {
            parmstru.parm.prot.cmd.flag1 = 0,
25        }
        else
        {
            parmstru.parm.prot.cmd.flag1 = 1;
        }
30        /* lock the channel group */
        parmstru.parm.prot.cmd.flag2 = 1;

        /* we have 3 channel groups to try - try right now to fix it

35        using the default parameters rcv_idx, xmit_relay and
        channel that came from do_scan */
        returned=NO-COMM ;
        if (temp_comm_parm_stru7.xmitRelay && try_cg!=default_cg)
        {
40            /* set the channel group to the old group */
            changroup = try_cg ;
            /* signal to acquire channel */
            parmstru.ctrl |=ACQUIRE_CHANNEL;
            while ((parmstru.ctrl&ACQUIRE_CHANNEL) && (!
45            (parmstru.ctrl&DO_BREAK)) && (scan_ctrl.scanLevel==Run && !
            scan_ctrl.pause))
            {
                scanwait() ;
            }
50            parmstru.channel.masterEdgewaitTime = SHORT_EDGEWAIT ;
            /* set turn around delay */
            good_resp=try_plc (6) ;

            if (good_resp==GOOD_REPLY || good_resp==GOT STATUS)
55            {
                returned=STATE_COMPLETE;
            }

            else
60            {

```

```

        returncd=NO_COMM;
    }
}

5      if (returncd==NO_COMM)
    {
        /* if we already have paramters, try them */
        if (temp_comm_parm_stru.xmitRelay)
        {
10         /* if we have aligned this meter already, try the
existing parms first */
            /* set the channel group to the default group */
            changroup = default_cg ;
            /* signal to acquire channel */
15         parmstru.ctrl |= ACQUIRE_CHANNEL;
            while ((parmstru.ctrl & ACQUIRE_CHANNEL) && (!
(parmstru.ctrl & DO_BREAK)) && (scan_ctrl.scanLevel==RUN && !
scan_ctrl.pause))
            {
20                 scanwait ()
            }

            /* try 6 times with the default cg before looking
for all possibilities */
25         good_resp=try_plc(6) ;

            if (good_resp==GOOD_REPLY | | good_resp==GOT_STATUS)
            {
30                 returncd=STATE_COMPLETE;
            }
            else
            {
                returncd=NO_COMM;
            }
35         }

        if (returncd == NO_COMM)
        {
            /* re-init the xmit and rcv relay to zero as a flag
40         that we don't have good parameters */
            temp_comm_parm_stru.rcvRely = 0;
            temp_comm_parm_stru.xmitRelay = 0;

            for (i=0; i<scan_stru.pters.xref_strup->ctrl.max_tries
45         && returncd==NO_COMM && (scan_ctrl.scanLevel==RUN && !scan_ctrl.pause)
; i++)
            {
                setpoll_retries = i ;
                /* first try the spec'd channel, then the other
50         if no reply */
                for (k=0; k<2 && (scan_ctrl.scanLevel==RUN && !
scan_ctrl.pause) ; K++ )
                {
                    /* loop through channel groups of 0 and
55         xref's changroup */
                    for (j=0, cg=0; j
<num_cg && returncd==NO_COMM; j++, cg=((j==1)?default_cg:try_cg))
                    {
80                 setpoll_changegroup = j ;

```

```

                                for (rcv_idx=0; rcv_idx
<NUM_RCV_IDXS&&returncd==NO_COMM && (scan_ctrl.scanLevel==Run && !
scan_ctrl.pause) ;rcv_idx++) -
                                {
5                                n = chan_switch_table[freq_range
(rcv_idx)] [k] ;
                                if (n>=0)
                                {
10                                /* only use this rcv_idx if the
                                mast_mask allows it to be used for master mode */
                                if (releaseCodep->
                                option.mast_ph_enable & (l<rcv_idx))
                                {
15                                /* set the relays */
                                temp_comm_parm_stru.rcvRelay=rcvtbl[rcv_idx] ;
                                temp_comm_parm_stru.xmitRelay=xmit_relay[rcv_idx] ;
                                temp_comm_stru.hardware
                                = 0 ;
20                                /* only if the xmit relay is
                                enabled */
                                if
                                (temp_comm_parm_stru.xmitRelay !=0)
                                {
25                                setpoll_rcv_idx =
                                rcv_idx ;
                                setpoll_chan = n ;
                                // loop through
30                                couplers, if any
                                temp_comm_parm_stru.hardware=0 ;
                                for (l=8; l>0; l>>=1)
                                {
35                                if (releaseCodep->
                                option.couplers_mask==0 | | temp_comm_parm_stru.xmitRelay!=0x8)
                                {
                                // only loop
                                once through this section if no couplers or not on T==8, set the coupler
40                                mask to 0
                                l=0 ;
                                }
                                else if
45                                (releaseCodep->option.couplers_mask && (releaseCodep->
                                option.couplers_mask&l)==0)
                                {
                                // have
                                couplers, on T==8, but this coupler is not installed - loop to find the
                                bit for an installed coupler
50                                continue;
                                }
                                else
                                {
                                // on T==8 and
55                                this coupler is installed, use l as the coupler number
                                }
                                // check for the
60                                need to change the channel, channel group or coupler number
                                if

```

```

[parmstru.channelNbr!=n || changroup != cg || 1!
=temp_comm_parm_stru.hardware)
{
// set the
5 coupler
parmstru.parm.comm_parm_stru.hardware=temp_comm_parm_stru.hardware = 1;
/* change the
channel */
10 temp_com_parm_stru.channelNbr=n;
// set the
channel group
changroup = cg ;
15 parmstru.newChannel=temp_comm_parm_stru.channelNbr;
// signal to
acquire channel or coupler
20 parmstru.ctr. |=ACQUIRE_CHANNEL;
while
((parmstru.ctrl&ACQUIRE_CHANNEL) && !(parmstru.ctrl&DO_BREAK)) &&
(scan_ctrl.scanLevel==RUN && ~!scan_ctrl.pause))
{
25 scanwait() ;
}
}
/* remember that the
30 meter is locked on this channel */
temp_comm_parm_stru.channelNbr = parmstru.channelNbr;
/* copy the
parameters over to the parmstru */
35 parmstru.parm.comm_parm_stru = temp_comm_parm_stru ;
parmstru.parm.prot.cmd.command = init_setpoll_cmd.command ;
40 parmstru.parm.prot.cmd.size = init_setpoll_cmd.size ;
/* set the channel
group */
parmstru.parm.prot.cmd.reg = scan_stru.ptrs.xref_strup->reg8.xref_
45 5.changroup;
parmstru.parm.prot.cmd.flag2 = 1;
/* unlock the
channel so that the meter can find us as we switch channels */
50 parmstru.parm.prot.cmd.flag1 = 0;
parmstru.channel.masterEdgewartime = SHORT_EDGEWAIT ; /* set turn
around delay */
55 good_resp=try_plc
(1);

```

```

    returned =
(good_res==GOOD_REPLY || good_resp==GOT_STATUS) ? UNLOCKED : NO_COMM ;
}
}
}
}
}
}
}
}
}
}
else
{
    returned=REINIT;
}
if (returned==STATE_COMPLETE || returned==UNLOCKED)
{
    set_poll.flag =
flag = (short) (backTime-mess_u.hdrx->event.time) ;
set_poll.comm_parm_stru =
"sfp->comm_parm_strup = temp_comm_parm_stru ;
set_poll.pollAddress = sfp->sp->poll_address ;
if (returned==UNLOCKED)
{
    /* no lock the channel on this channel */
    /* re-lock the channel number except channel 1 on xlow
*/
        // set up the command to lock the channel
parmstru.parm.prot.cmd.command = 0 ;
if (freq_range(parmstru.parm.comm_parm_stru.rcvRelay)
==PLC_PH_XLOW_FREQ && parmstru.channelNbr==0)
{
    parmstru.parm.prot.cmd.flag1 = 0;
}
else
{
    parmstru.parm.prot.cmd.flag1 = 1;
}
/* put in the acquired comm parameters */
parmstru.parm.comm_parm_stru = *sfap->_comm_parm_strup ;
/* set the channel group */
changroup = default_cg ;
/* signal to acquire channel */
parmstru.ctrl |= ACQUIRE_CHANNEL;
while ((parmstru.ctrl&ACQUIRE_CHANNEL) && (!
(parmstru.ctrl&DO_Break)) && (scan_ctrl.scanLevel==RUN && !
scan_ctrl.pause))
{
    scanwait();
}
/* send the command to lock the channel on this channel
*/
good_resp=try_plc(6) ;
/* make sure the meter got the command with the new
comm_par_stru */
if (good_resp==GOOD_REPLY || good_resp==GOT_STATUS)
{

```

```

        returned=STATE_COMPLETE ;
    }
5      else
    {
        /*assume that the meter is unlocked */
        parmstru.parm.prot.cmd.flag1 = 0;
    }
10    }
    }
    else
    {
15      set_poll.flag =
        flag = (short) (mess_u.hdrpx->event.time-backTime-2) ;
        set_poll.comm_parm_strup = zero_comm_parm_stru ;
        *sfap->comm_parm_strup = zero_comm_parm_stru ;
        set_poll.pollAddress = -1 ;
        returned=NO_COMM;
20    }
    reg8.setpoll_5.changroup = default_cg ;
    reg8.setpoll_5.channel = sfap->comm_parm_strup->channelNbr ;
    reg8.setpoll_5.locked = parmstru.parm.prot.cmd.flag1 ;
    if (0==putWord(&setpoll_messgp->reg8.ushorta[1], reg8.ushorta
25    [1]))
    {
        putLBuf ((unsigned long *) (&setpoll_messgp->set_poll),
(unsigned long *) (&set_poll)_messglen(setpollstru)) ;
    }
30    else
    {
        returned=REINIT;
    }
    }
35    else
    {
        returned=REINIT;
    }
    /* restore master edgewart time */
40    parmstru.channel.masterEdgewartTime = masterEdgewartTime_save ;
    return (returned) ;
}

SCANFN_RET init_alignph (scanfnargs *sfap)
45 {
    return (STATE_INITIALIZED) ;
}

SCANFN_RET alignph (scanfnargs *sfap)
50 {
    return (STATE_COMPLETE) ;
}

#define TPP 1
55 SCANFN_RET align_comm(scanfnargs *sfap)
{
    return (STATE_COMPLETE) ;
}

60 SCANFN_RET setpoll_init_1_statestr(scanfnargs *sfap)

```

```

{
    SCANFN_RET returned ;
    messu mess_u;
5
    sfap->stp->allowed_state_map |= SETPOLL;
    sfap->stp->requested_state_map |= SETPOLL;
    returned = init_l_statestr(sfap);
    if (returned == STATE_COMPLETE)
10
        {
            mess_u = scandatap[sfap->scan_slot+1][sfap->meter_state];
            if (mess_u.messp)
                {
                    if (mess_u.setpolleventp->
15 setpoll_messg.set_poll.com_parm__stru.xmitRelay)
                        {
                            /* copy over the communication parameters if the xmit
                            relay is set property */
                            *sfap->comm_parm_strup = mess_u.setpolleventp->
20 setpoll_messg.set_poll.com_parm_stru;
                        }
                        /* if poll address was not set to the correct value, set it
                        again */
                        if (mess_u.setpolleventp->
25 setpoll_messg.set_poll.pollAddress != sfap->sp->poll_address ||
                            /* if channel group was not set to the correct value,
                            set it again */
                            mess_u.setpolleventp->setpoll_messg.reg8.setpoll_
30 5.changroup != scan_stru.ptrs.xref_strup->reg8.xref_5.changroup ||
                                /* if meter was left unlocked, try setpoll again */
                                !mess_u.setpolleventp->setpoll_messg.reg8.setpoll_
                                5.locked
                                )
                                    {
45
40
55
60
                    sfap->stp->initialized_state_map &= ~SETPOLL;
                    sfap->stp->completed_state_map &= ~SETPOLL;
                    sfap->stp->requested_state_map |= SETPOLL;
                }
            }
        }
    return (returned);
}
SCANFN_RET alignph_init_l_statestr(scanfnargs *sfap)
{
45
    messu mess_u, setpoll_mess_u;
    SCAN_RET returned ;

    sfap->stp->allowed_state_map |= ALIGNPH;
    sfap->stp->requested_state_map |= ALIGNPH;
50
    returned = init_l_statestr(sfap);

    if (returned == STATE_COMPLETE)
        {
            mess_u = scandatap[sfap->scan_slot+1][sfap->meter_state];
55
            if (mess_u.messp)
                {
                    setpoll_mess_u = scandatap[sfap->scan_slot+1][SETPOLL_STATE]

                    if (setpoll_mess_u.hdrp && setpoll_mess_u.hdrp->
60
                        time.systemTime >= mess_u.hdrp->time.systemTime)

```

```

        {
            /* do nothing here, the setpoll is later than the
alignment */
5          /* keep the commparmstru data from setpoll over the data
from alignment */
        }
        else if (mess_u.aligneventp->align_messg.align.flag>=0)
        {
10         if (mess_u.alighnventp->align_mess.align.channelNbr
<NUMCHANS)
            {
                sfap->comm_parm_strup->xmitRelay =
mess_u.aligneventp->align_messg.align.xmit_relay;
15         sfap->comm_parm_strup->rcvRelay =
mess_u.aligneventp->align_messg.align.rcv_relay;
                sfap->comm_parm_strup->channelNbr =
mess_u.aligneventp->align_messg.align.channelNbr;
                /* make this state allowed and initialized and
20         completed if complete */
                /* also the set poll state */
                sfap->stp->initialized_state_map|=ALIGNNPH;
                sfap->stp->completed_state_map|=ALIGNNPH;
            }
25         else
            {
                sfap->stp->initialized_state_map&=~ALIGNNPH;
                sfap->stp->completed_state_map&map&=~ALIGNNPH;
                sfap->stp->requested_state_map|=ALIGNNPH;
30         }
            }
        }
    }
    return (returned)
35 }

void setpoll_default_scan_state()
{
40     Xref_temp.re_[SETPOLL_STATE].interval_minutes = 1440;
    Xref_temp.re_scan[SETPOLL_STATE].offset_minutes = 10;
    Xref_temp.ctrl.allowed_states |= SETPOLL ;
}

45 void alignph_default_scan-state()
{
    Xref_temp.re_scan[ALIGNNPH_STATE].interval_minutes = 1440;
    Xref_temp.re_scan[ALIGNNPH_STATE].offset_minutes = 10;
50     Xref_temp.ctrl.allowed_states |= ALIGNNPH ;
}

void alignph_warm_start()
{
55     scan/vls *cmdp;

    cmdp=scan_stru.cmd+ALIGNNPH_STATE;

    cmdp->do_scan=alignph;
    cmdp->init_scan=init_alignph;
60     cmdp->reset_flash=reinit_flash_alignph;

```

```

cmdp->rebuild_RAIC=alignph_init_1_statestr;
copymem (16L,cmdp->description, "align cornm") ;
copymem (16L,cmdp-> control_cmd, "da") ;

5
    /* if Delta is used, connect each phase to the previous phase with
    an inversion */
    /* test the hdwr to see if the hardware is there for that phase,
    though */
10    if (releaseCodep->option.masterXmitDelta)
        {
            xmit_relay [0] = (releaseCodep->option .plc_ph1)? 0x15:0;
            xmit_relay [1] = (releaseCodep->option .plc_ph2)? 0x23:0;
            xmit_relay [2] = (releaseCodep->option .plc_ph3)? 0x46:0;
15        }
        else
        {
            xmit_relay [0] = (releaseCodep->option .plc_ph1)? 0x1:0;
            xmit_relay [1] = (releaseCodep->option .plc_ph2)? 0x2:0;
20            xmit_relay [2] = (releaseCodep->option .plc_ph3)? 0x4:0;
        }

    /* the fourth phase is just a datalink phase, not for connection to
    LT power lines */
25    /* for MT lines the coupler is outside the ST - so it's like a
    datalink */
    xmit_relay [3]=(releasecodep->option.plc_ph4)?0x8 : 0;
}

30 void setpoll_warm_start ()
{
    scanlvls *cmdp;

    cmdp=scan_sru.cmd+SETPOLL_STATE;
35
    cmdp->do_scan=setpoll;
    cmdp->init_scan=init_setpoll;
    cmdp->rebuild_RAIC=setpoll_init_1_statestr;
    cmdp->reset_flash=reinit_flash_setpoll;
40    copymem (16L, cmdp->description, "set poll") ;
    copymem (16L, cmdp->control_cmd, "xr -p") ;
}

void alignph_init_statestr(void)
45 {
    scanfnargs sfa ;

    sfa.meter_state=ALIGNPH_STATE;
    loop_sfap(alignph_init_1_statestr, (&sfa));
50 }

void setpoll_init_statestr(void)
{
    scanfnargs sfa ;
55
    sfa.meter_state=SETPOLL_STATE;
    loop_sfap(setpoll_init_1_statestr,(&sfa)) ;
}

60 SCANFN_RET reinit_flash_alignph(scanfnargs *sfap)

```

```

{
    messu mess_u ;

5     SCANFN_RET returned = STATE_INCOMPLETE ;
    setpollmessg *setpoll_messgp ;
    reg8stru reg8 ;
    messghdr messgHdr ;

10    /* set up a record for the setpoll event */
    messgHdr=getMessgHdr(SCAN_DATA,messglen(setpollmessg)) ;
    messgHdr.time.rtc = sfap->sp->serno;
    mess_u=putNextDailyMessgHdr(&messgHdr) ;
    if (mess_u.messp)
15        {
            setpoll_messgp = &mess_u.setpolleventp->setpoll_messg;
            reg8.ulong=0;
            reg8.gen_scan.fnIndex = ALIGNPH_STATE;
            if (0==putULWord(&setpoll_messgp->reg8.ulong,reg8.ulong))
20                {
                    storePointers (mess_u , DAILY_EVENTS,ONLY_ONE_EVENT,NO_WAIT) ;
                    putWord((unsigned short *) &mess_u.setpolleventp->
setpoll_messg.set_poll.flag, (unsigned short) (-2)) ;
                }
25        }
    }
    return (returned) ;
}

30 SCANFN_RET reinit_flash_setpoll (scanfnargs *sfap)
{
    messu mess_u ;
    SCANFN_RET returned = STATE_INCOMPLETE ;
    setpollmessg *setpoll_messgp;
35    setpollstru_set_poll ;
    messghdr messgHdr ;

    /* set up a record for the setpoll event */
    messgHdr=getMessgHdr (SCAN_DATA,messglen(setpollmessg)) ;
    messgHdr.time.rtc = sfap->sp->serno;
40    mess_u=putNextDailyMessgHdr(&messgHdr);
    if (mess_u.messp)
        {
            setpoll_messgp = &mess_u.setpolleventp->setpoll_messg;
45            sfap->reg8.setpoll_5.fnIndex = sfap->meter_state;
            if (0==putULWord(&setpoll_messgp->reg8.ulong,sfap->reg8.ulong))
                {
                    set_poll.flag = -2 ;
                    set_poll.comm_parm_stru = zero-comm_parm_stru ;
50                    set_poll.comm_parm_stru.channelNbr = sfap->reg8.setpoll_
5.channel1;
                    set_poll.pollAddress = 0xffff ;
                    putLBuf ((unsigned long *) (&setpoll_messgp->set_poll),
(unsigned long *) (&set_poll),messglen(setpollstru)) ;
55                    storePointers(mess_u,DAILY_EVENTS,ONLY_ONE_EVENT,NO_WAIT) ;
                }
            }

    return (returned) ;

60 }

```

5

BAUDPLL.C

```

10  /*
    baud pll routines
    version 1 created 5/24/1999 sas

    version 2 modified 9/8/1999 sas
    1) do not hang up in baudpllHardwareReset if line frequency capture
15  not working
    2) check max/min on plcbrint to keep within 40 to 70 Hz range
    3) baudP11 -- nco capture is read in the subsecond to insure that
    all cycles are locked, not just at 1 second boundaries

20  version 2.00 for flash 2 9/8/2000 sas
    version 2.01
    1) fix baudInitFlag
    2) restore integrator to line frequency if it goes too far astray
    - before it was set to min or max

25  version 2.02 2/20/2003 sas
    1) tighten up the min/max limits - they were way too loose
    2) fix baudInitFlag
    */
30  #include "mtrlink.def"
    #include "sysclk.def"
    #include "fs1004.def"
    #include <math.h>
    #include "baudpll.h"
35
    #pragma region ("data=secondBack")
    void (*plcBaudSecondp) (void)=plcBaudSecond ;
    #pragma region ("data=data")

40  /*
    N.B. !!! plcBaudHardwareReset must be executed BEFORE
    mtrasmphHardware Reset
    put a !000.0 instead of # in the mod file for this module
    */
45  #pragma region ("data+hardwareRest")
    void (*plcBaudHardwareResetp) (void) = plcBaudHardwareReset ;
    #pragma region ("data=data")

    /* global plc baudrate generator integrator - 64 bits */
50  phaccum plcbrint ;
    unsigned char baudInit Counter, baudInitFlag;
    /* loop constants */
    #define KCS (-565L)
    #define KHS (-10794L)
55  /* number of seconds after jamming frequency control word before baud is
    considered init'd */
    #define INIT_INIT_COUNTER 40
    #define BAUD_NCO_UPPER 0xF0000000
    #define BAUD_NCO_LOWER 0x10000000
60  const unsigned long Kcs = -KCS ;

```

```

const unsigned long Khs = -KHS ;

void setMaxMin (void) ;
unsigned long max_plcbrint.min_plcbrint ;
5
/* timer between initialization and sanity checks */
#define BAUD_PLL_TIMER 240
unsigned int baudpll_timer ;

10 void plcBaudSubsecond (void)
{
}

void plcBaudSecond (void)
15 {
    /*
        routine to do the baud calculation in assembly language plcbrint is
a        phaccum <defined in flash_8K.def> 64 bit unsigned int NCOread is a
20        32 bit signed long. We take its absolute value and remember the sign for
        later
        Khs and Kcs are unsigned longs plcbrfcw is an unsigned long result
25        to put into the fs1004.plc.baudrate (Lo then Hi in parts)

        The calculation is as follows

30        sign = 0 ;
        if (NCOread<0)
        {
            sign = 1;
            NCOread = -NCOread ;
35        }

        if (sign)
        {
            plcbrint += NCOread*Kcs ;
40            temp_phaccum = plcbrint ;
            temp_phaccum += NCOread*Khs ;
            plcbrfcw = temp_phaccum.high ;
        }
        else
45        {
            plcbrint -= NCOread*Kcs ;
            temp_phaccum = plcbrint ;
            temp_phaccum -= NCOread*Khs ;
            plcbrfcw = temp_phaccum.high ;
50        }
    */
    unsigned long plcbrfcw ;
    long NCOread ;
    int sign ;
55    phaccum temp_phaccum,
    *temp_phaccum=&temp_phaccum,
    *plcbrintp=&plcbrint;

    long tempRead ;
60    unsigned long D2save ;

```

```

NCO read = fs1004.plc.baudrateNCO ;
if (NCOread<0)
5   {

    sign = 1 ;
    tempRead =- NCO read ;
10   }
    else
    {
        sign = 0 ;
        tempRead = NCOread ;
15   }
/* add when sign negative because KCS and KHS are negative */
if (sign)
{
    temp_phaccum = plcbrint ;
20   /*plcbrint += NCO read*Kcs ; */
    asm (
        " MOVE.L      {temp_phaccum},A0      ",
        " MOVE.L      {Kcs}, D0              ",
        " MOVE.L      {tempRead}, D1          ",
25   " MOVE.L      D2, {D2save}              ",
        #include      "multa64.def"
        " MOVE.L      {D2save}, D2            "
    );
    plcbrint = temp_phaccum ;
30   /*temp_phaccum +=NCOreads*Khs ; */
    asm (
        " MOVE.L      {temp_phaccum},A0      ",
        " MOVE.L      {Kcs}, D0              ",
        " MOVE.L      {tempRead}, D1          ",
35   " MOVE.L      D2, {D2save}              ",
        #include      "multa64.def"
        " MOVE.L      {D2save}, D2            "
    );
}
40   else
    {
        temp_phaccum = plcbrint ;
        /*plcbrint -= NCOread*Kcs ; */
        asm (
45   " MOVE.L      {temp_phaccum},A0      ",
            " MOVE.L      {Kcs}, D0              ",
            " MOVE.L      {tempRead}, D1          ",
            " MOVE.L      D2, {D2save}              ",
            #include      "multa64.def"
50   " MOVE.L      {D2save}, D2            "
        );
        plcbrint = temp_phaccum ;
        /*temp_phaccum -= NCOread*Khs ; */
        asm (
55   " MOVE.L      {temp_phaccum},A0      ",
            " MOVE.L      {Kcs}, D0              ",
            " MOVE.L      {tempRead}, D1          ",
            " MOVE.L      D2, {D2save}              ",
            #include      "multa64.def"
60   " MOVE.L      {D2save}, D2            "
        );
    }
}

```

```

        );
    }
    plcbrfcw = temp_phaccum.high ;
    /* every BAUD_PLL_TIMER seconds reset Max/min plcbrint and check
5   integrator for sanity */
    if (++baudpll_timer>BAUD_PLL_TIMER)

    {
        baudpll_timer = 0 ;
10    set MaxMin() ;
        if (plcbrfcw<min_plcbrint || plcbrfcw>ma_plcbrint)
        {
            /* fix integrator - it went astray */
            /* calculate and set the initial value of the integrator
15    based upon lineFrequency measured by mtrsamp*/
            plcbrint.high = (unsigned long) (ldexp (lineFrequency*
            (fs1004.plc.baudratePrescale+1)/SYSTEM_CLOCK, 34)) ;
            plcbrint.low=0;
            plcbrfcw=plcbrint.high;
20    baudInitCounter= INIT_BAUD_INIT_COUNTER;

            baudInitFlag=FALSE;
        }
    }

25    fs1004.plc.baudrateLo=(unsigned short) (plcbrfcw) ;
    fs1004.plc.baudrateHi=(unsigned char) (plcbrfcw>>16) ;

    if (baudInitCounter>0)
30    {
        baudInitCounter--;
    }
    else if (baudInitFlag==FALSE)
    {
35    baudInitFlag = TRUE;
    }

}

40 void setMaxMin (void)
{
    max_plcbrint = (unsigned long) (ldexp (lineFrequency*1.005*
    (fs1004.plc.baudratePrescale+1)/SYSTEM_CLOCK,34)) ;
    min_plcbrint = (unsigned long) (ldexp(lineFrequency*0.995*
45    (fs1004.plc.baudratePrescale+1)/SYSTEM_CLOCK,34)) ;
}

void plcBaudHardwareReset (void)
{
50    unsigned long rtc1, rtc2;
    unsigned long plcbrfcw ;
    /*int timeout ; */
    fs1004.plc.baudratePrescale=23;          /* divide by prescale +
1 */
55

    /* calculate and set the initial value of the integrator based upon
lineFrequency measure by mtrsamp*/
    plcbrint.high = (unsigned long) (ldexp(lineFrequency*
    (fs1004.plc.baudratePrescale+1)/SYSTEM_CLOCK,34)) ;
60    plcbrint.low=0

```

```

    plcbrfcw=plcbrint.high;
    baudInitCounter= INIT_BAUD_INIT_COUNTER;
    baudInitFlag = FALSE;
    setMaxMin() ;
5    /* clear the baudrate NCO */
    fs1004.plc.baudrateNCO = 0L;

    rtc1=rtc2= RTC;
    /* wait for powerline zero crossing */
10    while (!fs1004.power.captureDone && ((rtc2-rtc1) &RTC_MASK)<3)
        {
            rtc2 = RTC;
        }
    /* initialize the baudrate NCO at the zero crossing to achieve
15    initial phase lock */
    /* the NCO starts only after a byte is written to baudrateHi */
    fs1004.plc.baudrateLo=(unsigned short) ((plcbrfcw) &0xFFFF) ;
    fs1004.plc.baudrateHi=(unsigned char) ((plcbrfcw) >>16) ;
    fs1004.power.captureDone=0;
20    baudpll_timer = 0 ;
}

```

25

BAUDPLL.H

```

30    /*
    version 2.00 for flash 2 9/8/2000 sas
    */
    void plcBaudSecond(void) ;
    void plcBaudHardwareReset (void) ;
35    void plcBaudSusecond (void) ;

```

CLAIMS

We claim:

1. A system comprising:
a master data clock source;
5 one or more transponders; and
a plurality of remote power line transceivers;
wherein all of said plurality of transceivers are connected to a common alternating
current power distribution grid; and
wherein each of said plurality of transceivers has a location and is operable to
10 monitor a voltage waveform of a power line prevailing at said location.

1/227

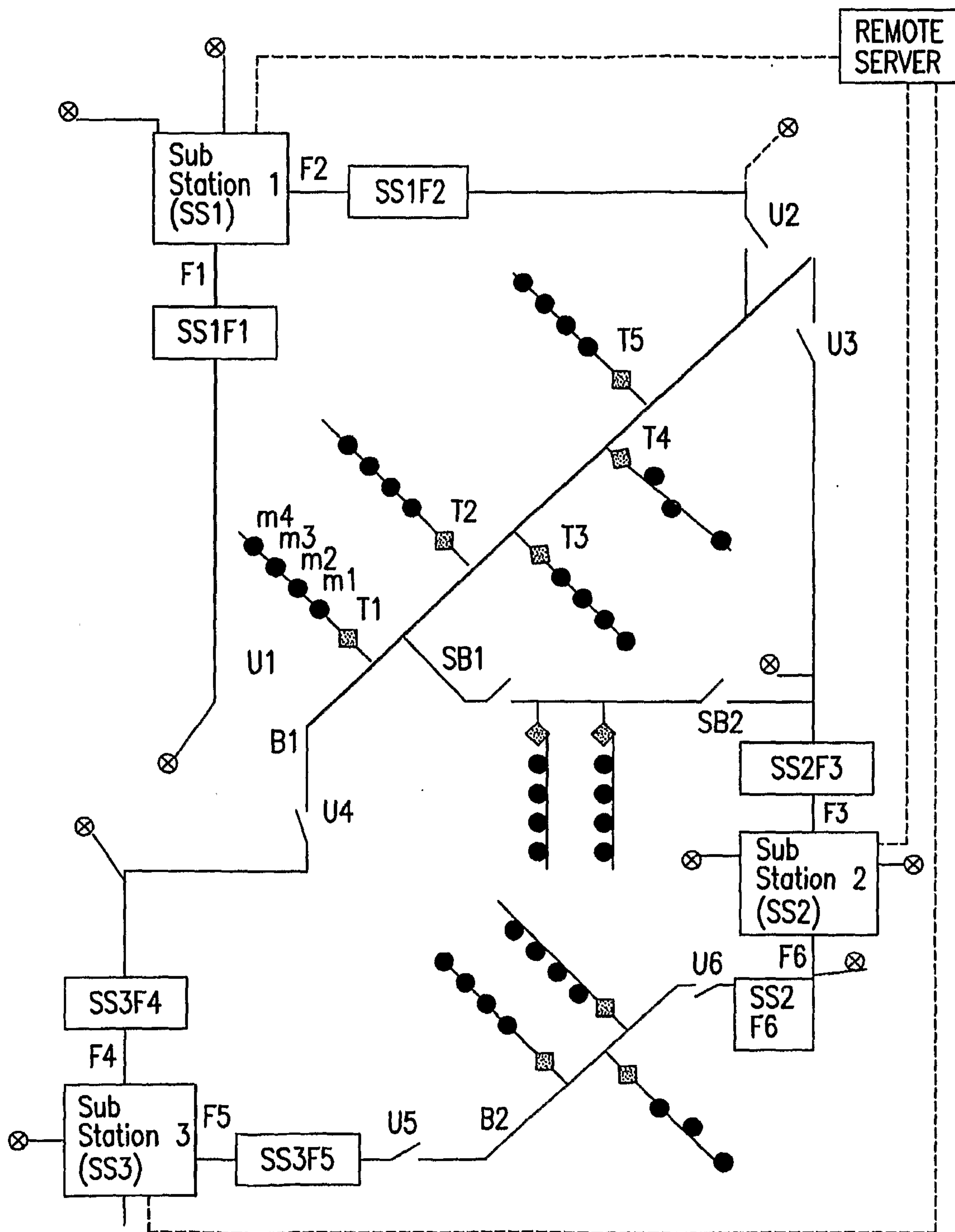


FIG. 1

2/227

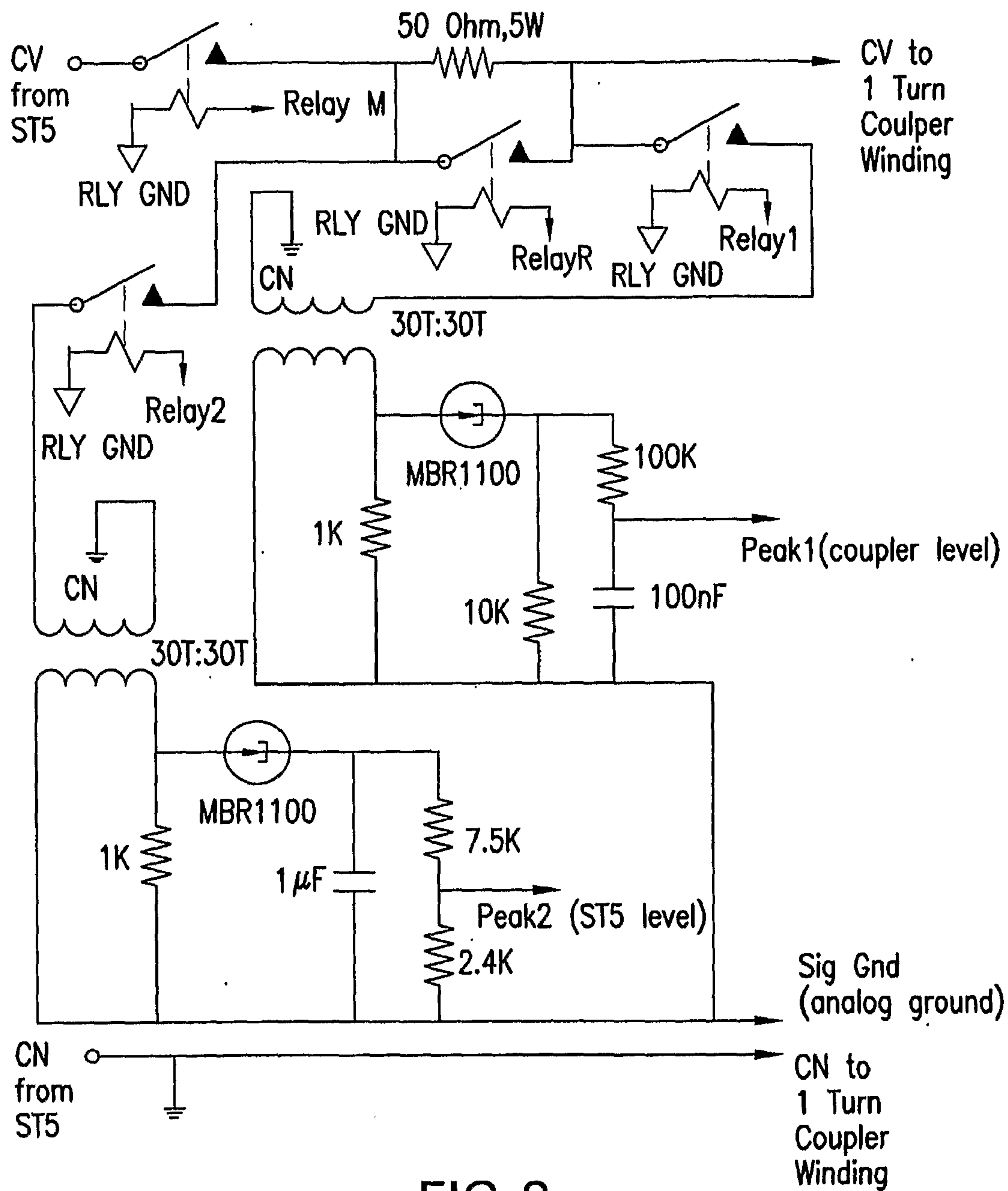


FIG.2

3/227

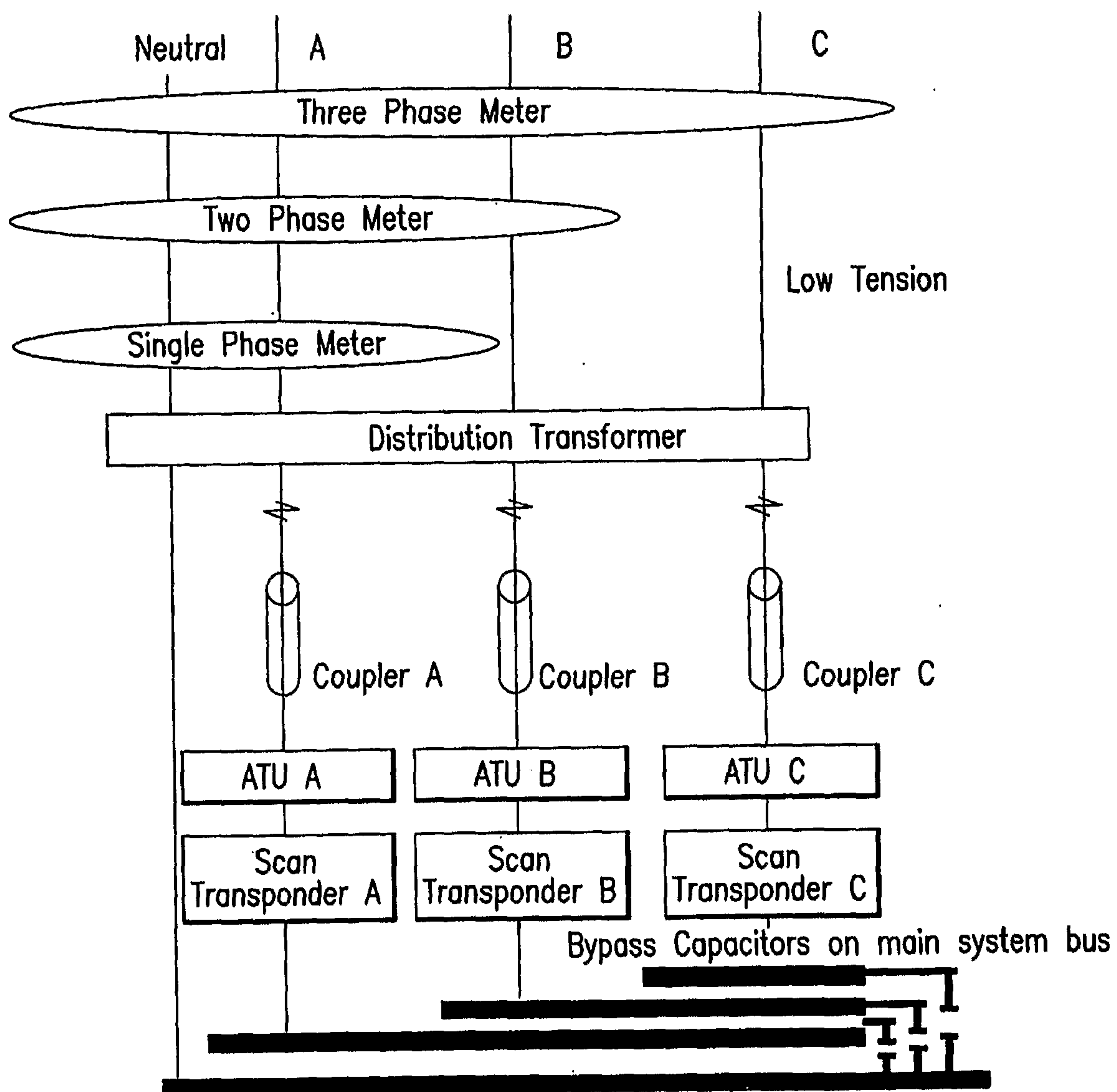


FIG.3

4/227

0kHz-25kHz BAND

Number of Taps	65
Stop Band Attenuation	71.23dB
Pass Band Upper Frequency	25kHz
Stop Band Lower Frequency	35kHz
Sampled in	60*4096
Sampled Out	30*2048

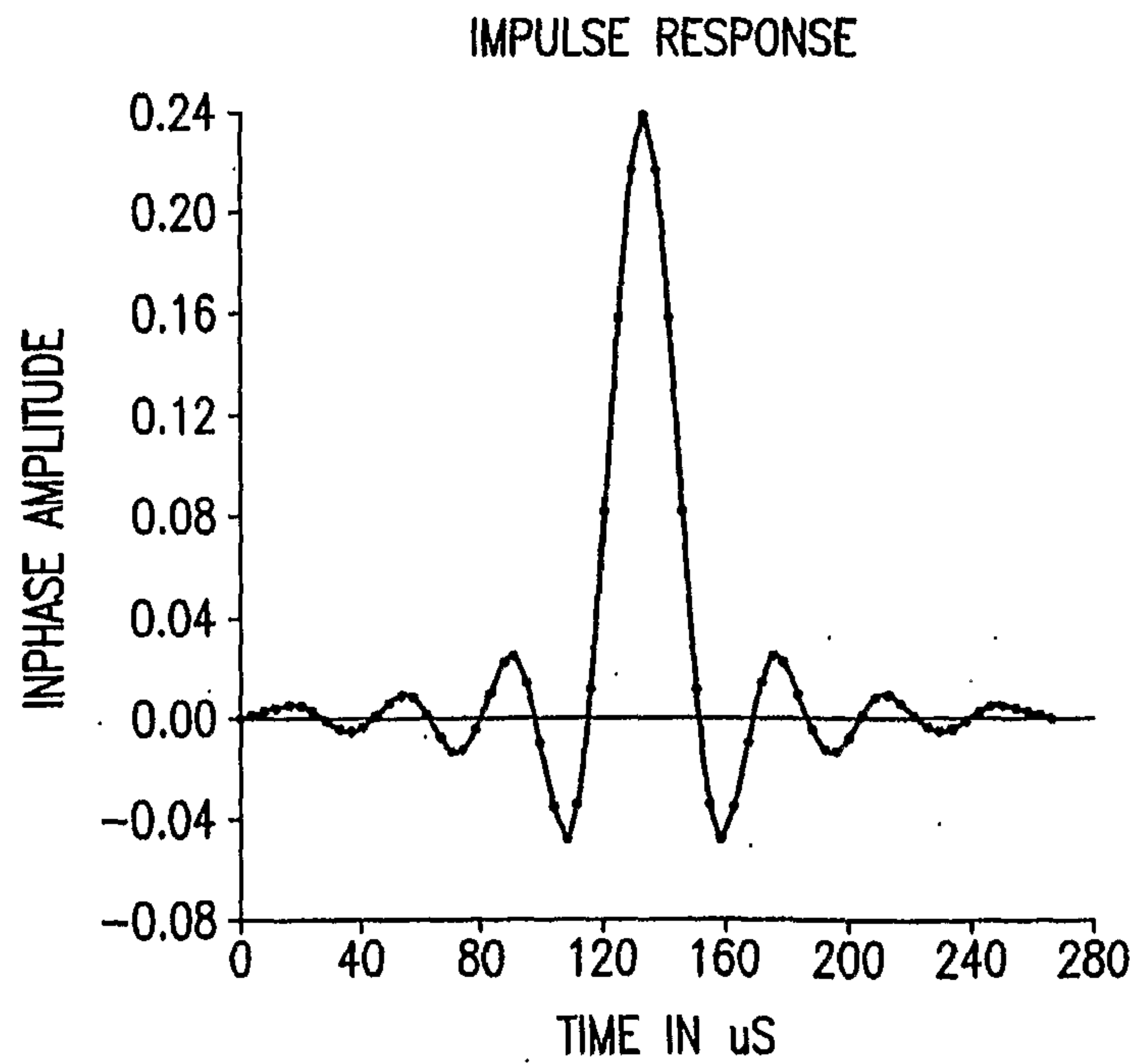
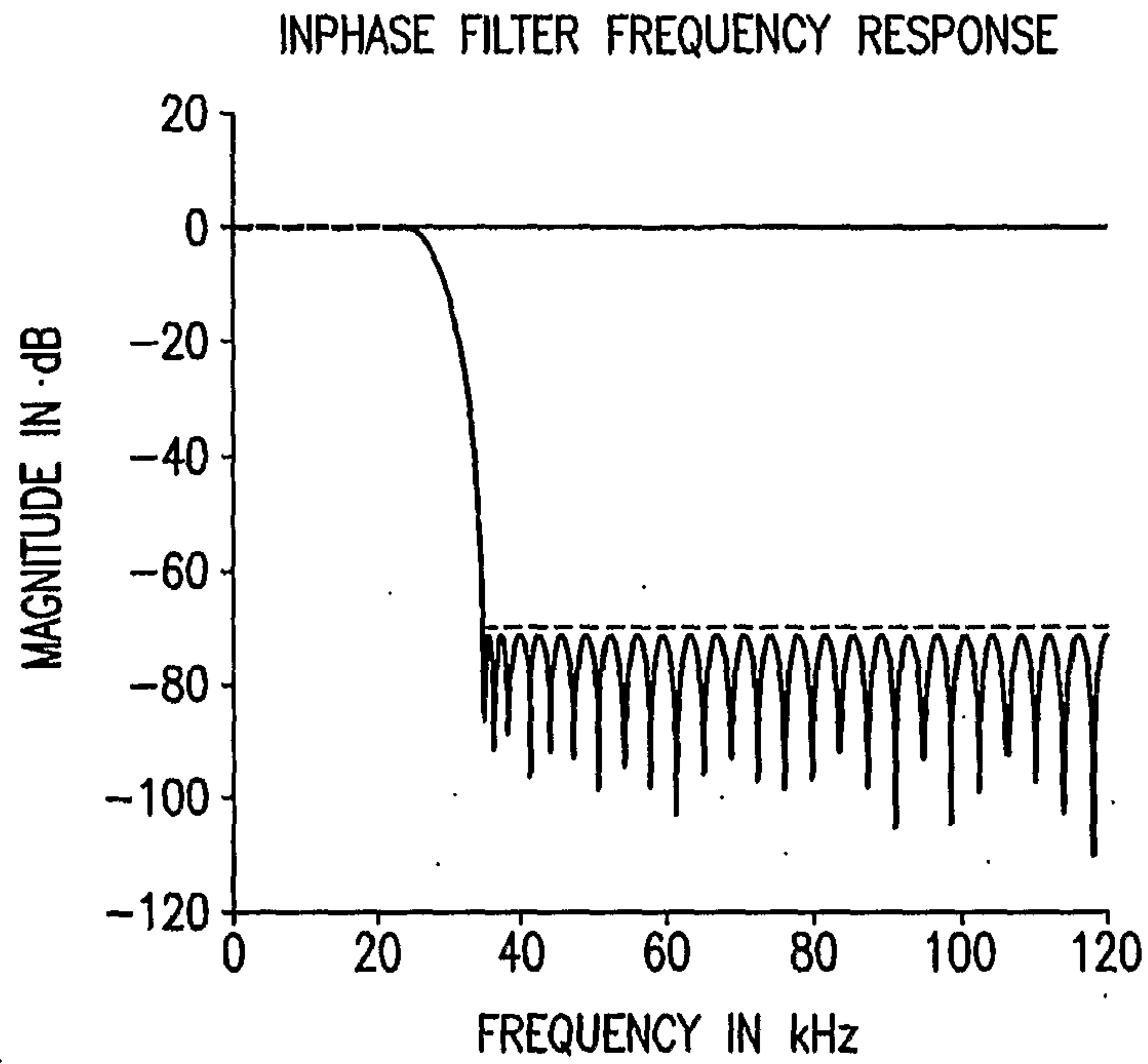
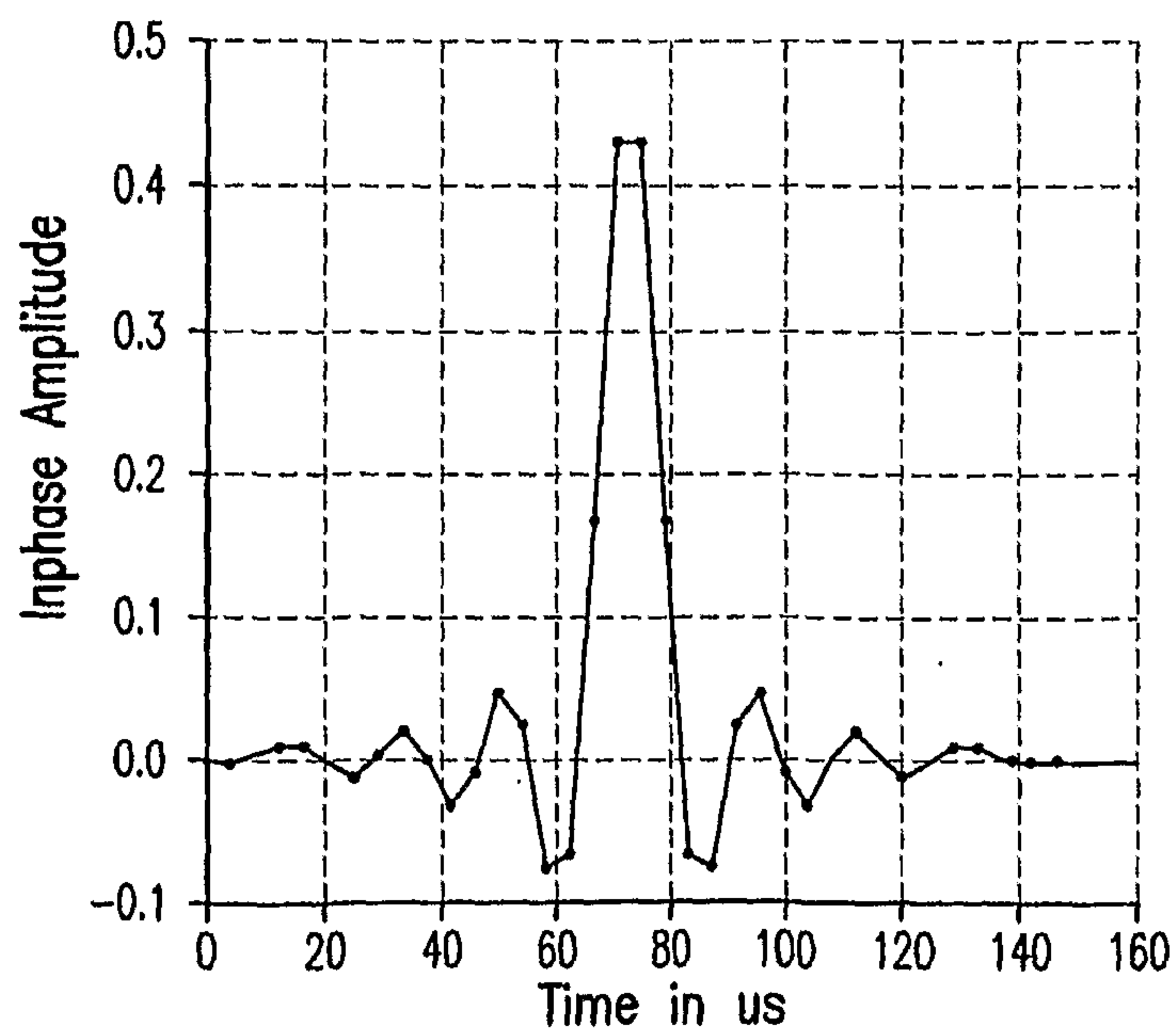
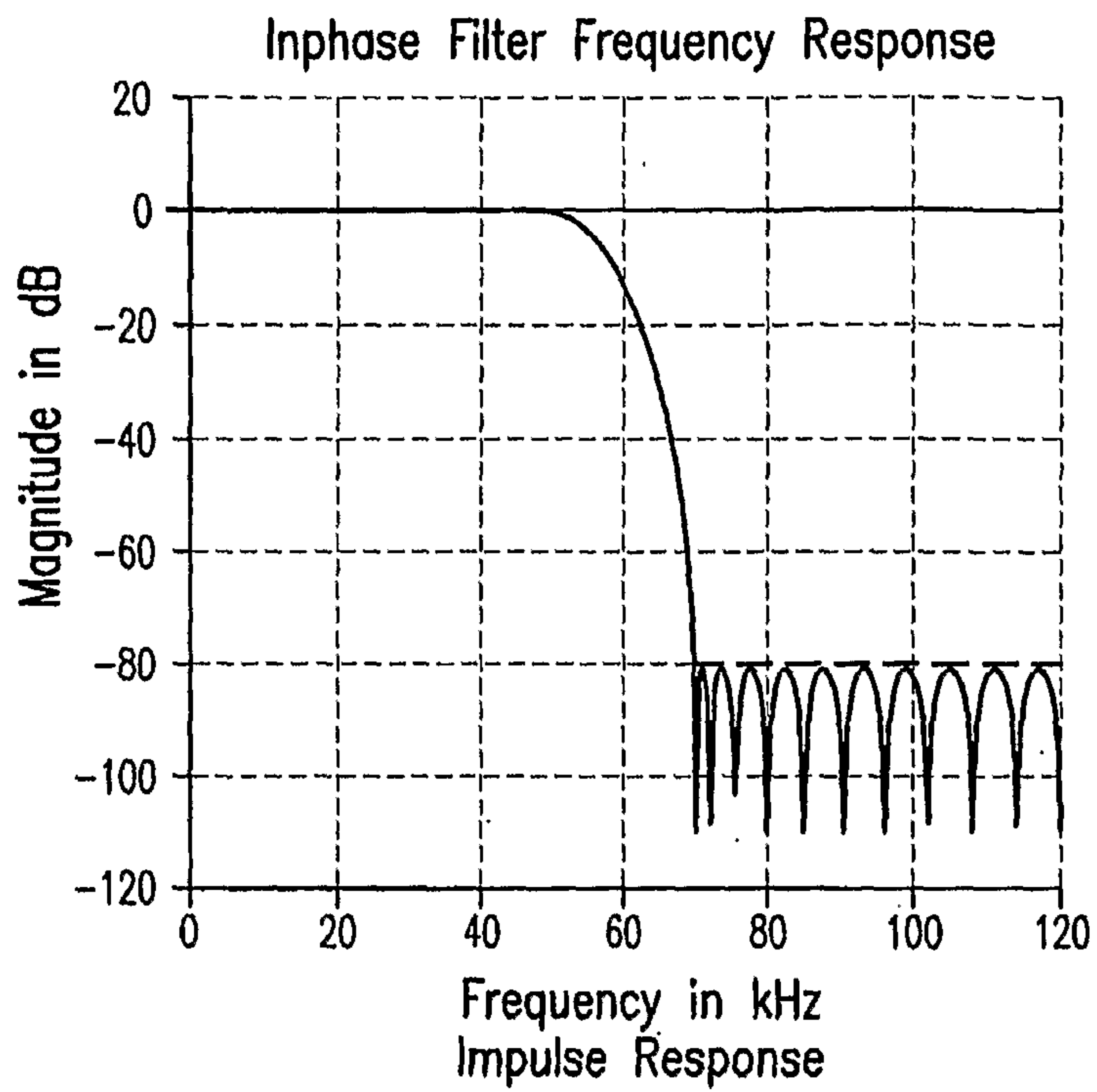


FIG.4

5/227

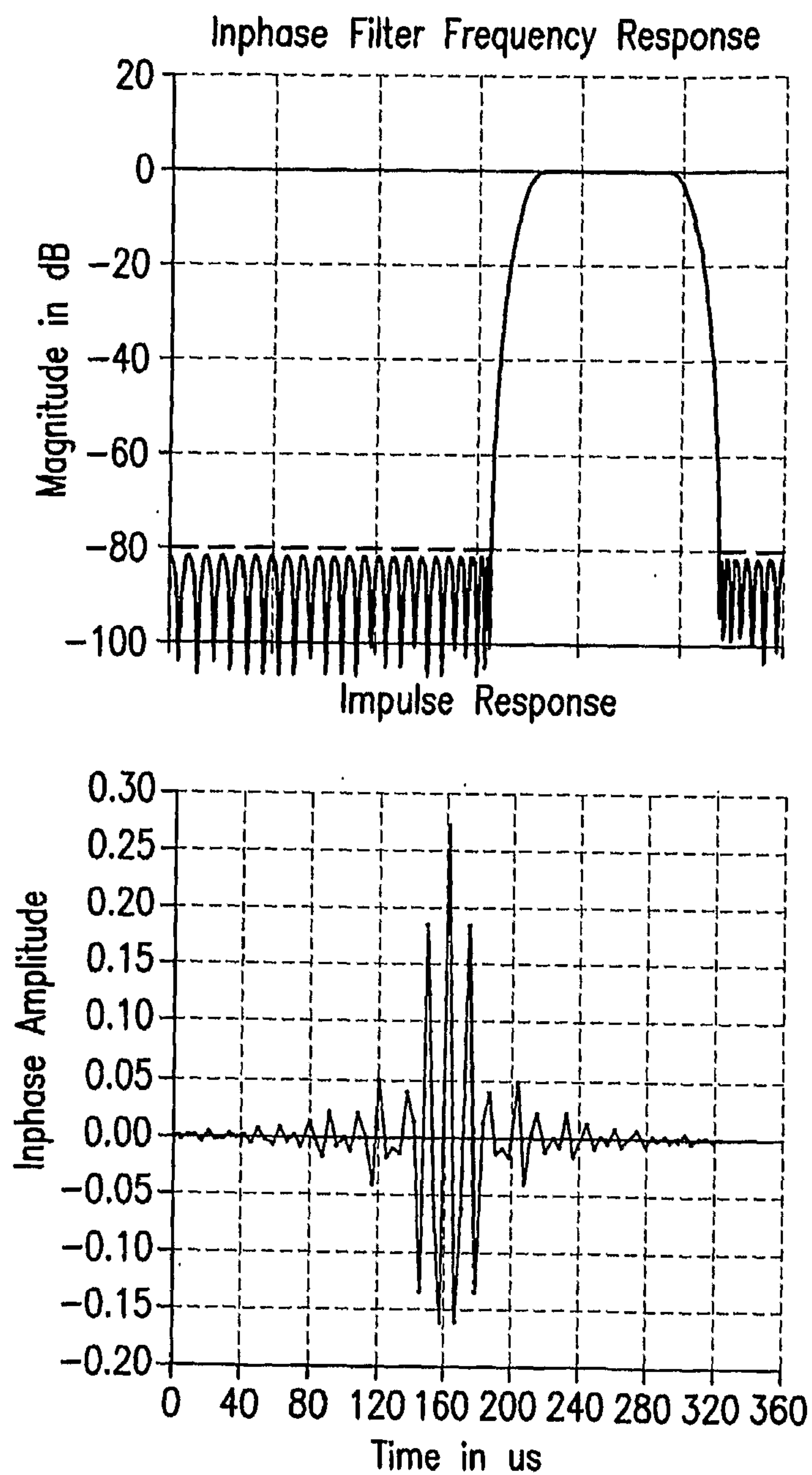
25kHz-50kHz Band

Number of Traps	36
Stop Band Attenuation	81.15dB
Pass Band Upper Frequency	50kHz
Stop Band Lower Frequency	70kHz
Sampled in	60*4096
Sampled Out	60*2048

**FIG.5**

6/227

Number of Taps	79
Stop Band Attenuation	81.15dB
Pass Band Lower Freq	25kHz
Pass Band Upper Freq	45kHz

**FIG.6**

7/227

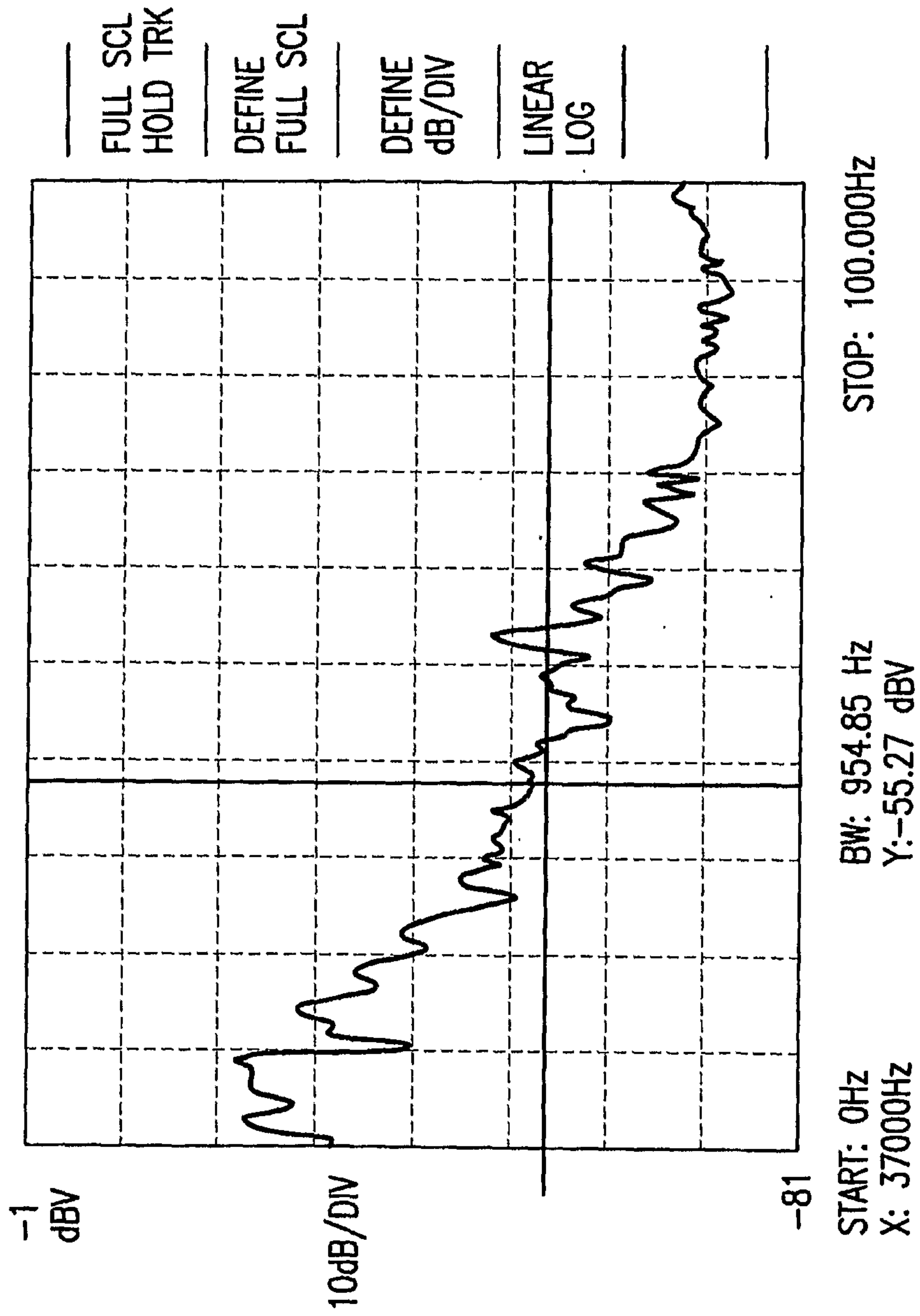


FIG. 7

8/227

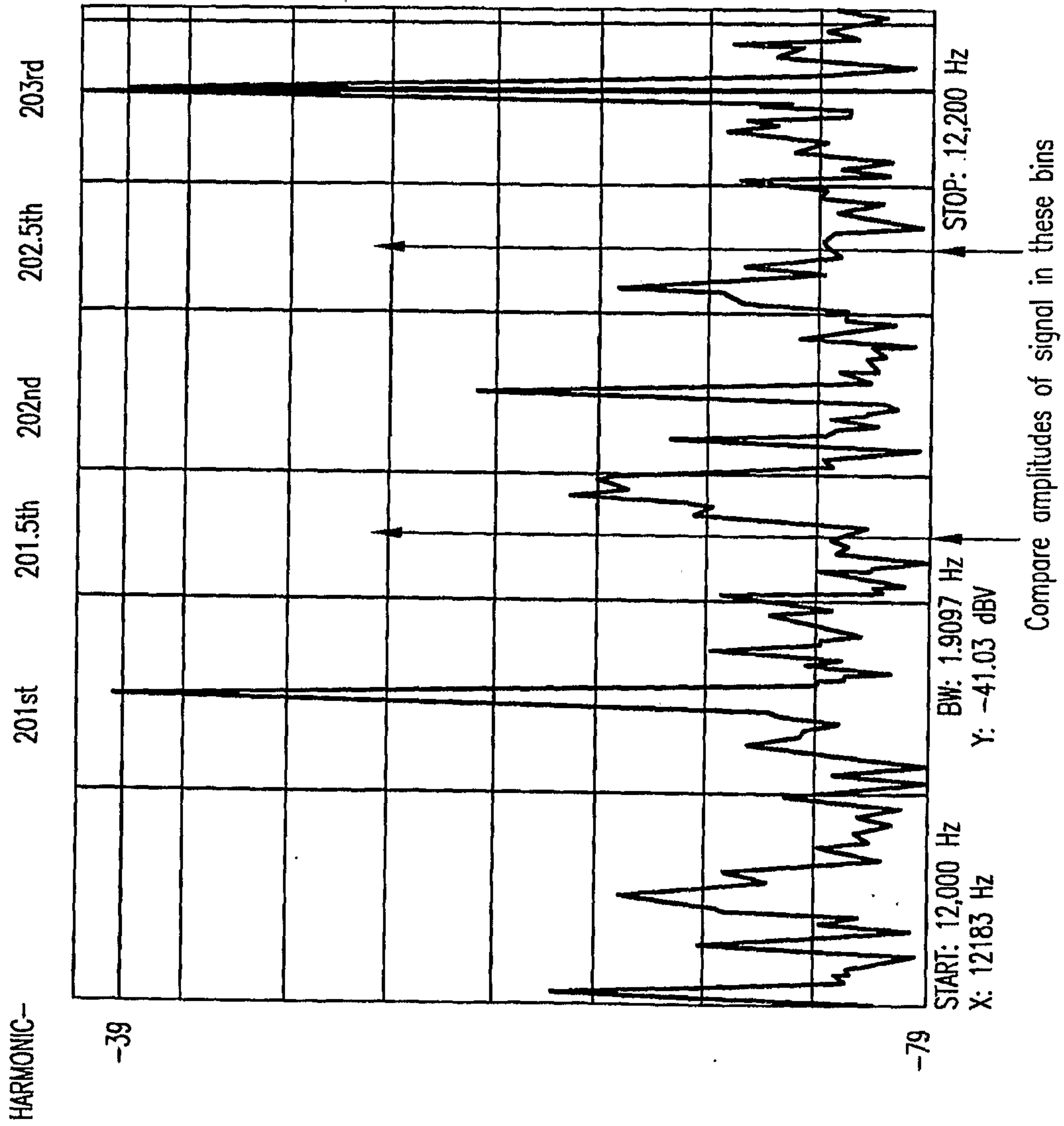


FIG. 8

9/227

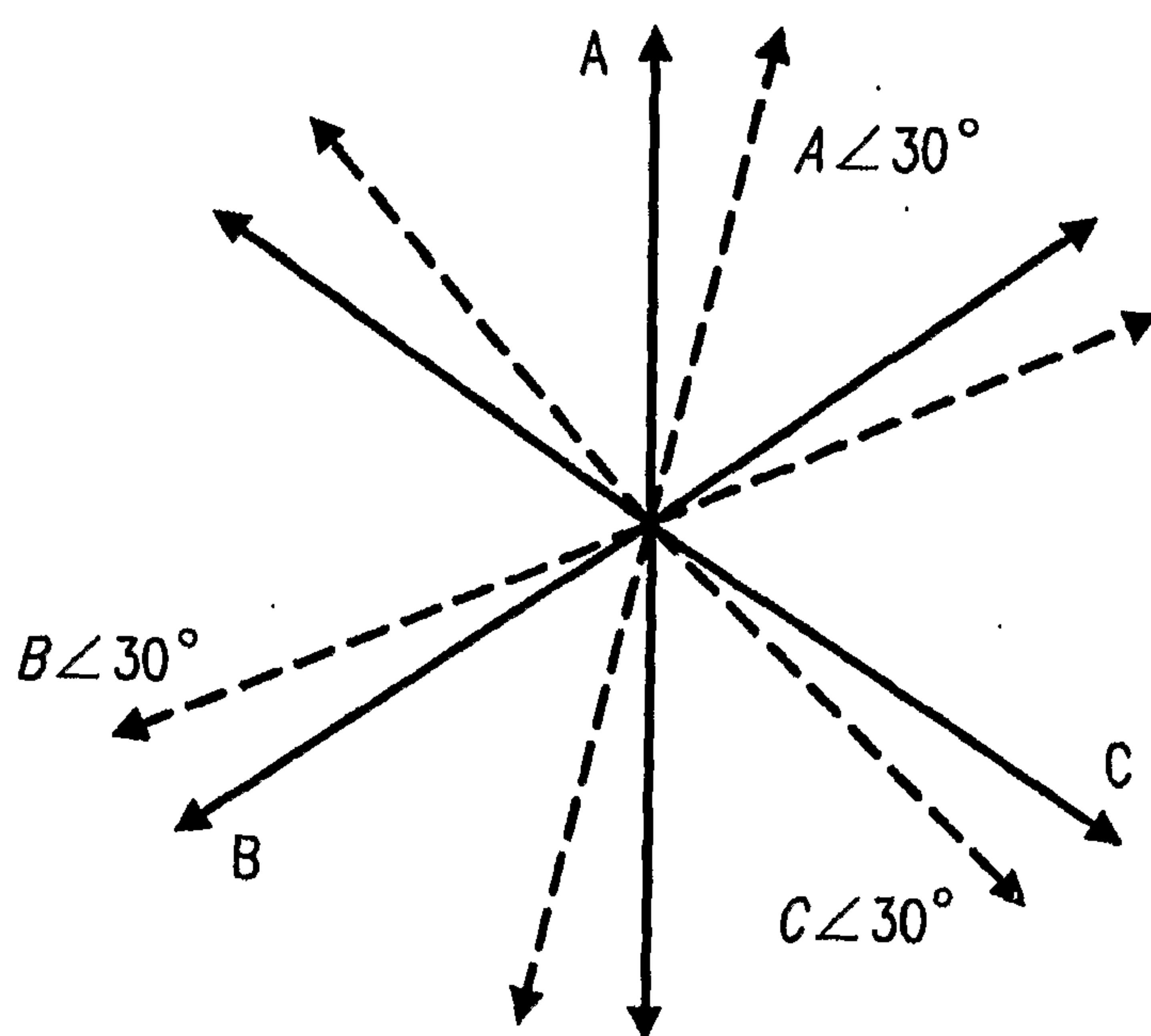


FIG.9

10/227

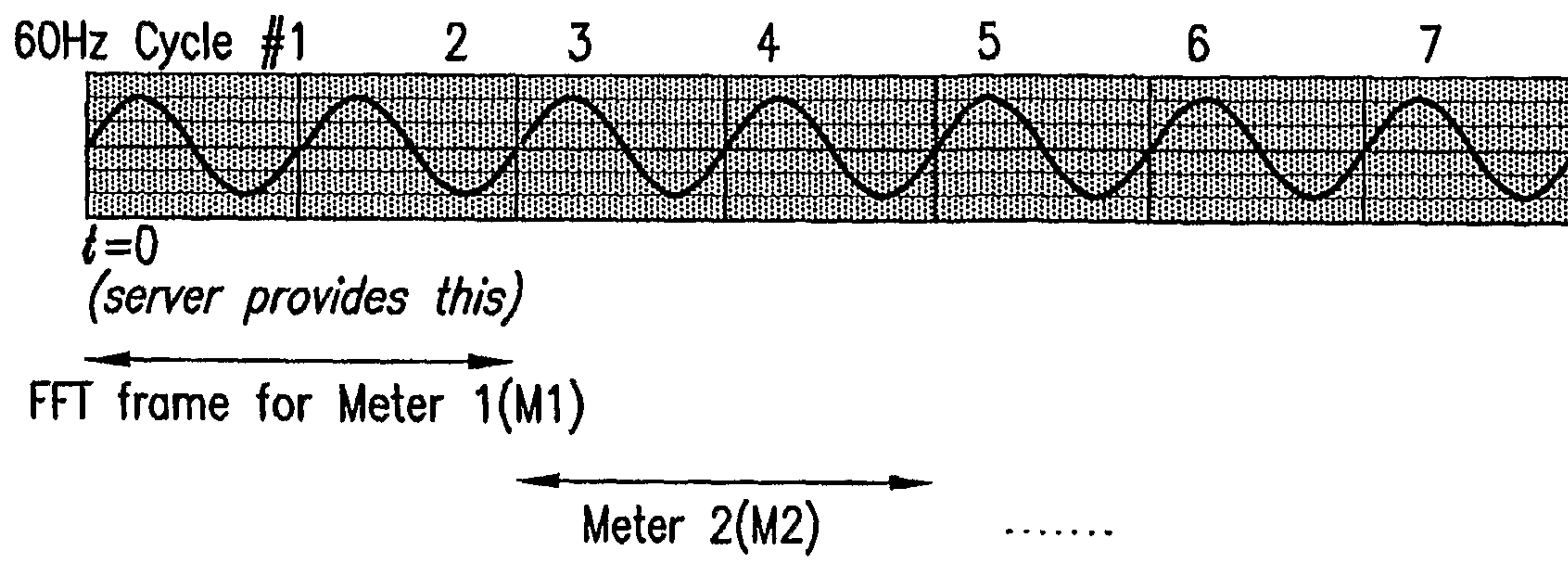


FIG.10

11/227

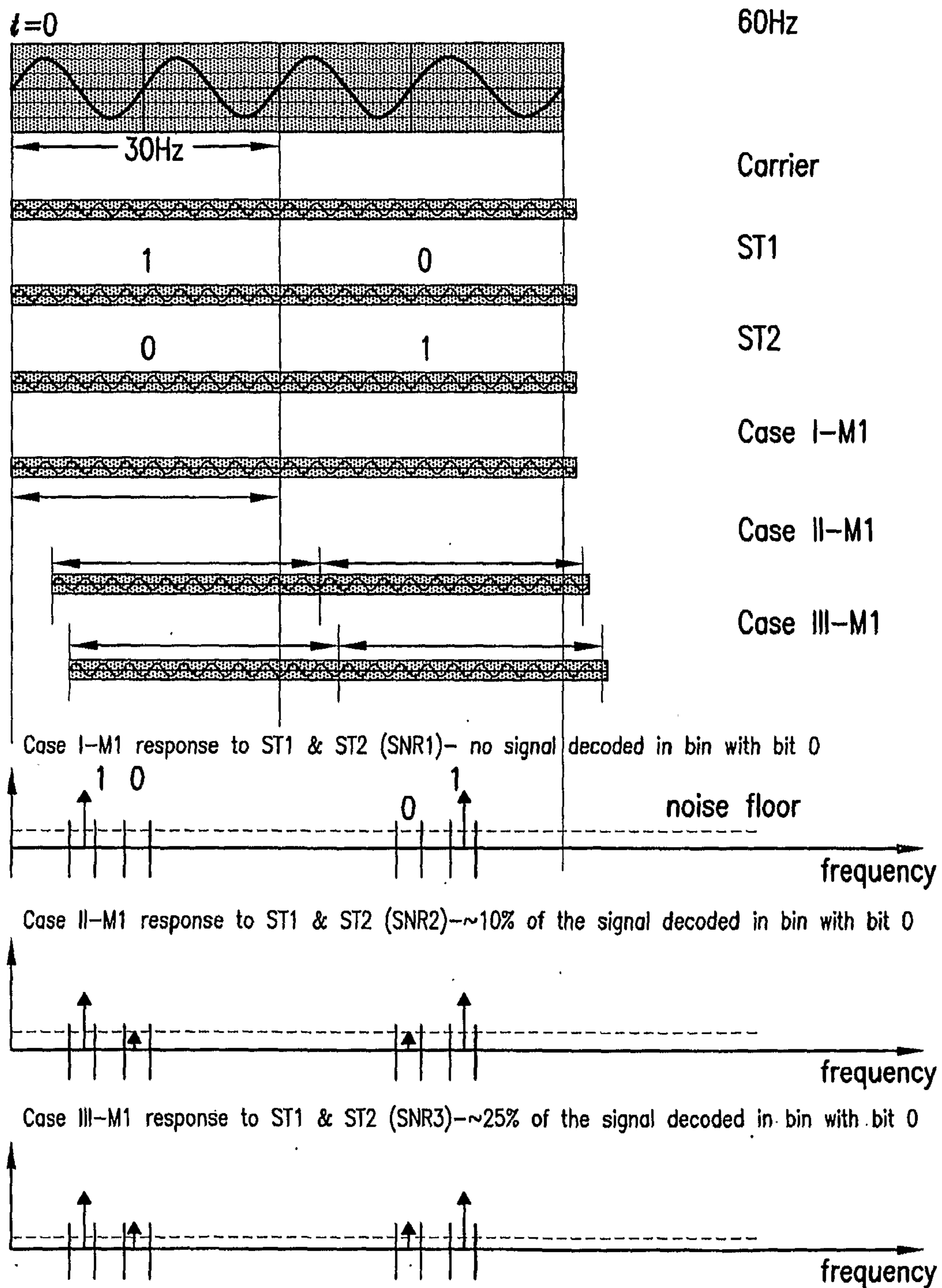


FIG. 11

12/227

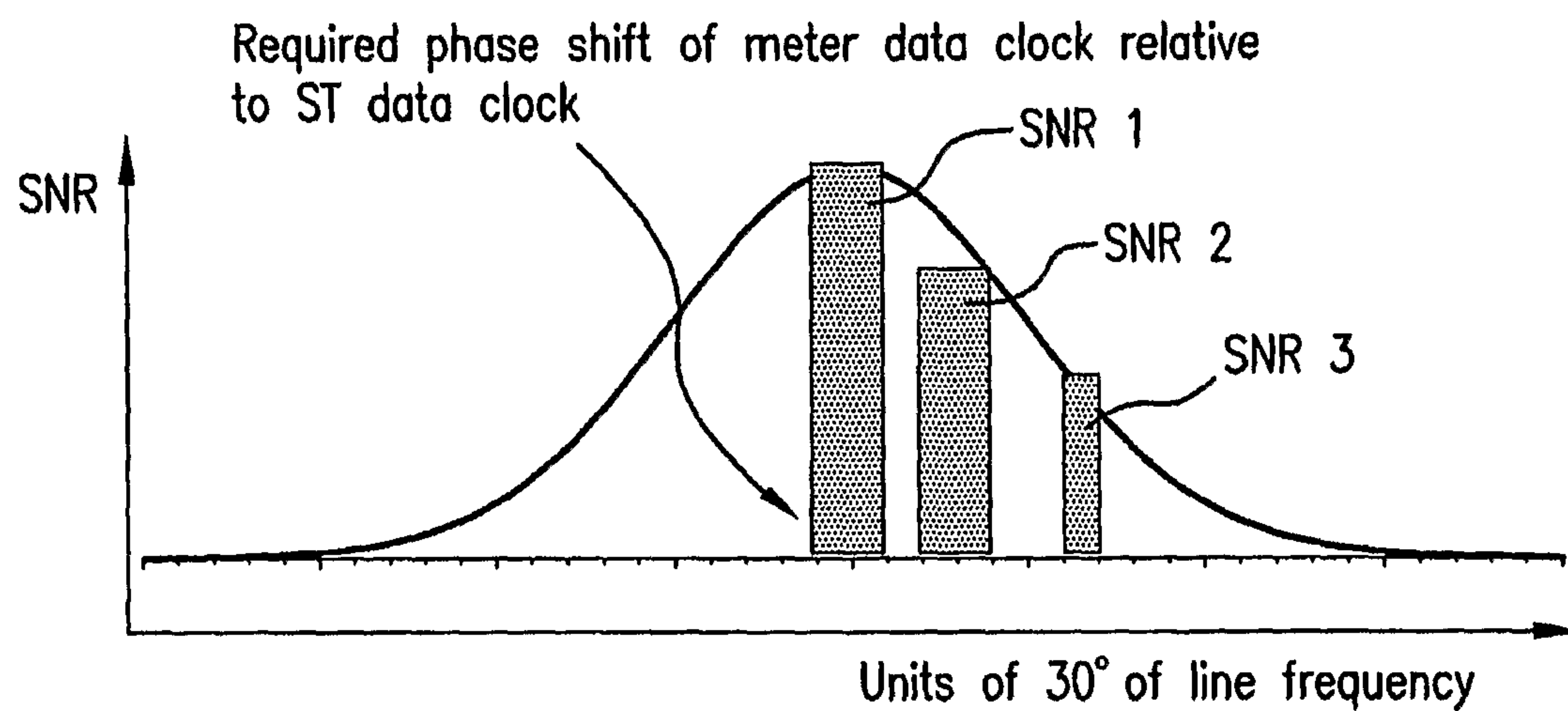


FIG.12

13/227

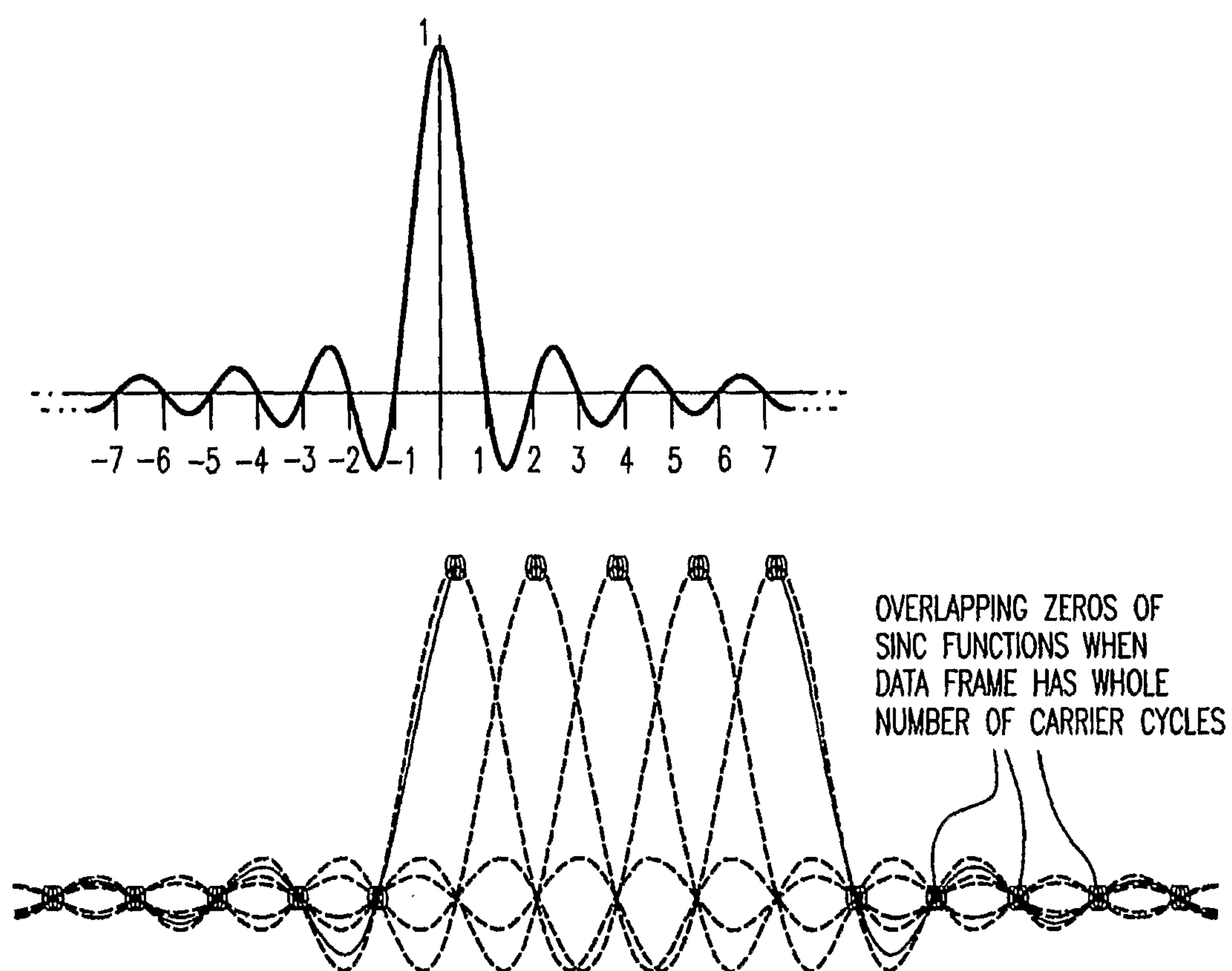


FIG.13

14/227

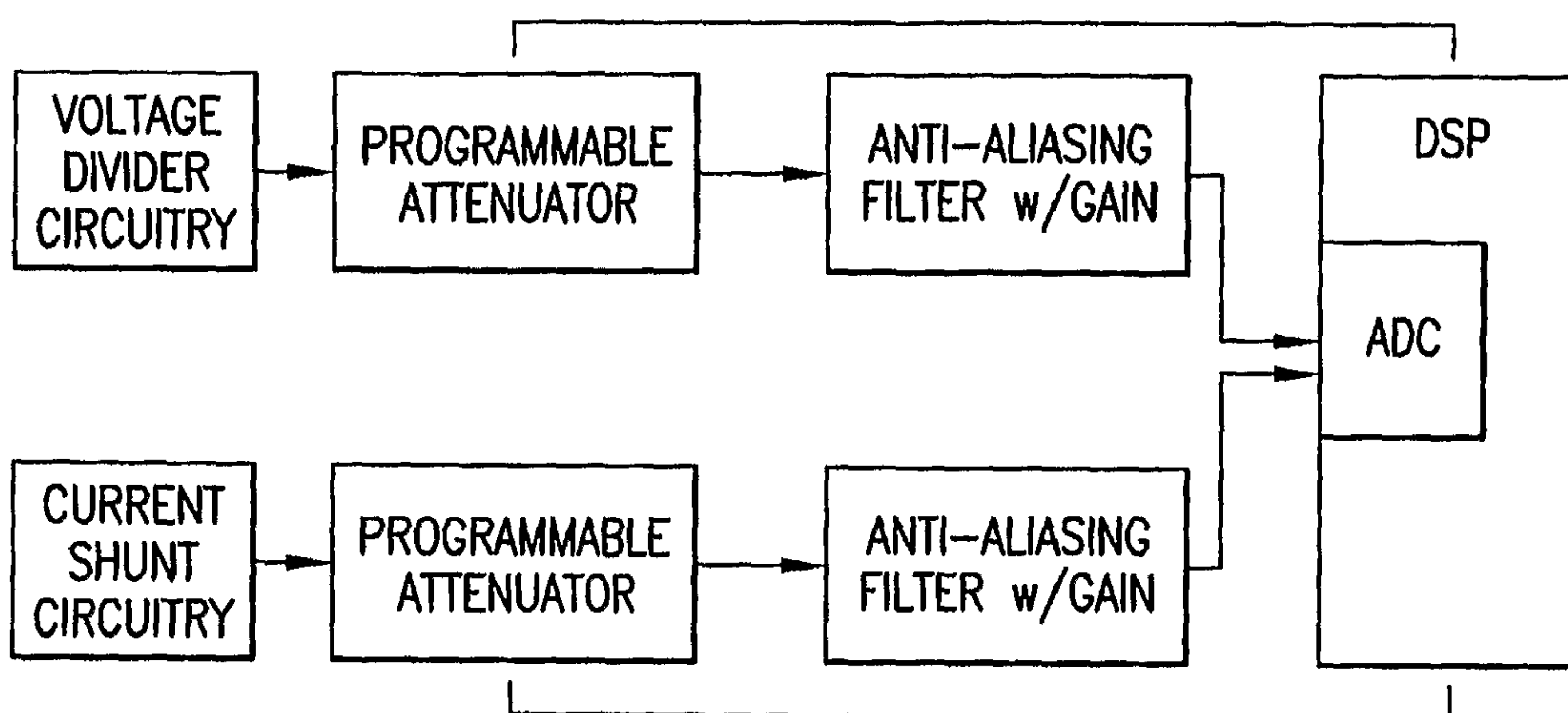


FIG. 14

15/227

Number of Taps	29
Stop Band Attenuation	80.453dB
Pass band Upper Freq	3kHz
Stop band Lower Freq	12kHz

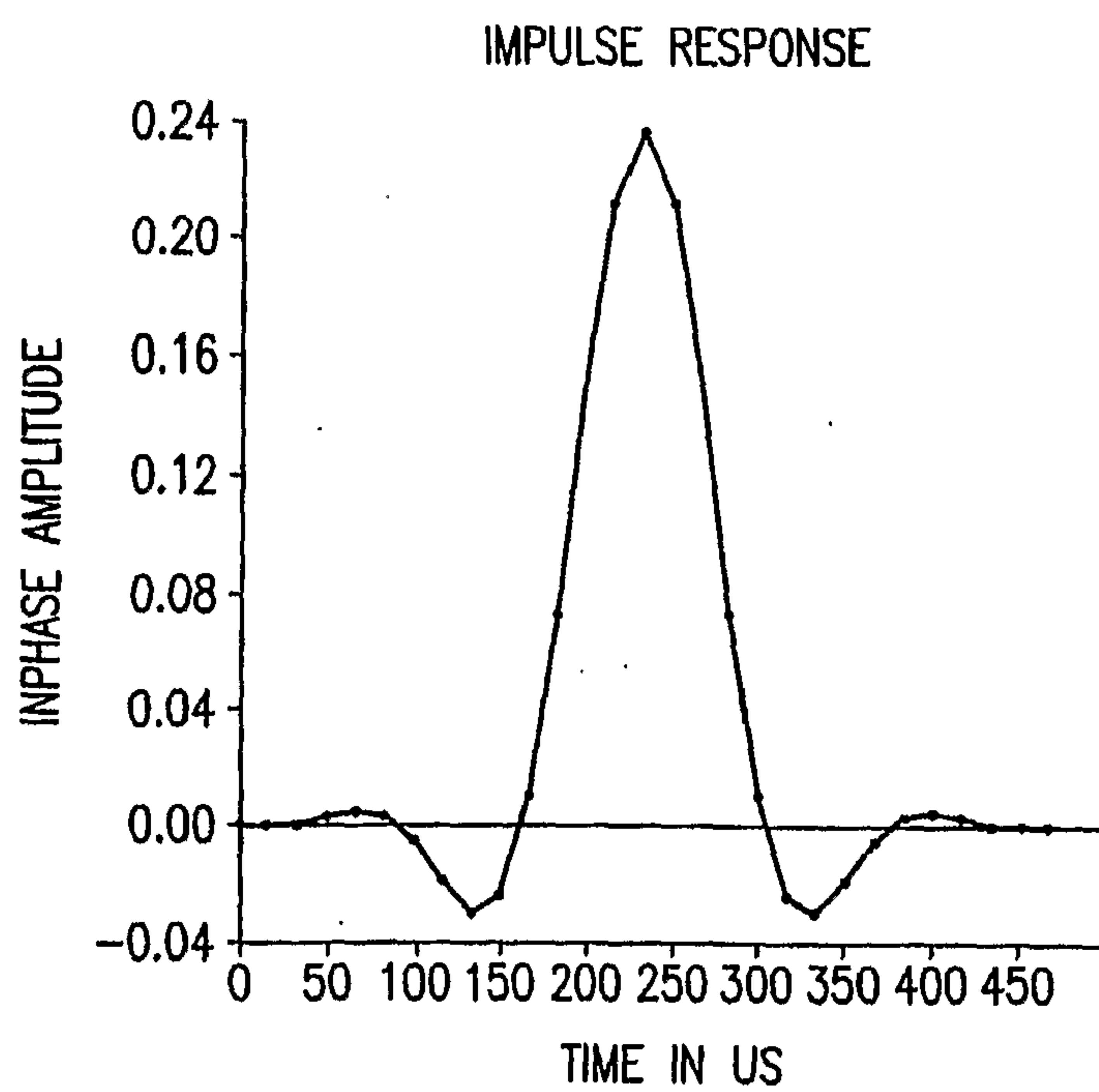
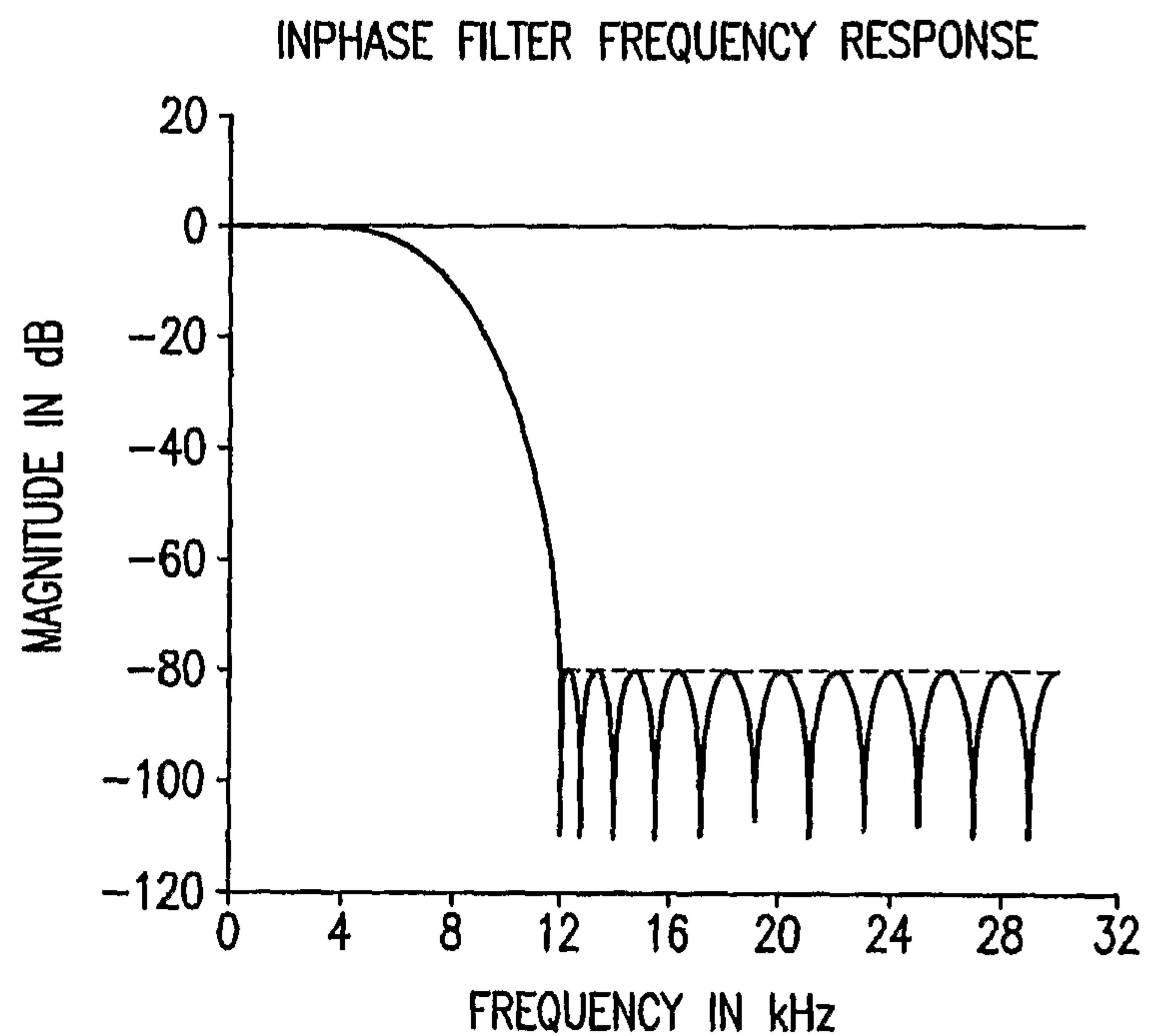


FIG. 15

16/227

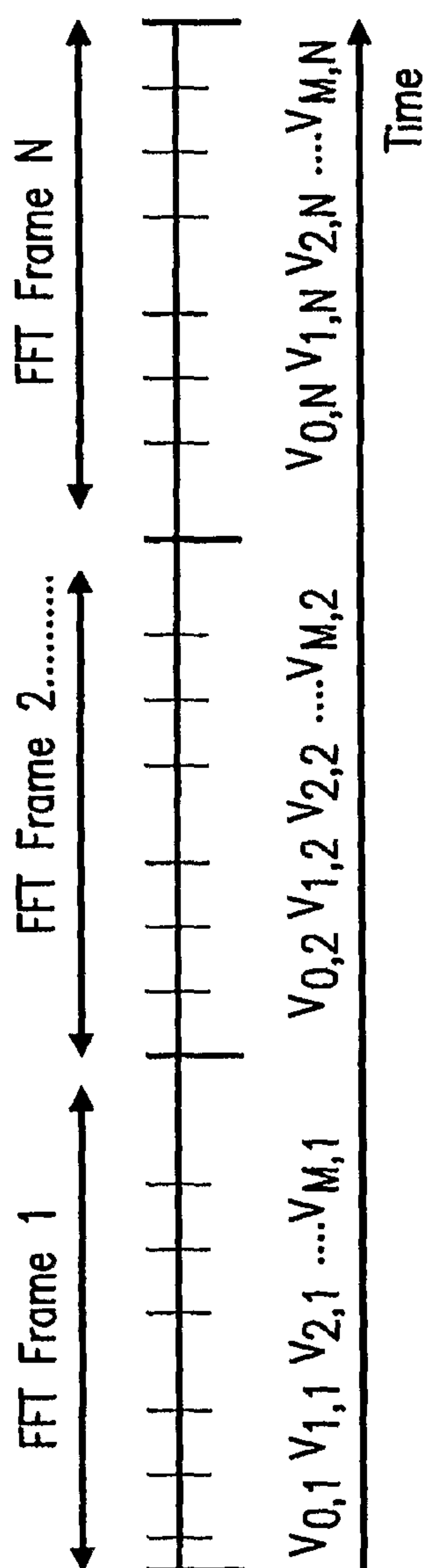


FIG.16

17/227

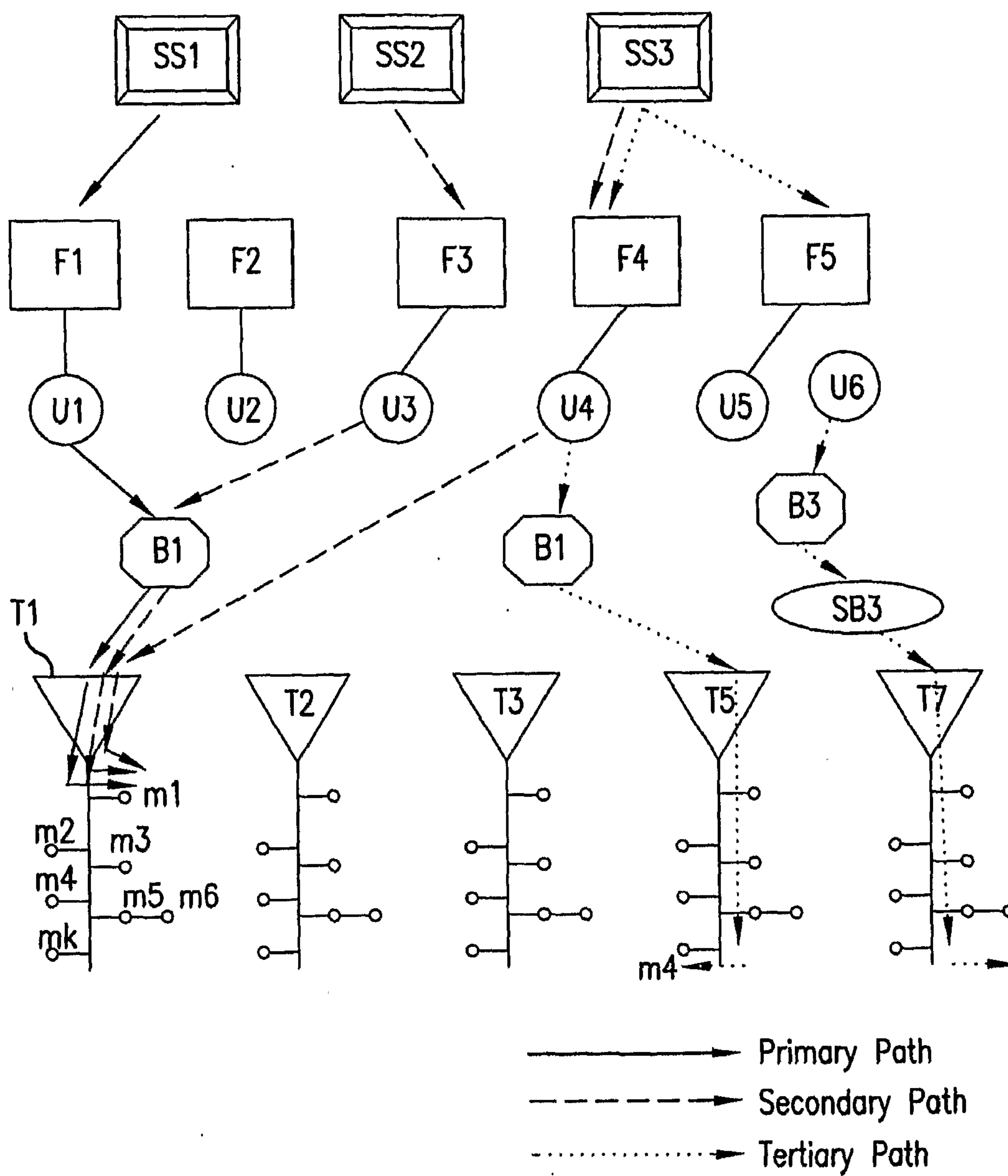


FIG.17

18/227

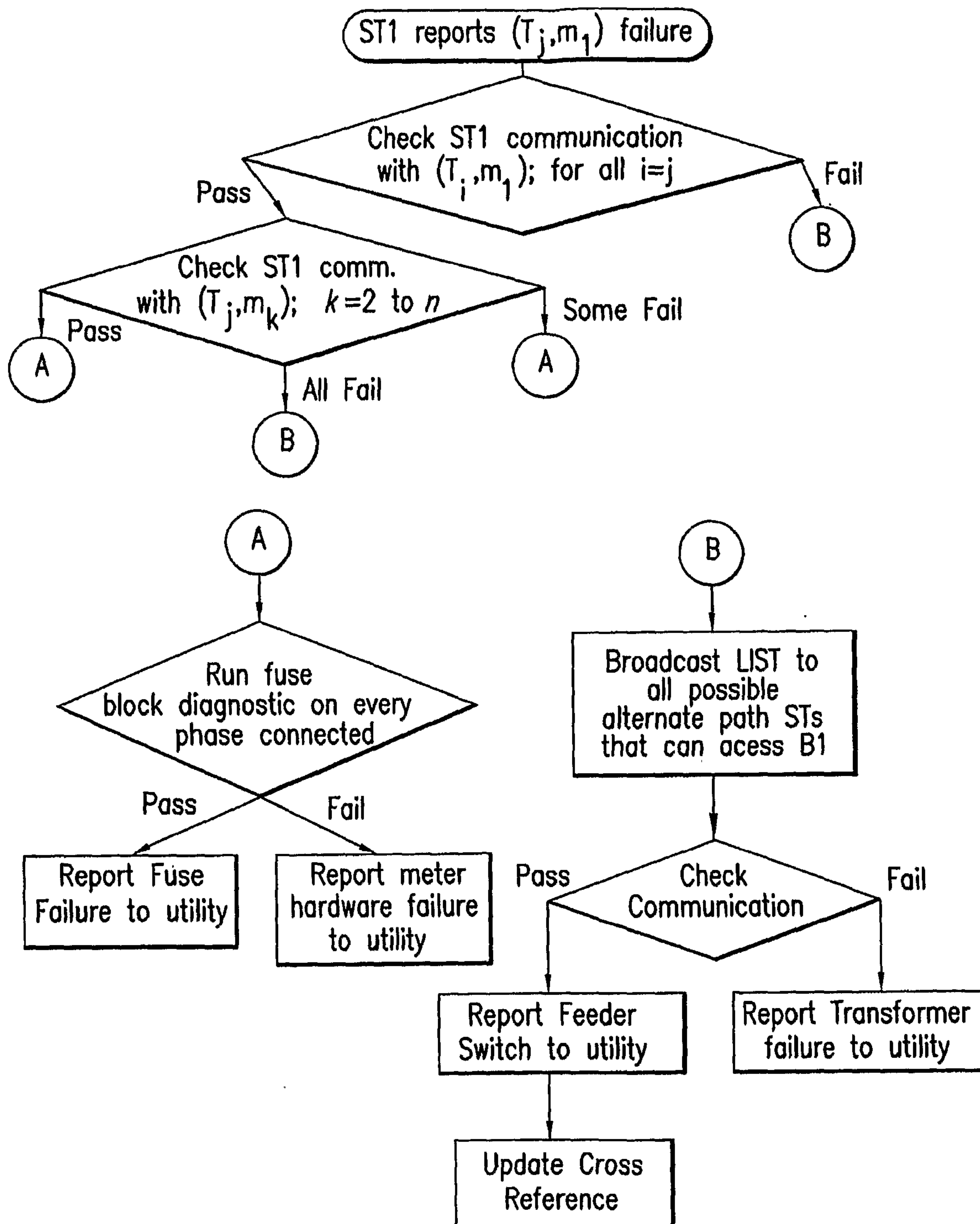


FIG.18

19/227

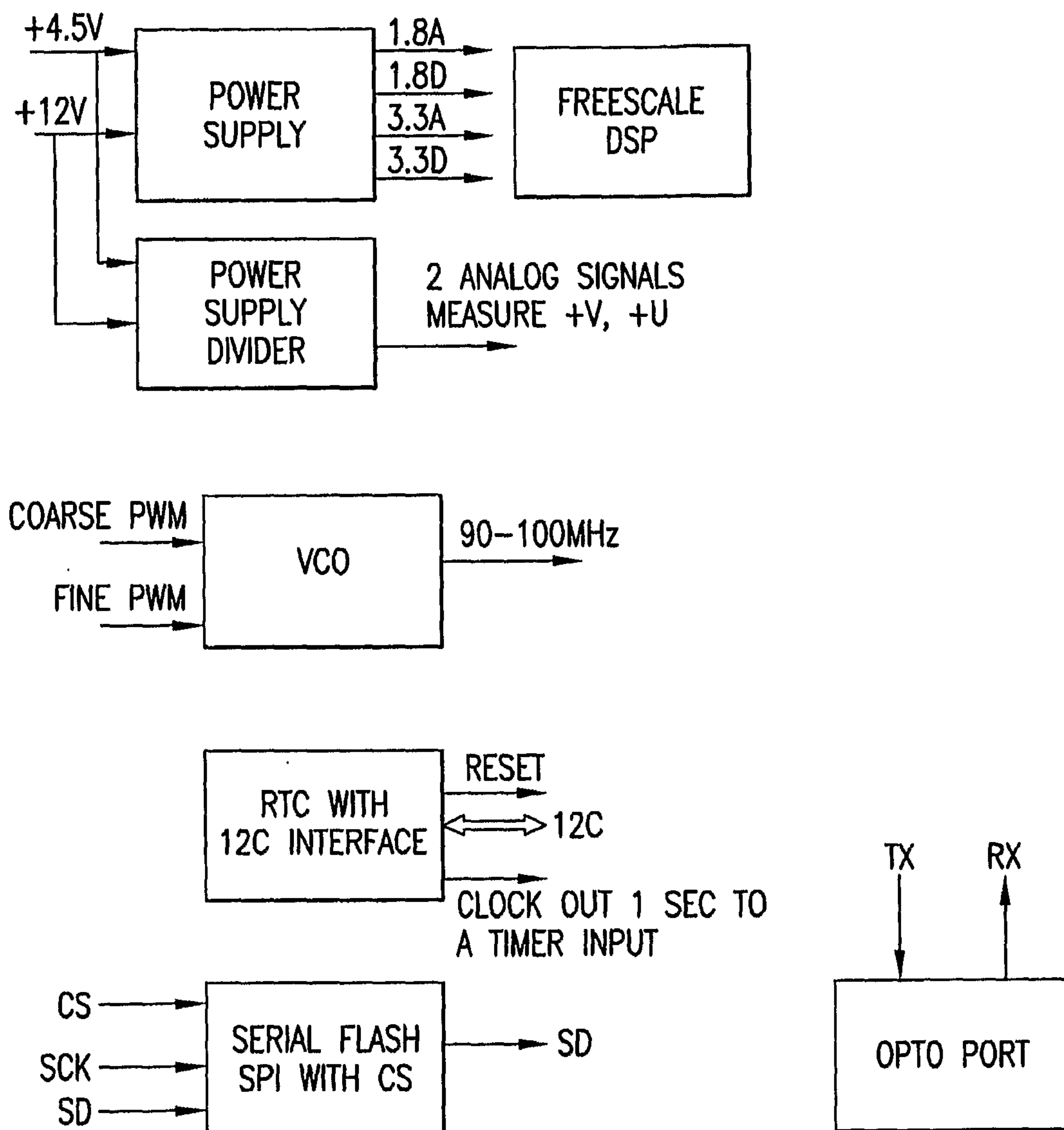
ID HARDWARE BLOCK DIAGRAM

FIG.19A

20/227

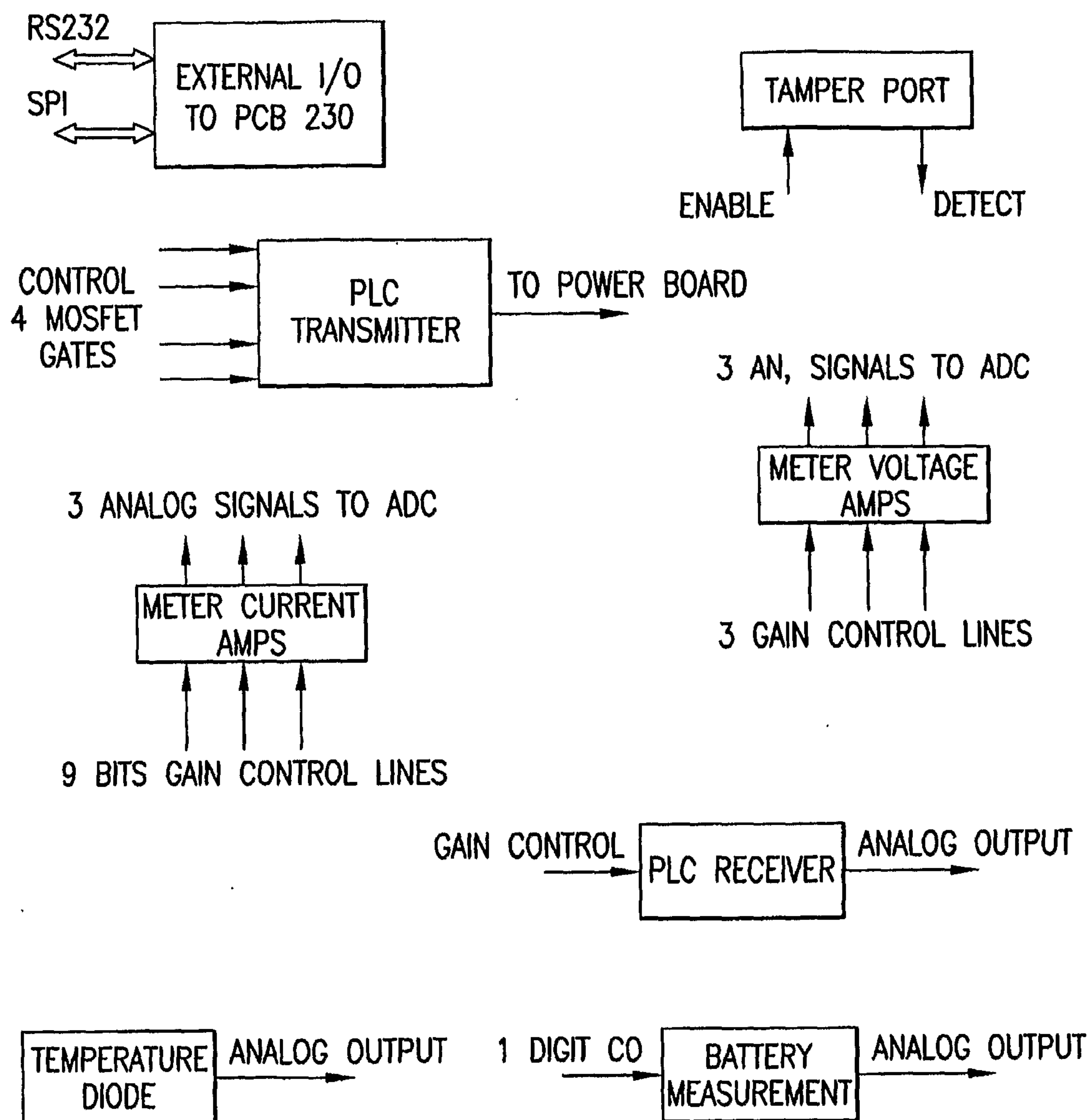


FIG.19B

21/227

FIG.20A-1	FIG.20A-2	FIG.20A-3
FIG.20A-4	FIG.20A-5	FIG.20A-6
FIG.20A-7		

FIG.20A

22/227

FIG. 20A-1

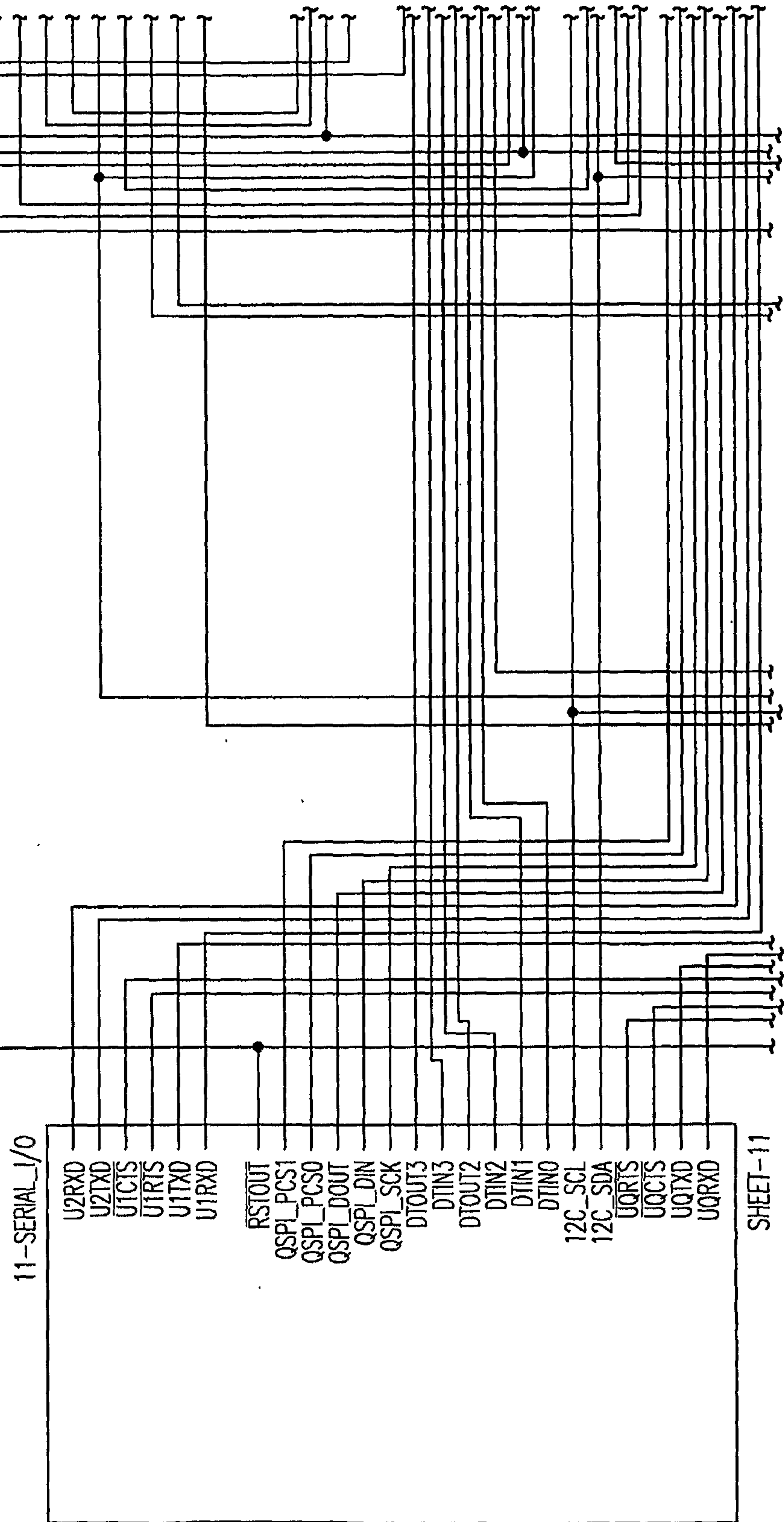
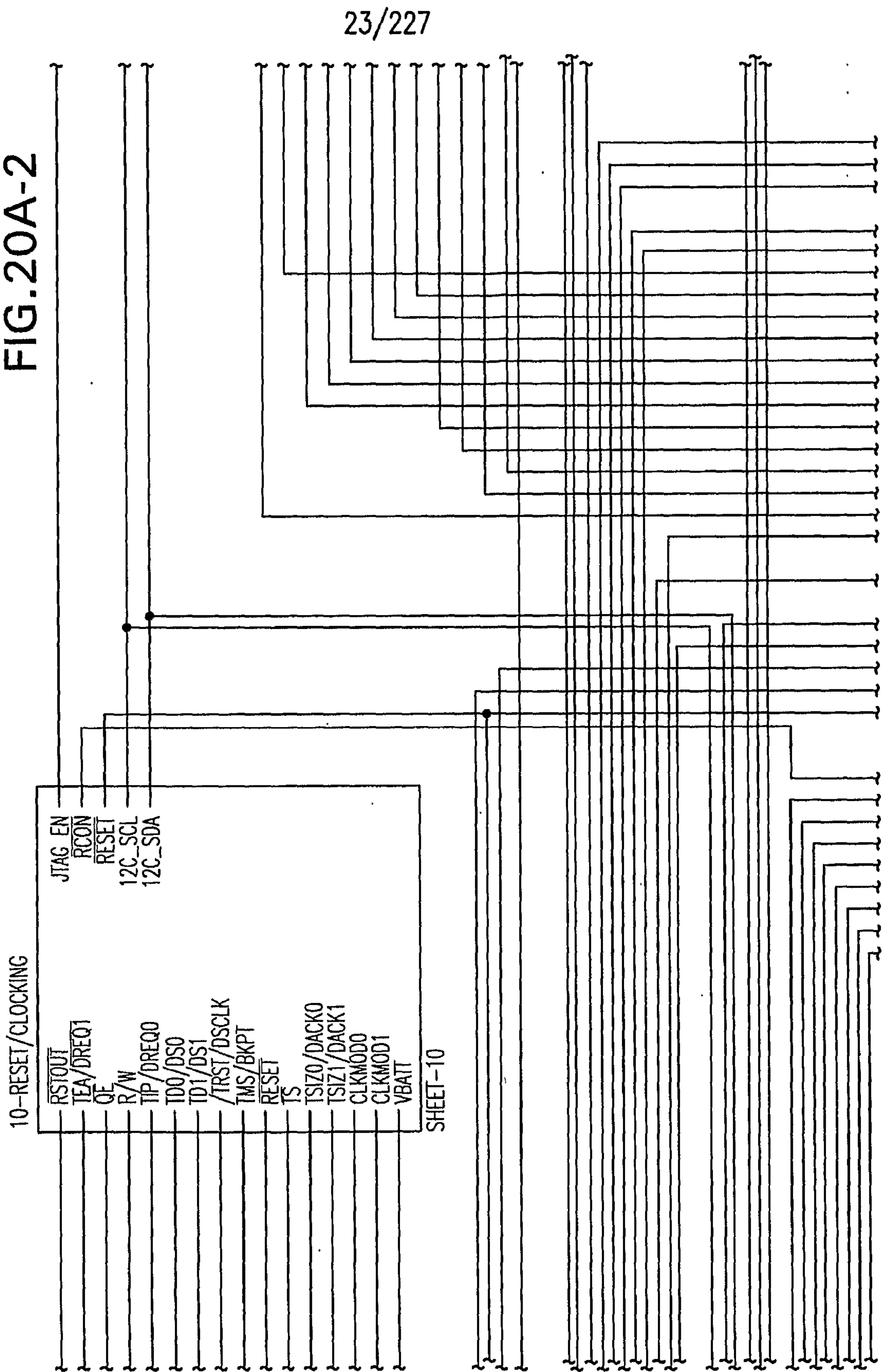


FIG. 20A-2



24/227

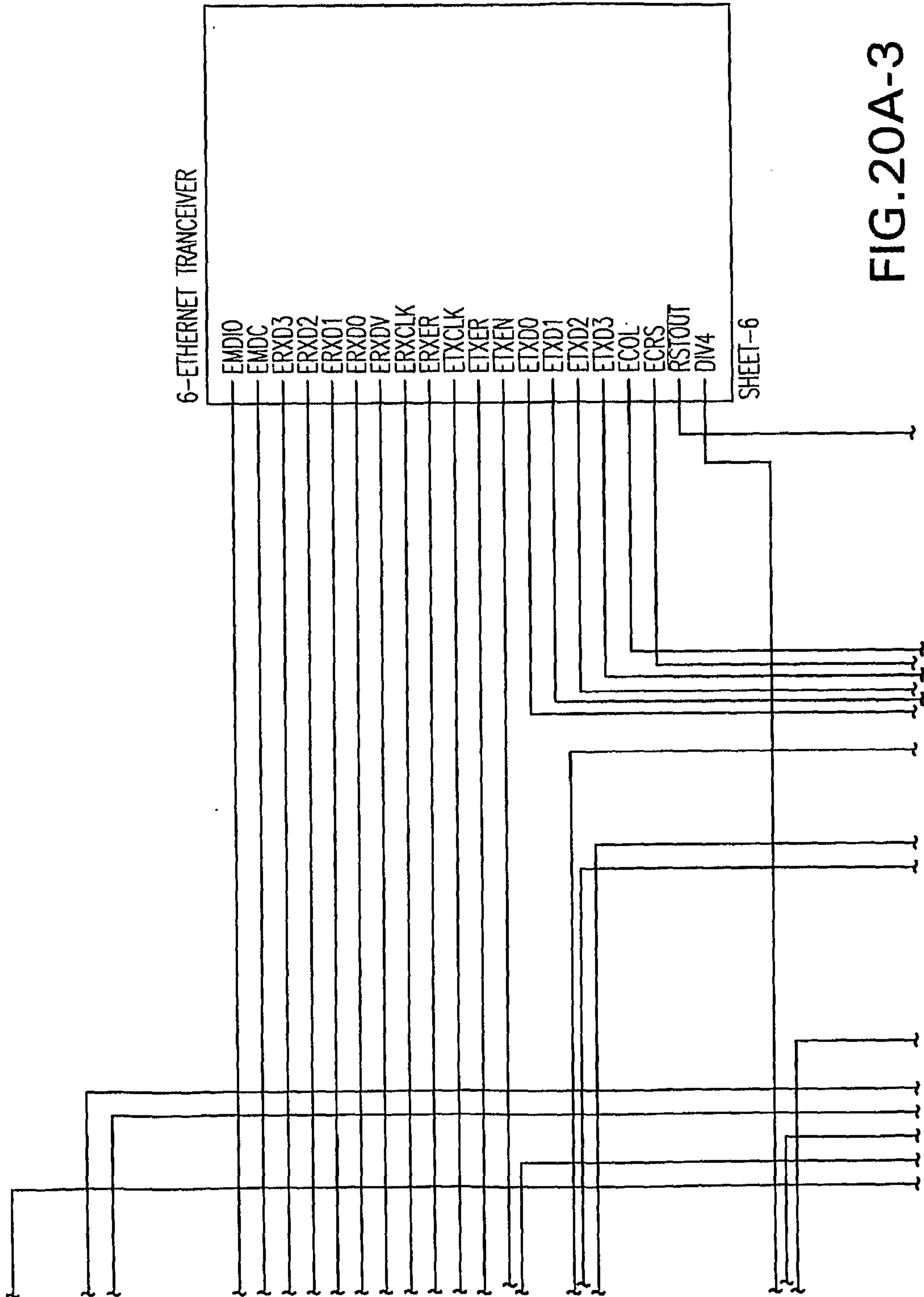
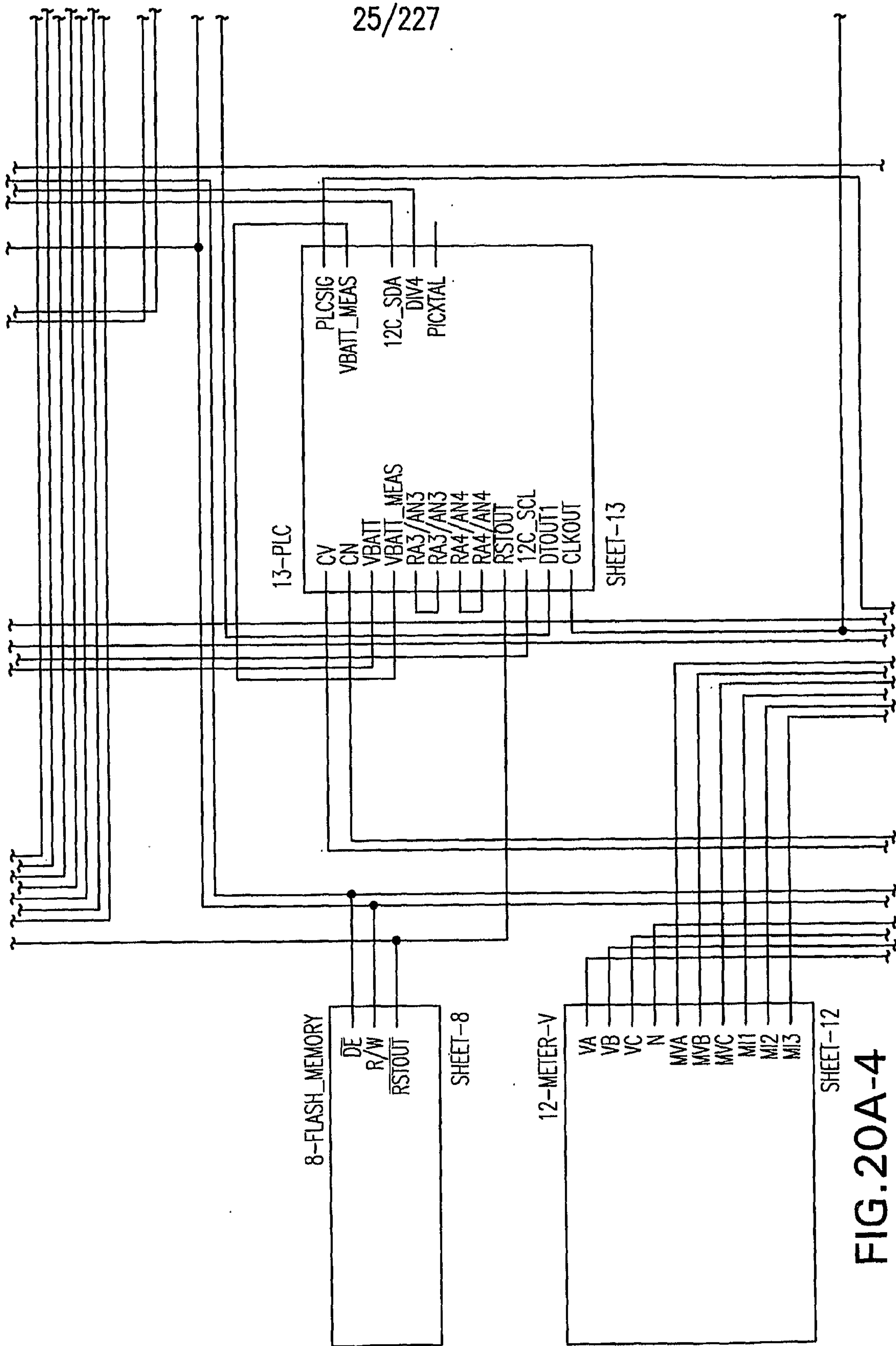


FIG. 20A-3



26/227

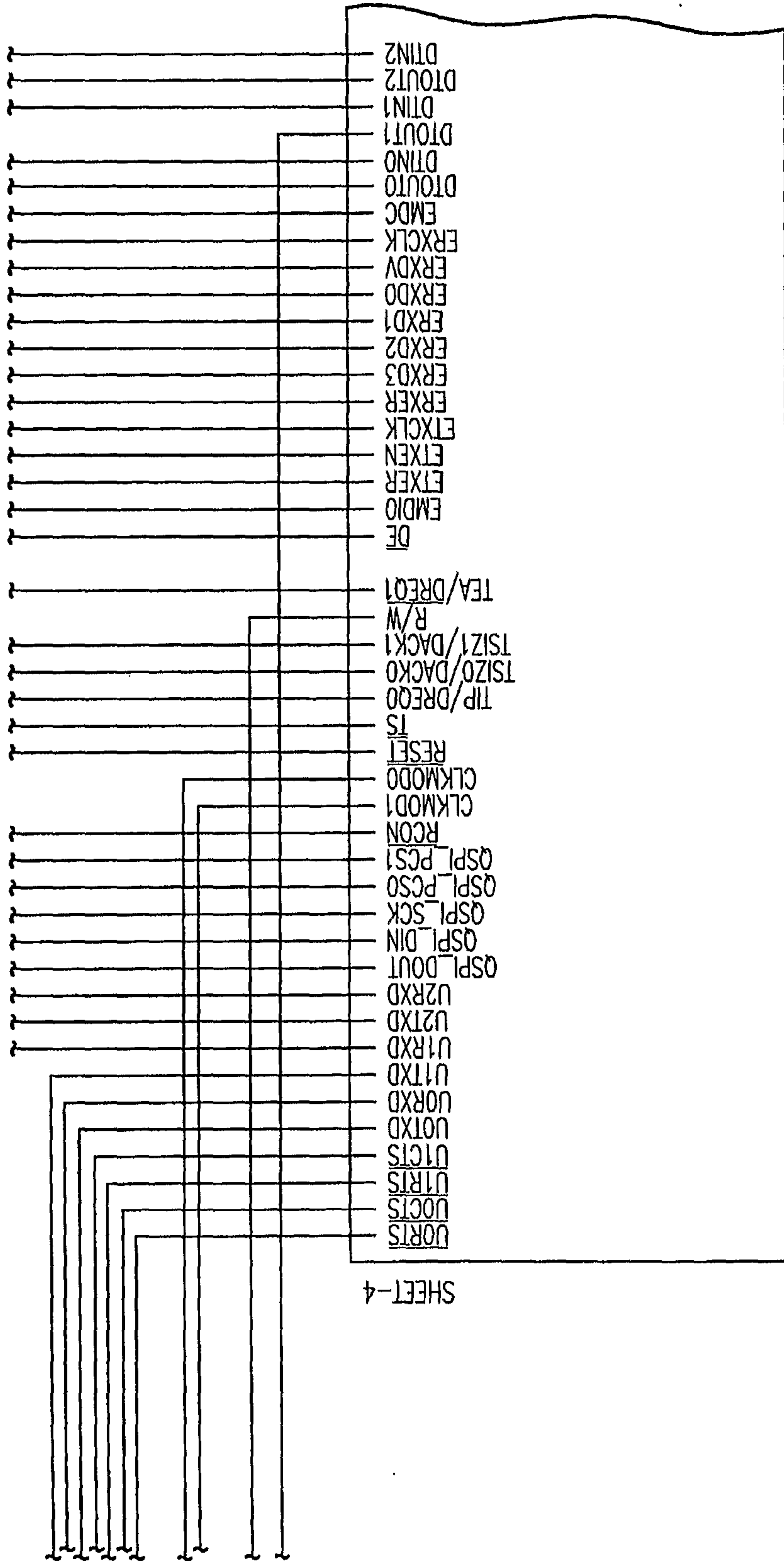


FIG. 20A-5

27/227

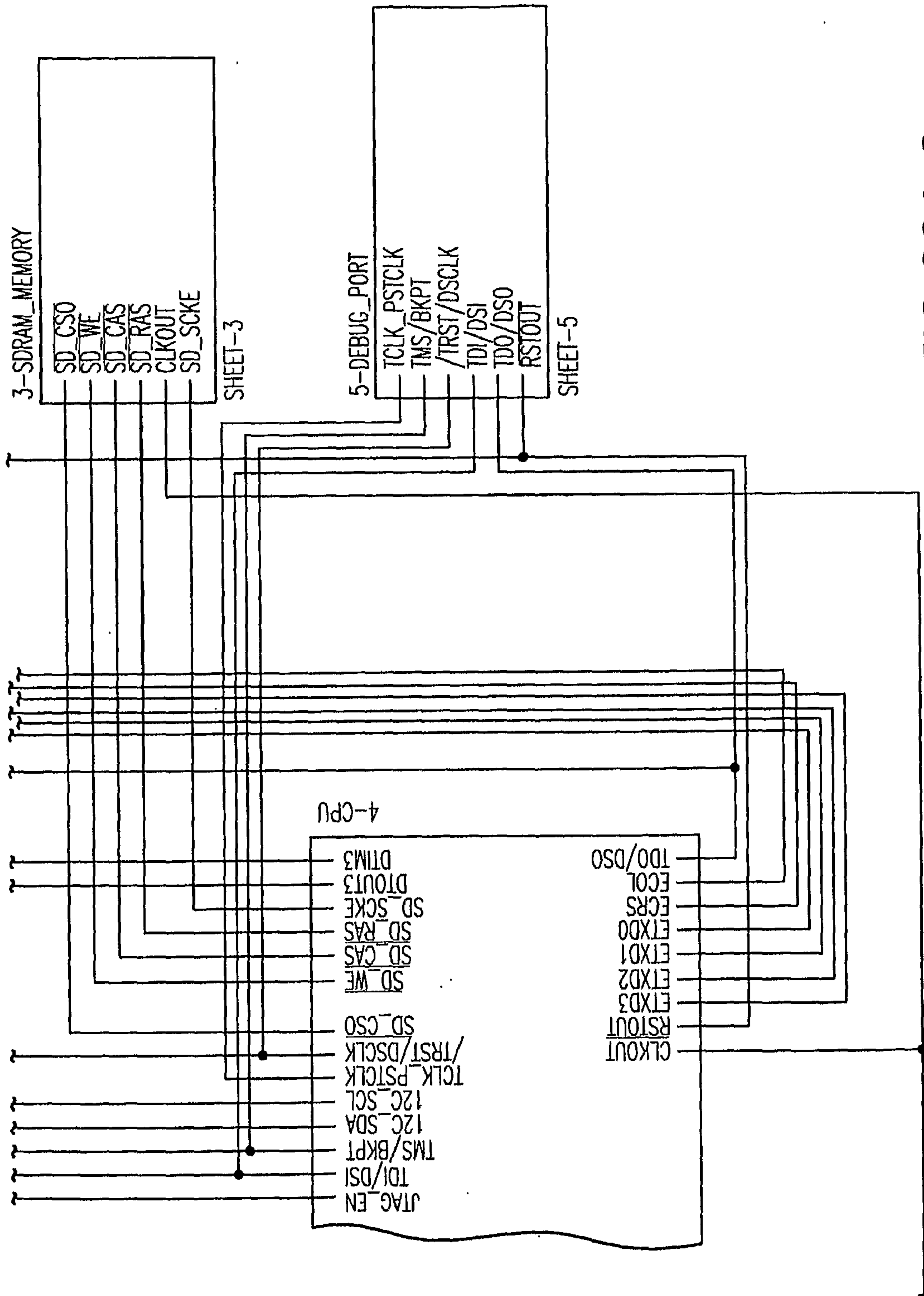


FIG. 20A-6

28/227

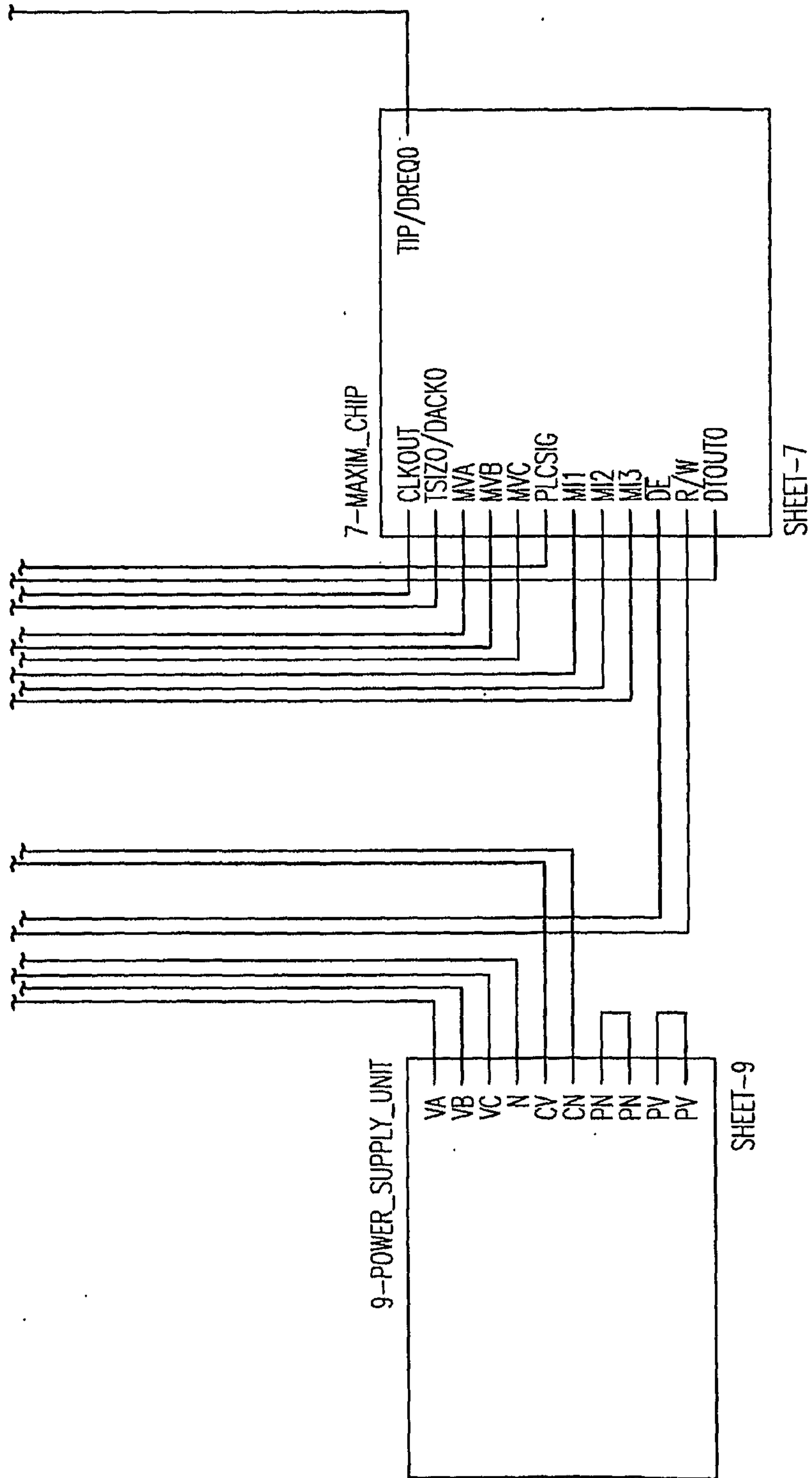


FIG. 20A-7

29/227

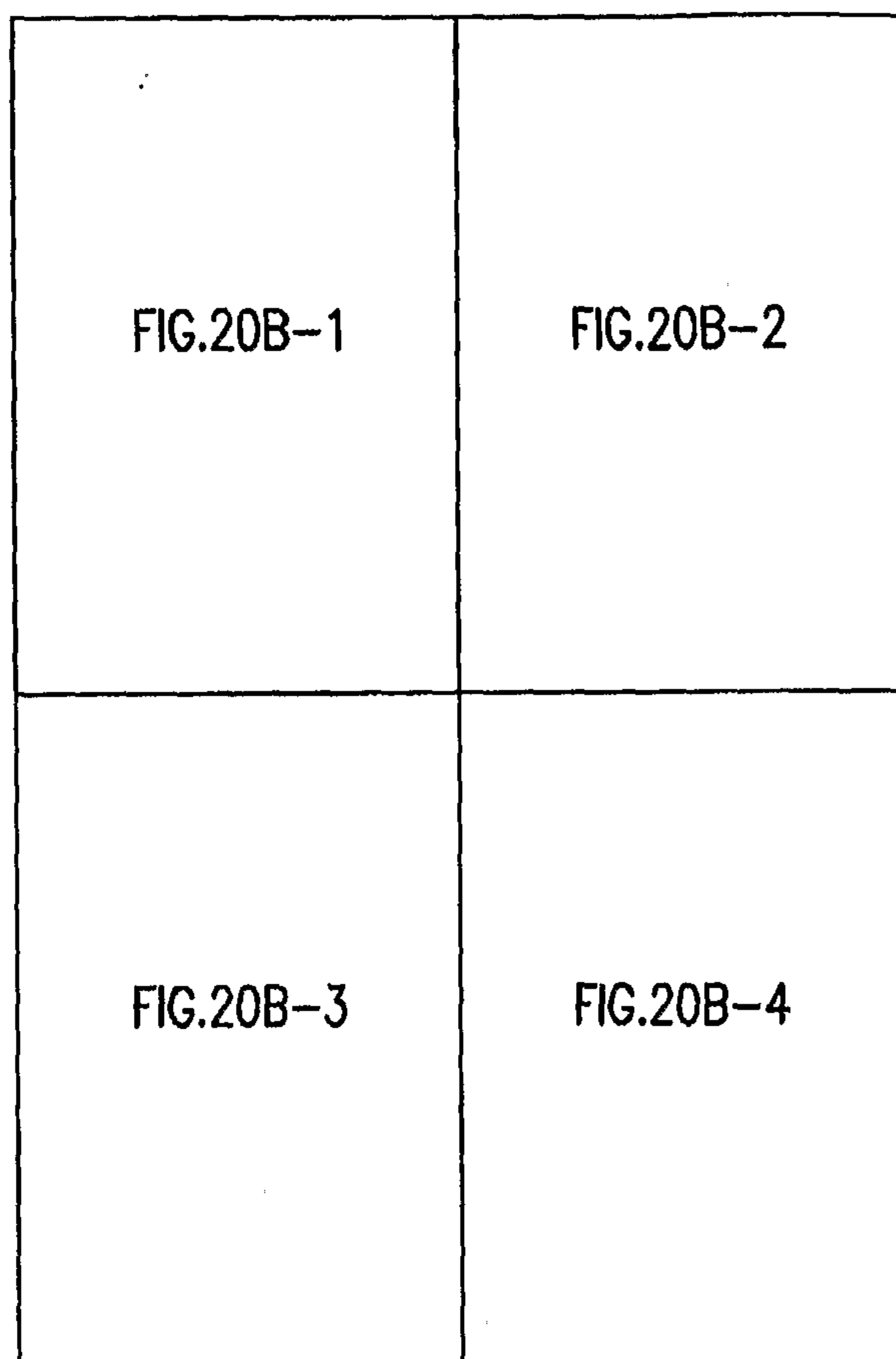
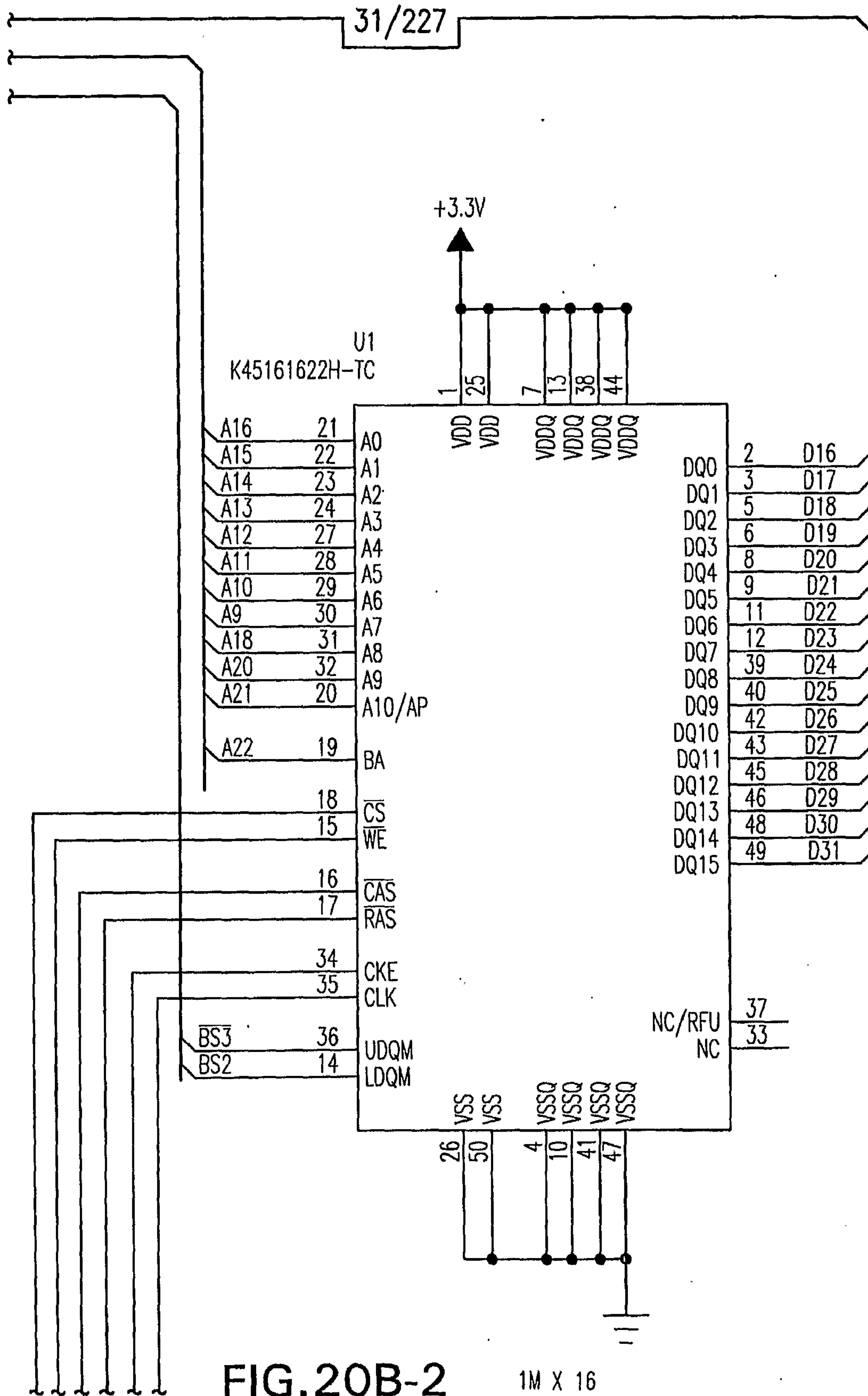


FIG.20B





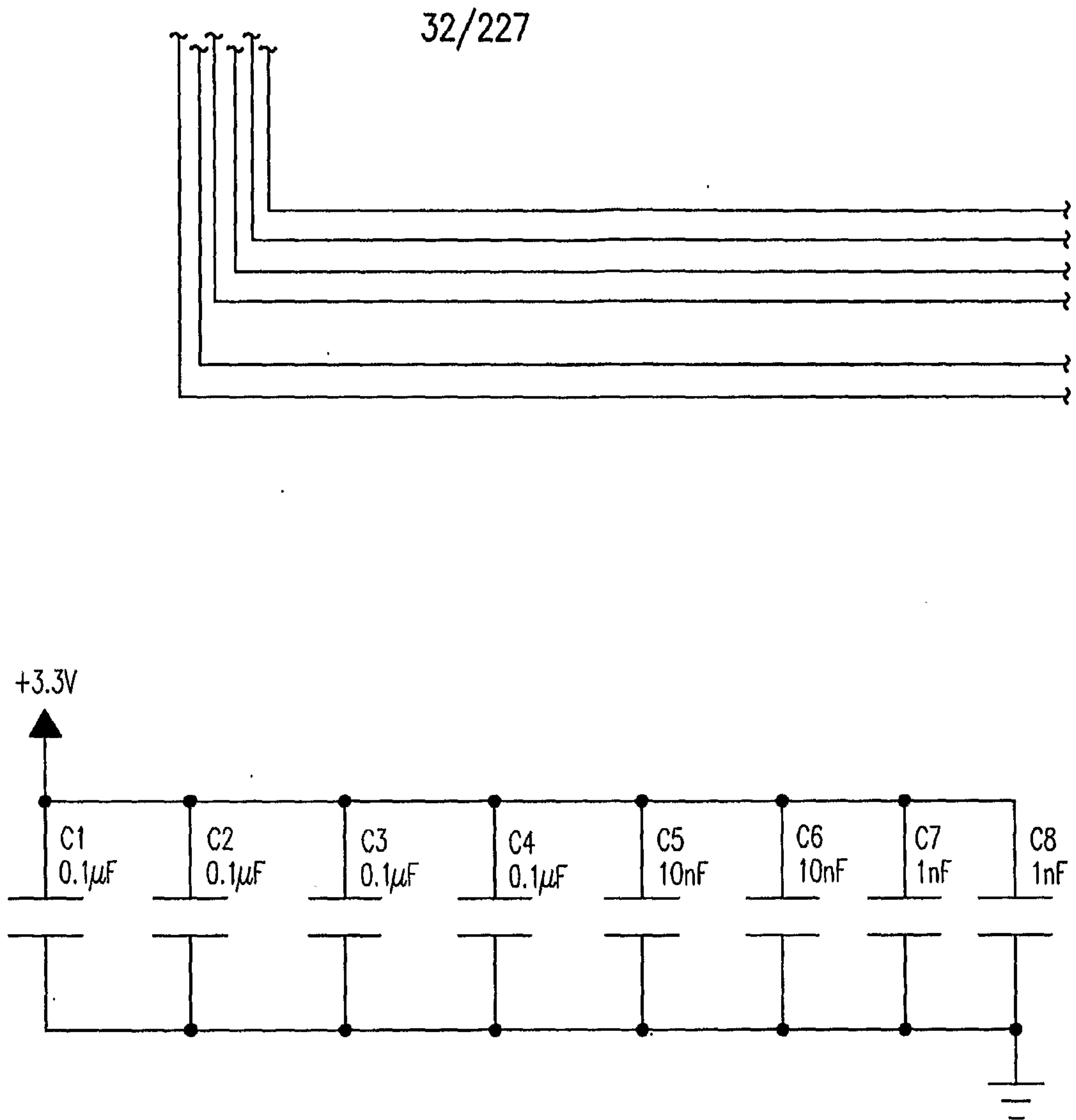


FIG. 20B-3

33/227

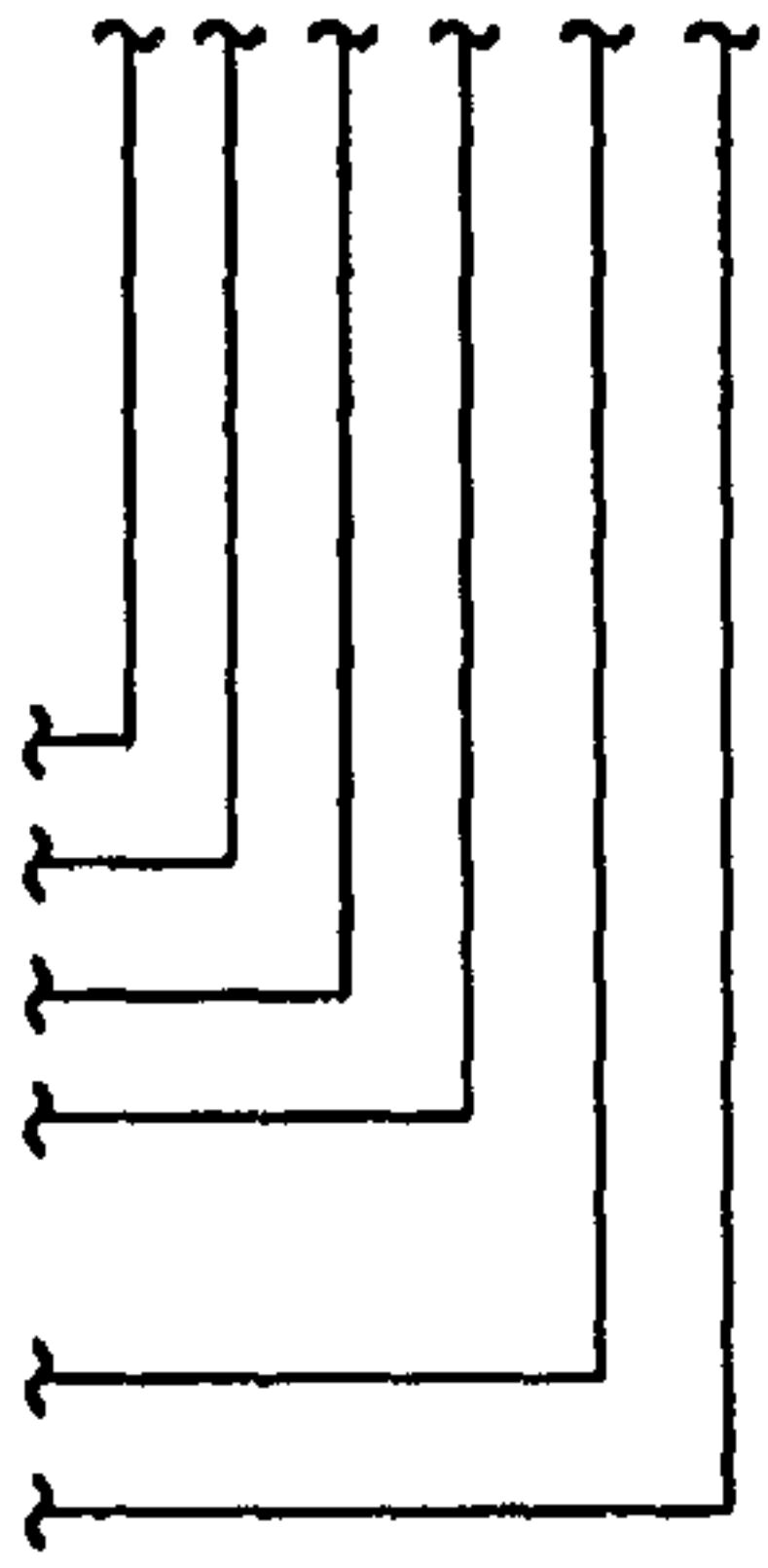


FIG.20B-4

FIG.20C-1	FIG.20C-2	FIG.20C-3	FIG.20C-4
FIG.20C-5	FIG.20C-6	FIG.20C-7	FIG.20C-8

FIG.20C

35/227

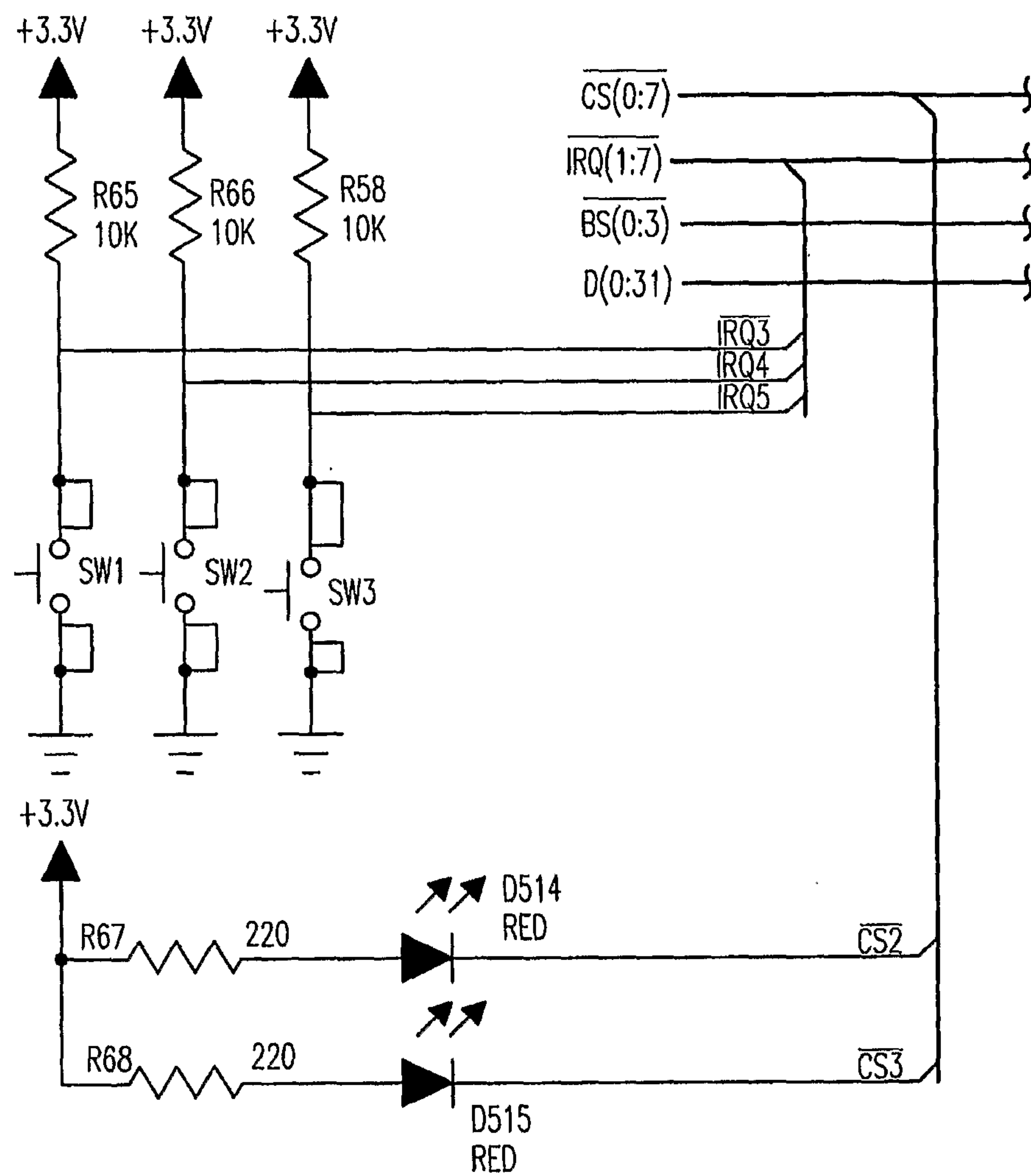


FIG. 20C-1

36/227

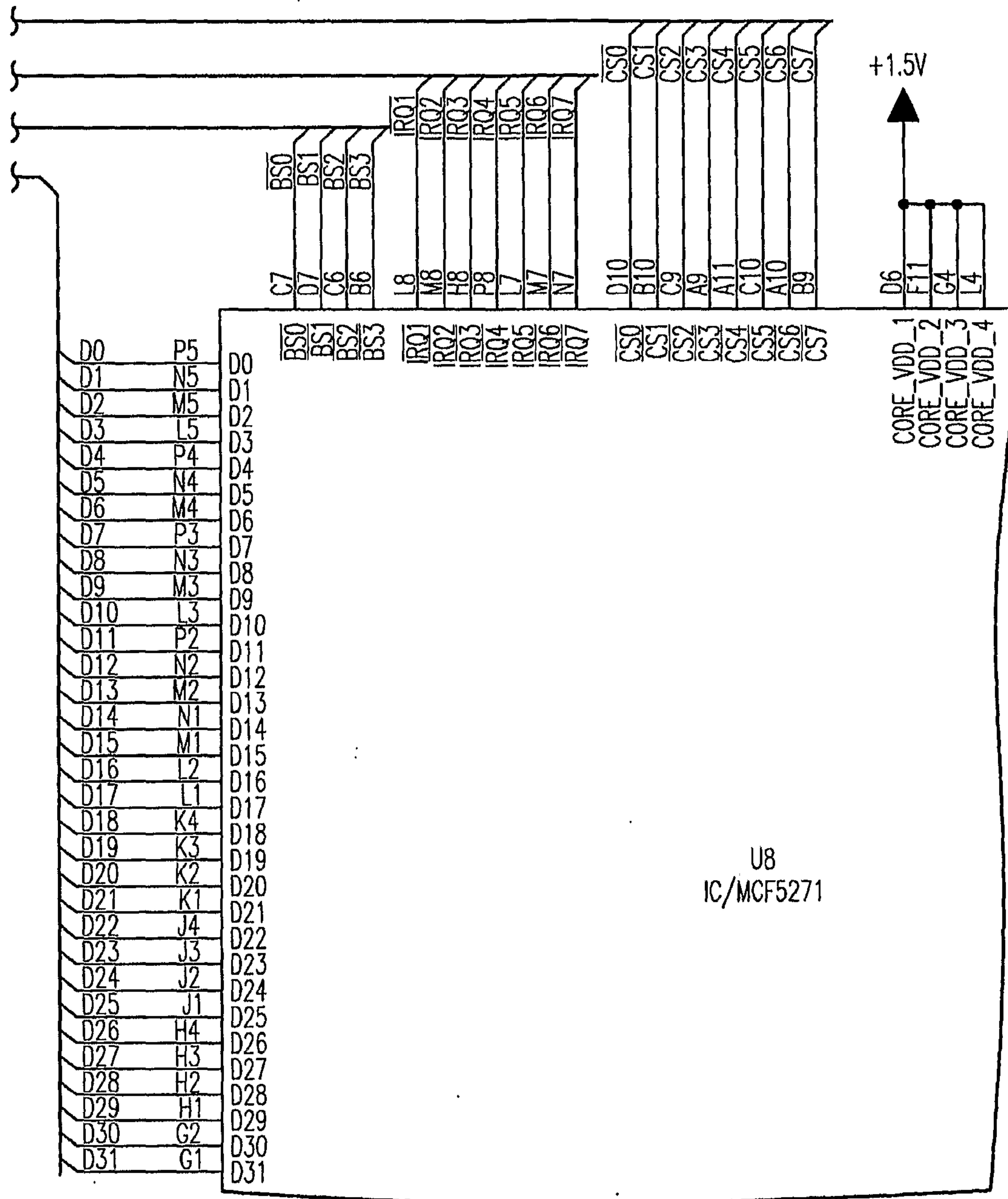


FIG.20C-2

37/227

NOTE: PLACE C42, C43, C64 AND L1 AS CLOSE TO PINS L14 & M13 AS POSSIBLE USING A SEPARATE GROUND PLANE.

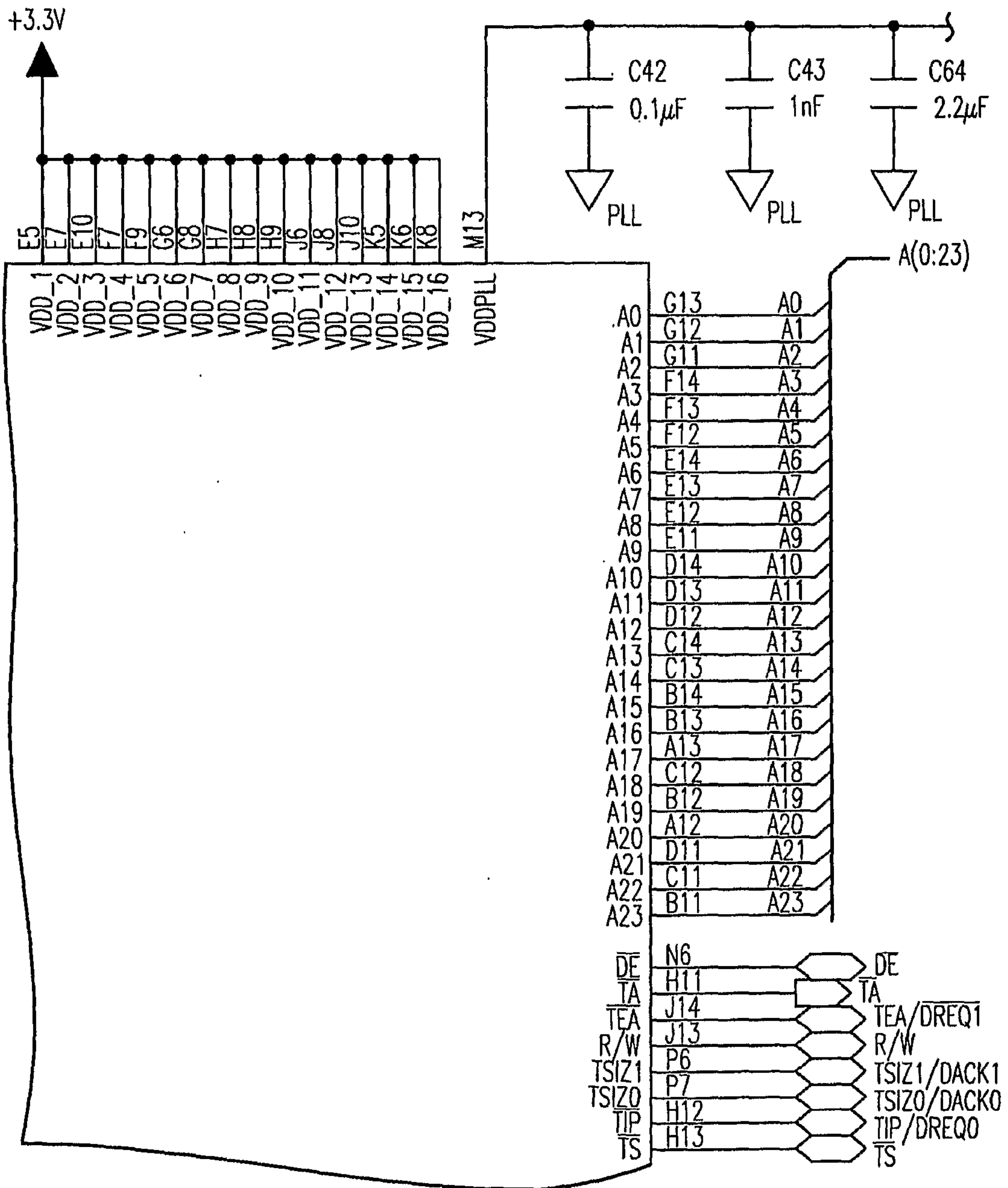


FIG.20C-3

38/227

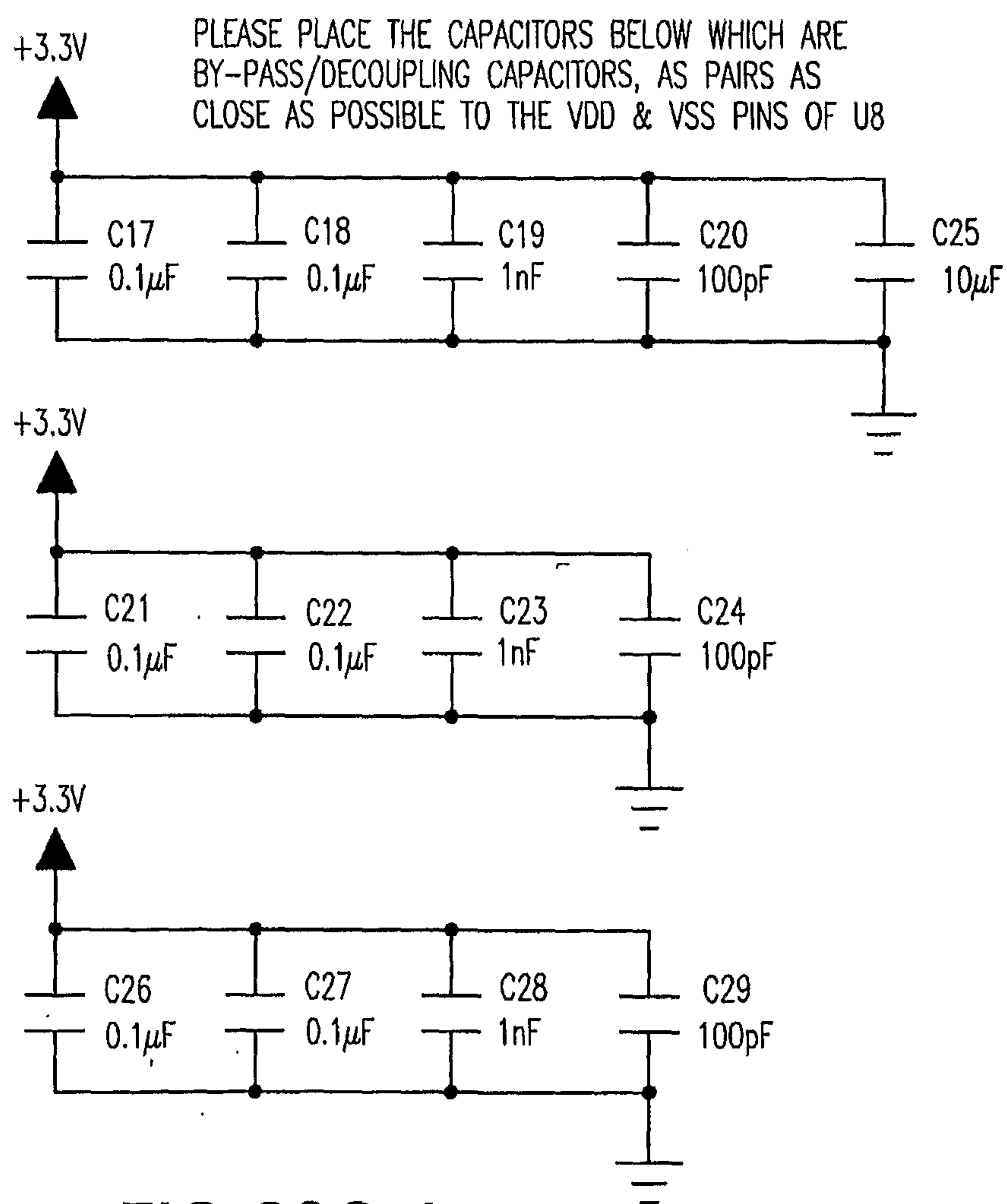
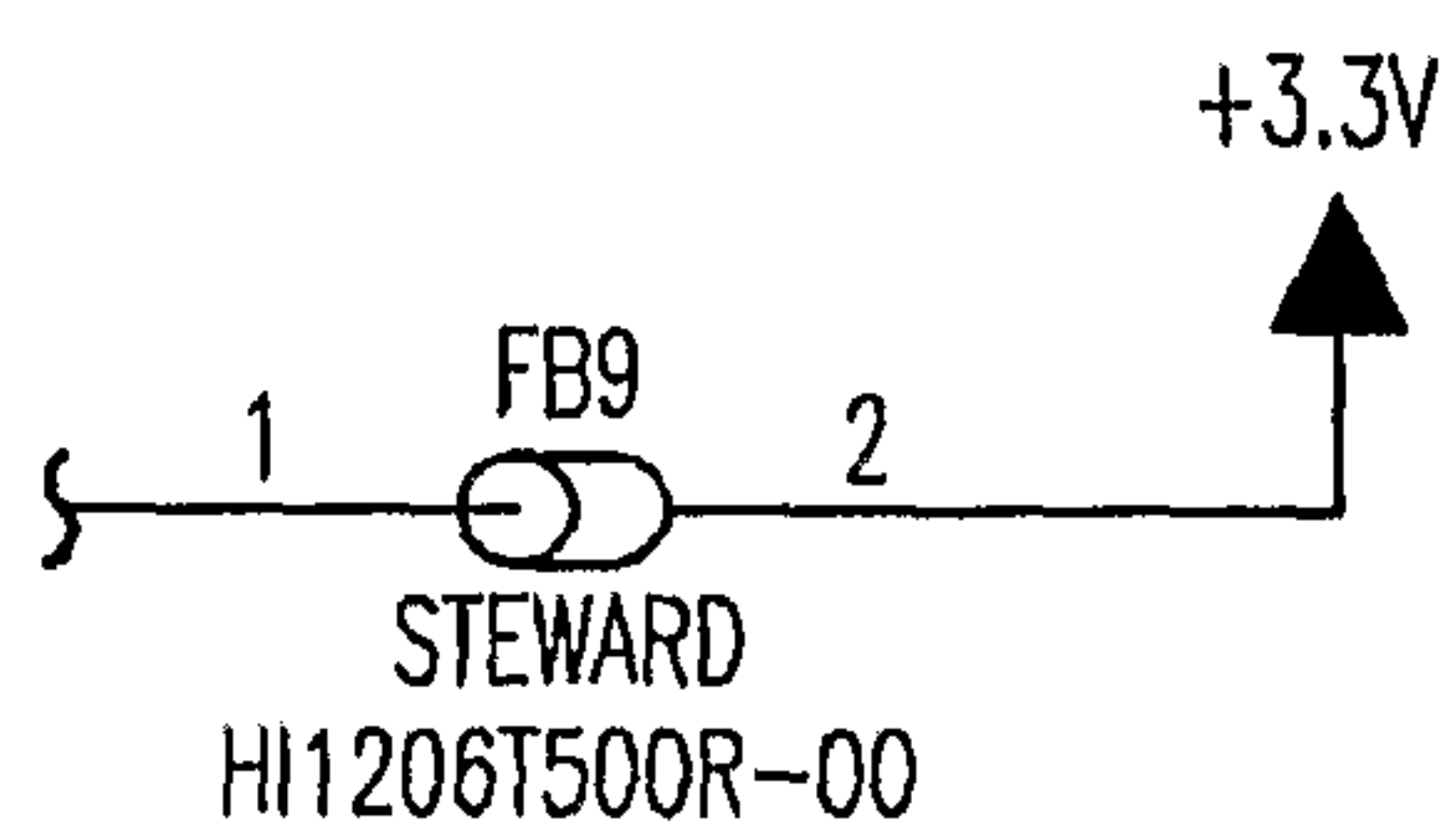
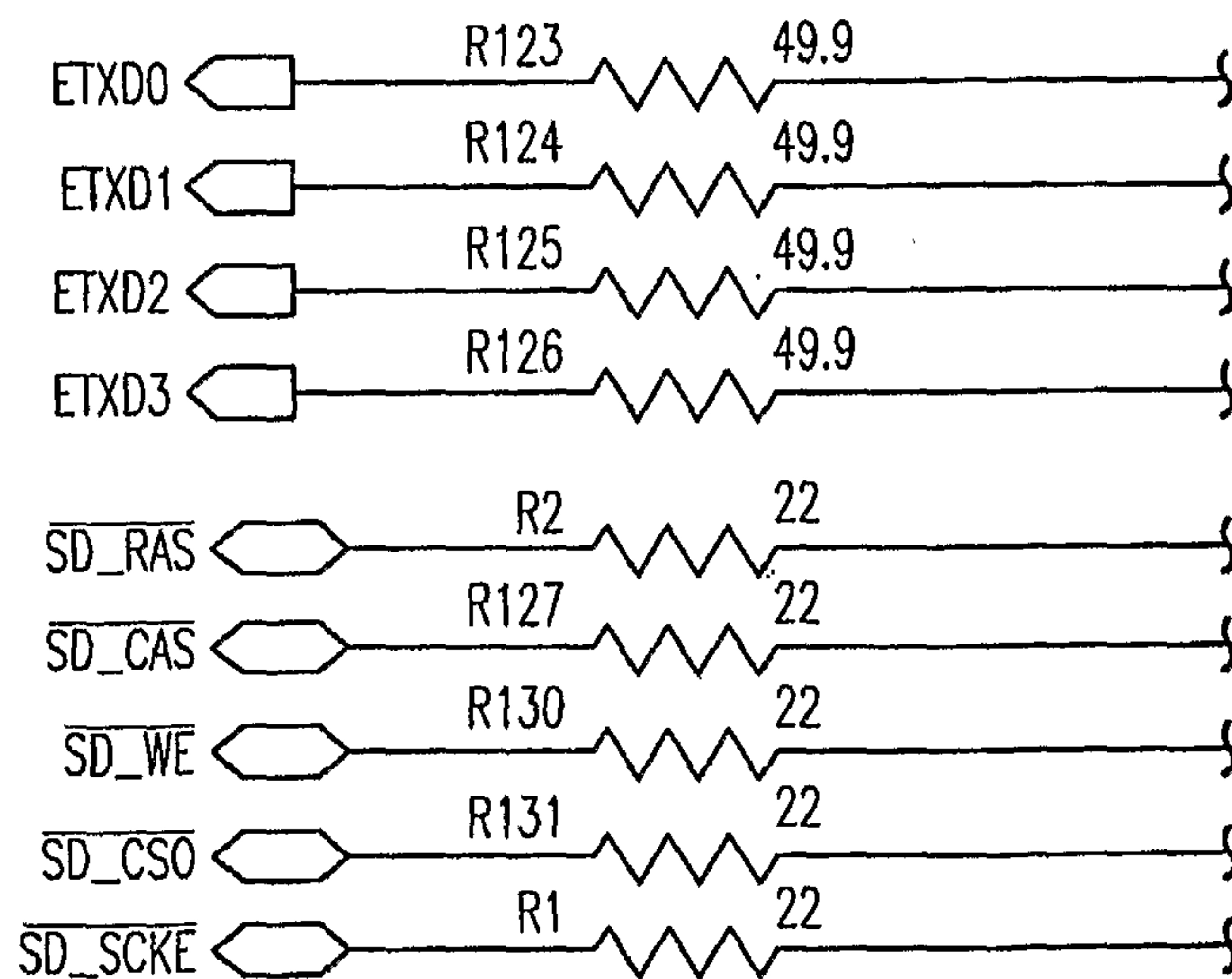
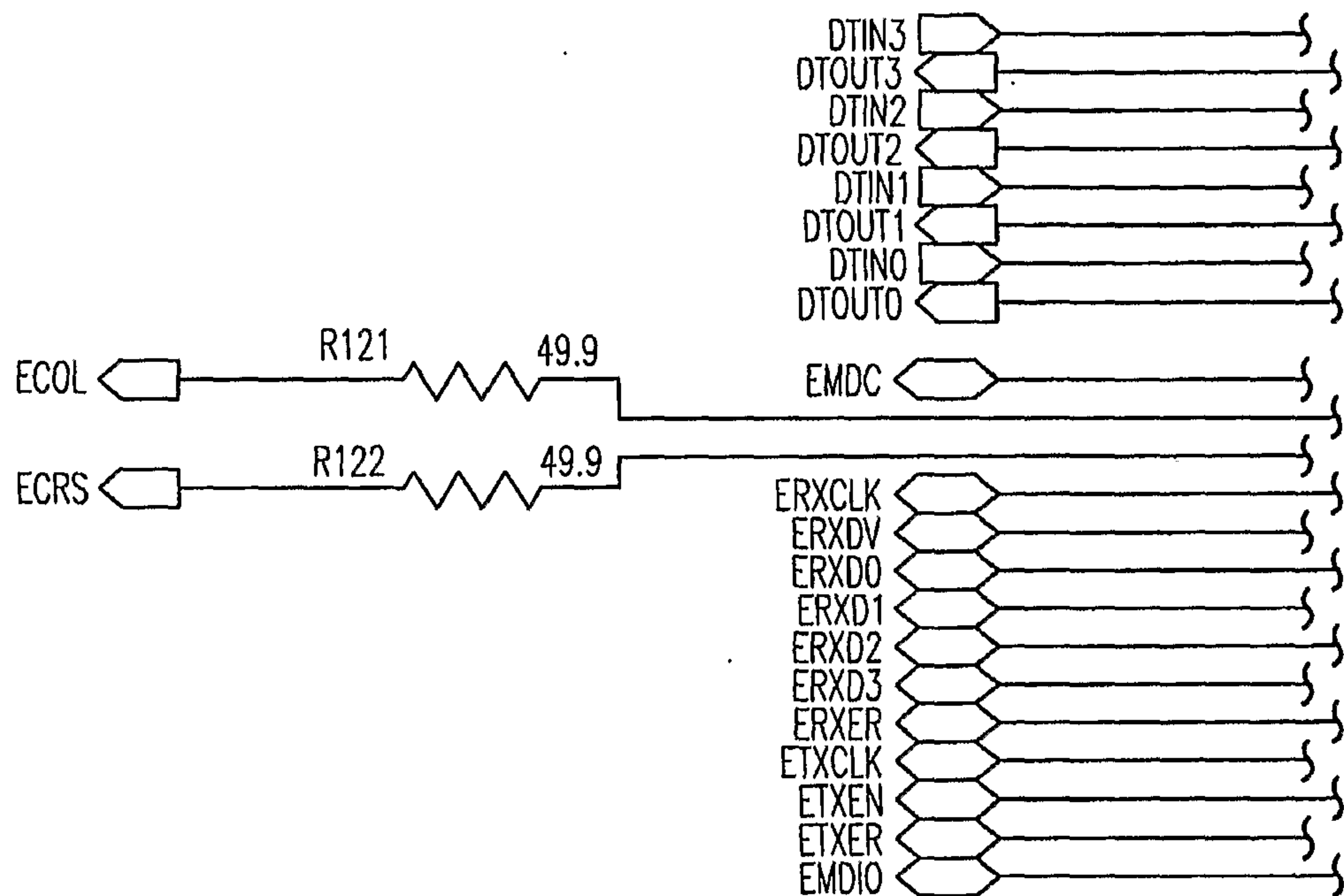


FIG.20C-4

39/227



PLACE RESISTORS AS CLOSE AS POSSIBLE TO PINS OF U8

FIG.20C-5

40/227

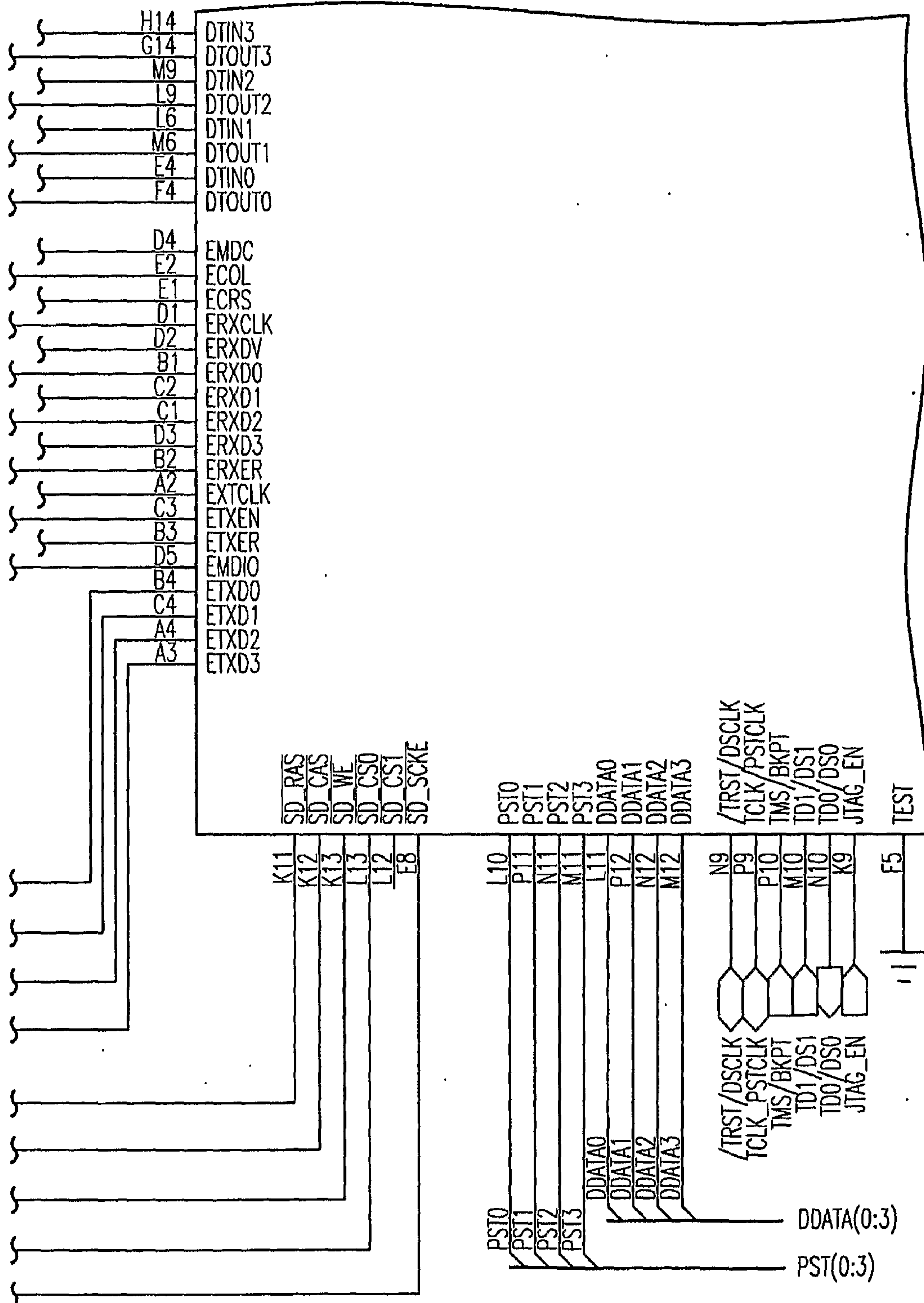


FIG. 20C-6

41/227

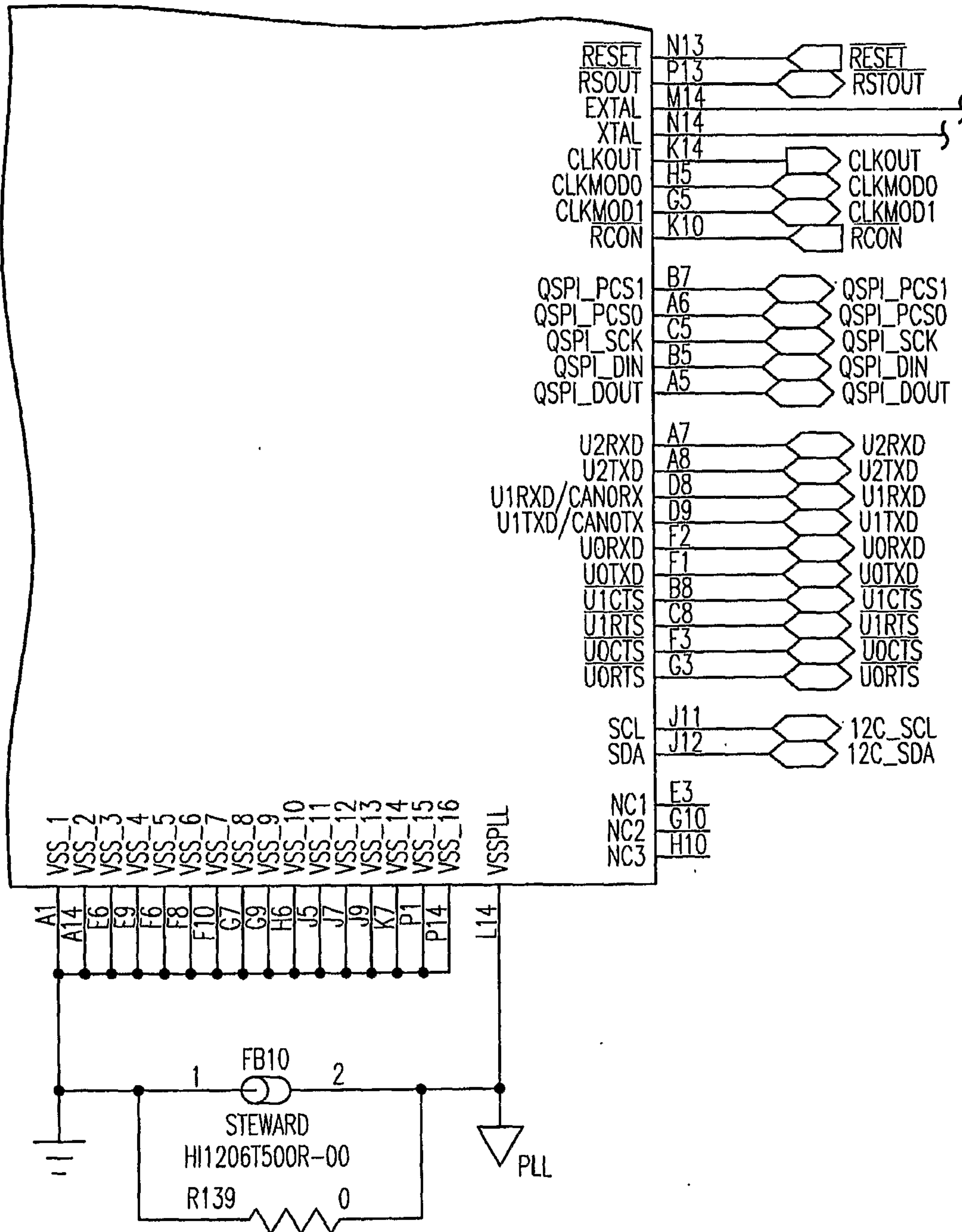


FIG.20C-7

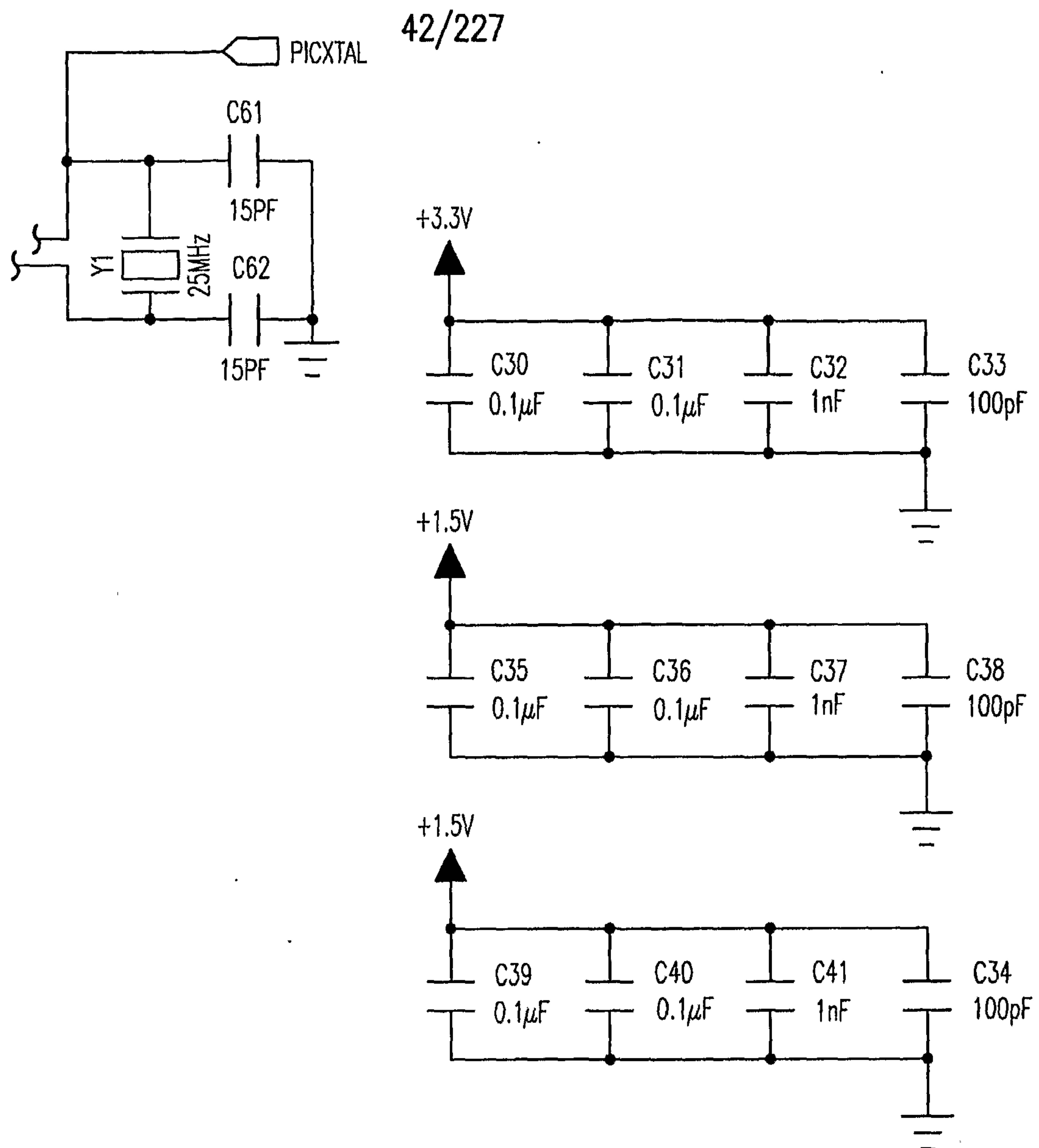


FIG.20C-8

43/227

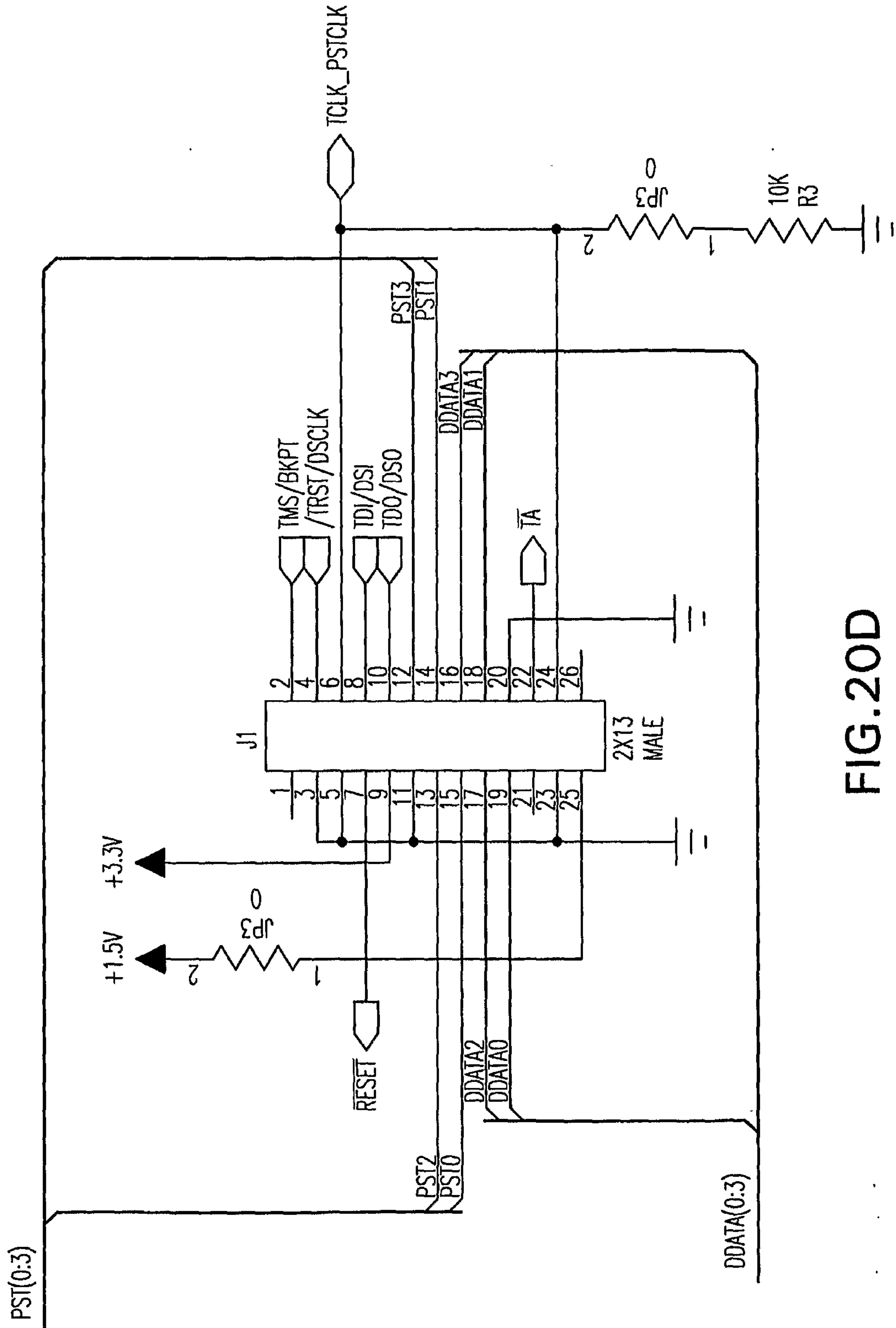


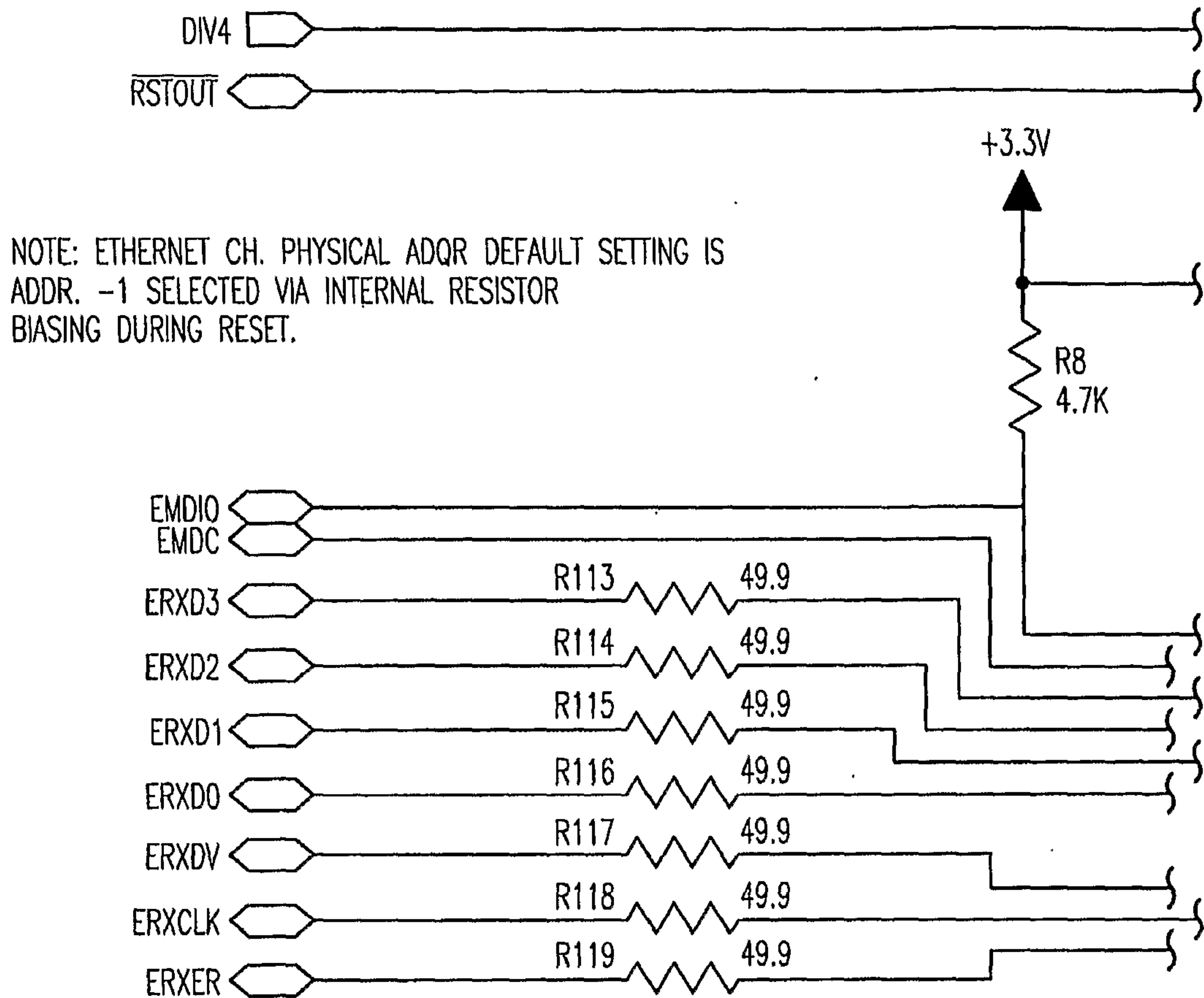
FIG.20D

44/227

FIG.20E-1	FIG.20E-2	FIG.20E-3	FIG.20E-4
FIG.20E-5	FIG.20E-6	FIG.20E-7	

FIG.20E

45/227



PLACE R113 THRU R120 AS CLOSE TO U11 AS POSSIBLE

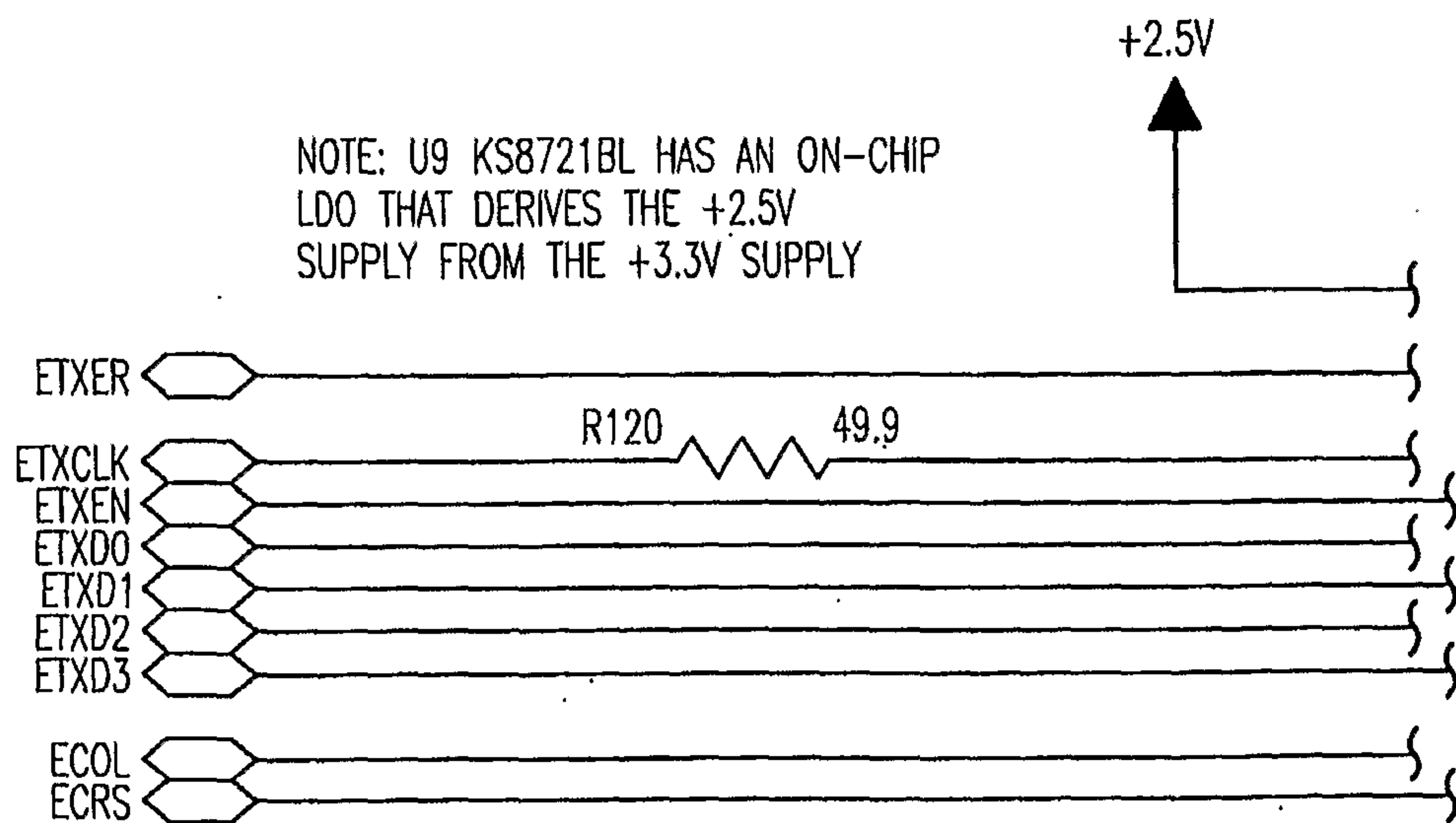


FIG.20E-1

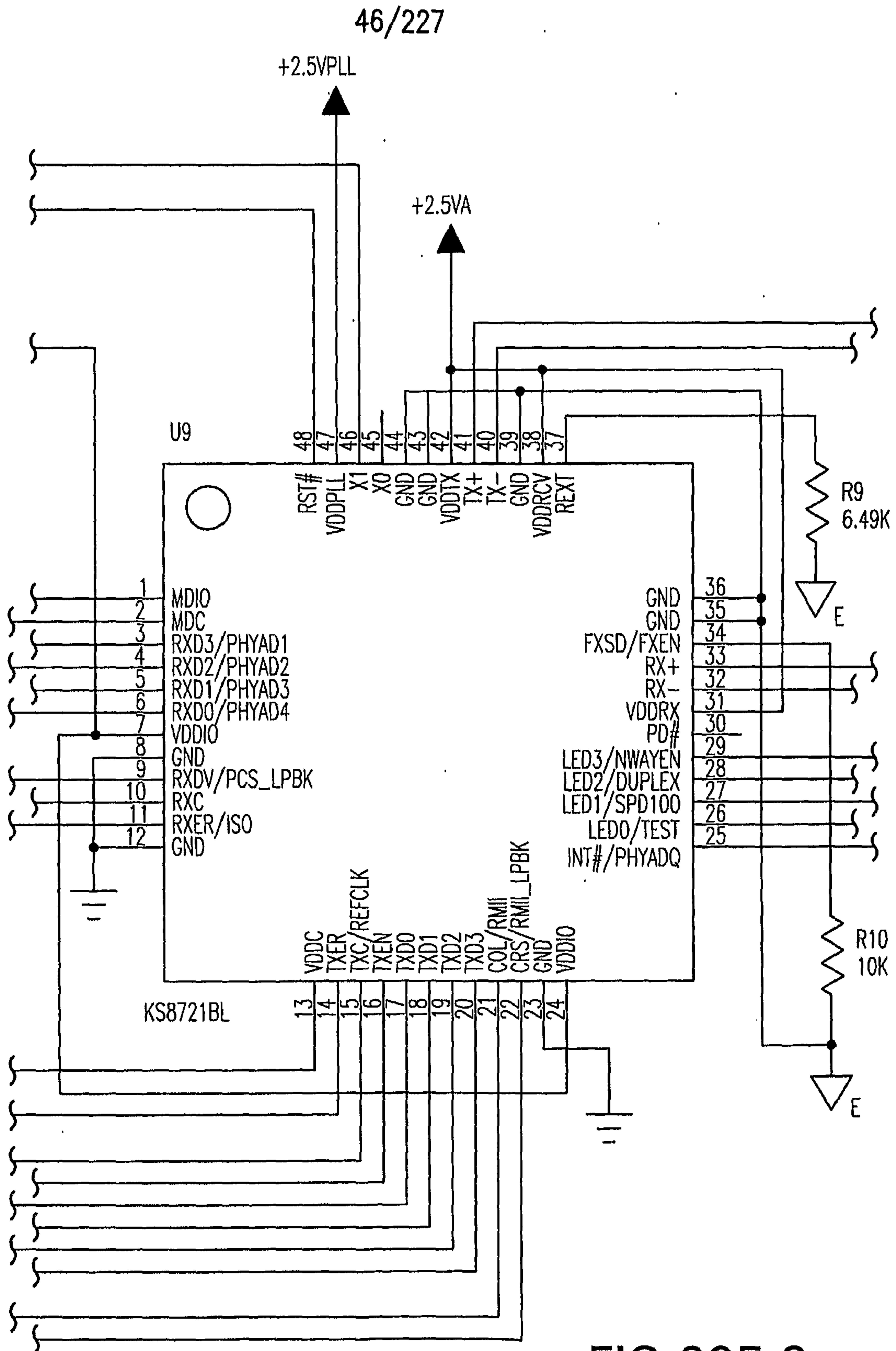
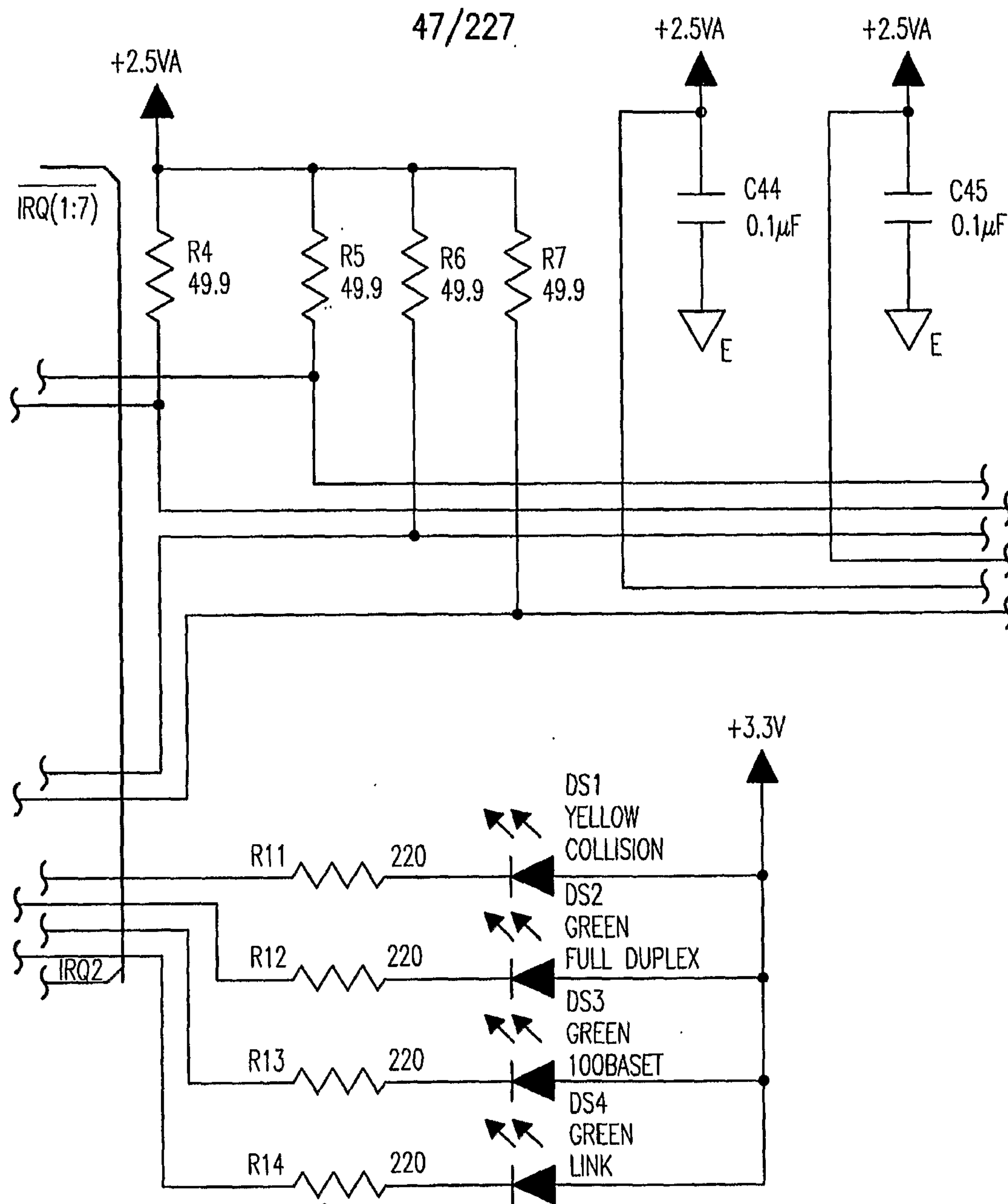


FIG.20E-2



PLACE SILK SCREEN LED LABELS NEXT TO D1 THRU D4

FIG. 20E-3

48/227

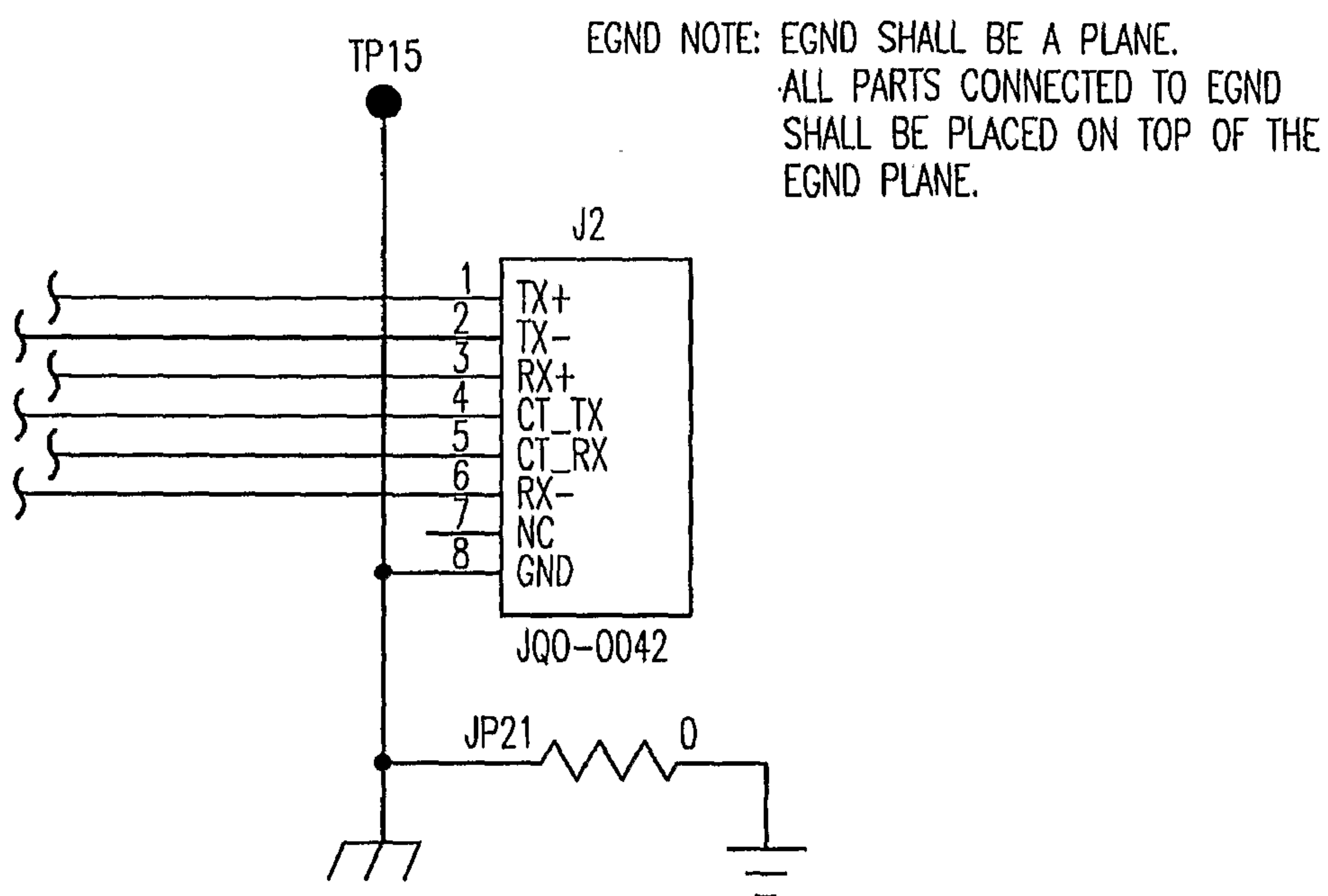


FIG.20E-4

49/227

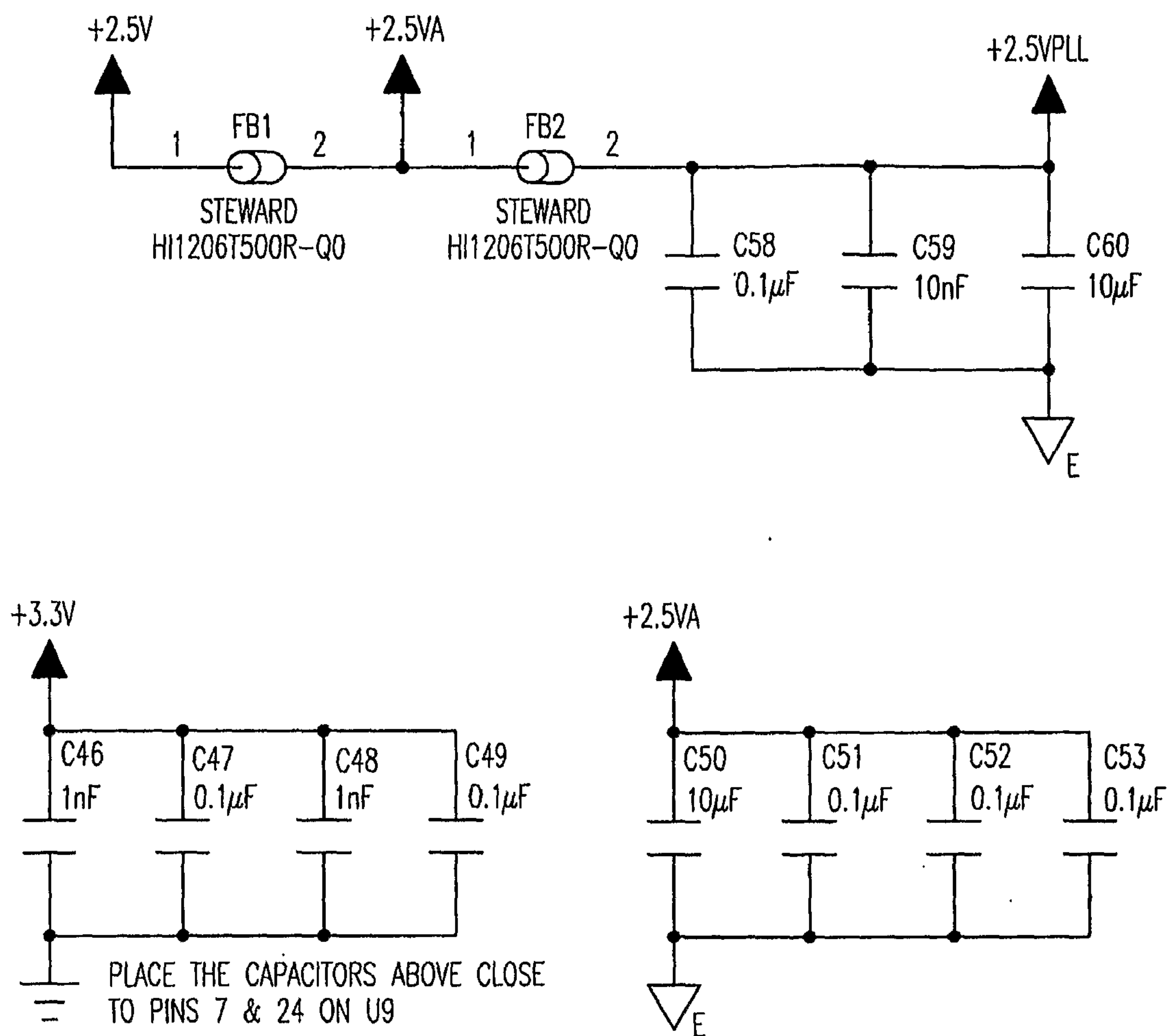


FIG.20E-5

50/227

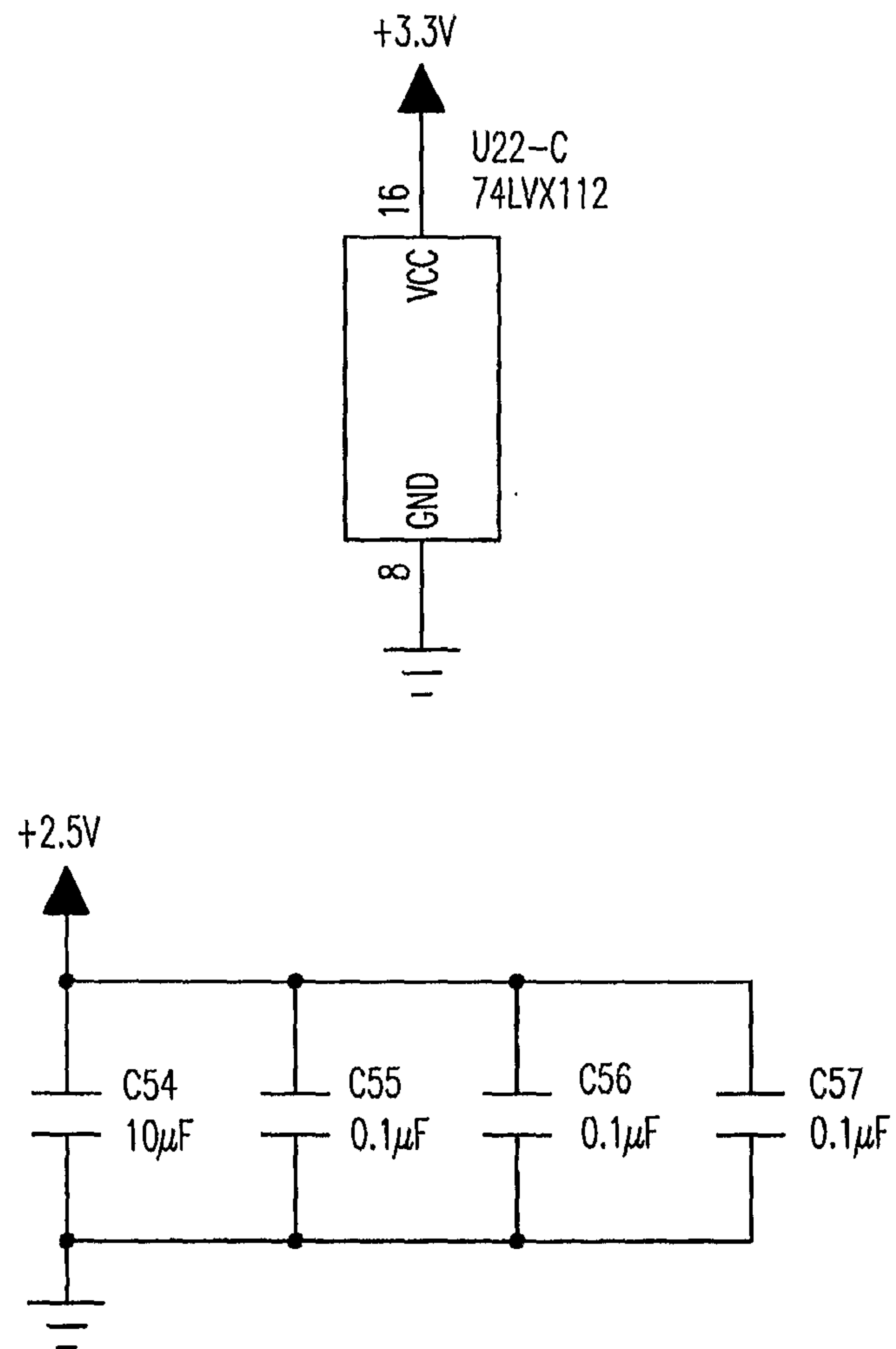


FIG.20E-6

51/227

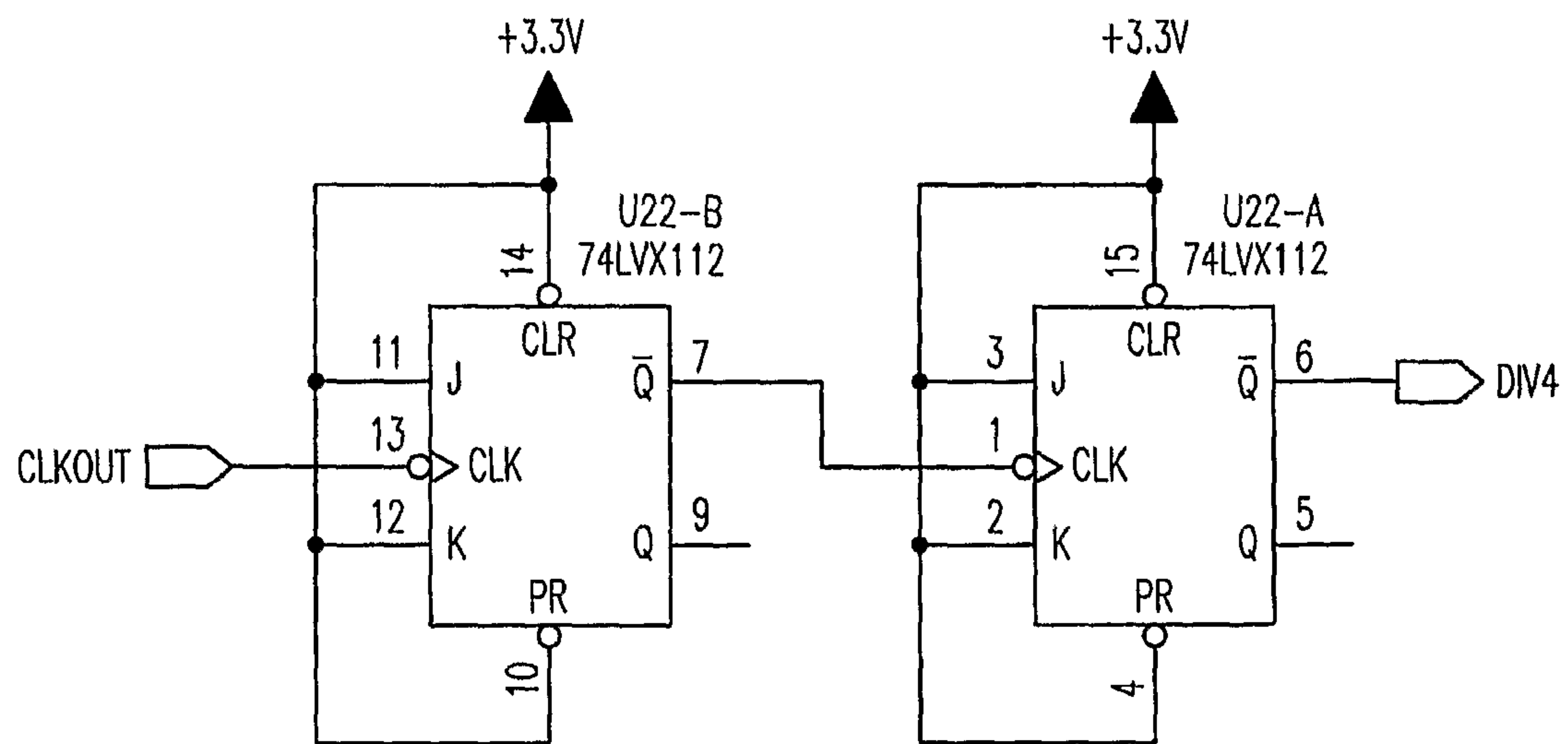


FIG.20E-7

52/227

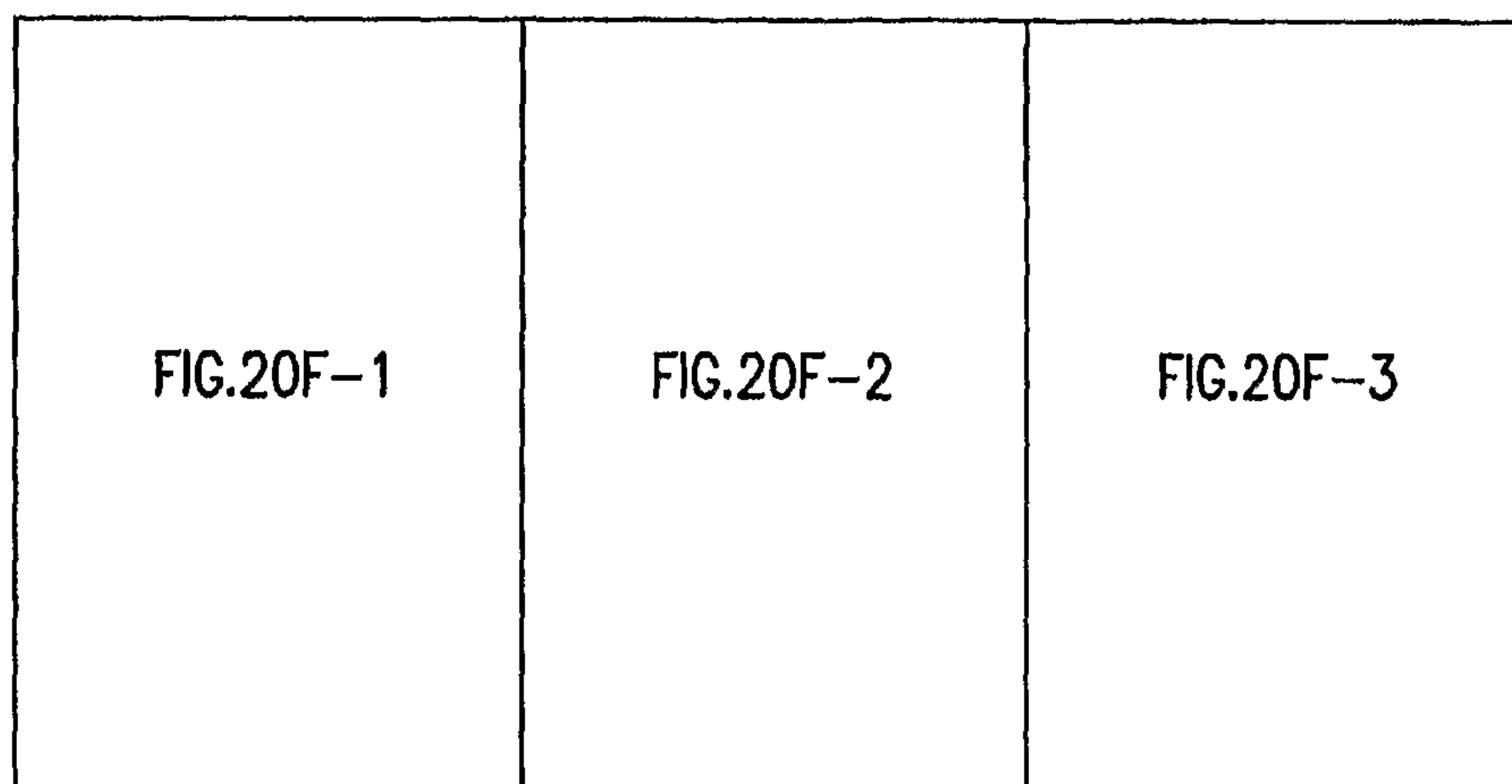
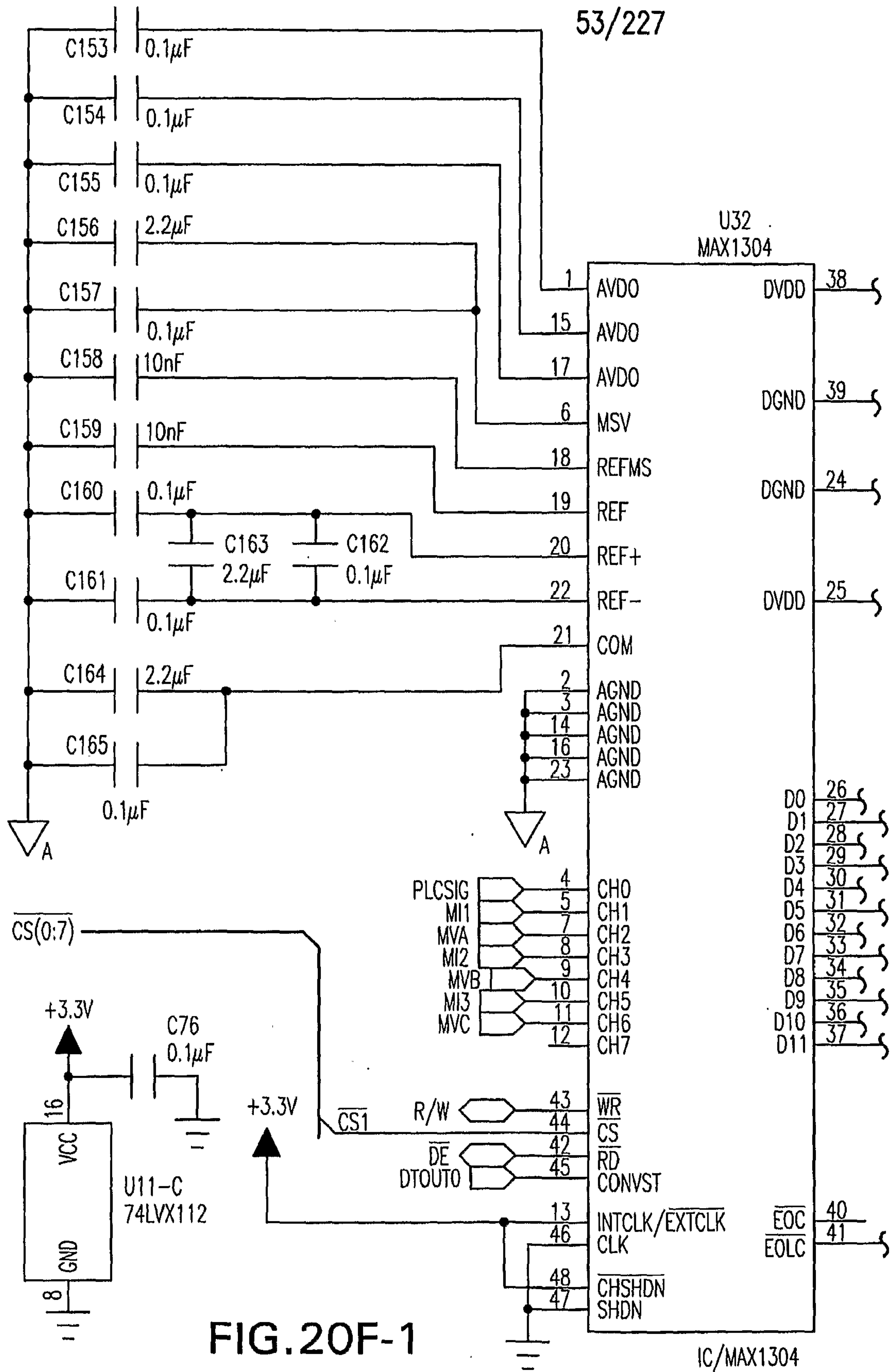


FIG.20F



54/227

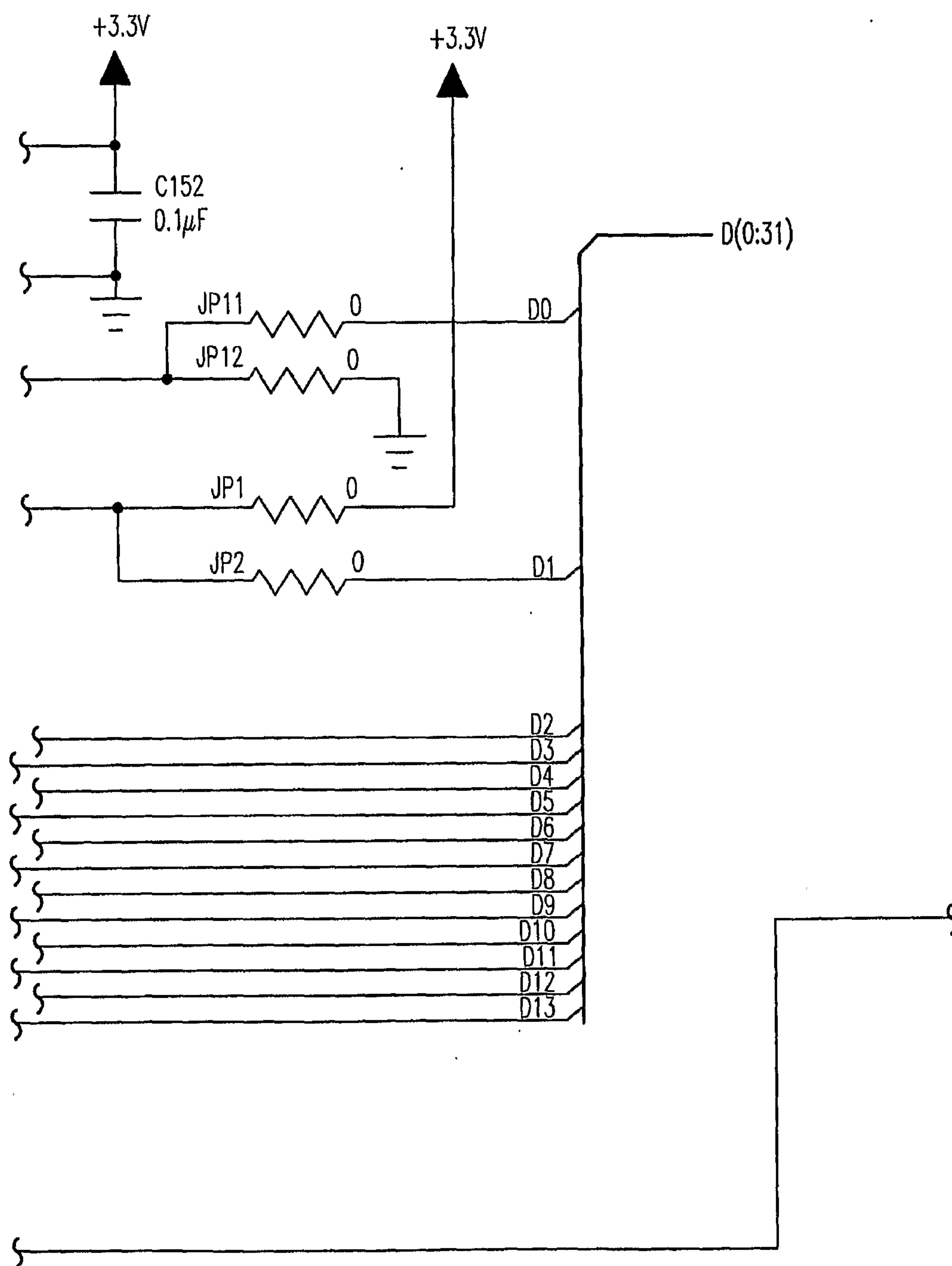


FIG.20F-2

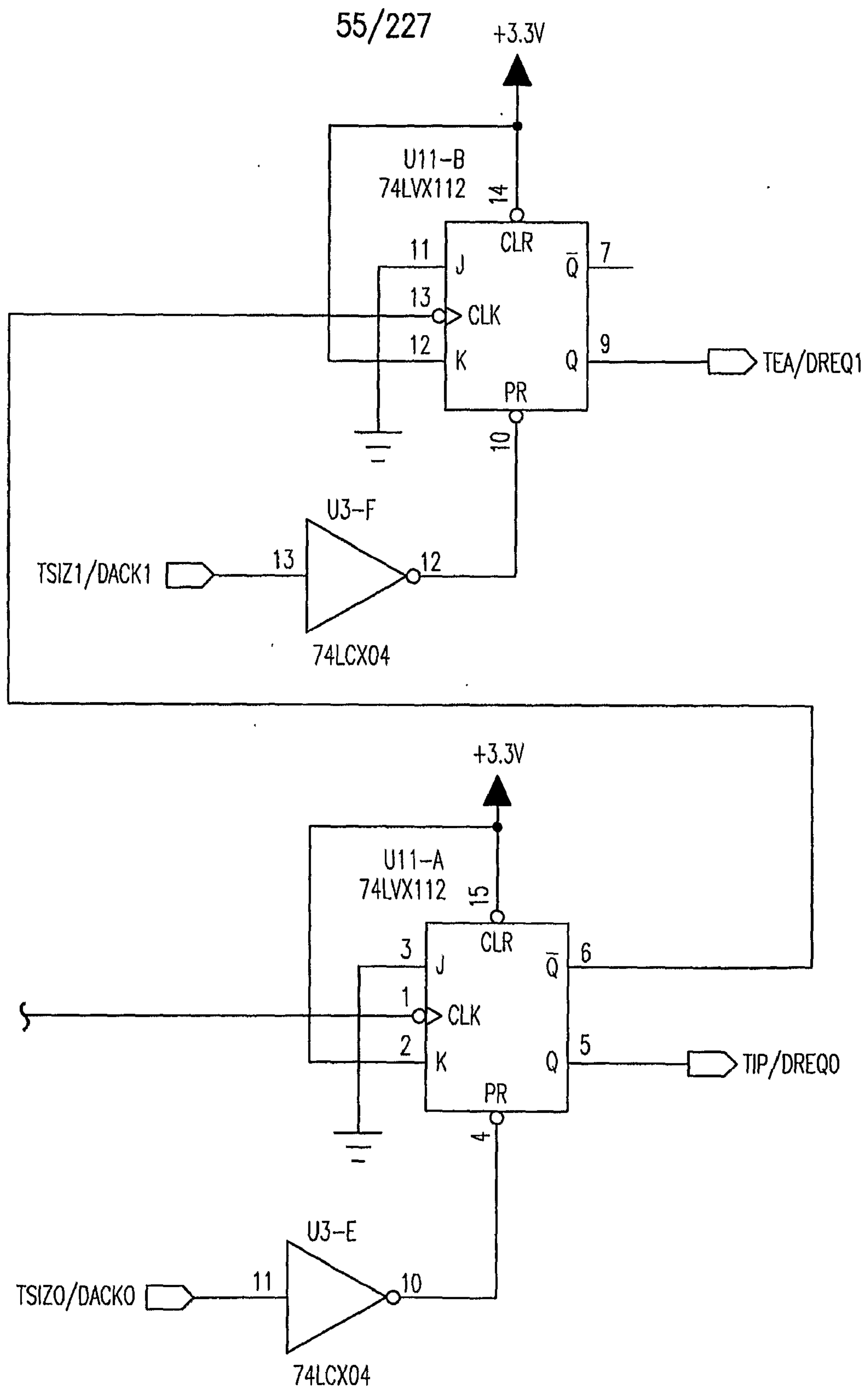


FIG.20F-3

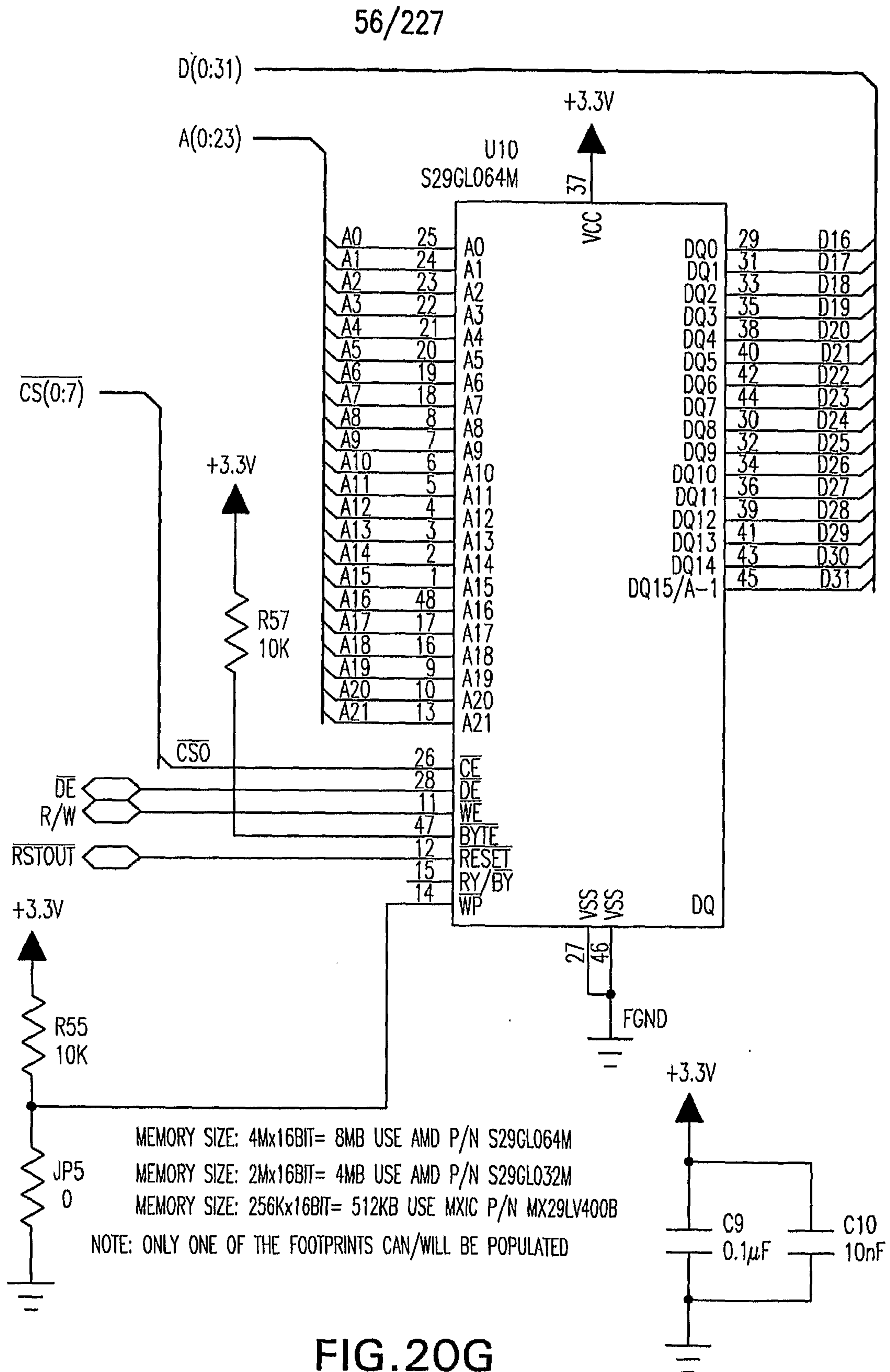


FIG.20G

57/227

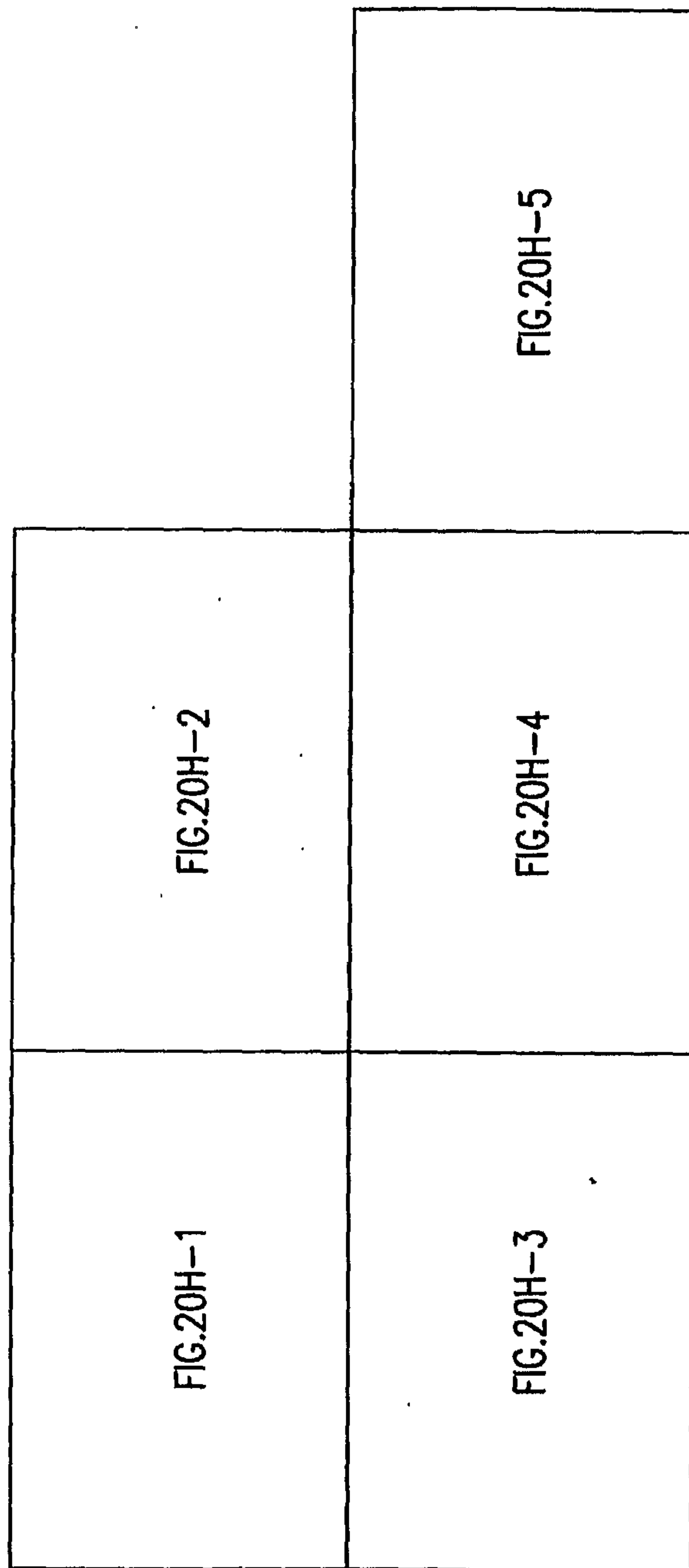


FIG.20H

59/227

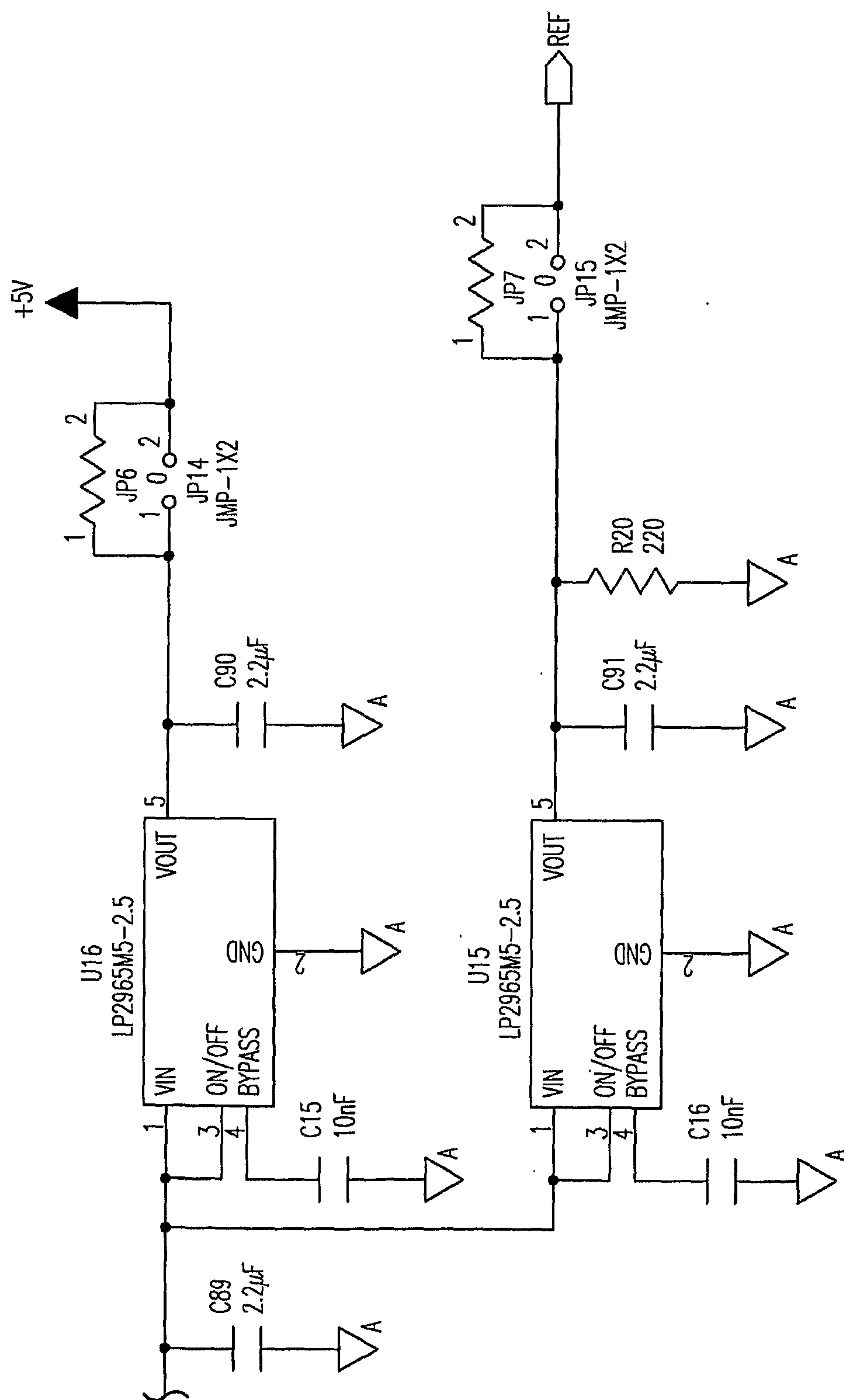


FIG.20H-2

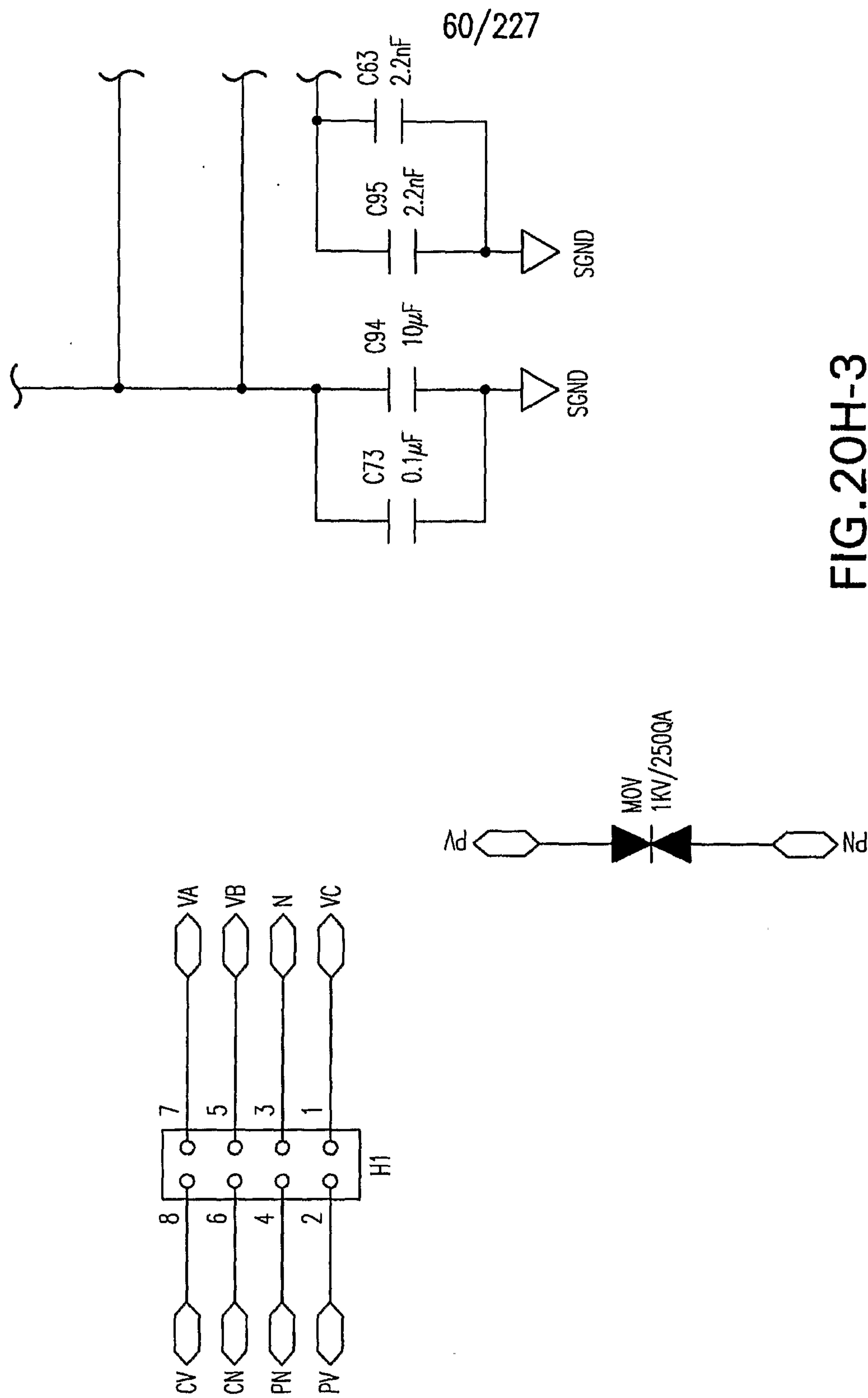


FIG.20H-3

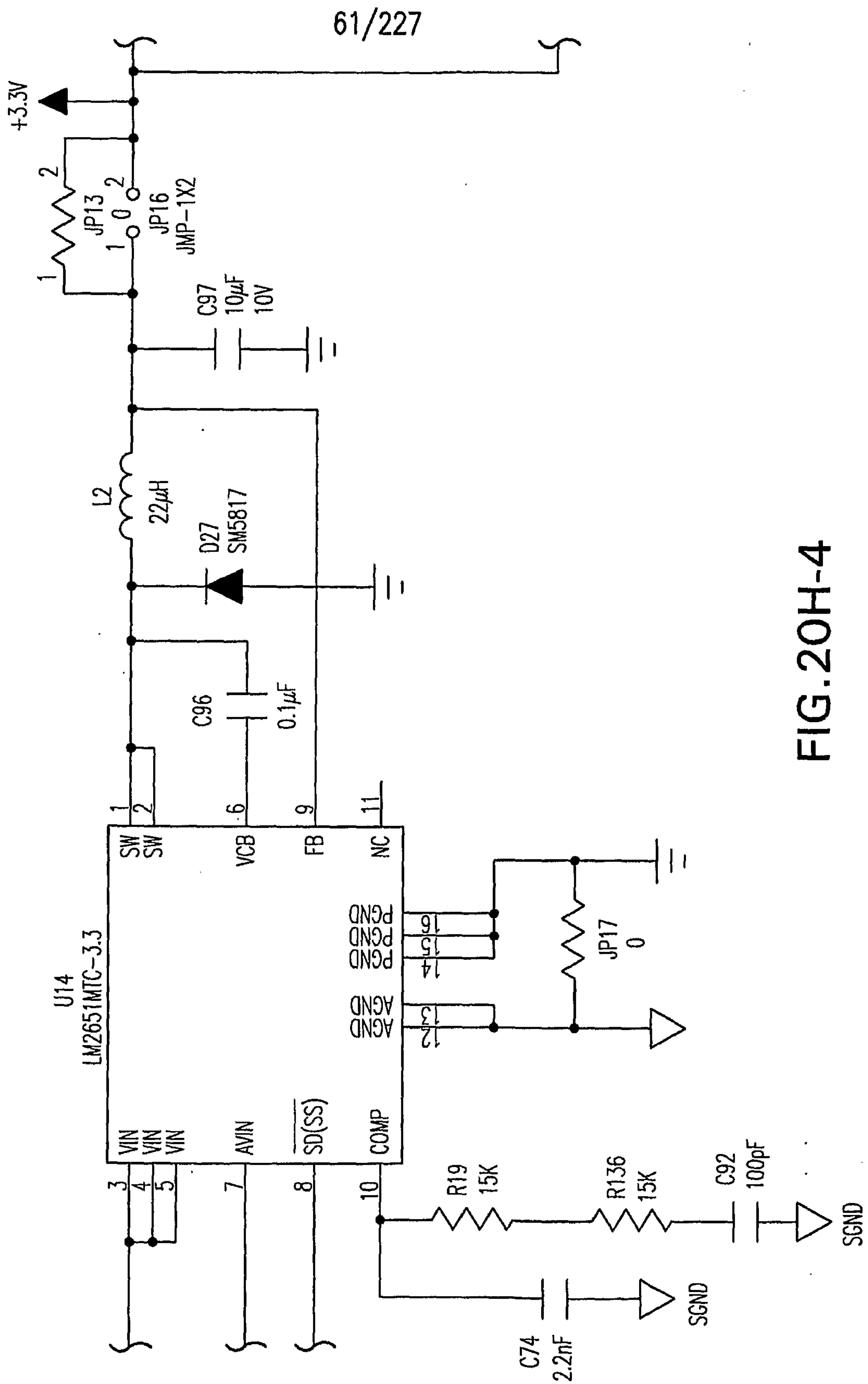


FIG. 20H-4

62/227

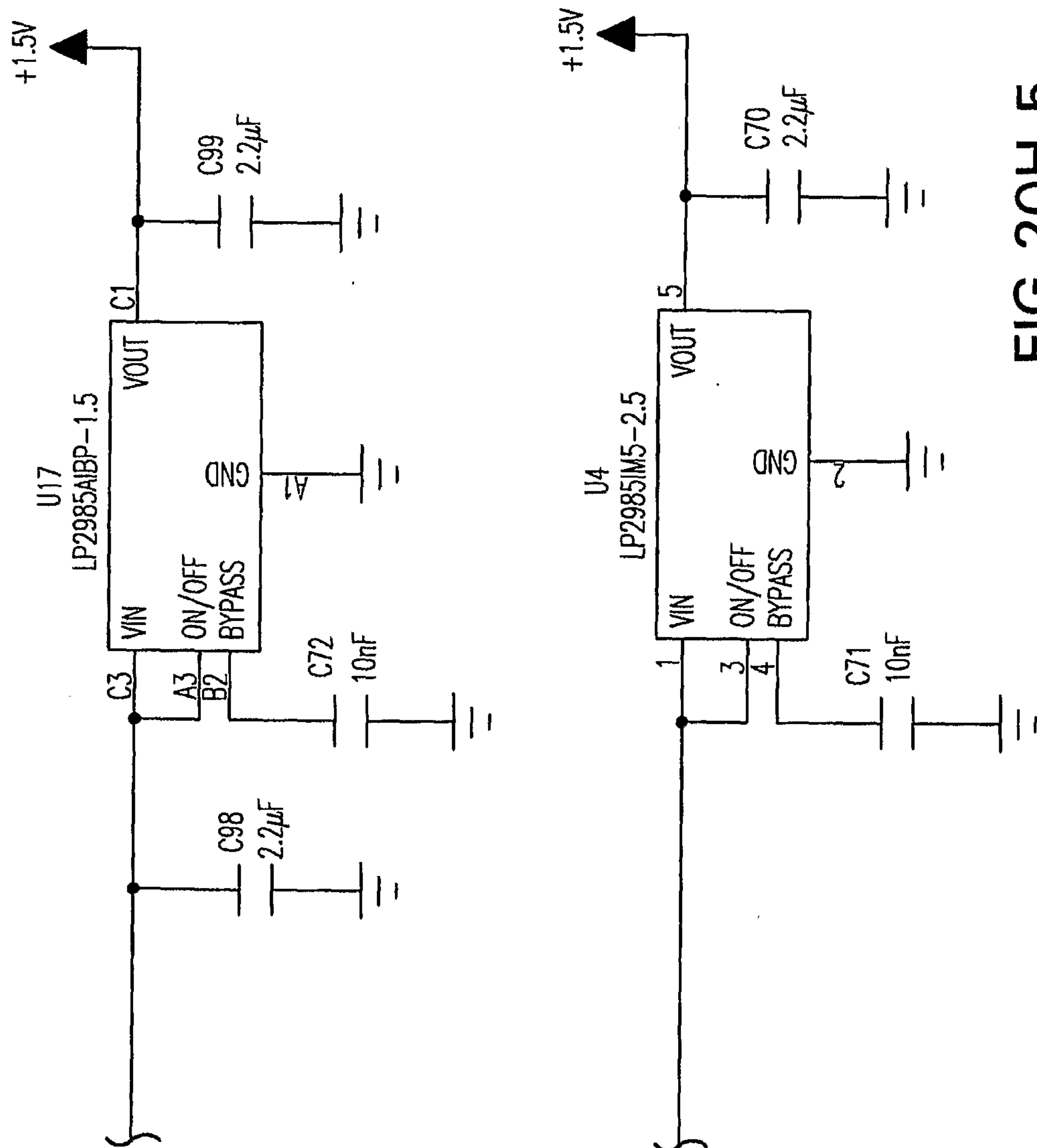


FIG. 20H-5

63/227

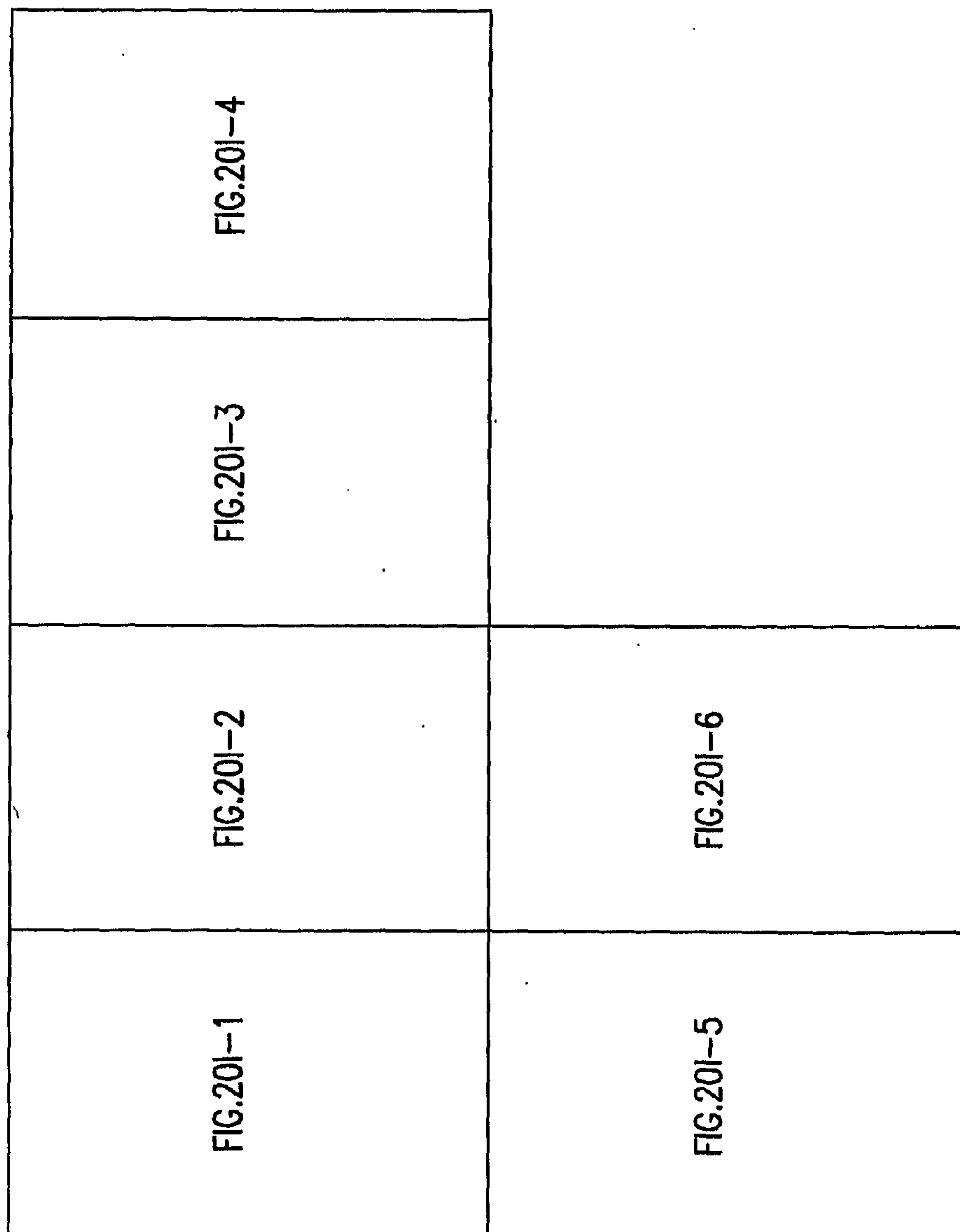


FIG.201

64/227

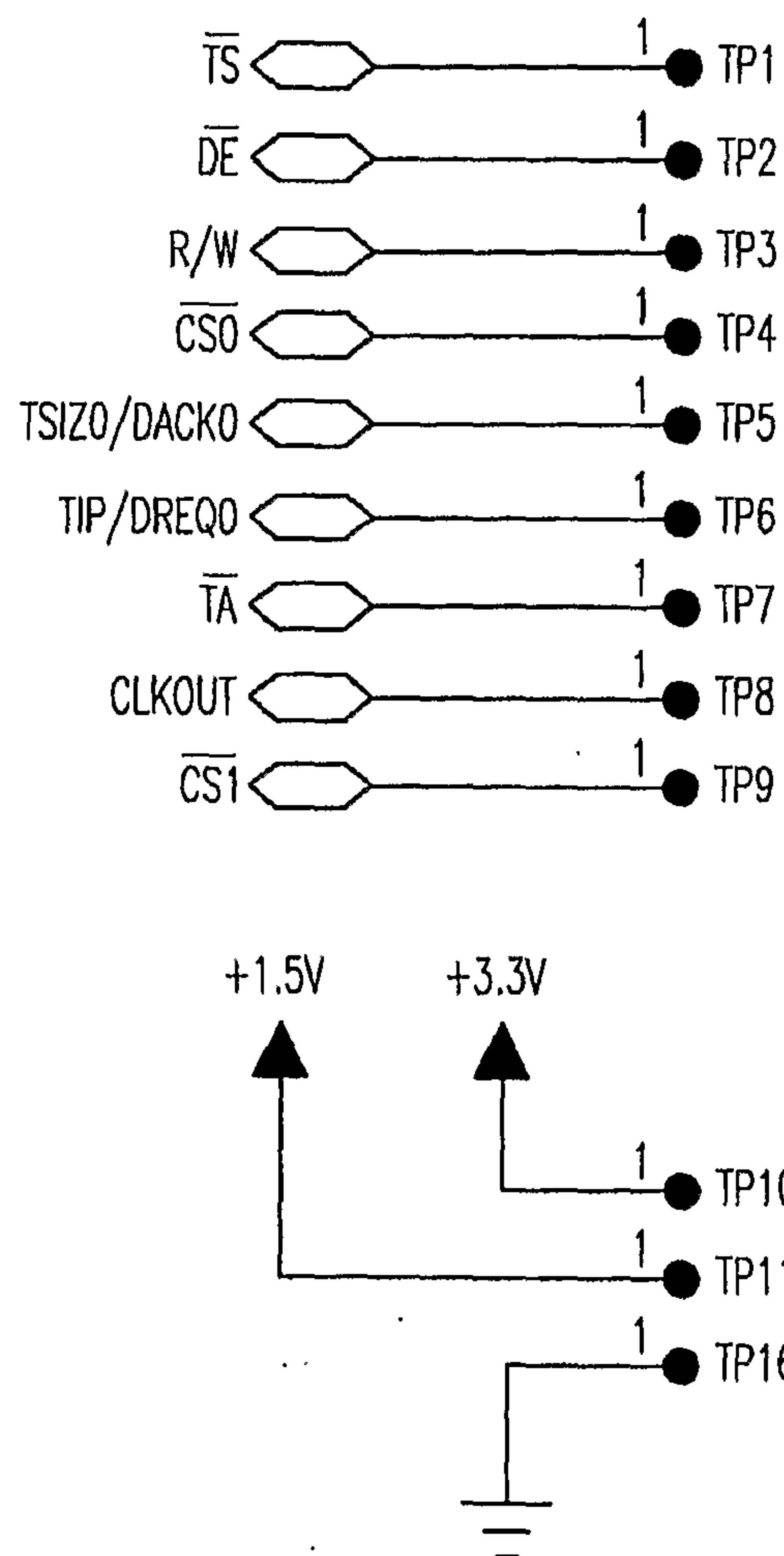


FIG.20I-1

65/227

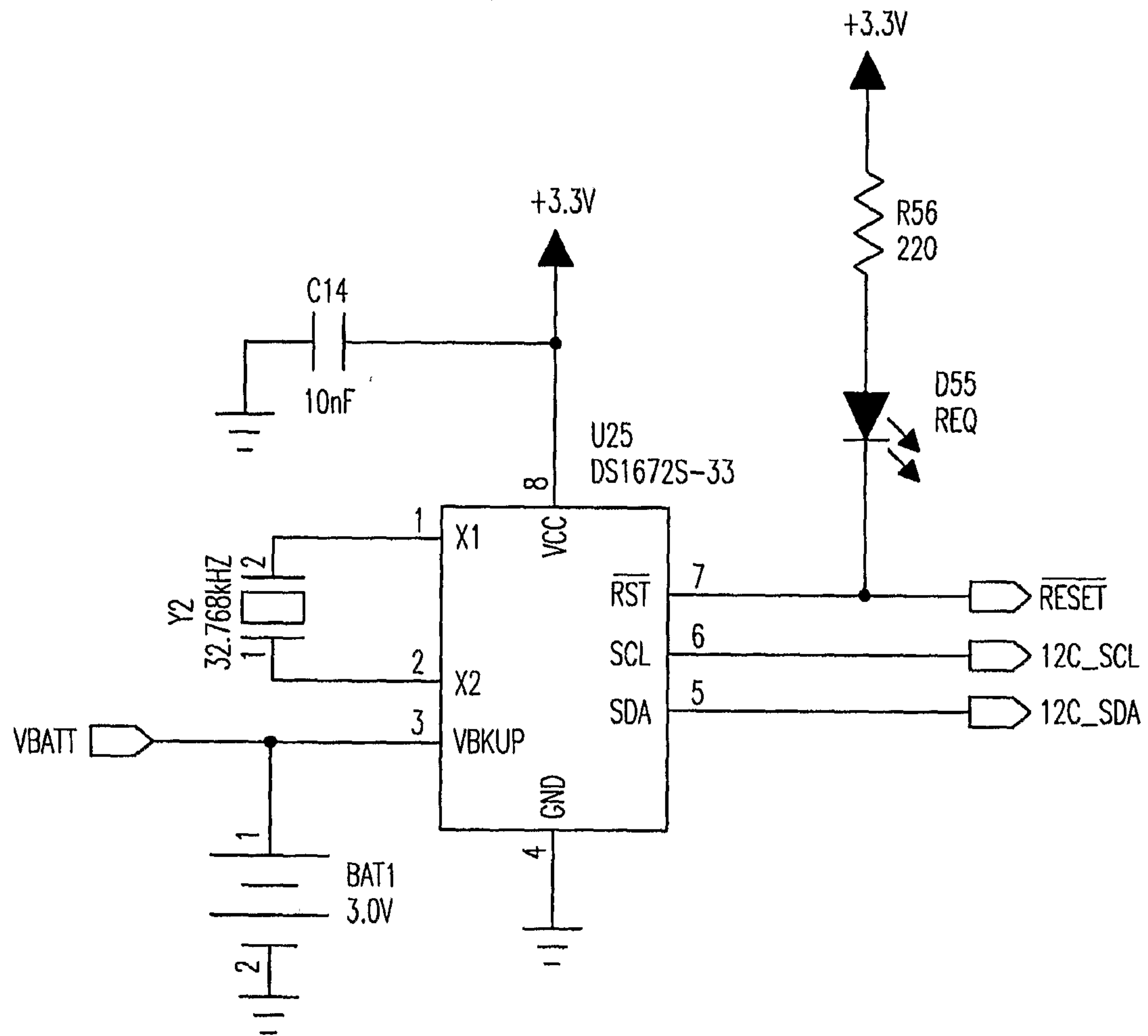


FIG.20I-2

66/227

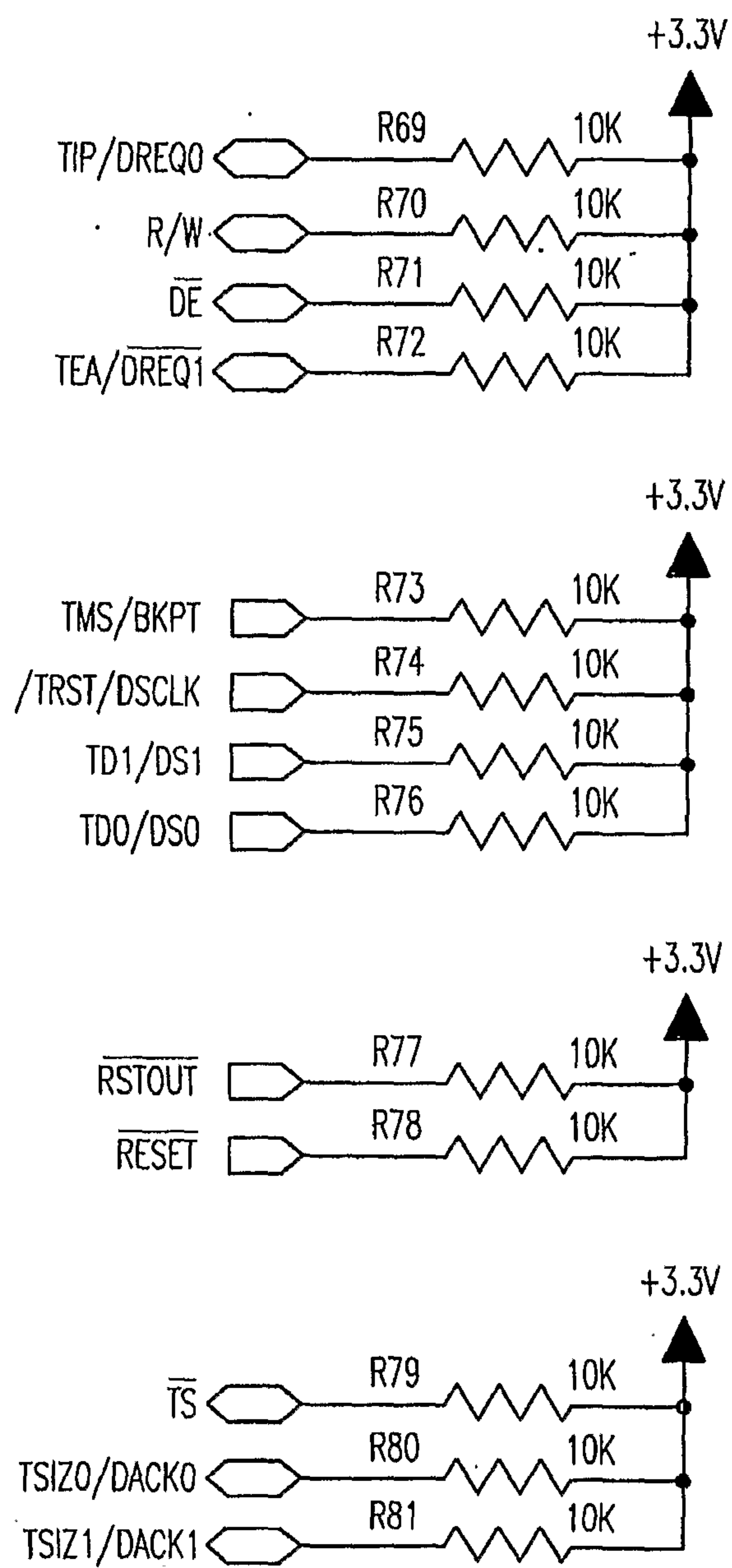


FIG.20I-3

67/227

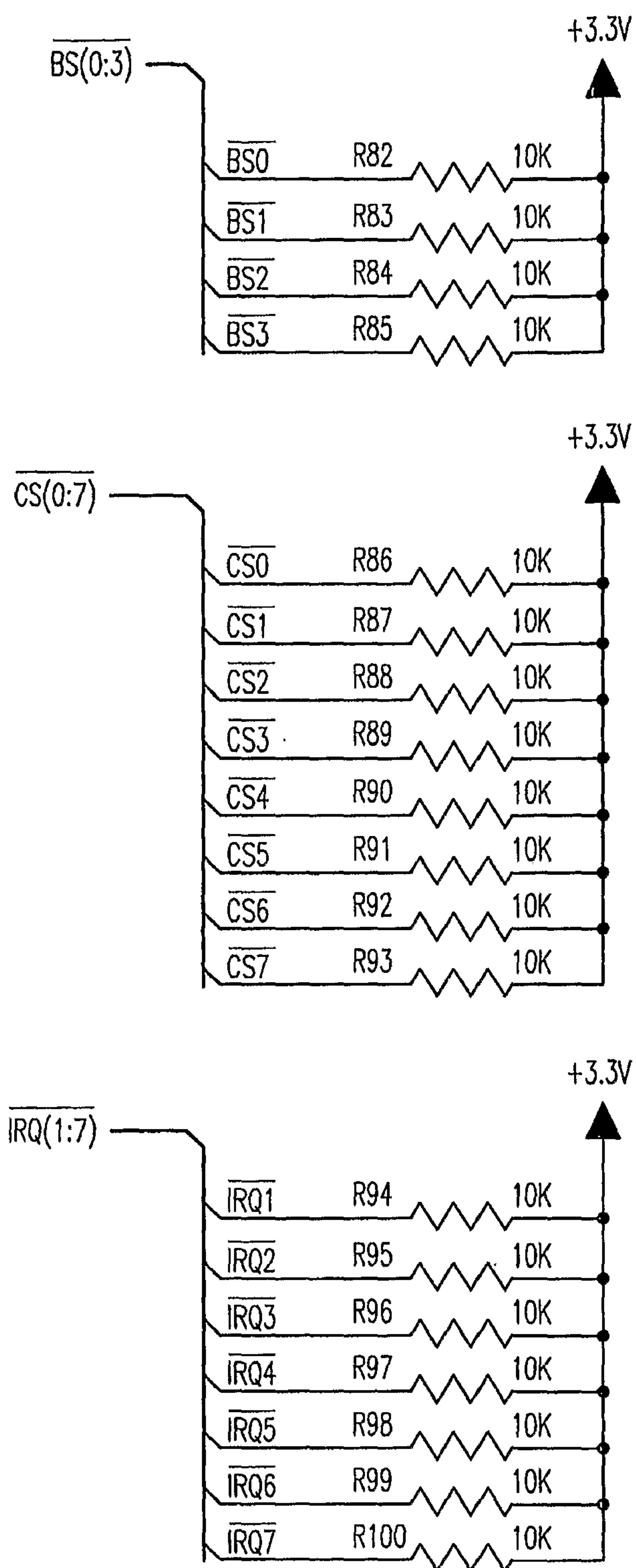


FIG.20I-4

68/227

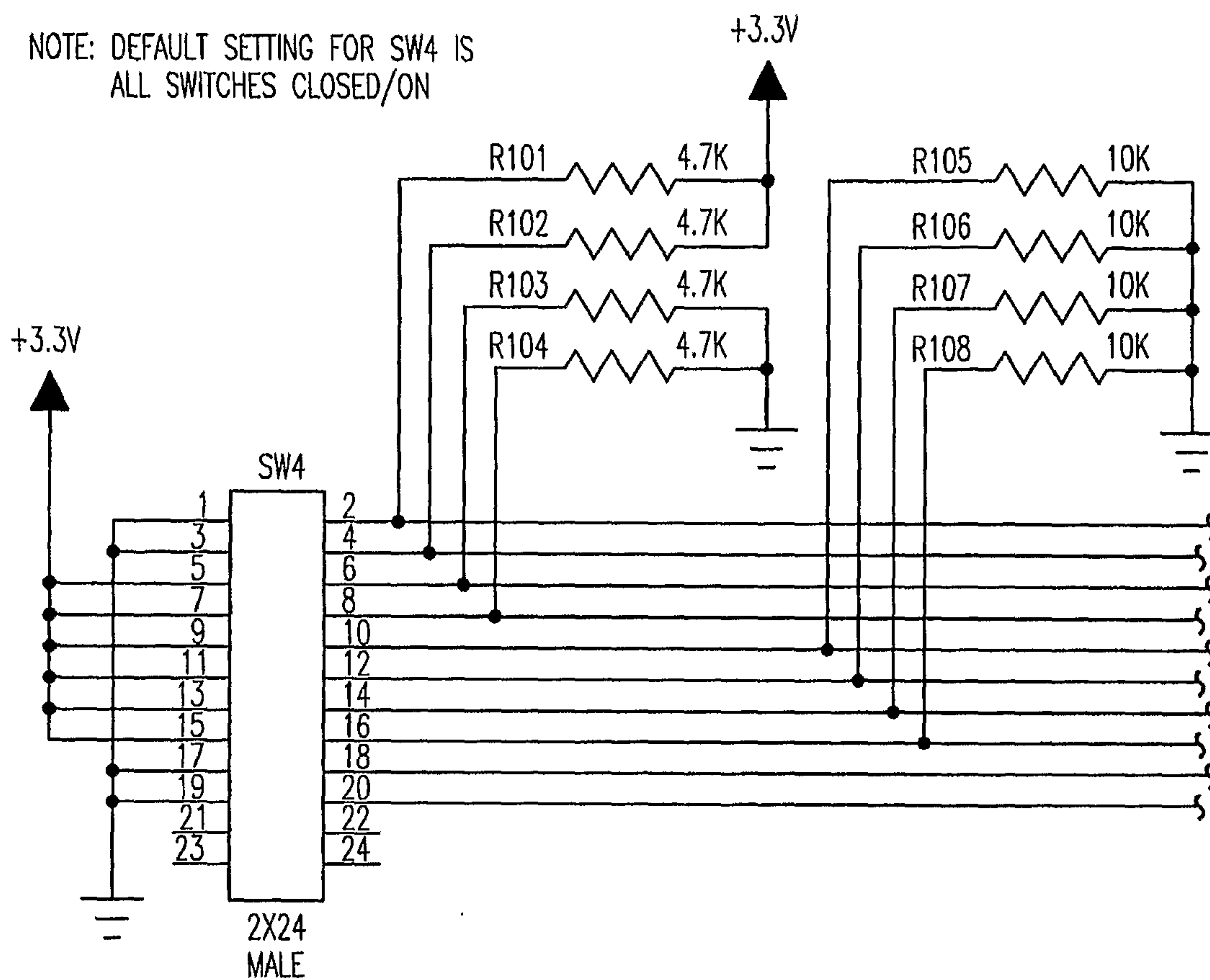
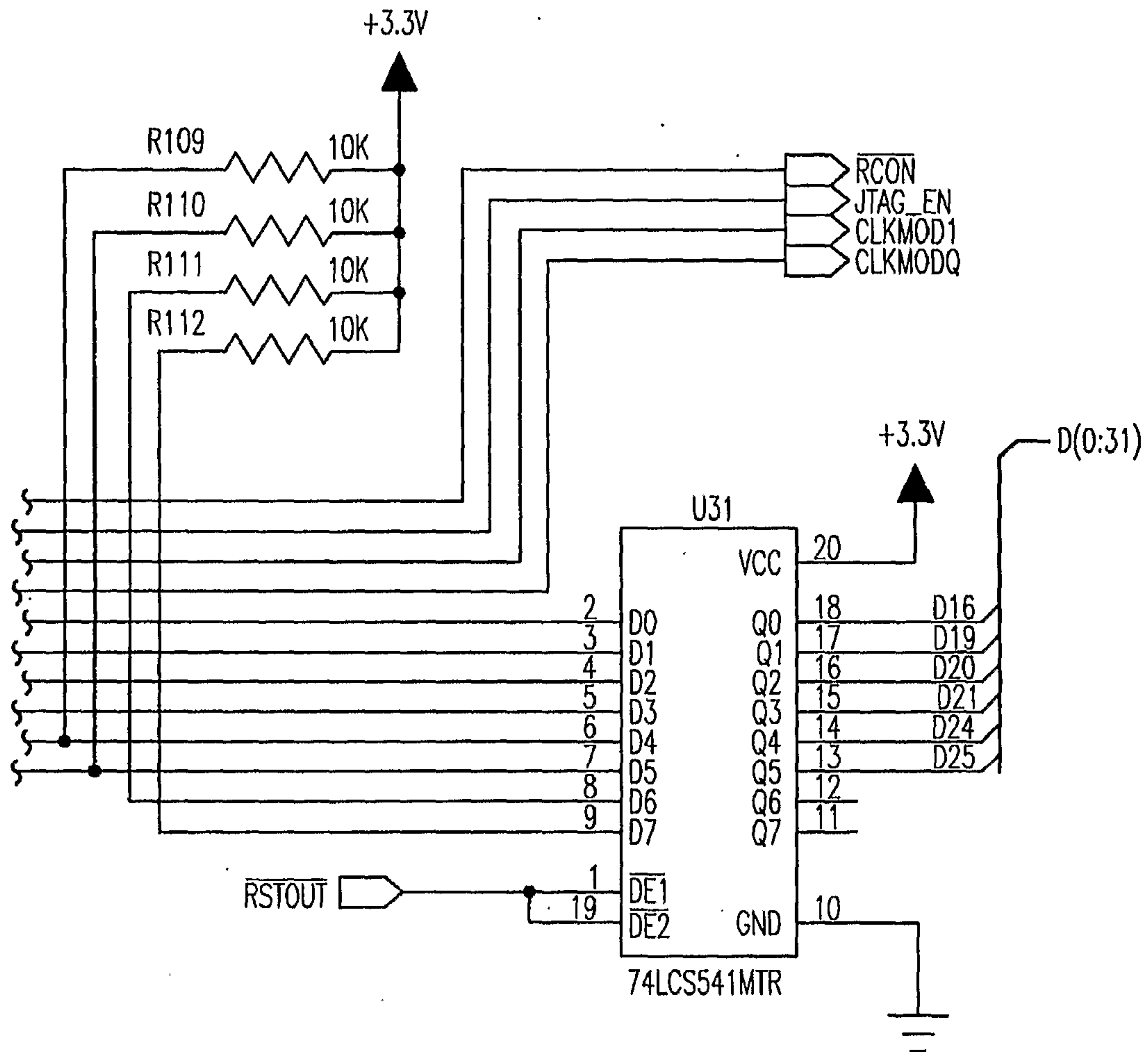


FIG.20I-5

69/227



IMPORTANT NOTE: THE /RSTOUT SIGNAL MUST BE USED TO DRIVE THE OUTPUT ENABLE PINS OF U19 TO ALLOW THE D16, D17, D18, D19, D21, D24, D25, & D26 SIGNALS TO BE LATCHED CORRECTLY BY THE MCF5270/1 FOR CONFIGURATION AT RESET

FIG.20I-6

70/227

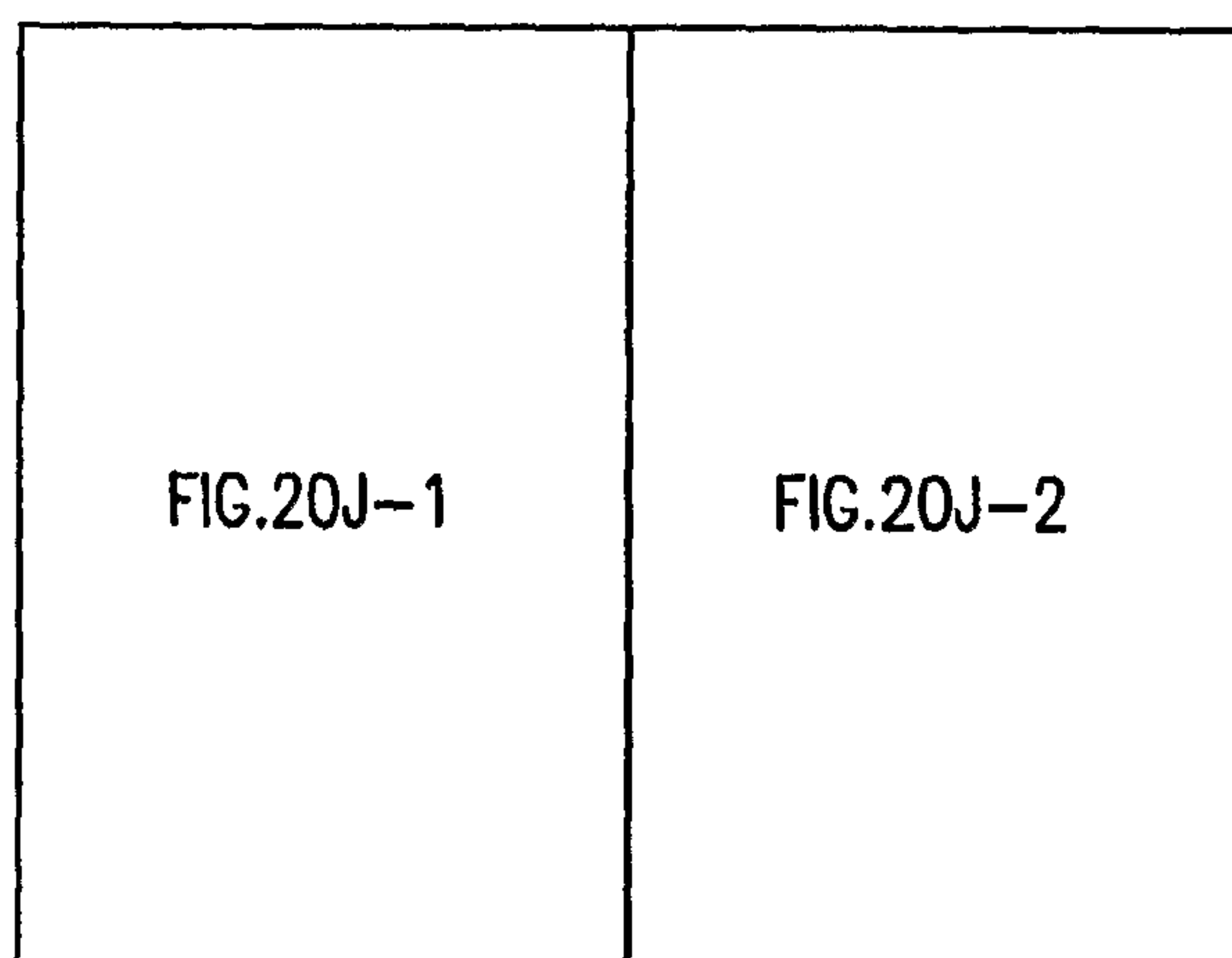
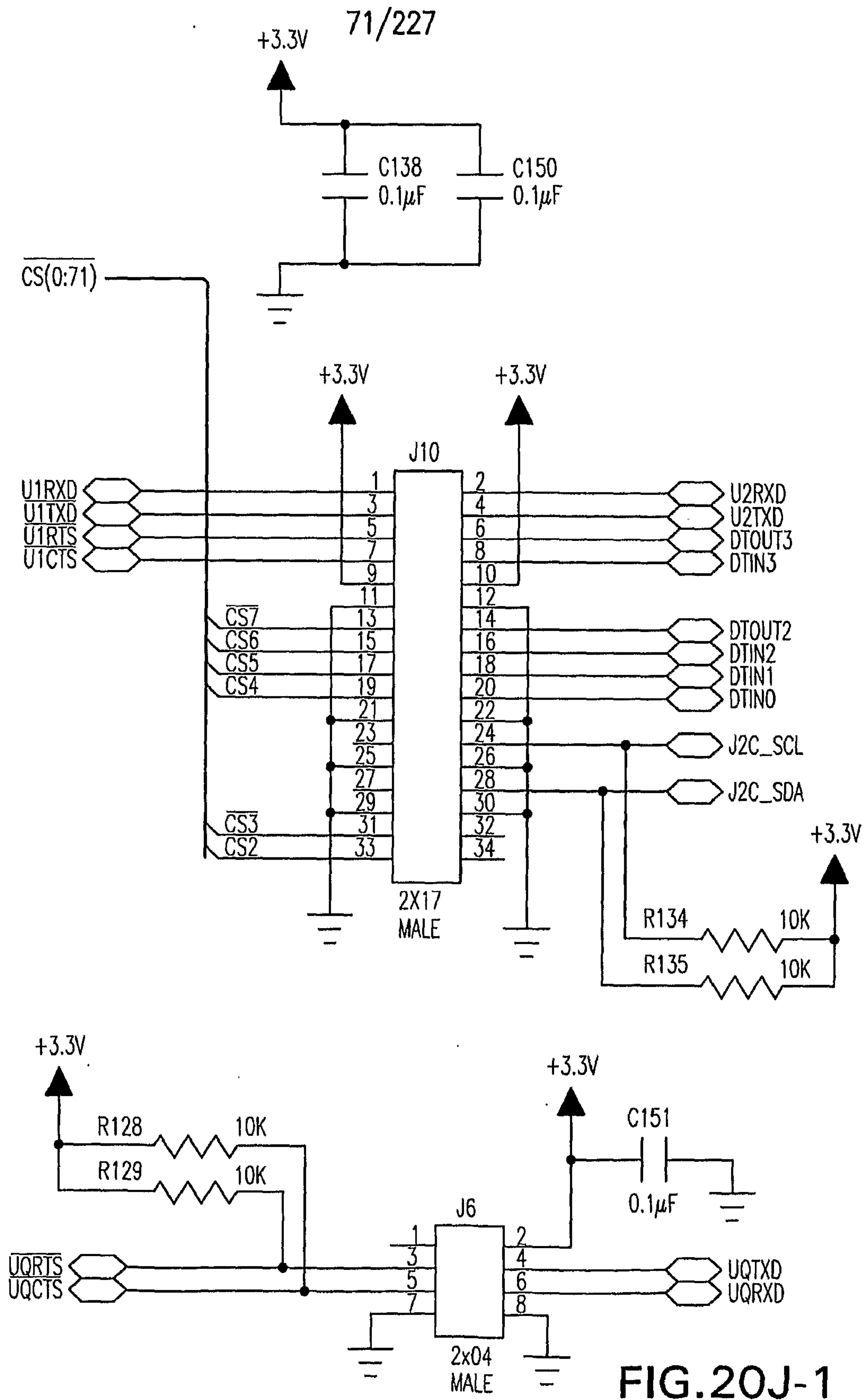


FIG.20J



72/227

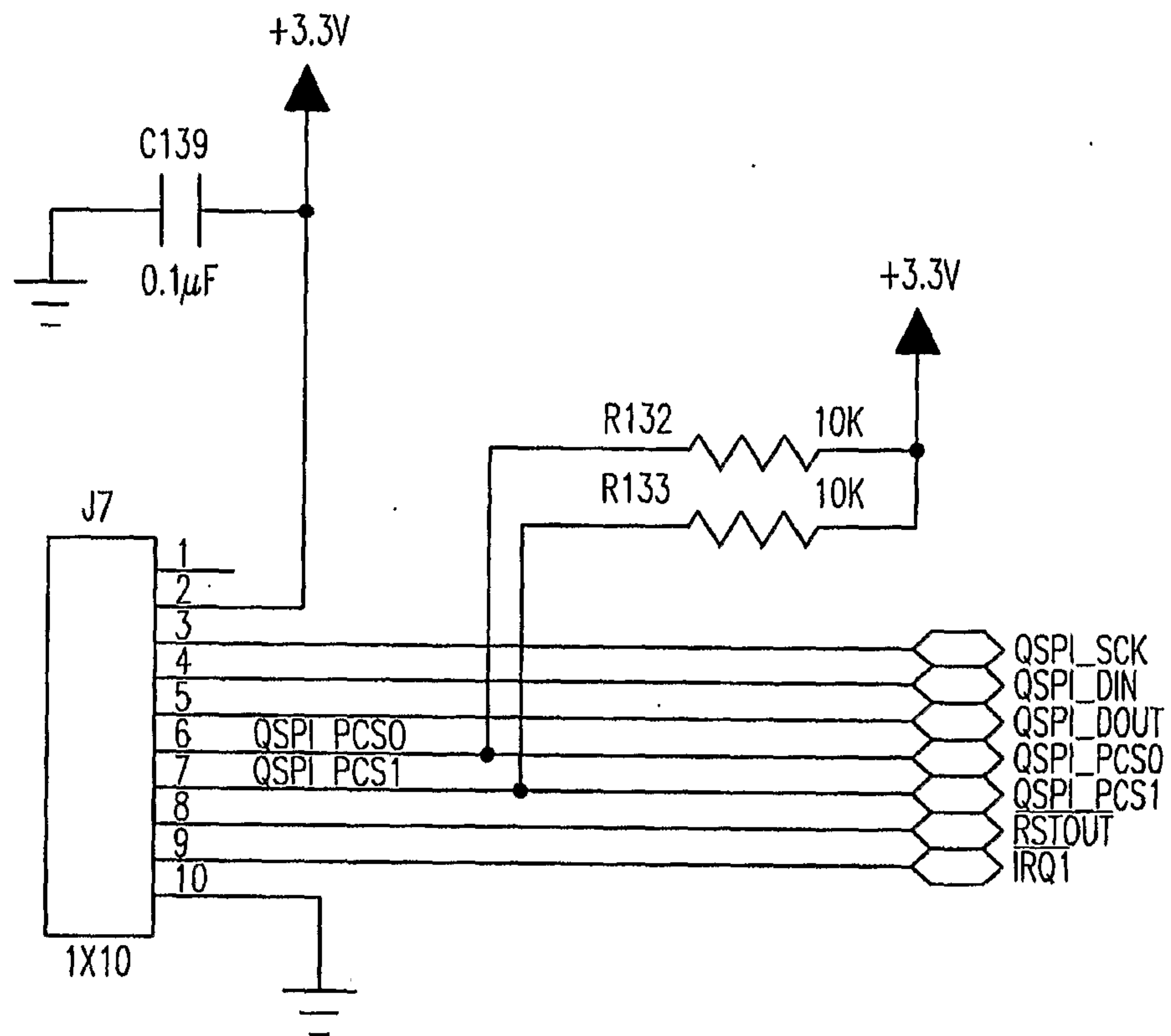


FIG.20J-2

73/227

FIG.20K-1	FIG.20K-2
FIG.20K-3	FIG.20K-4
FIG.20K-5	FIG.20K-6
FIG.20K-7	

FIG.20K

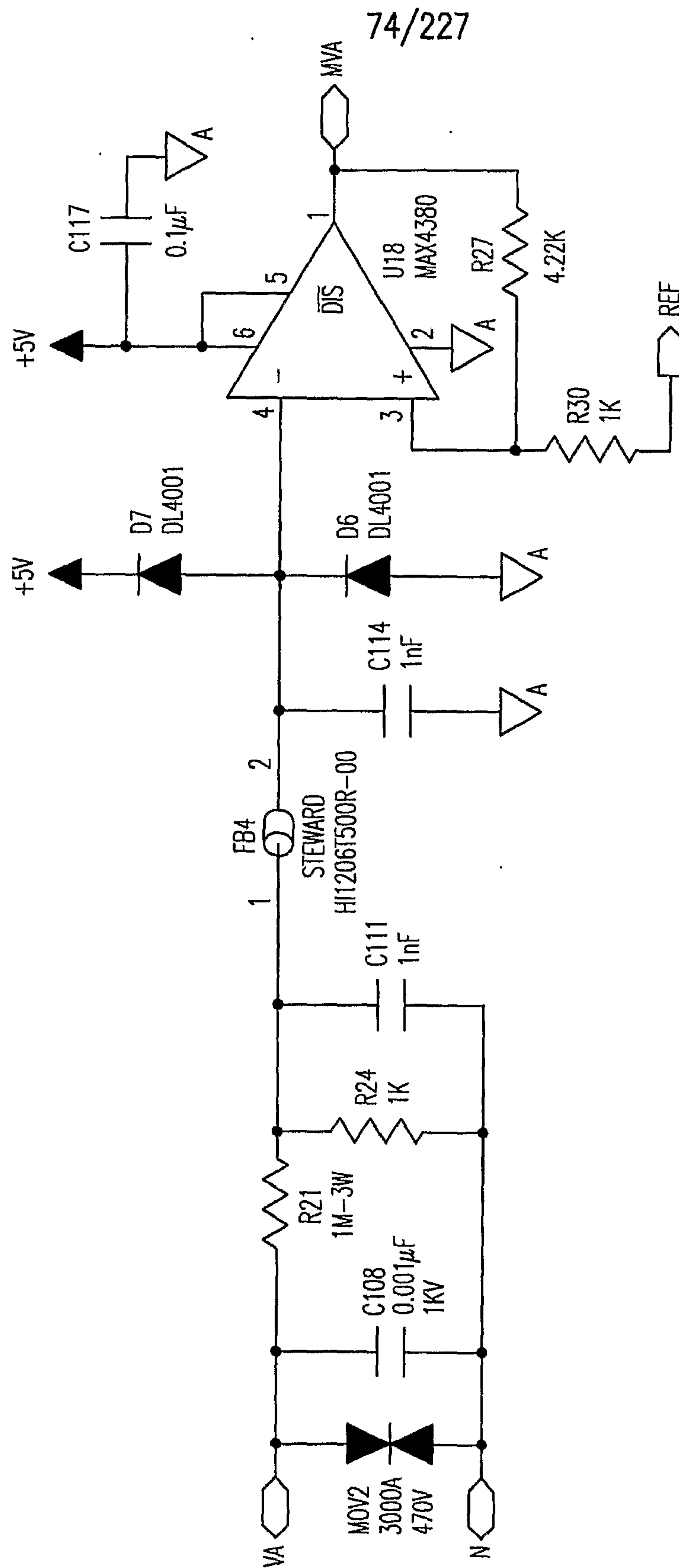


FIG.20K-1

75/227

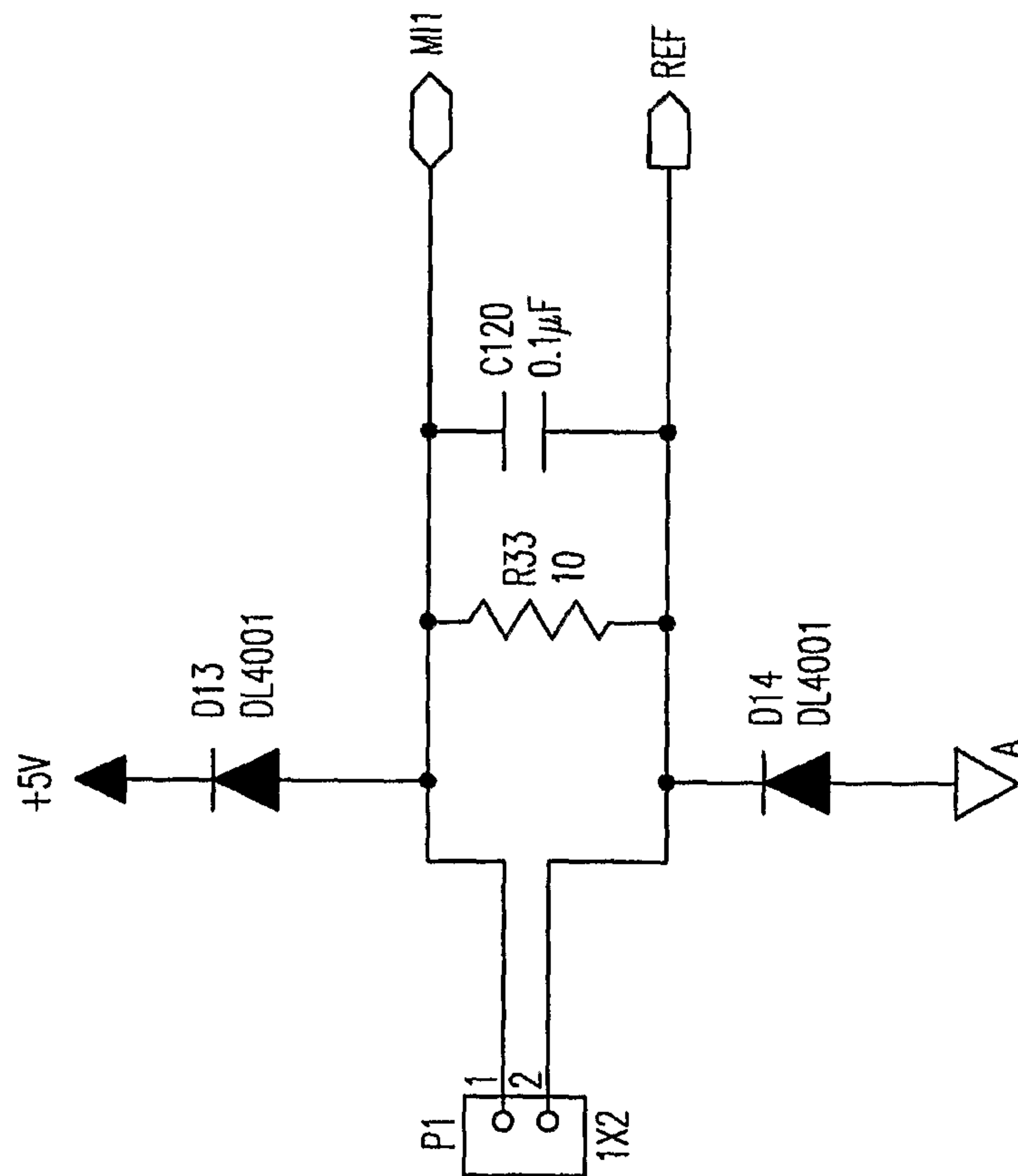


FIG. 20K-2

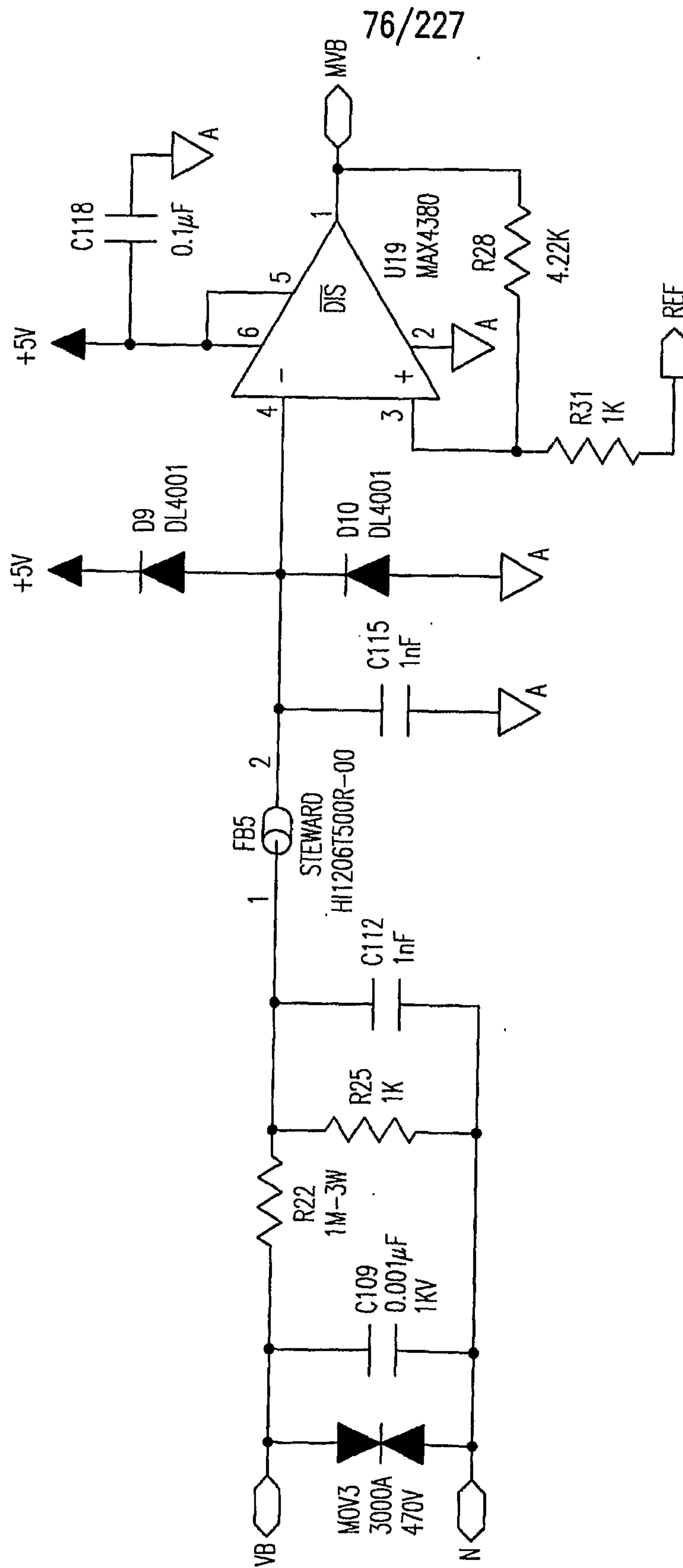


FIG. 20K-3

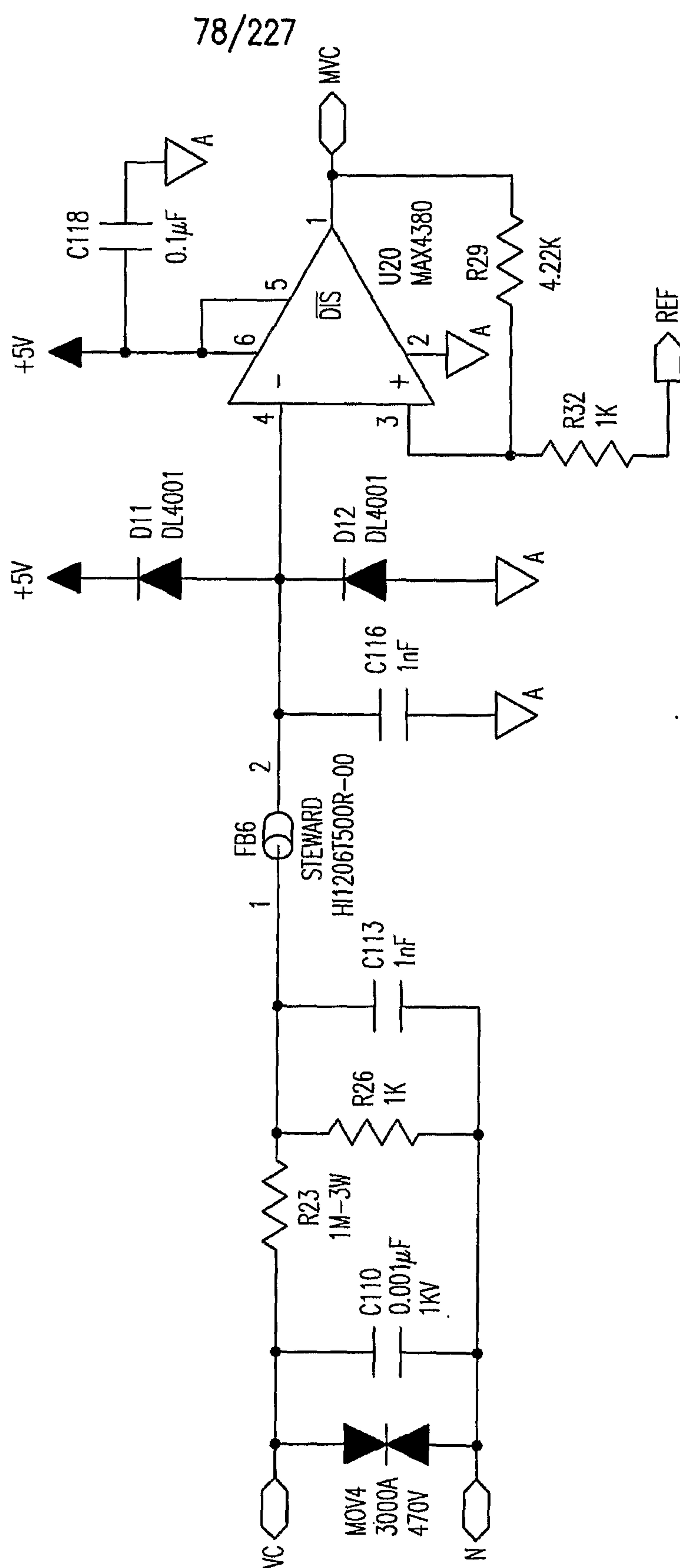


FIG. 20K-5

79/227

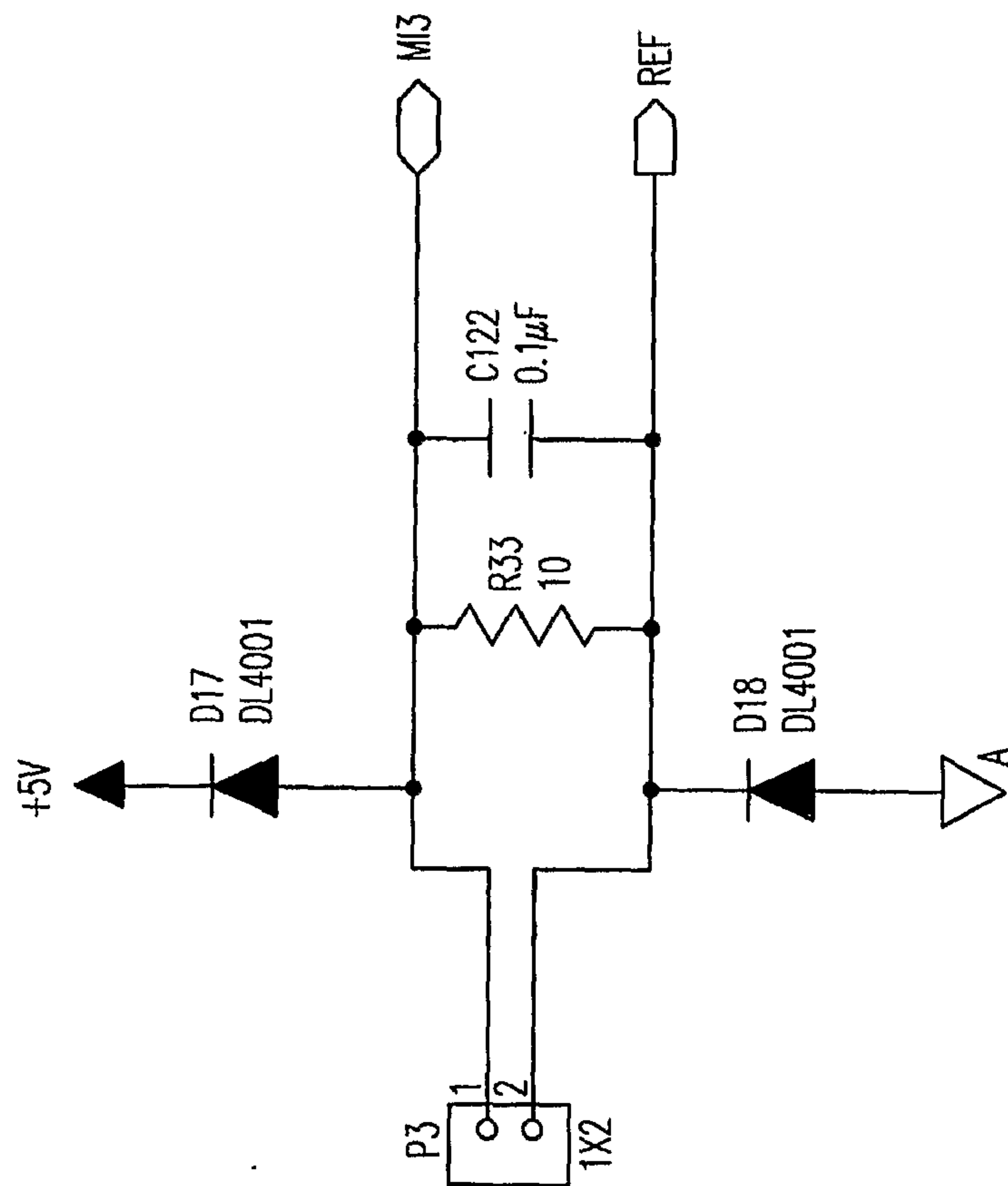


FIG. 20K-6

80/227

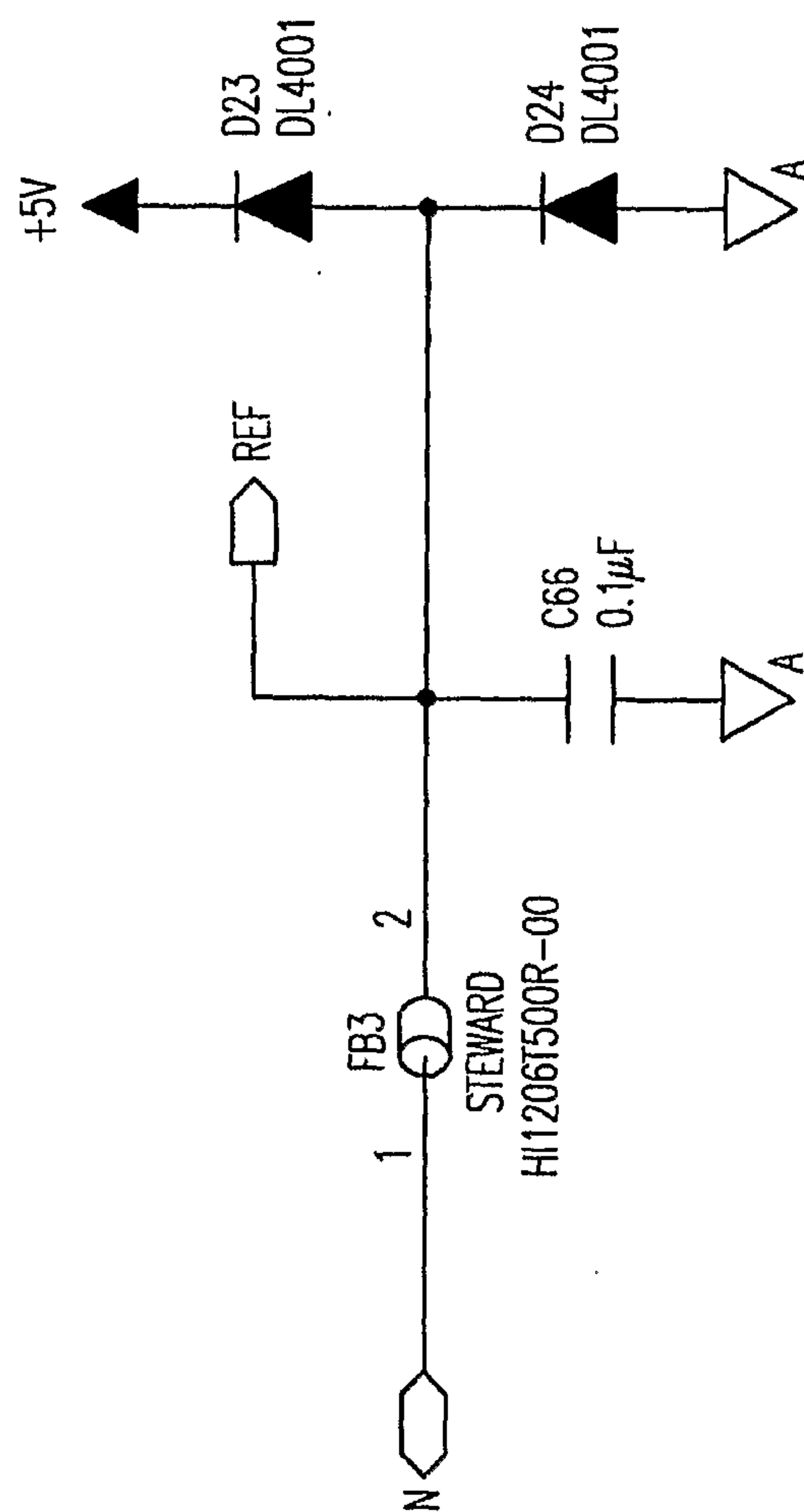


FIG. 20K-7

81/227

FIG.20L-1	FIG.20L-2	FIG.20L-3	FIG.20L-4
FIG.20L-5	FIG.20L-6	FIG.20L-7	FIG.20L-8

FIG.20L

82/227

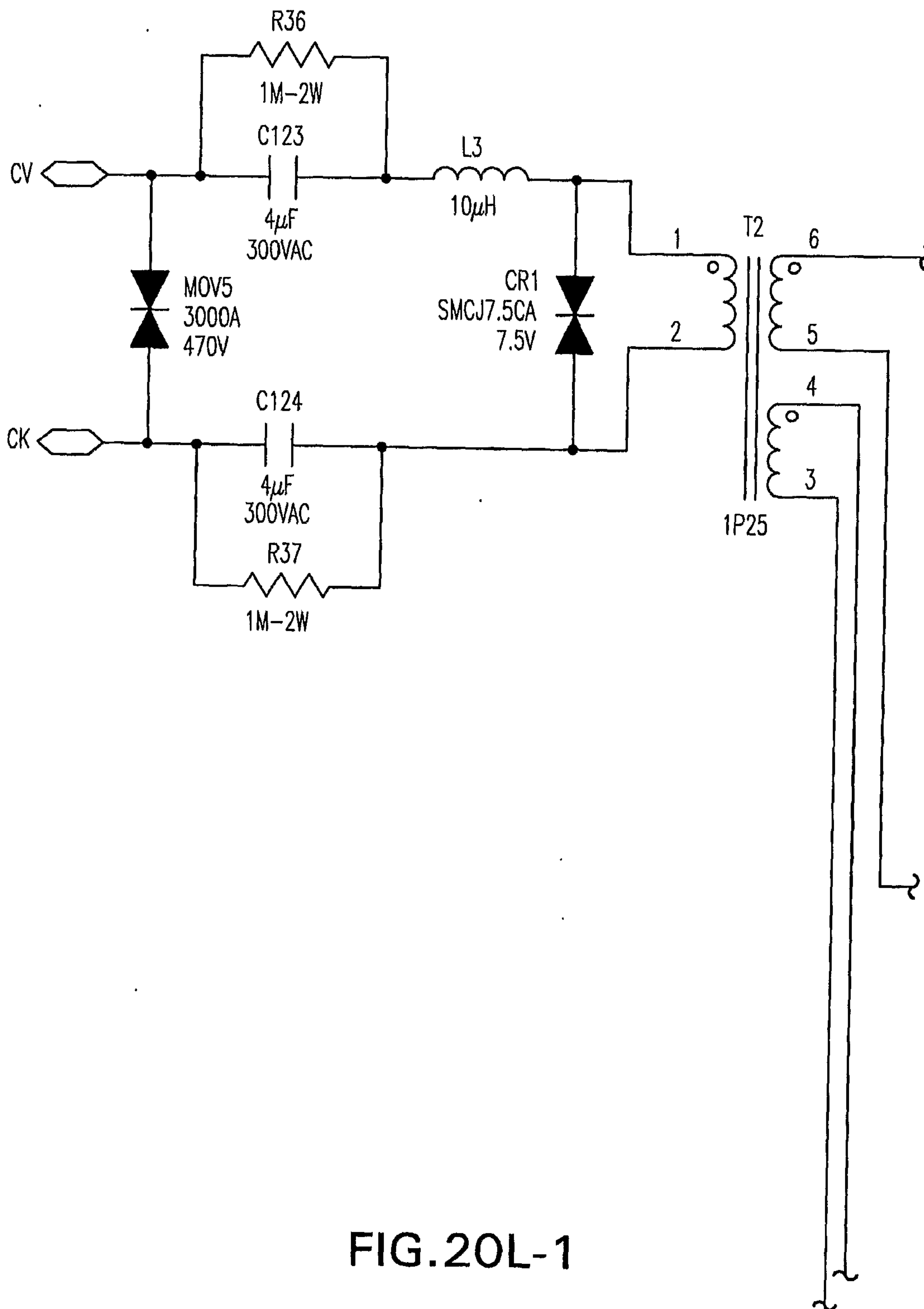


FIG. 20L-1

83/227

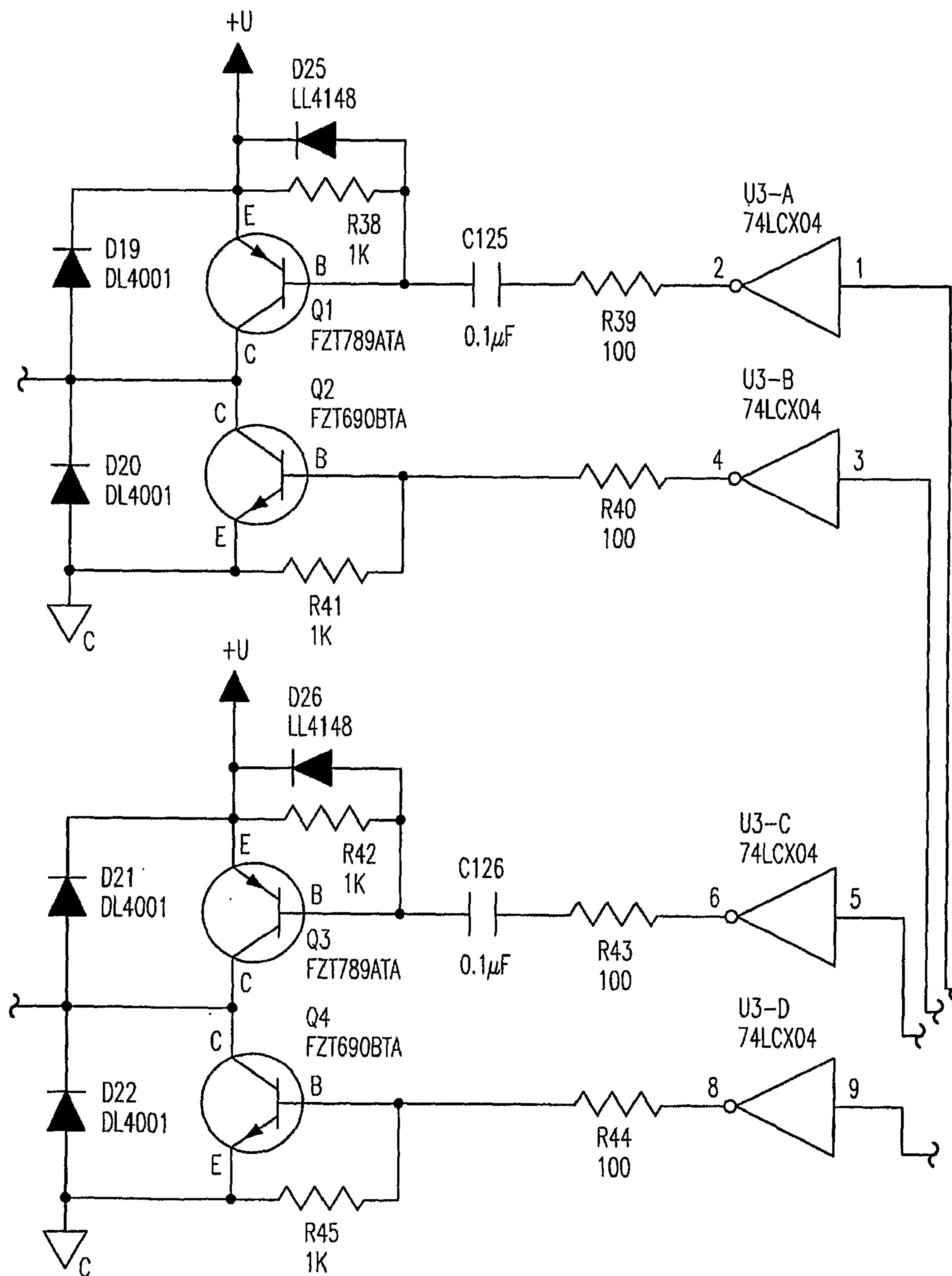


FIG. 20L-2

84/227

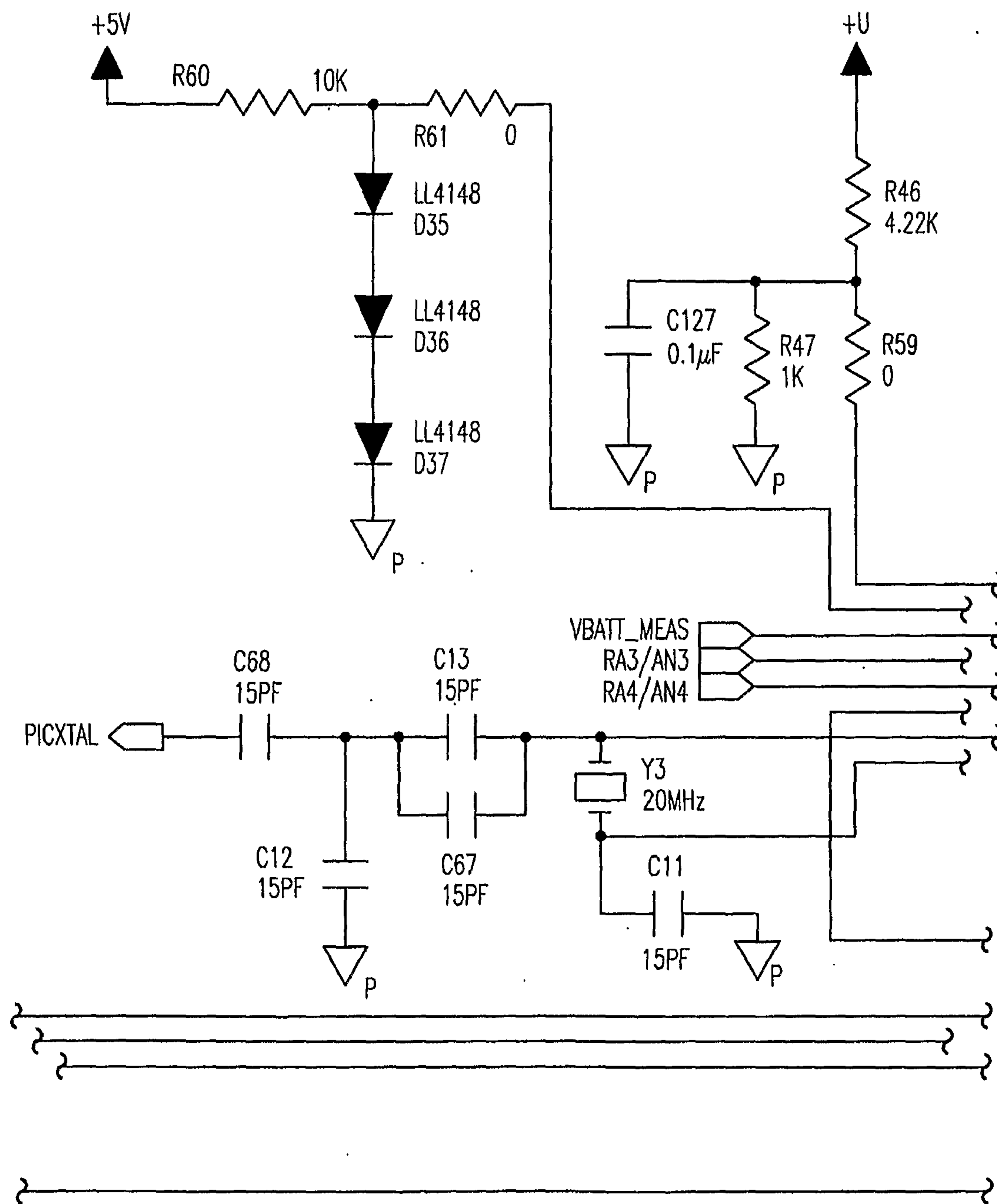


FIG.20L-3

85/227

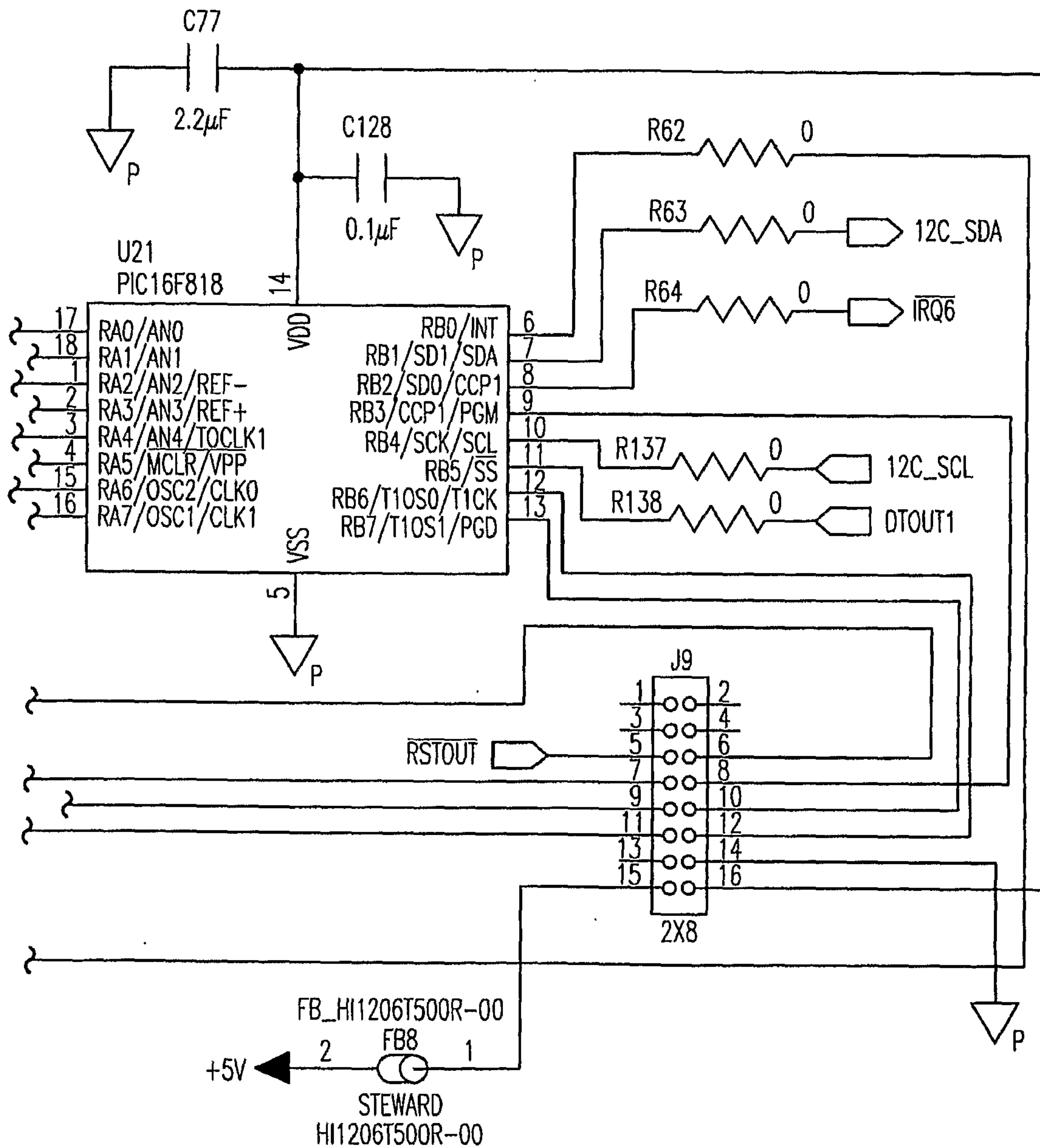


FIG.20L-4

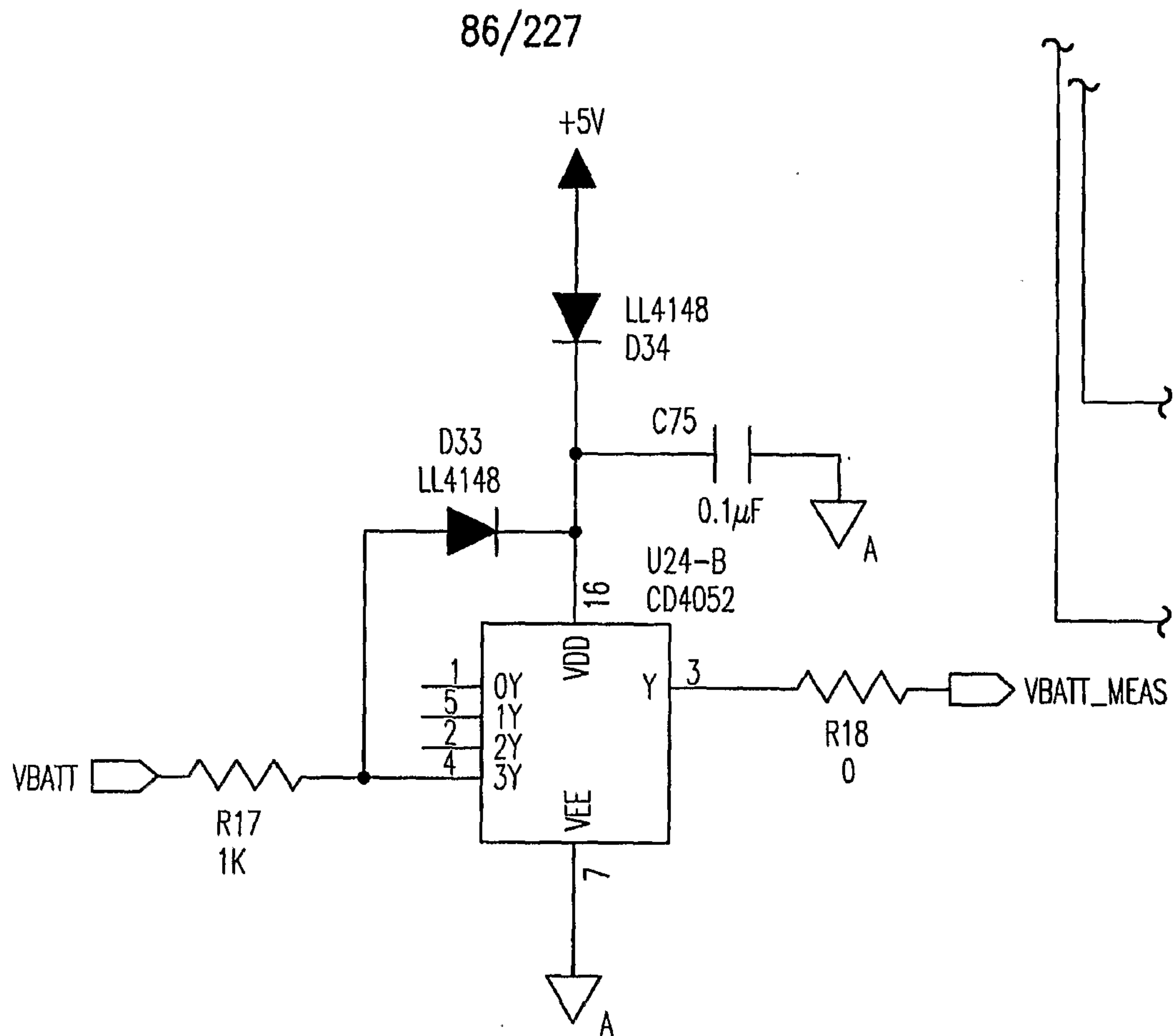


FIG.20L-5

87/227

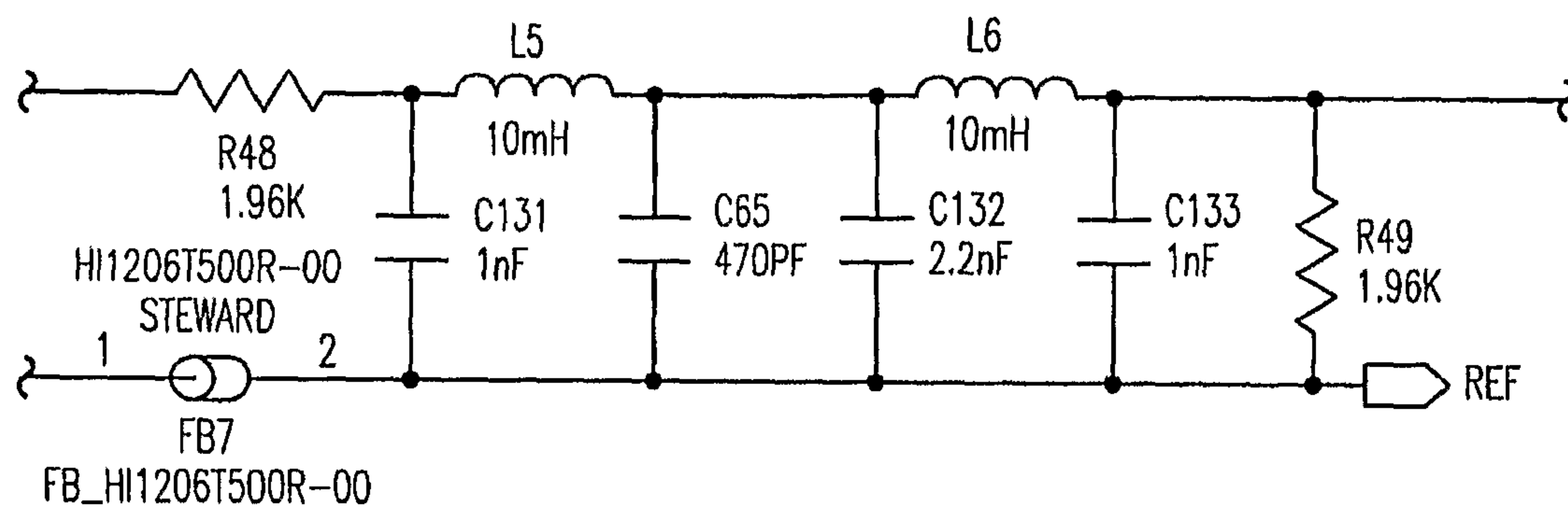


FIG.20L-6

88/227

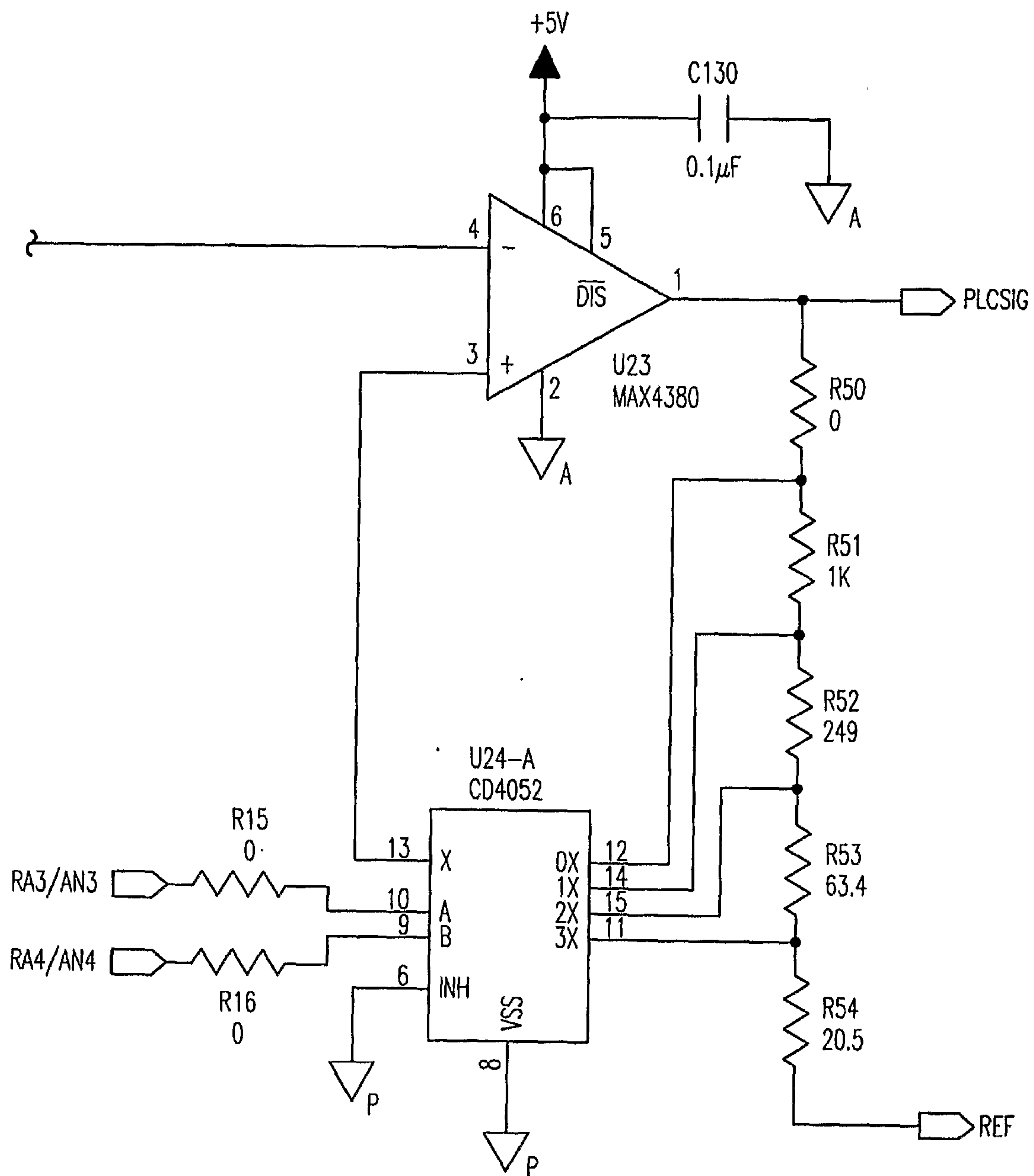


FIG. 20L-7

89/227

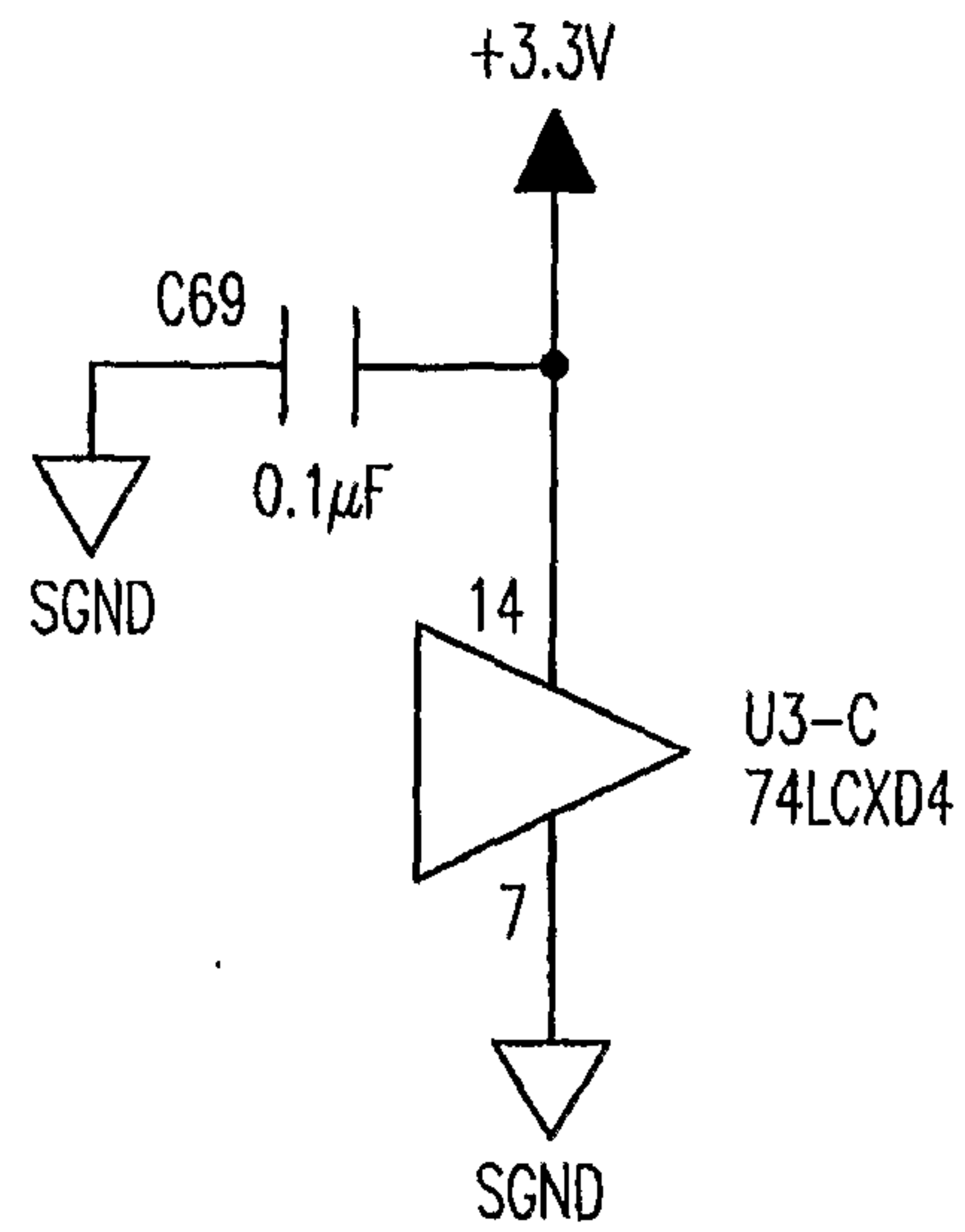
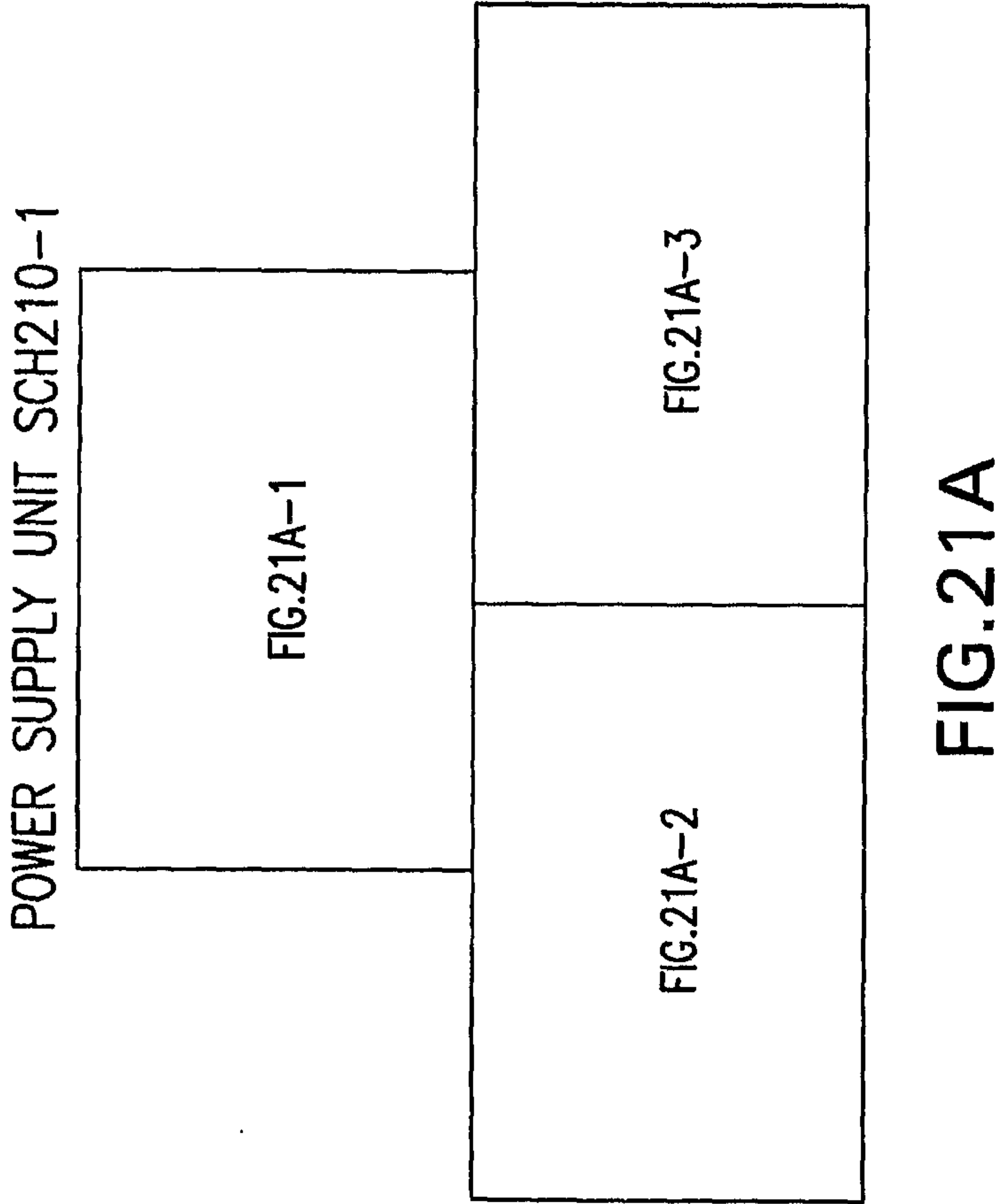


FIG.20L-8

90/227



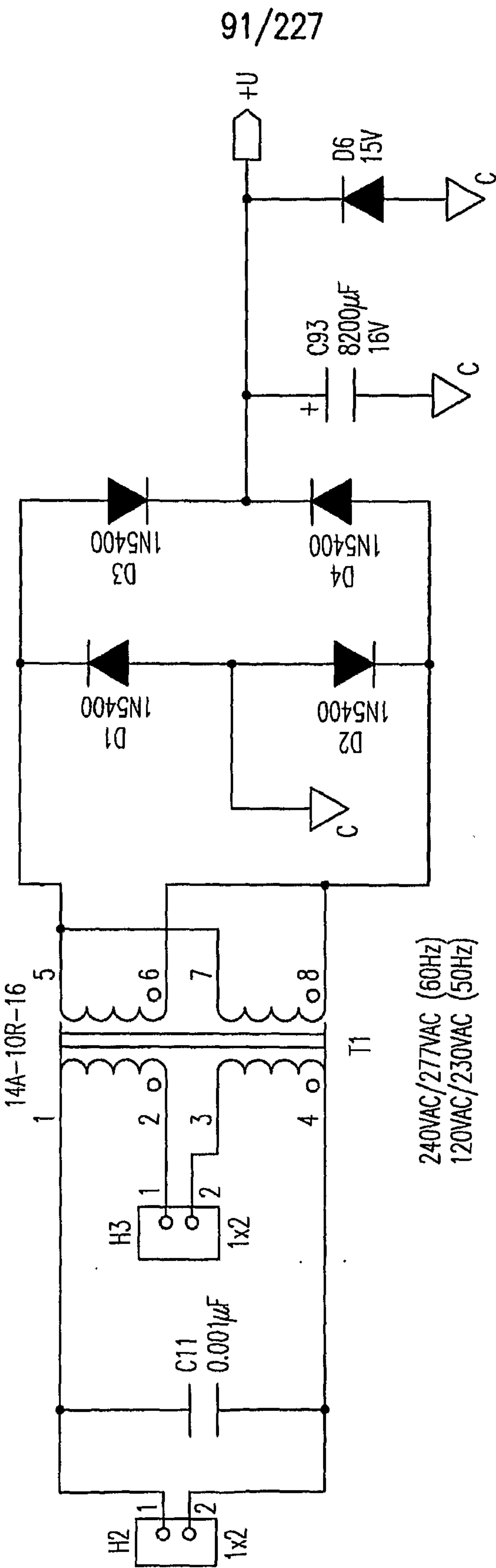


FIG. 21A-1

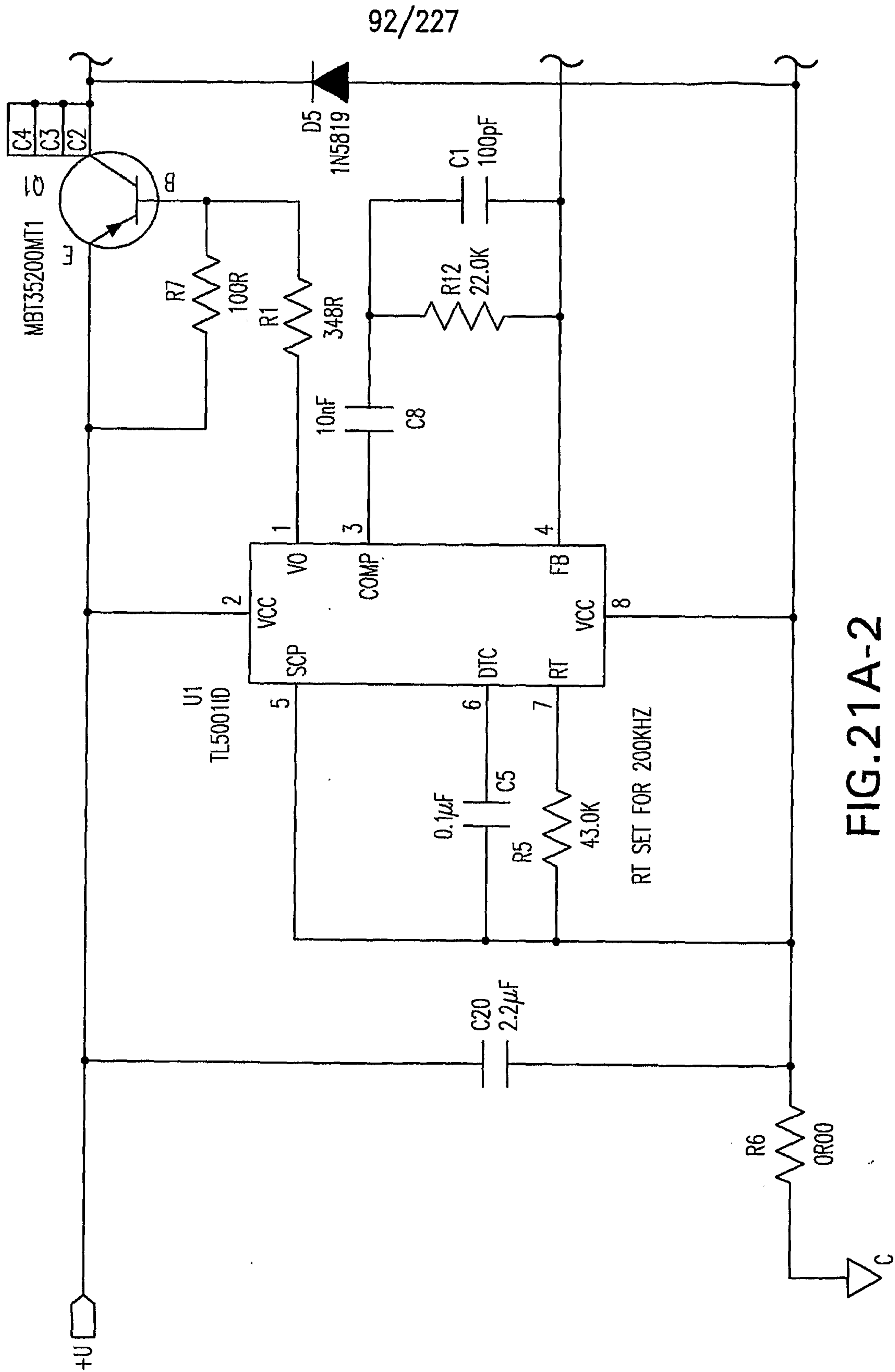


FIG.21A-2

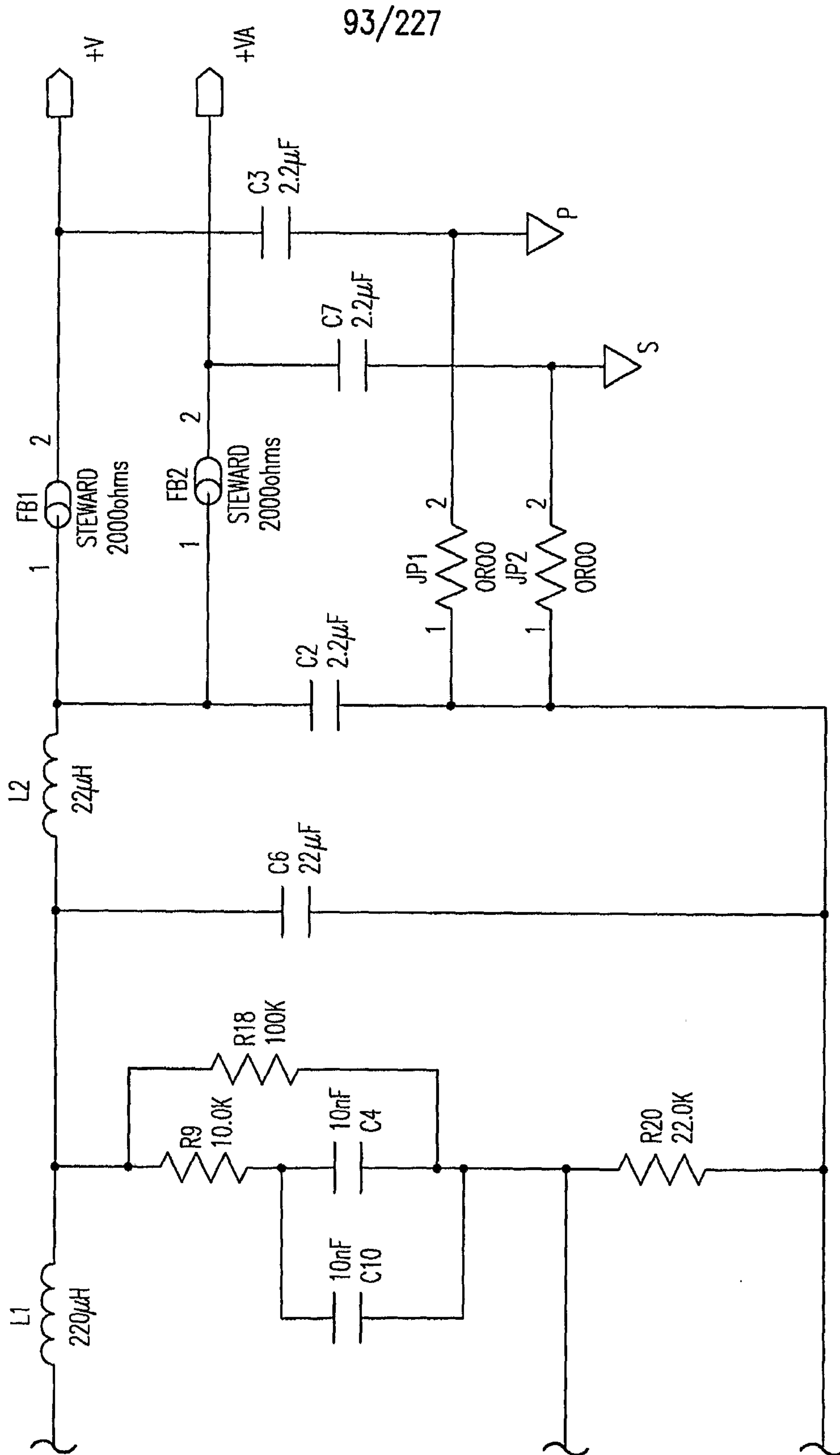
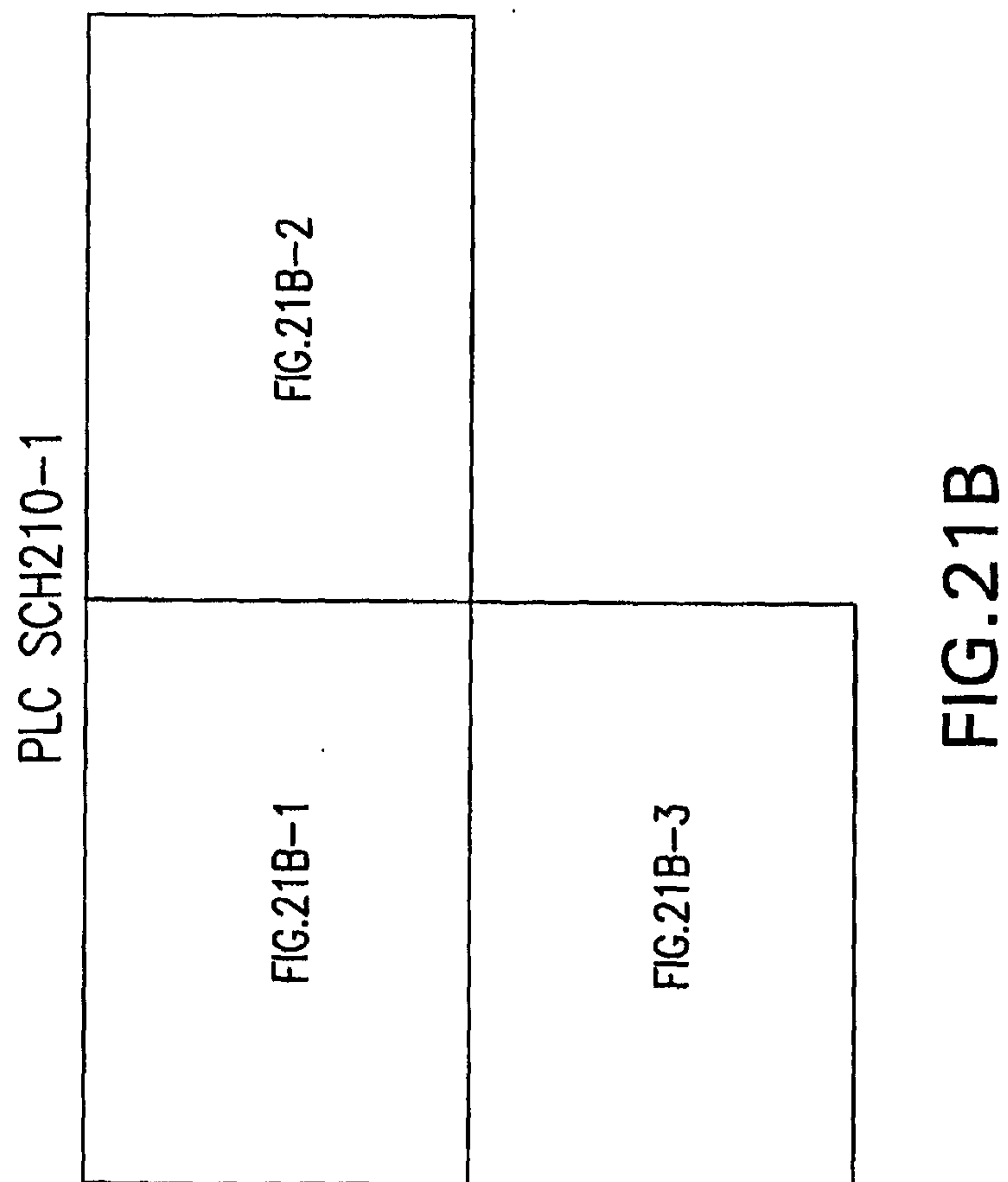


FIG. 21A-3

94/227



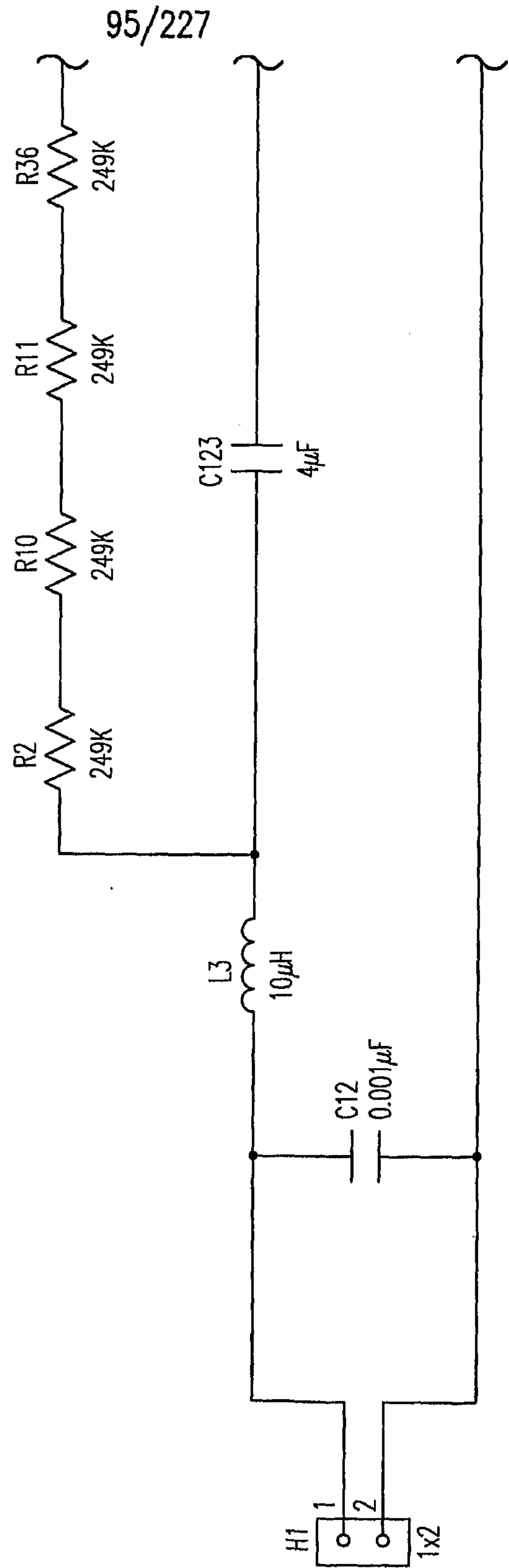


FIG.21B-1

96/227

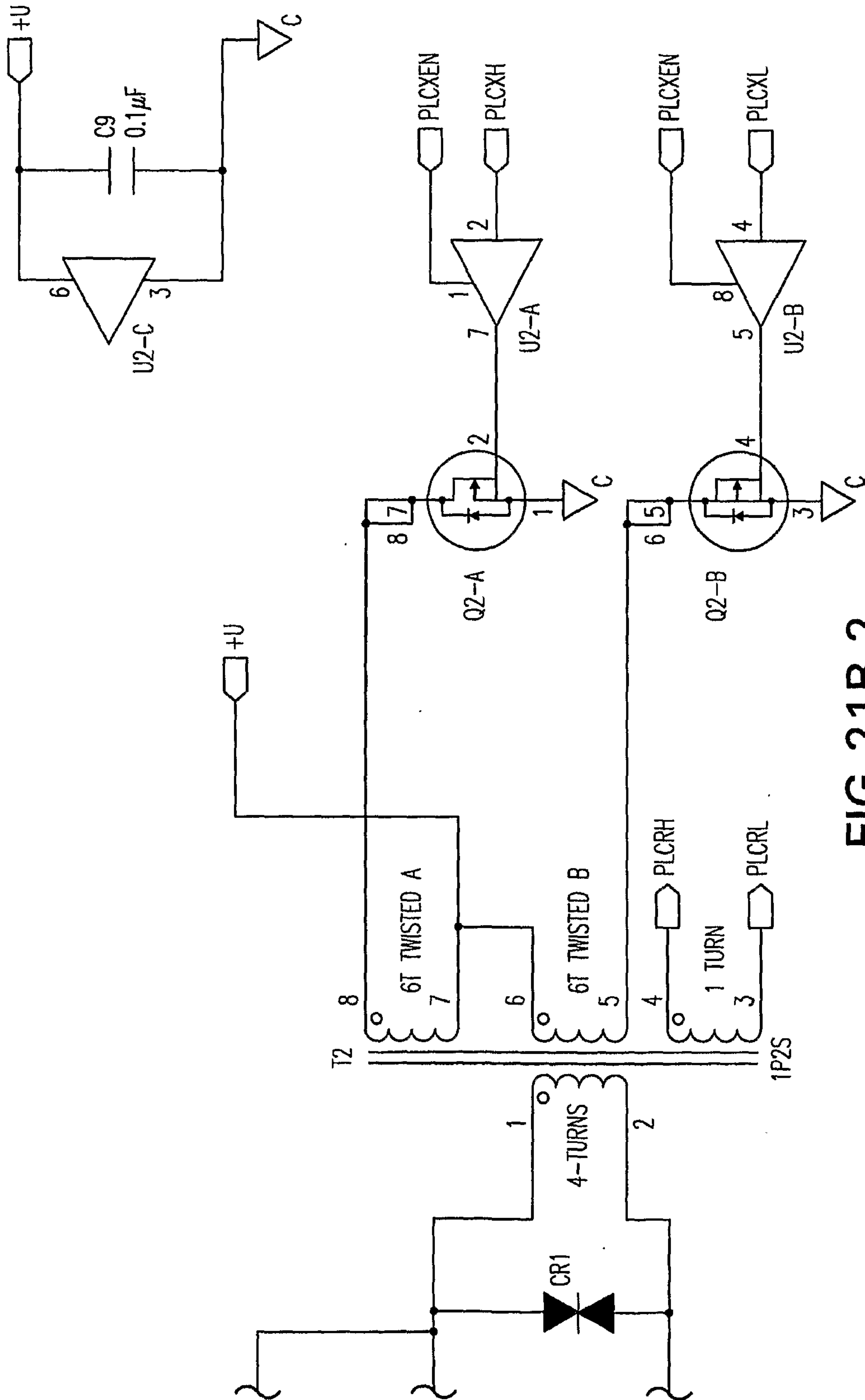


FIG. 21B-2

97/227

HEADER TO CPU BOARD

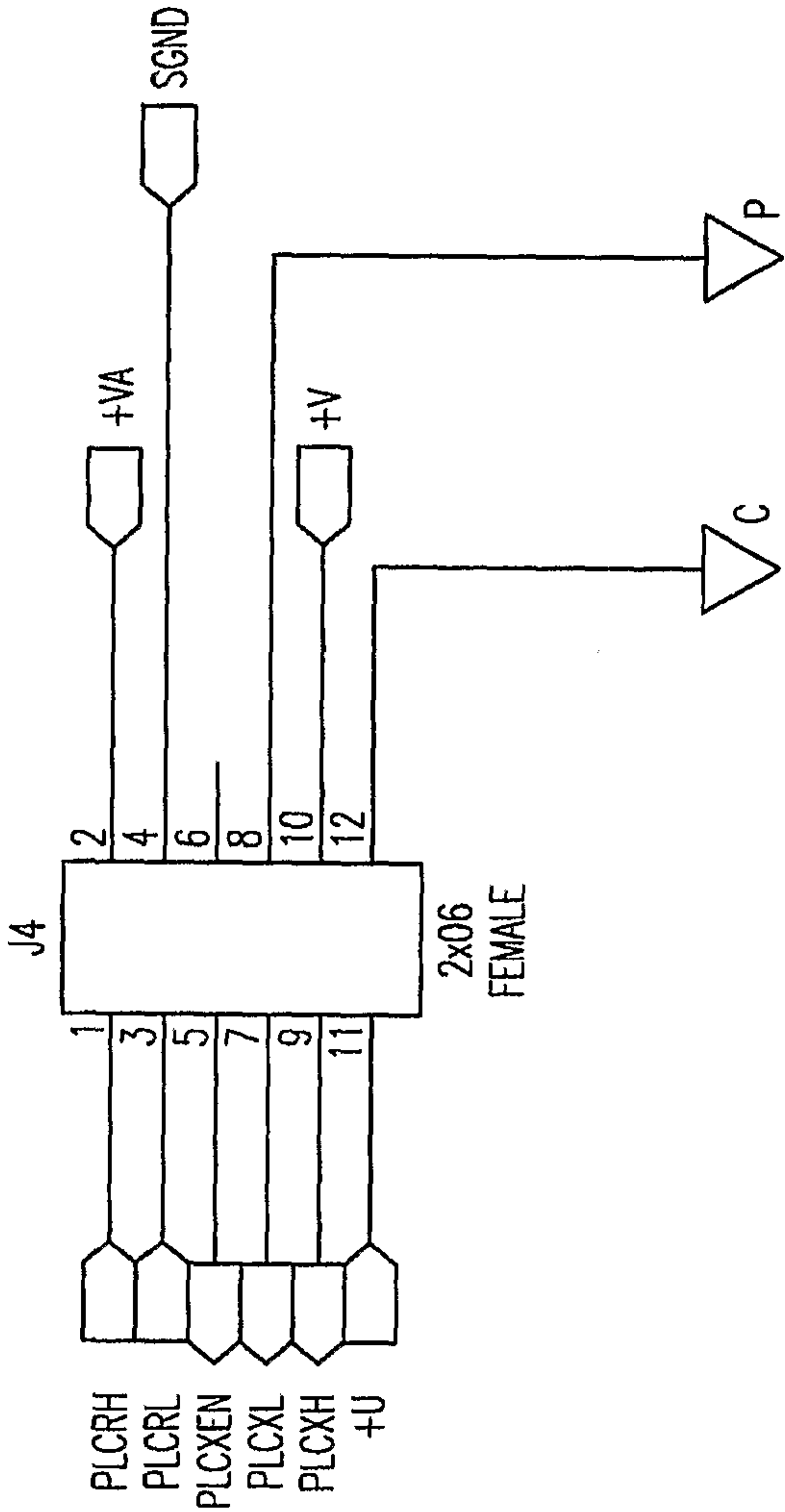
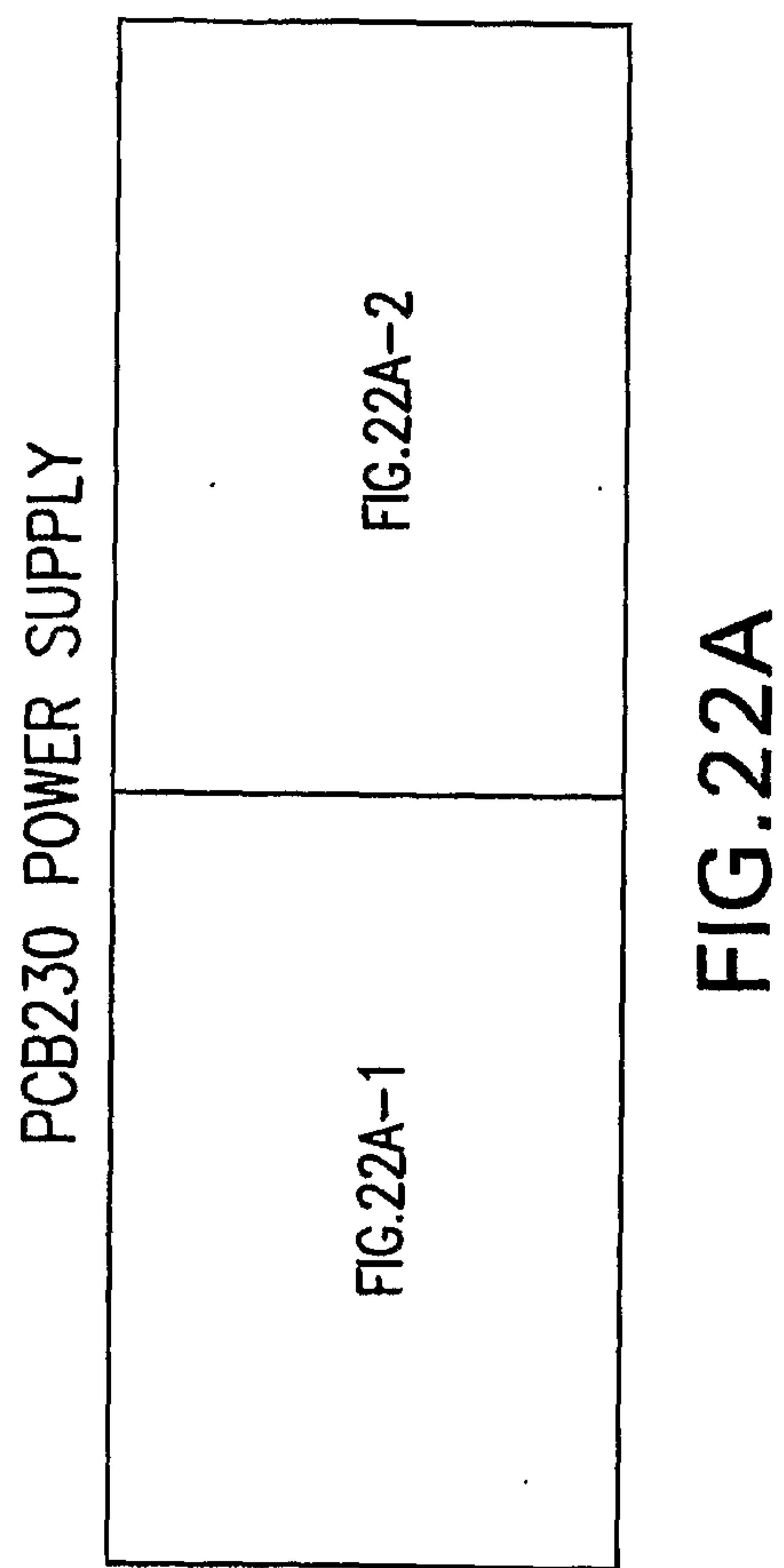
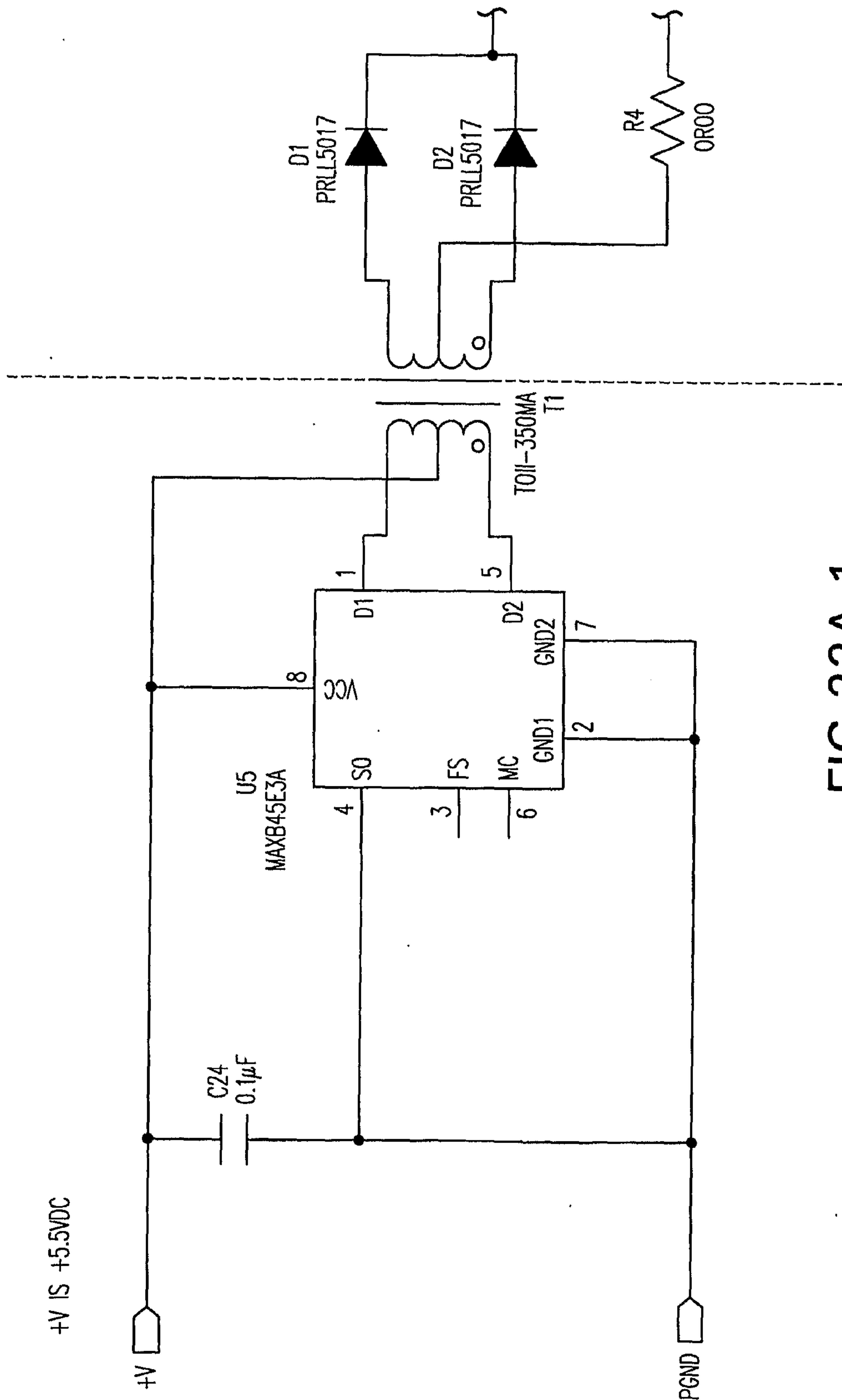


FIG.21B-3

98/227



99/227



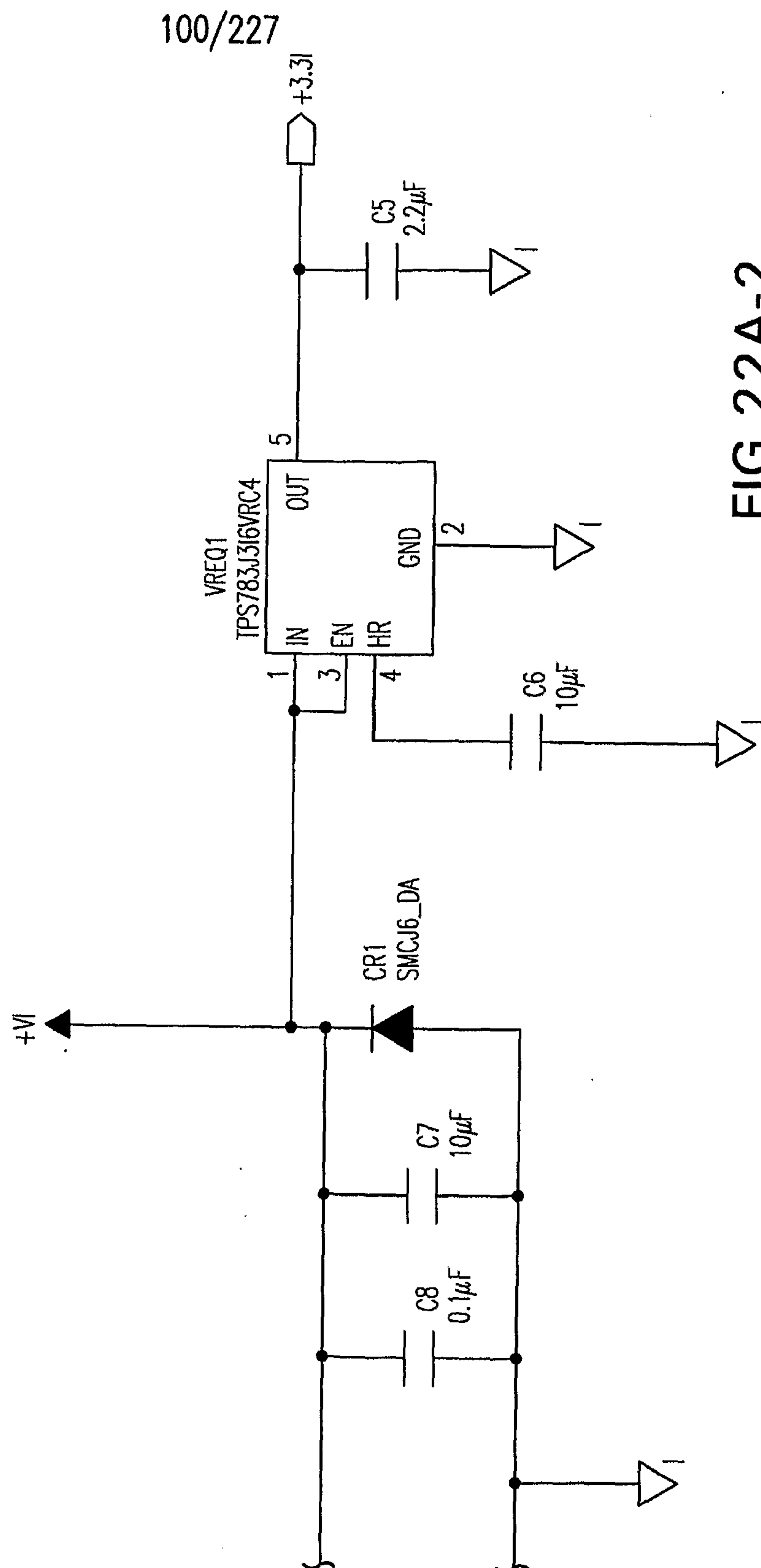
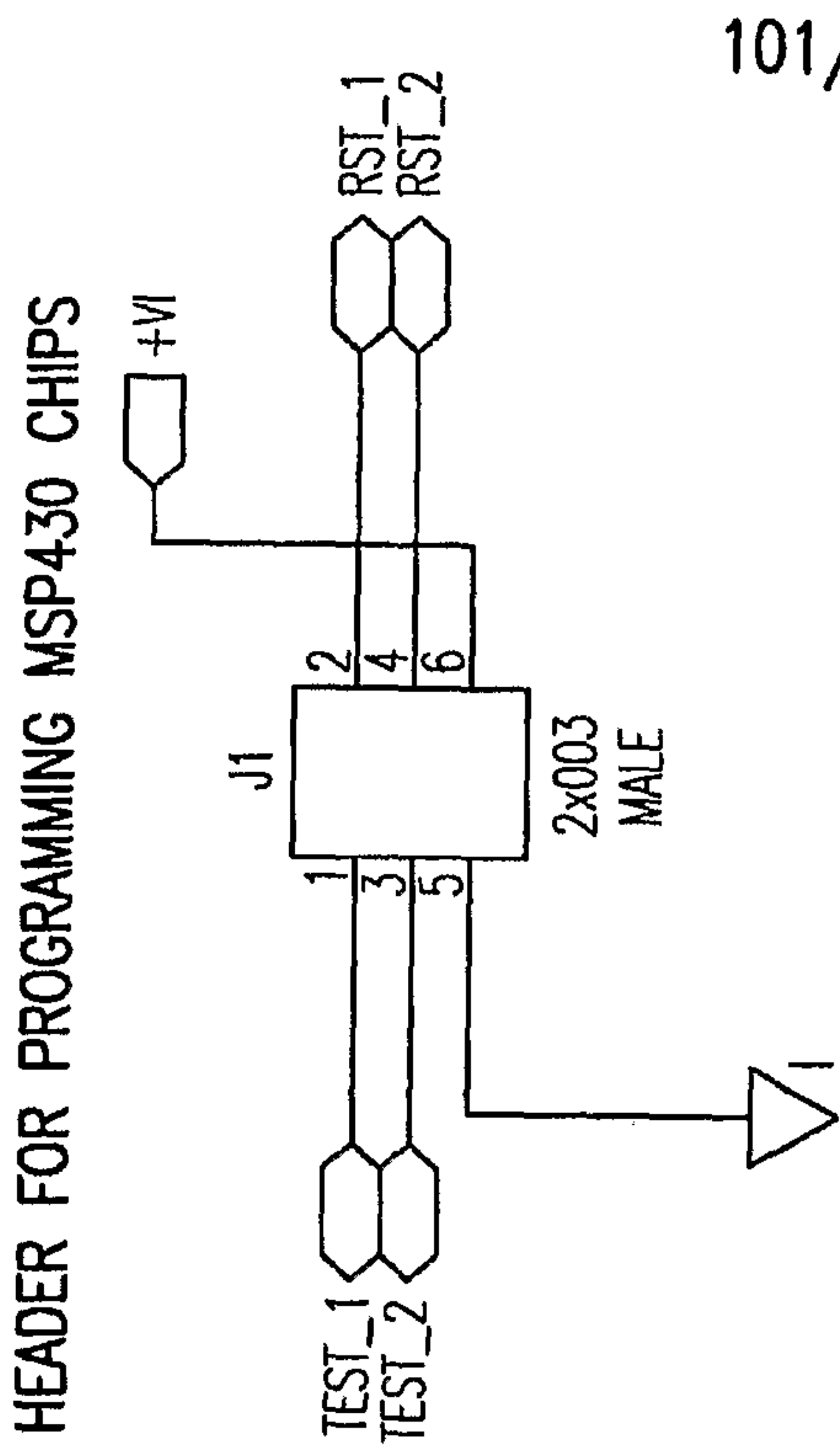
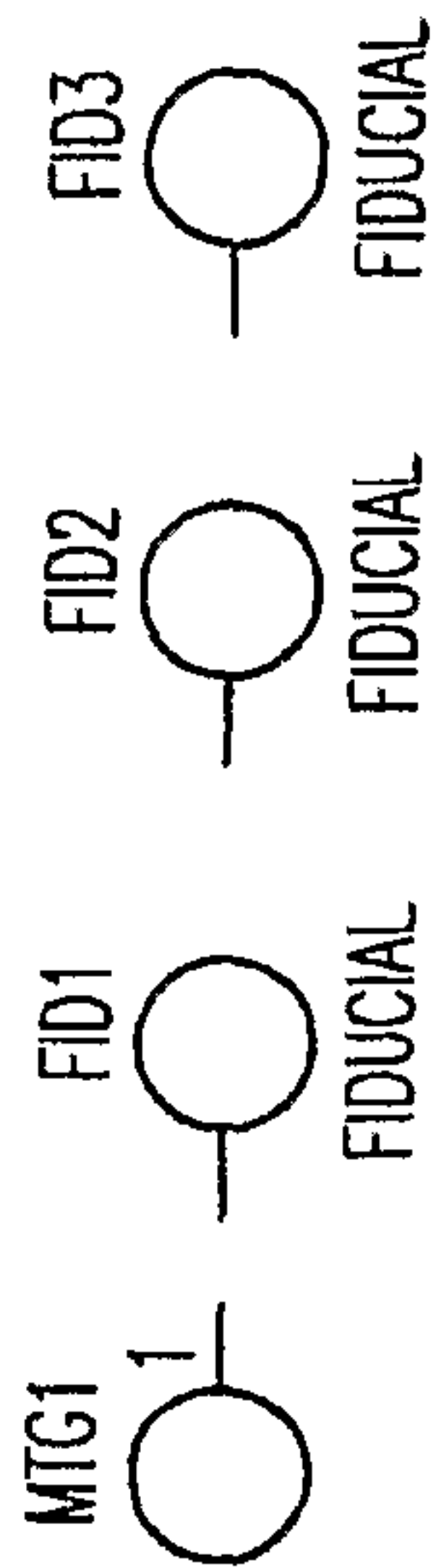
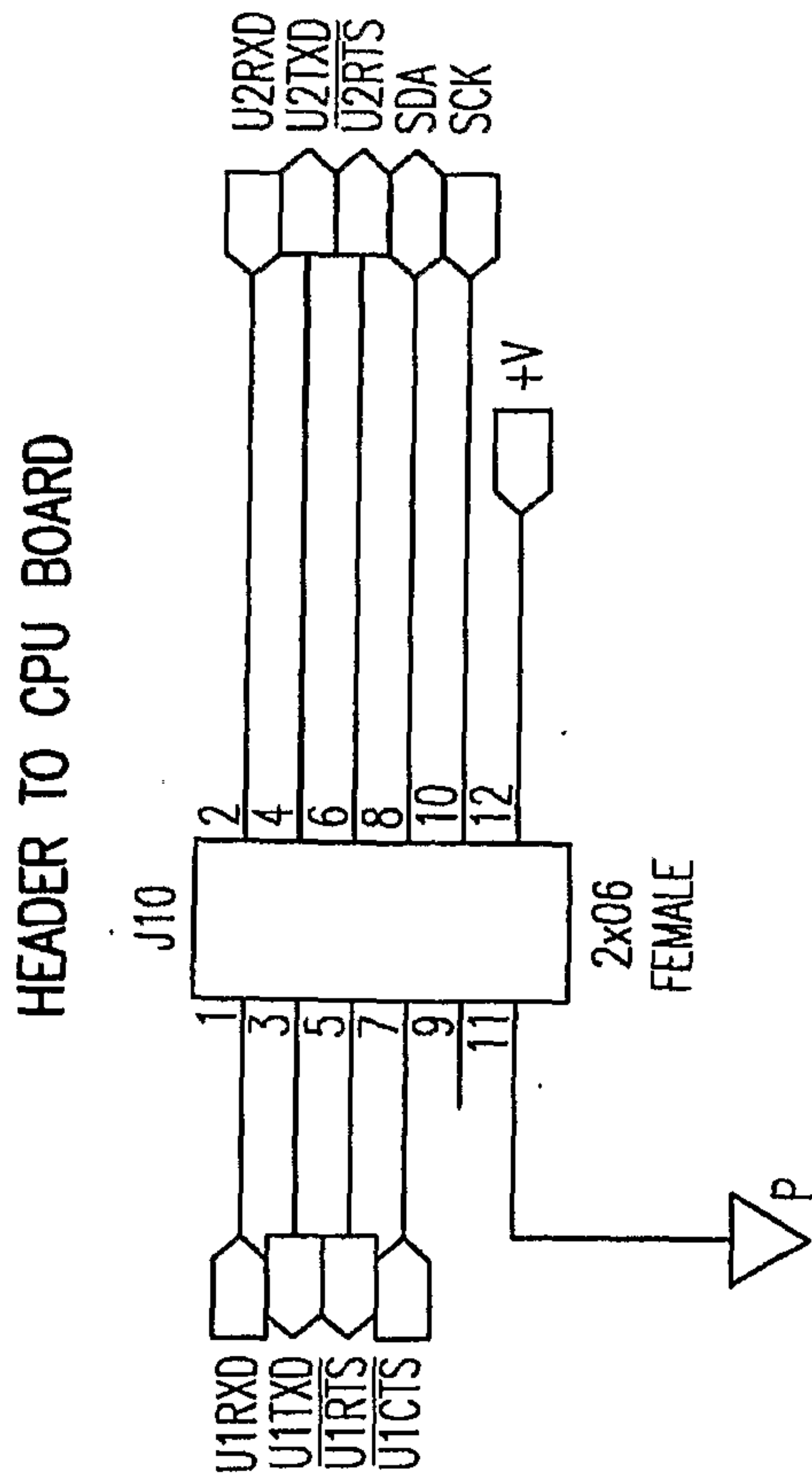


FIG. 22A-2



101/227



IO BOARD SCH230-1

FIG.22B

102/227

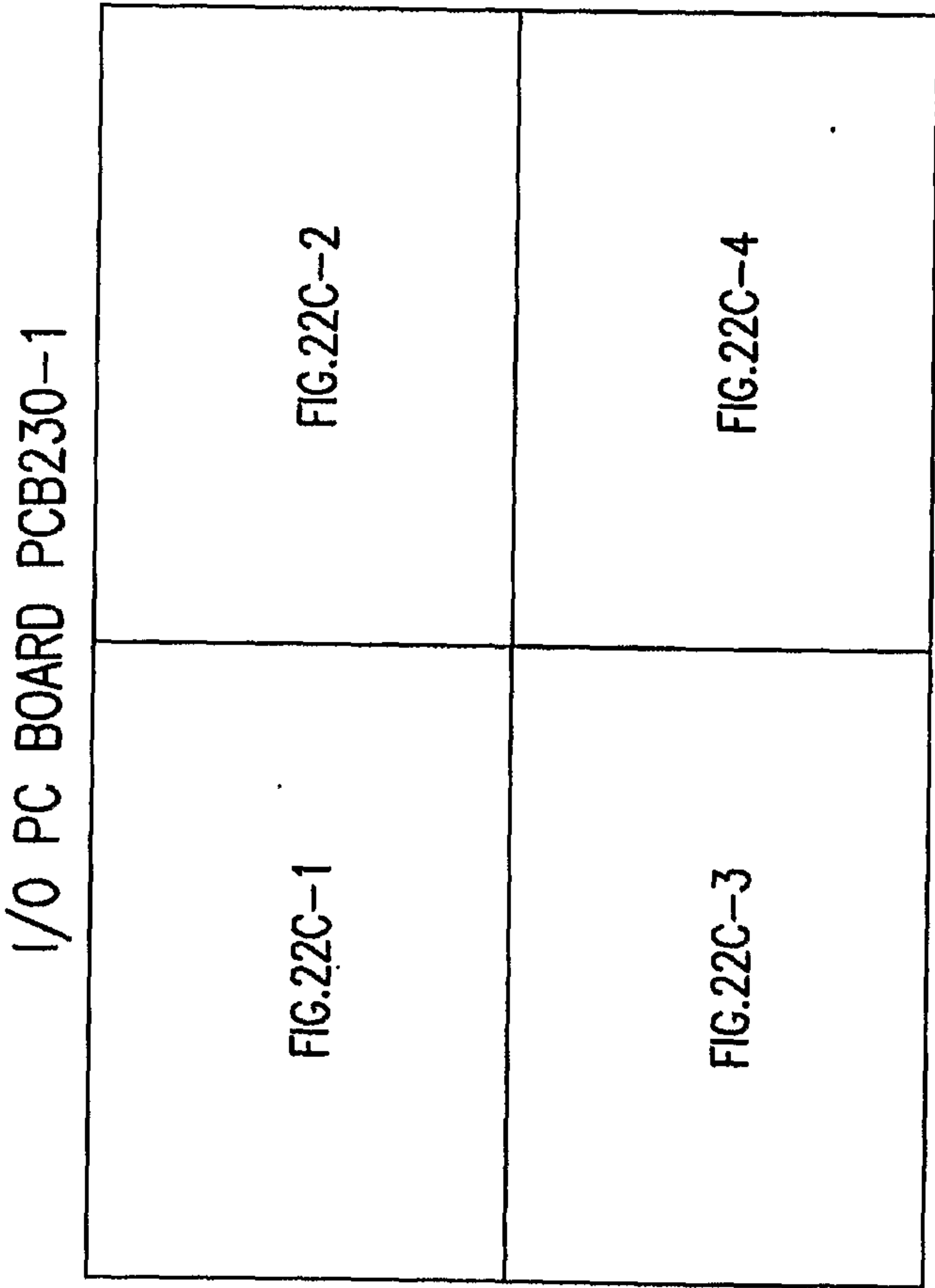
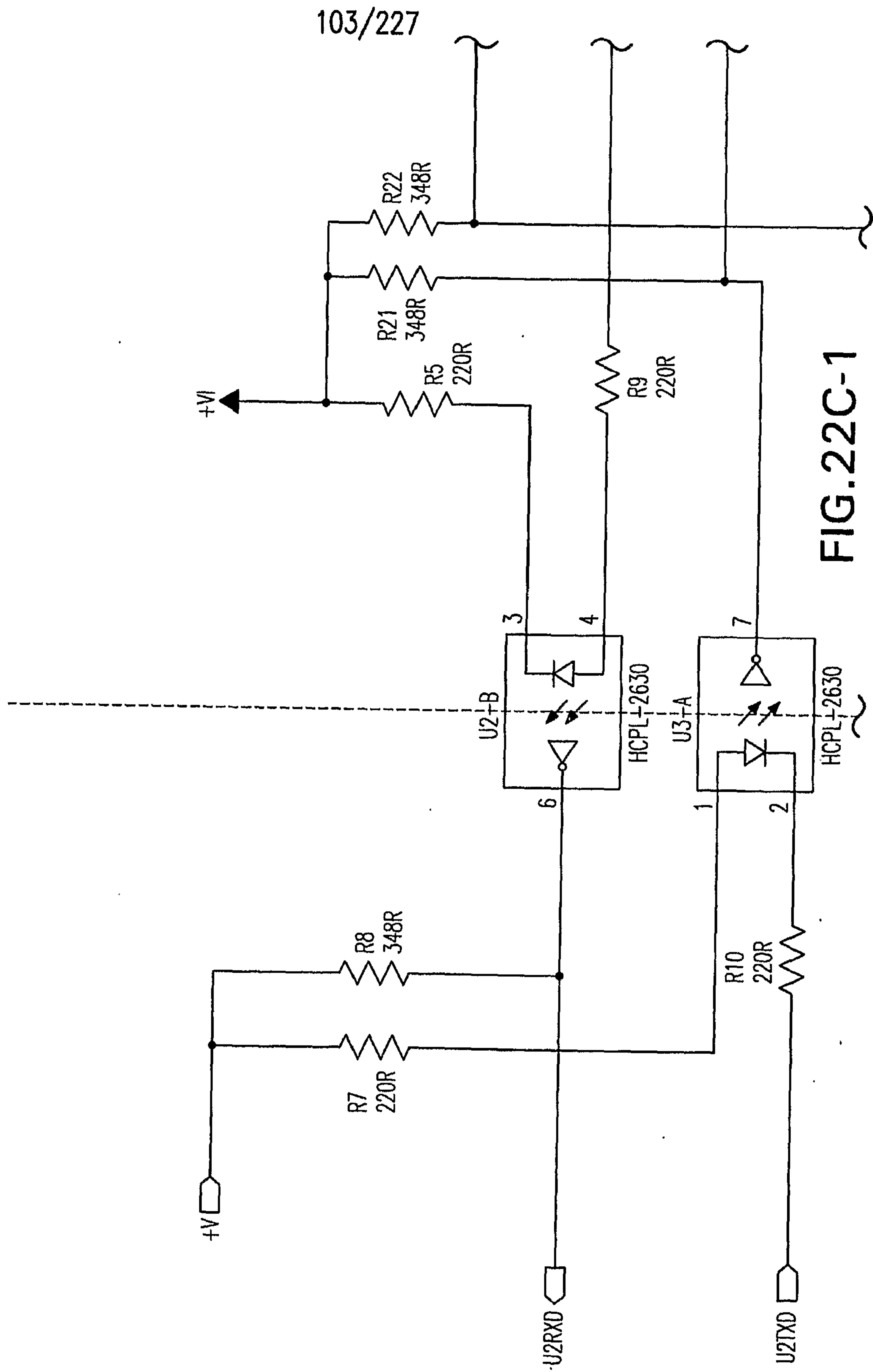
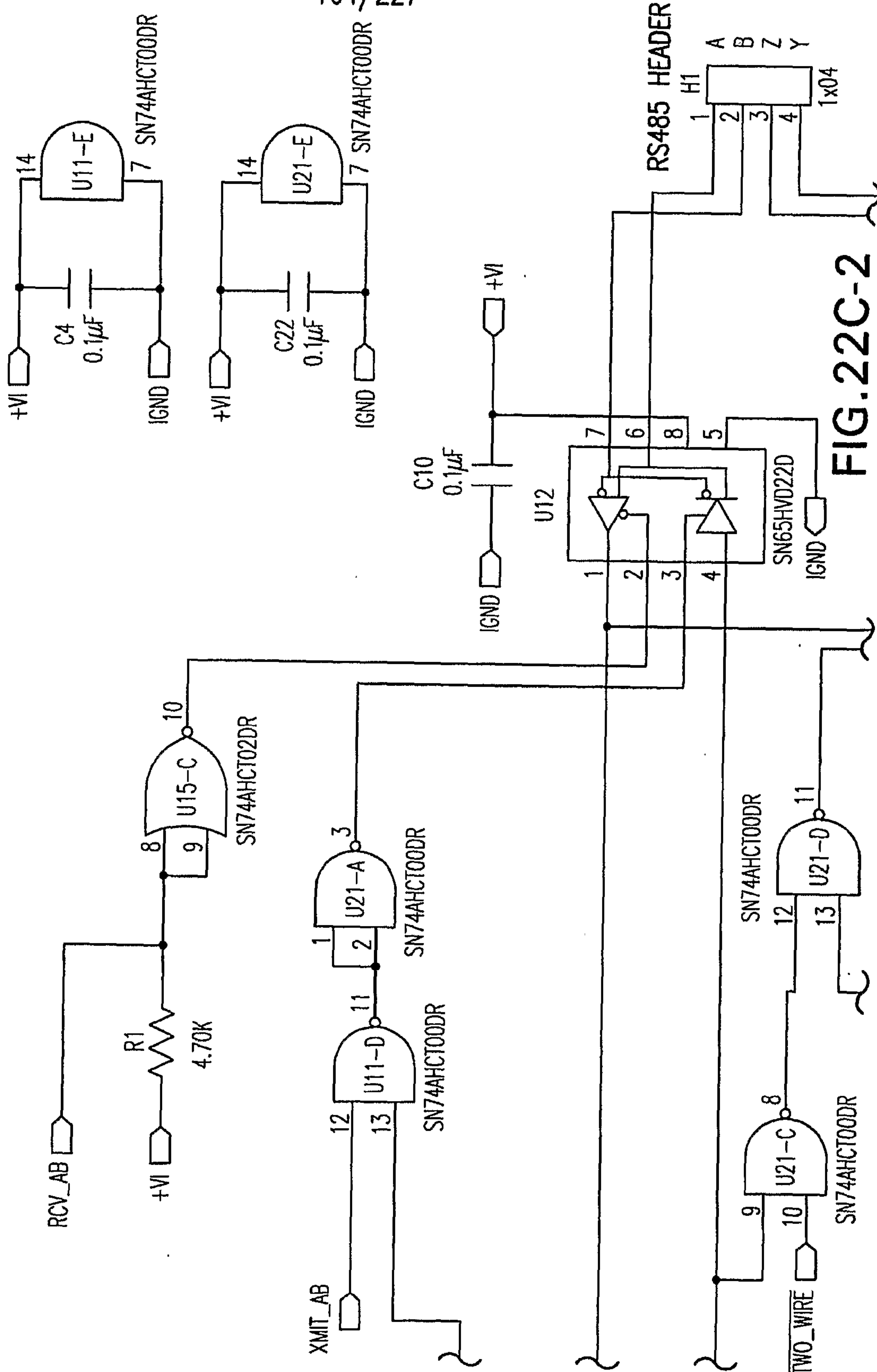


FIG.22C



104/227



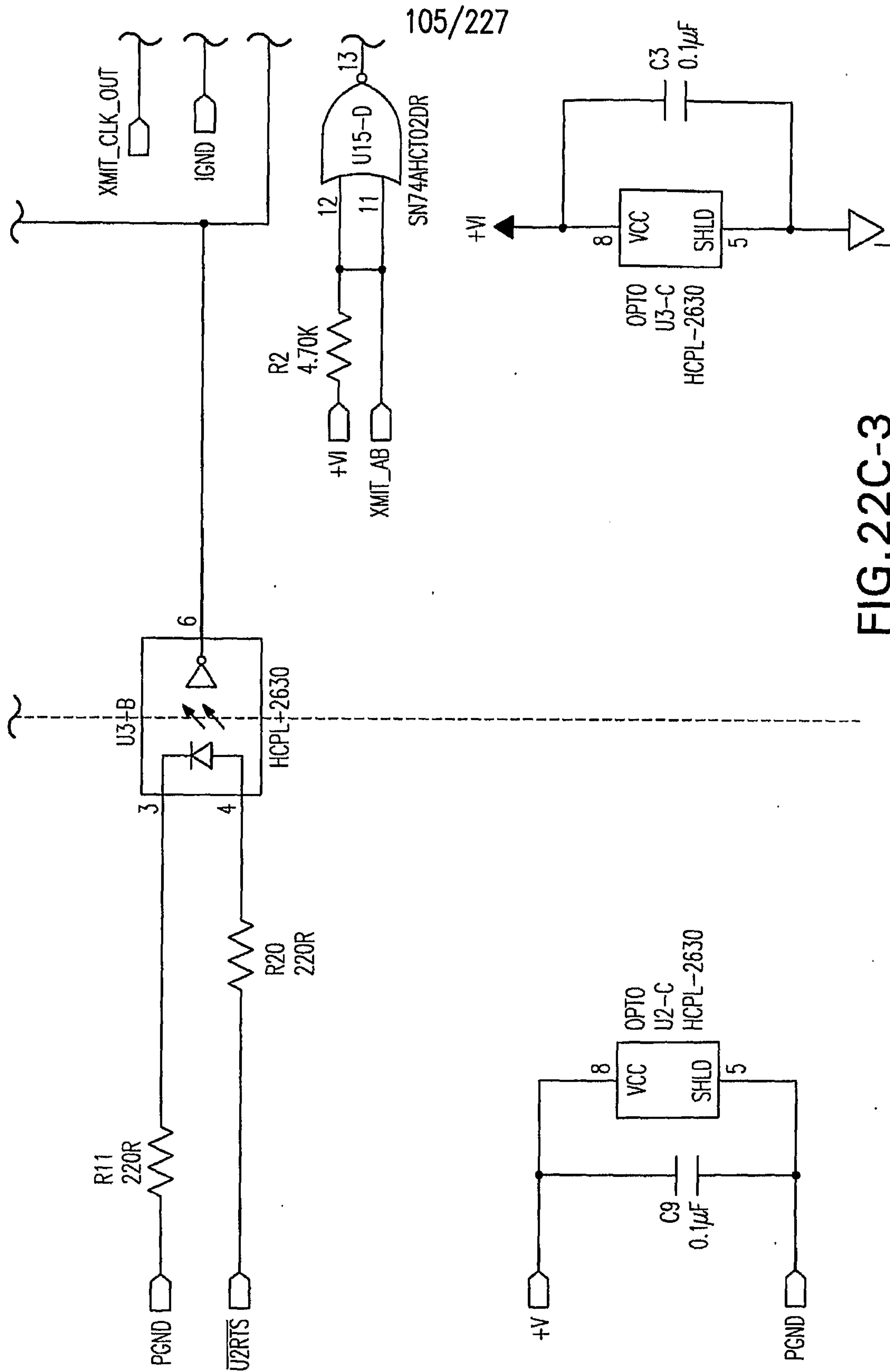


FIG. 22C-3

106/227

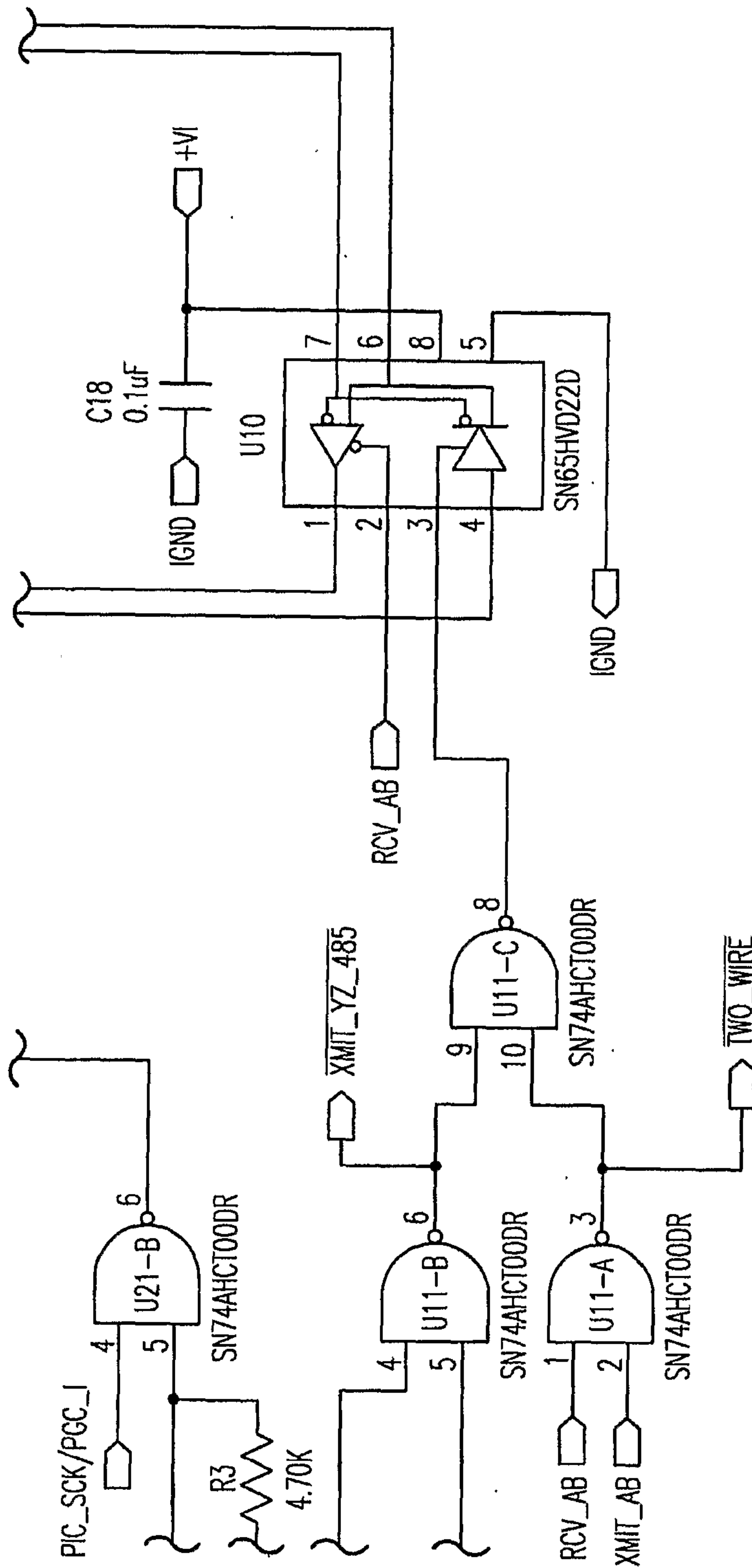
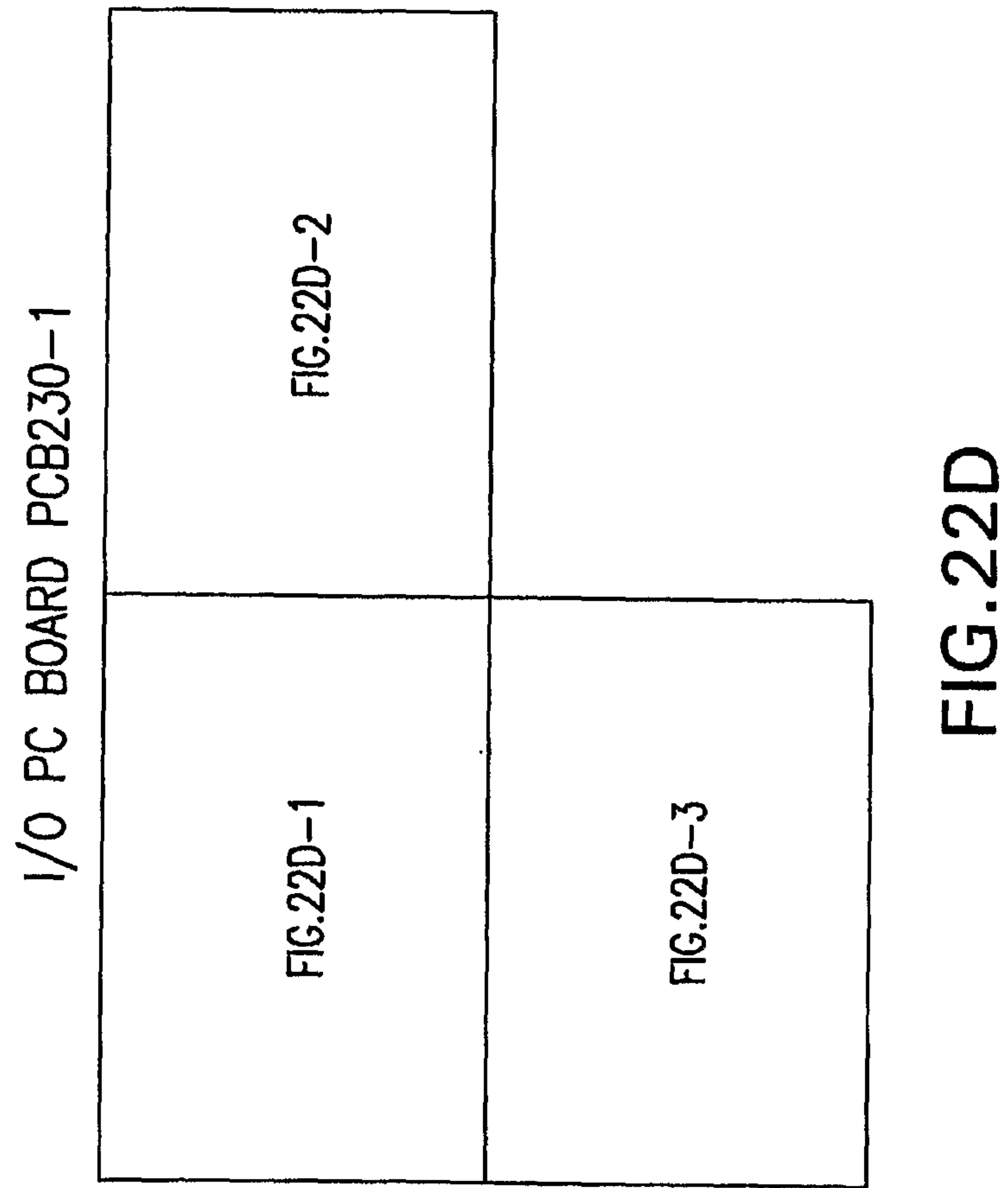


FIG. 22C-4

107/227



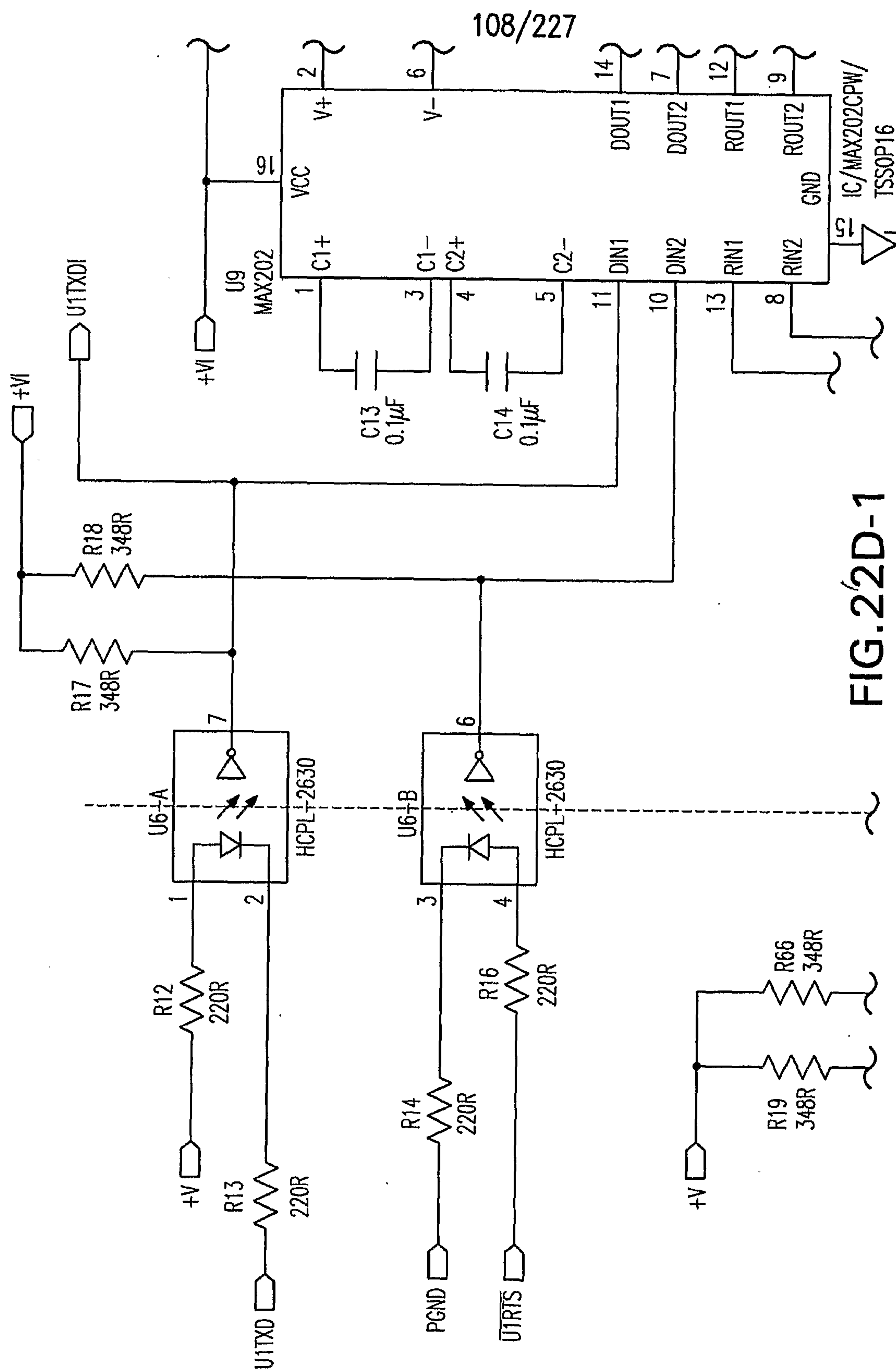


FIG. 22D-1

109/227

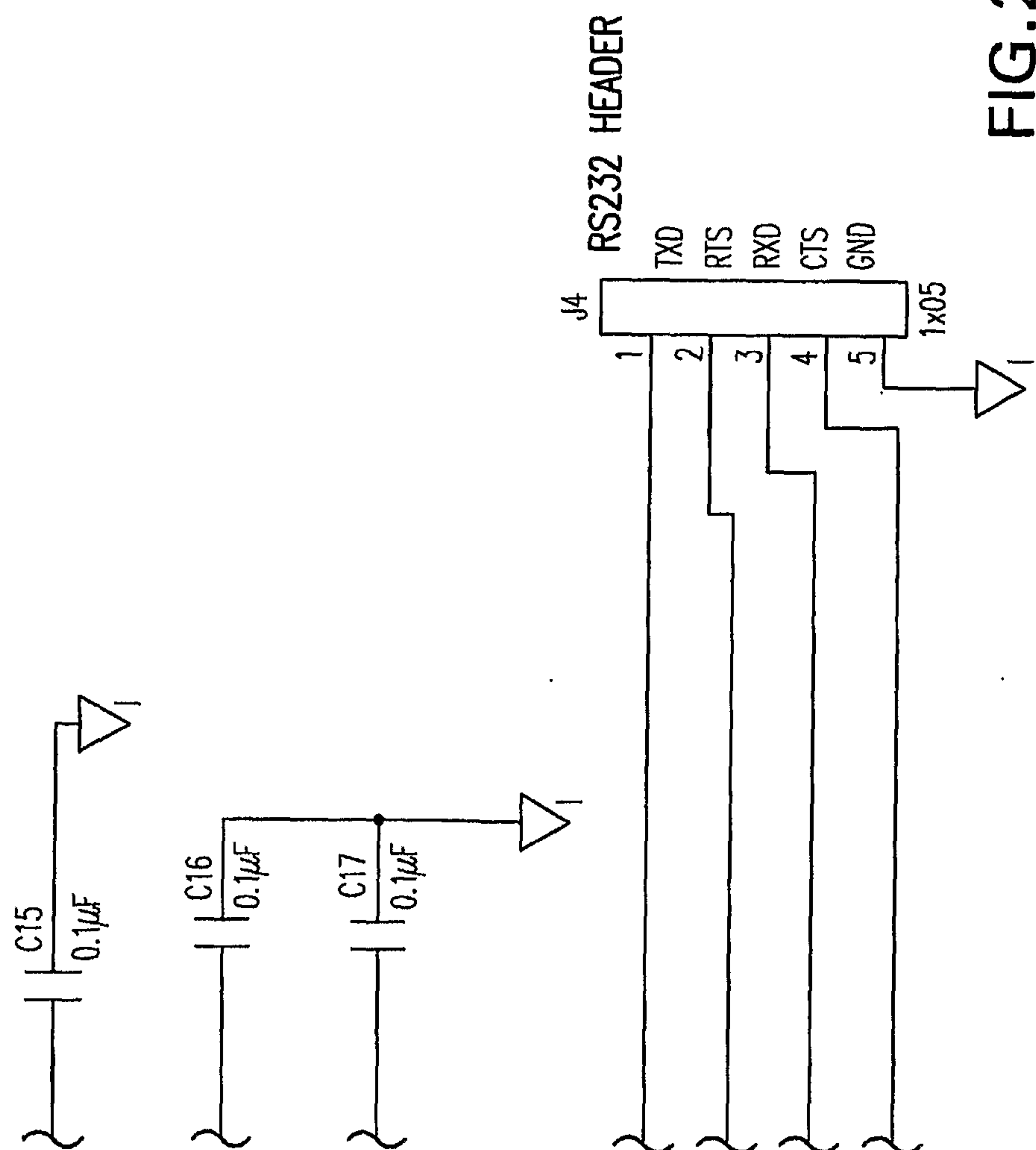
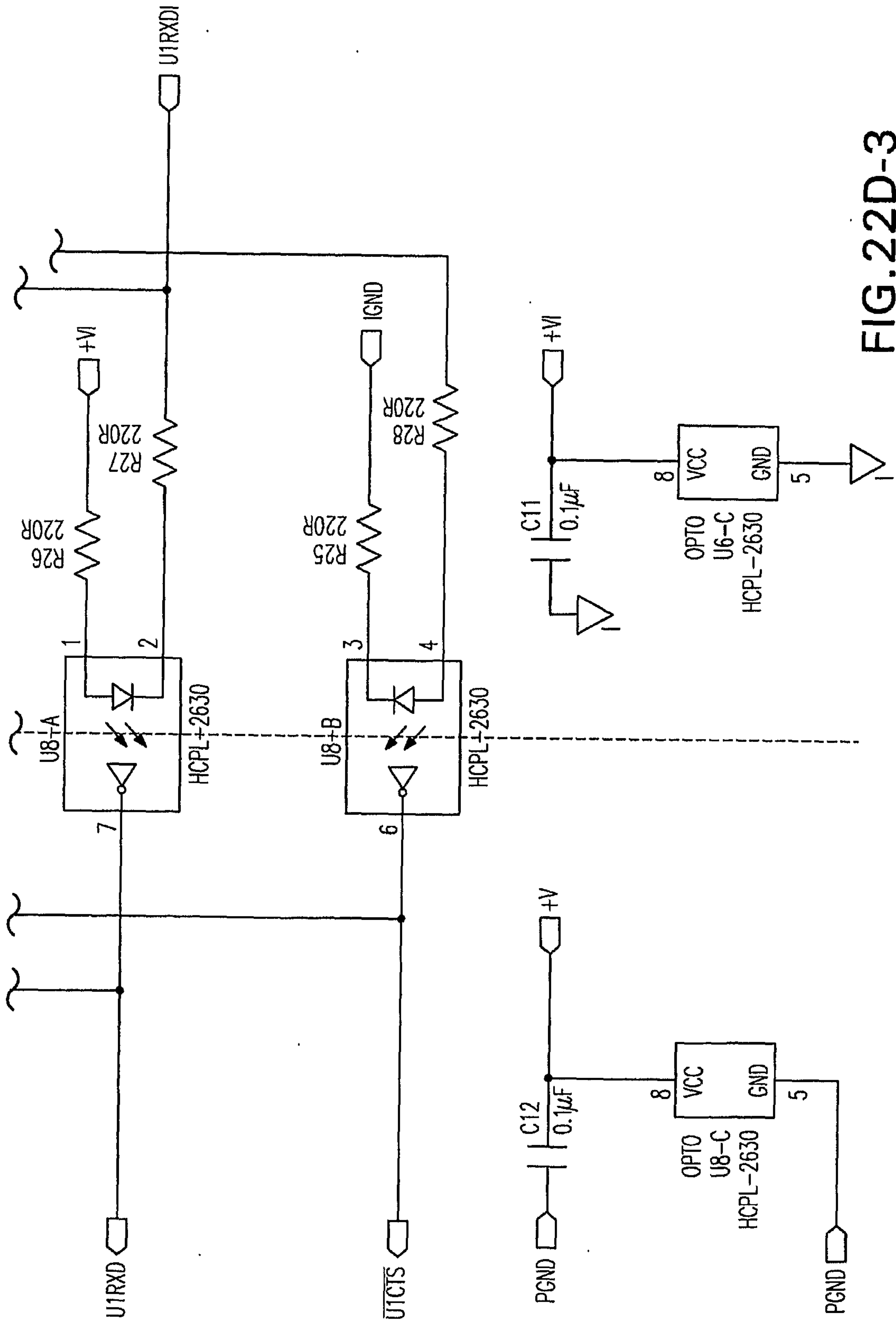
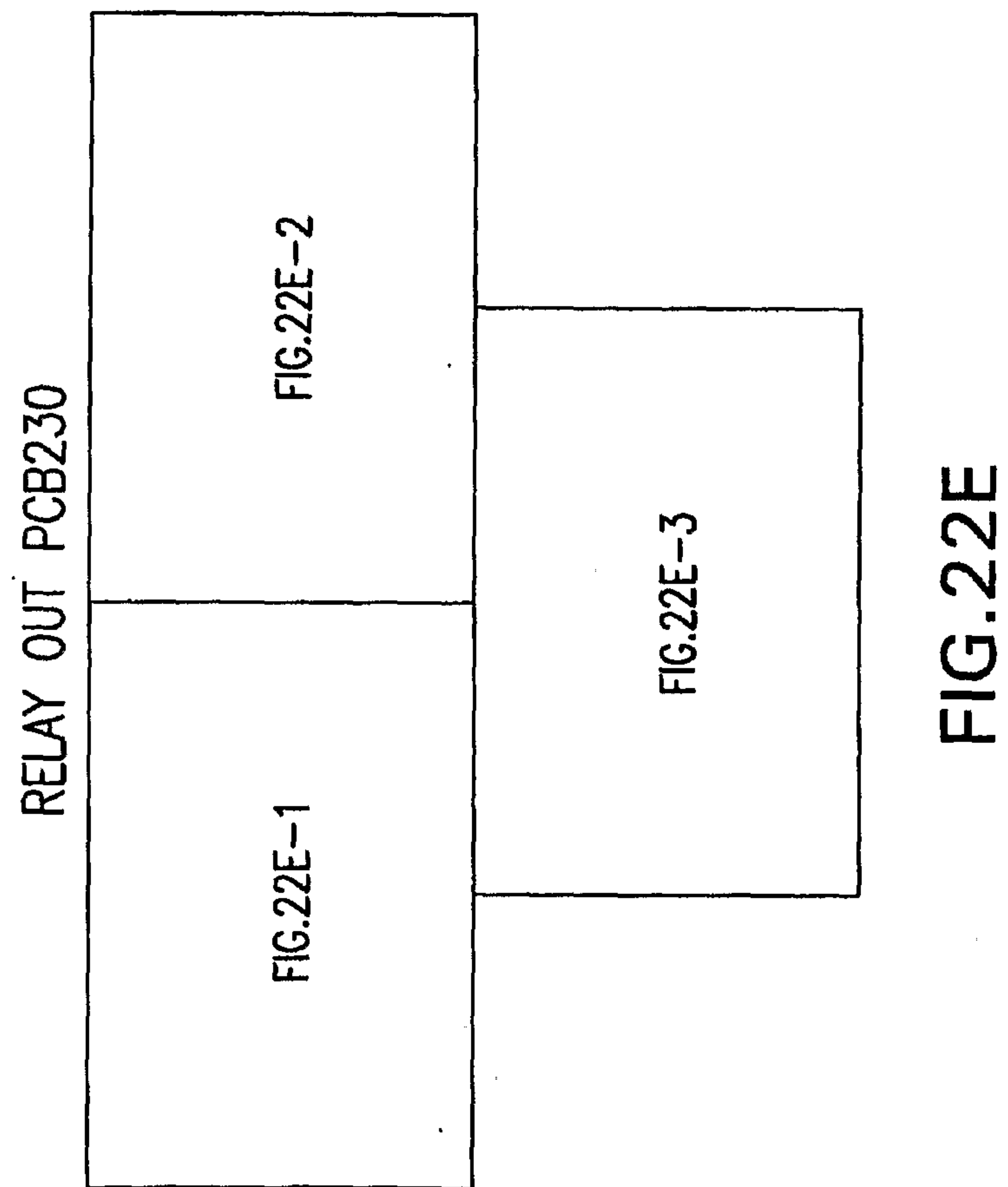


FIG. 22D-2

110/227



111/227



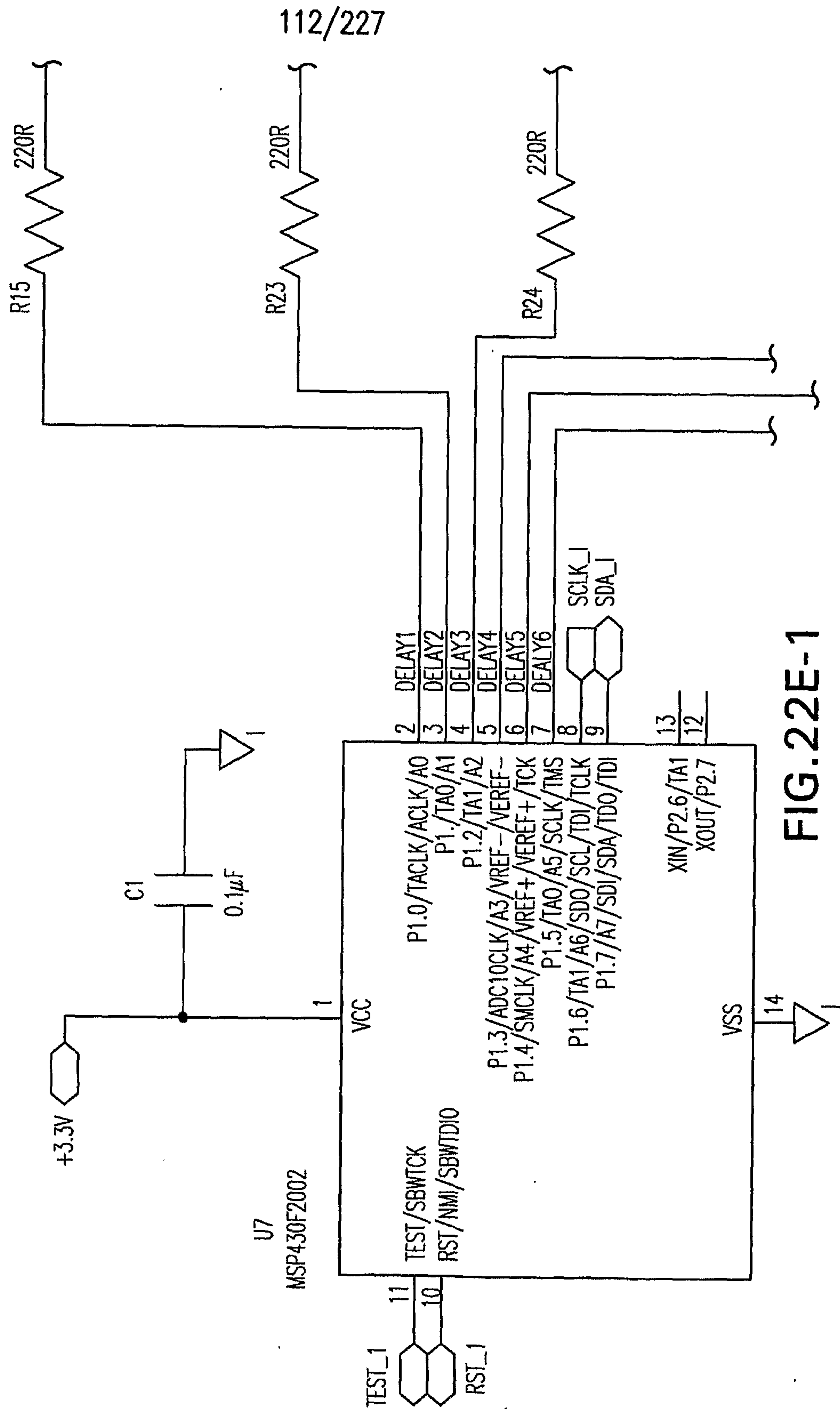


FIG. 22E-1

113/227

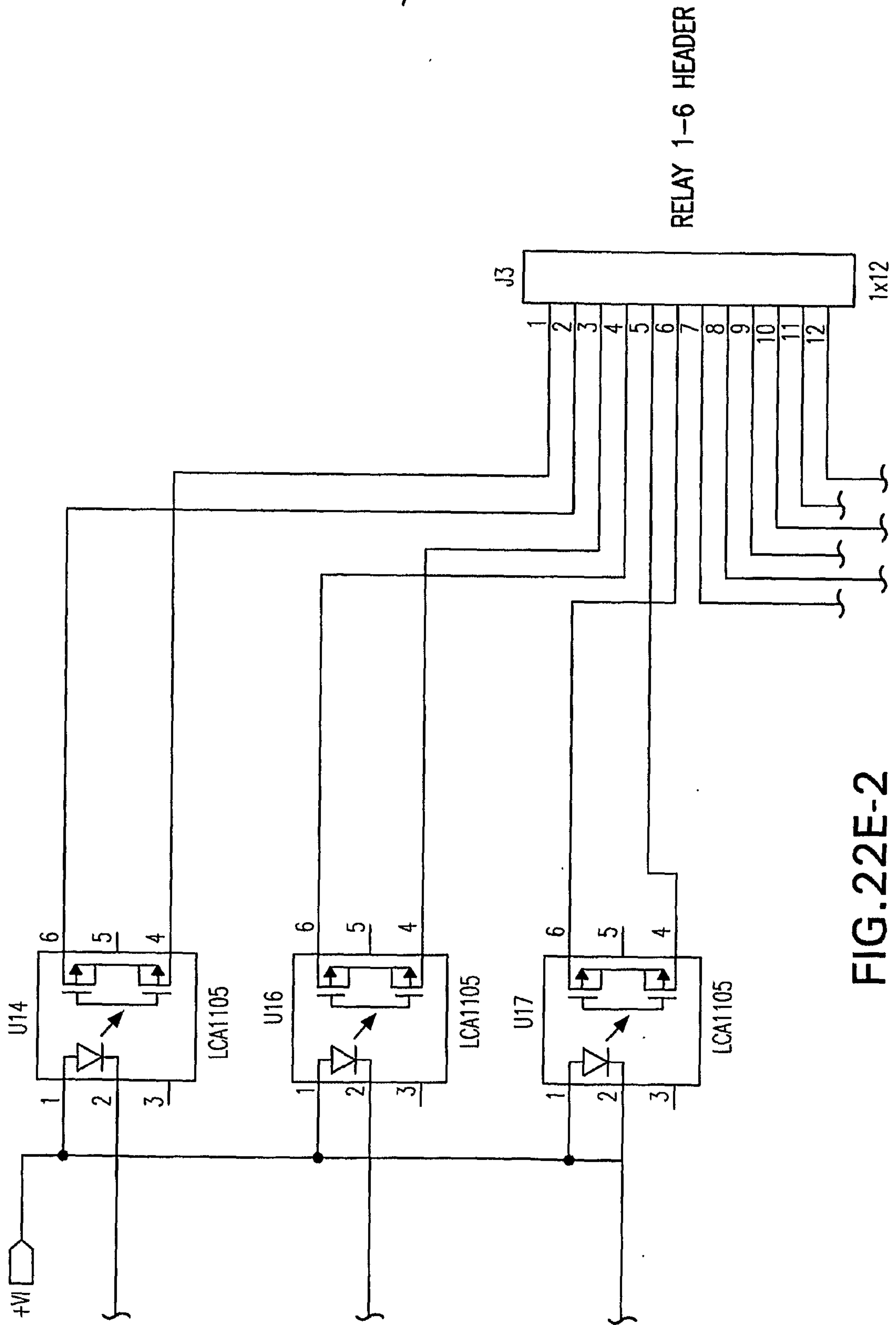


FIG. 22E-2

114/227

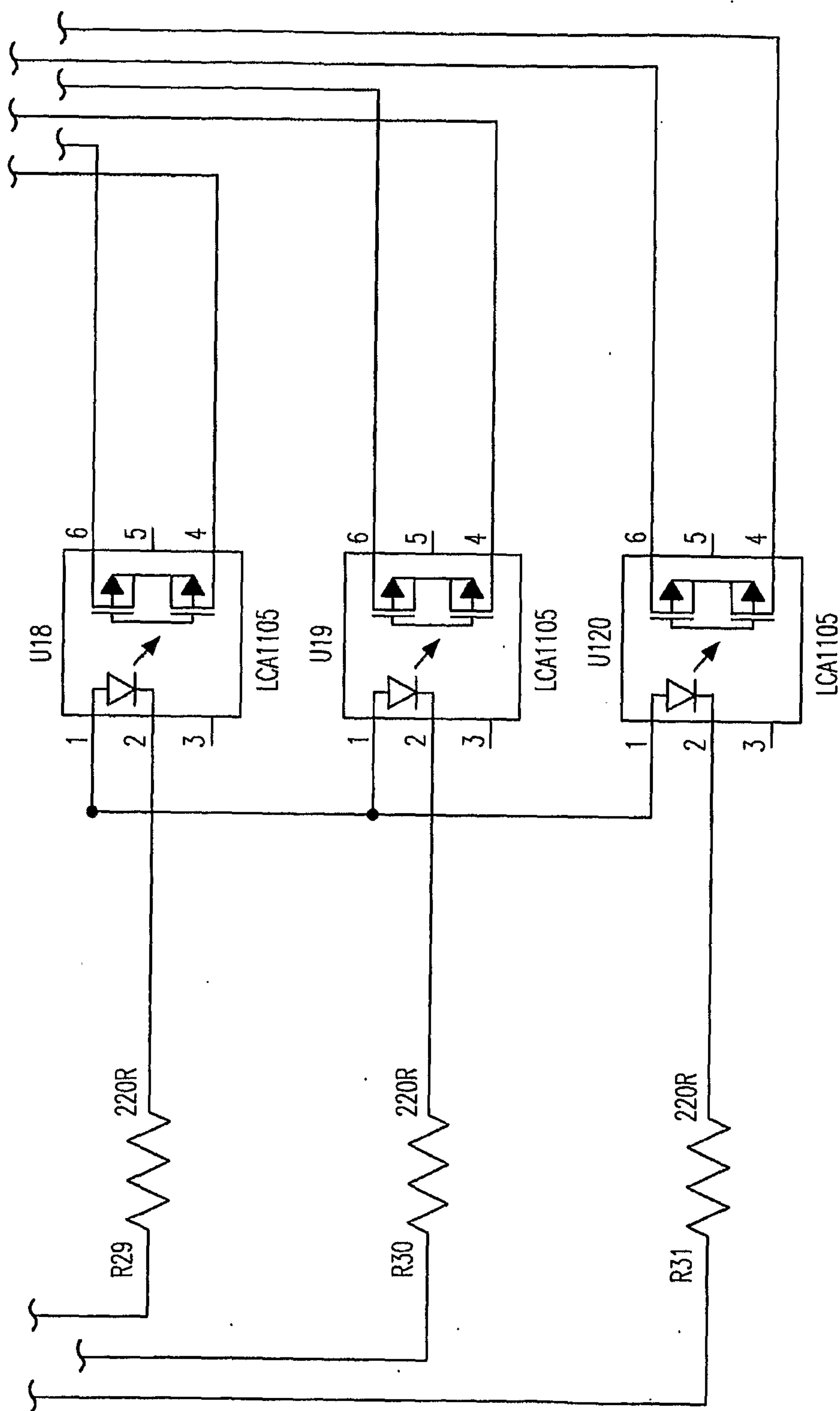
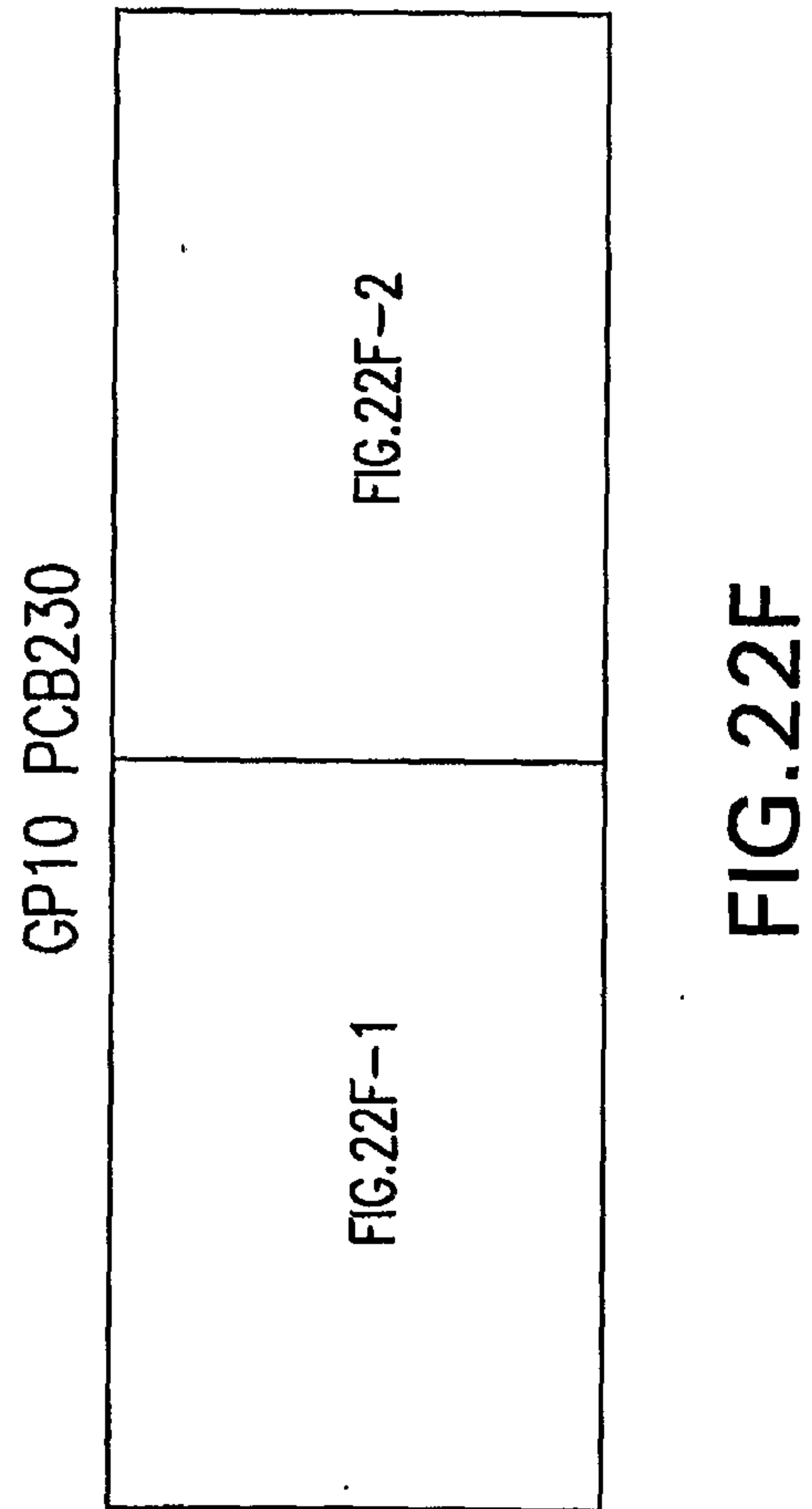


FIG. 22E-3

115/227



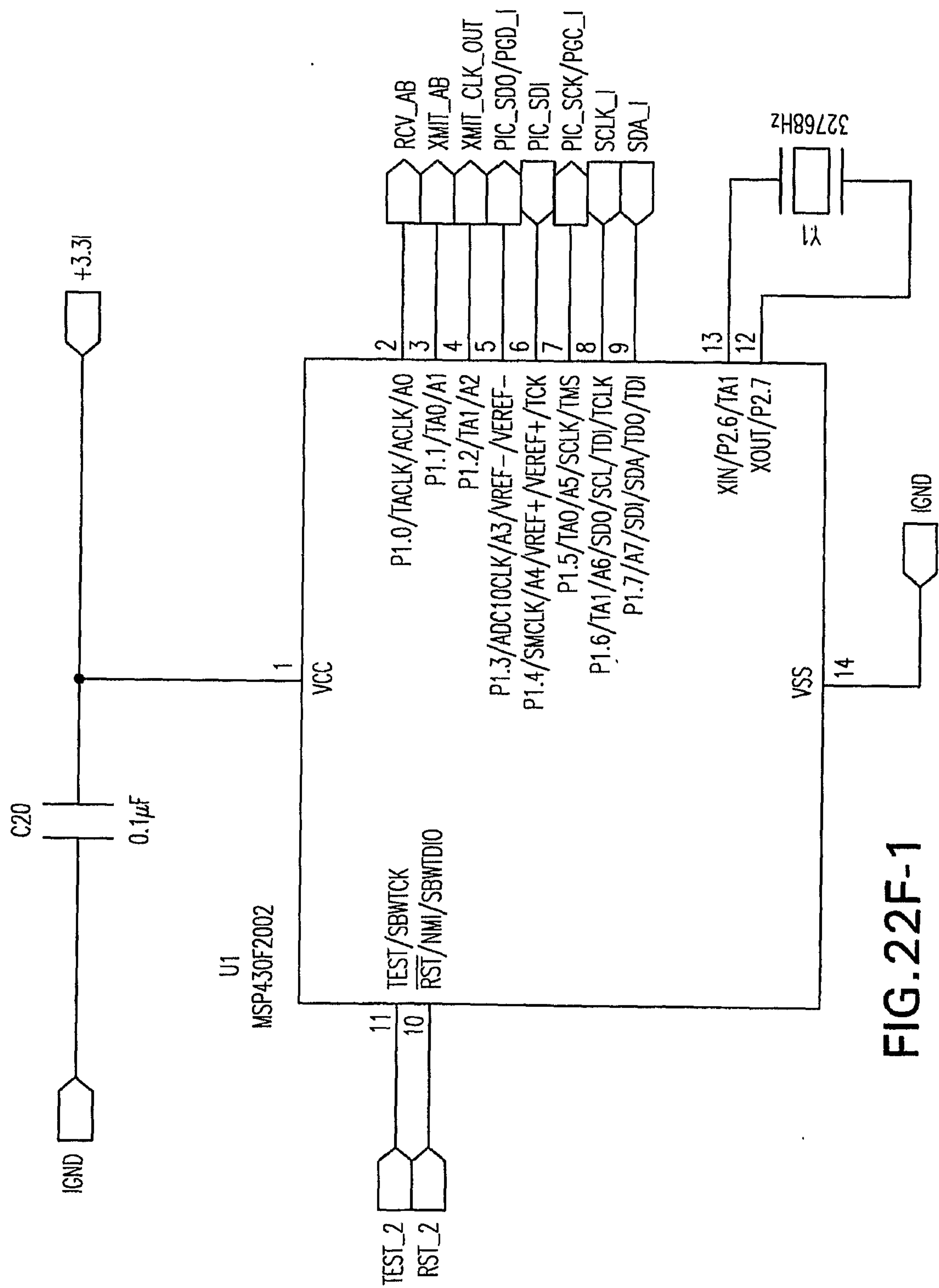


FIG. 22F-1

117/227

QLC PIC BUS HEADER

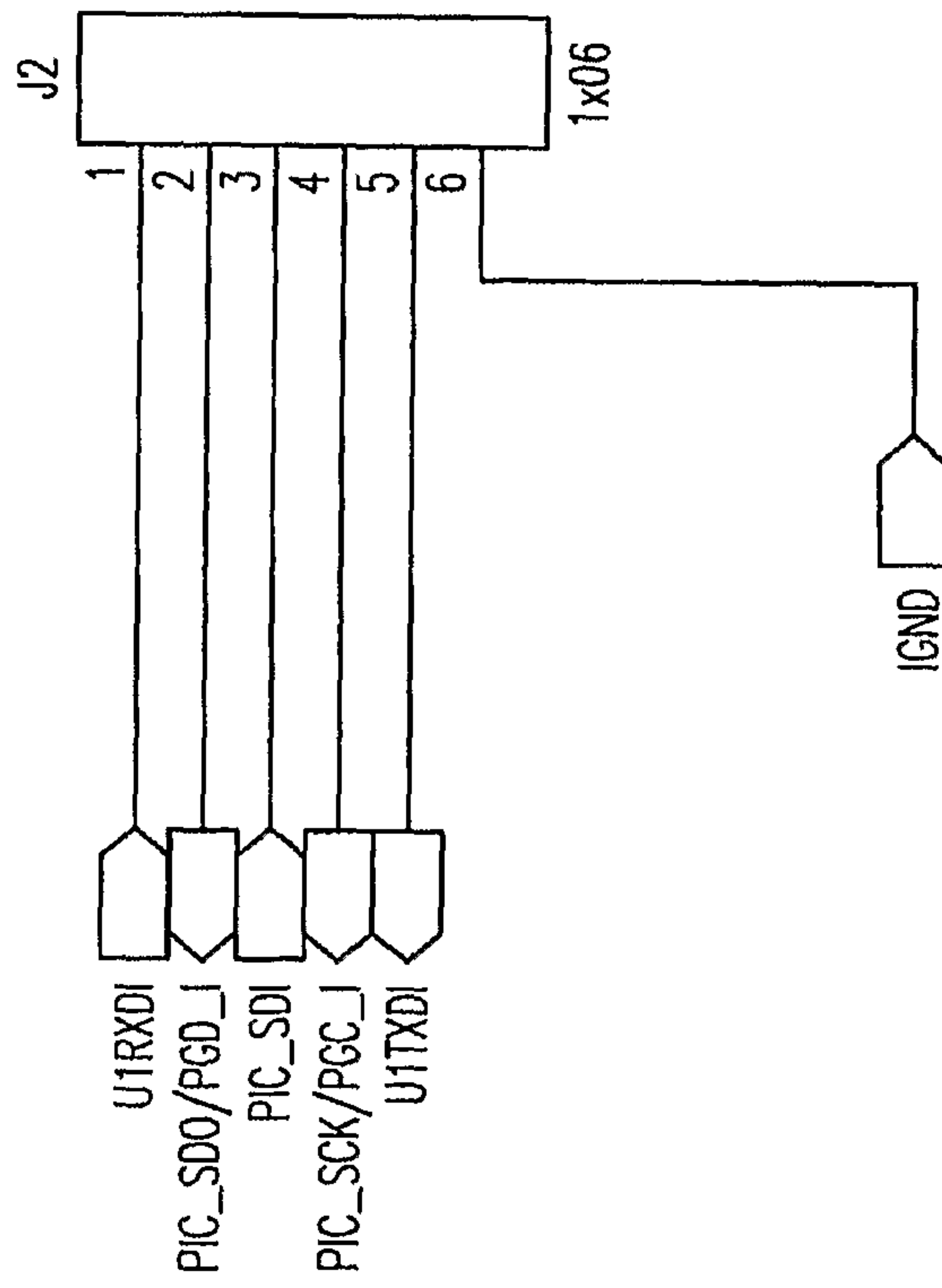
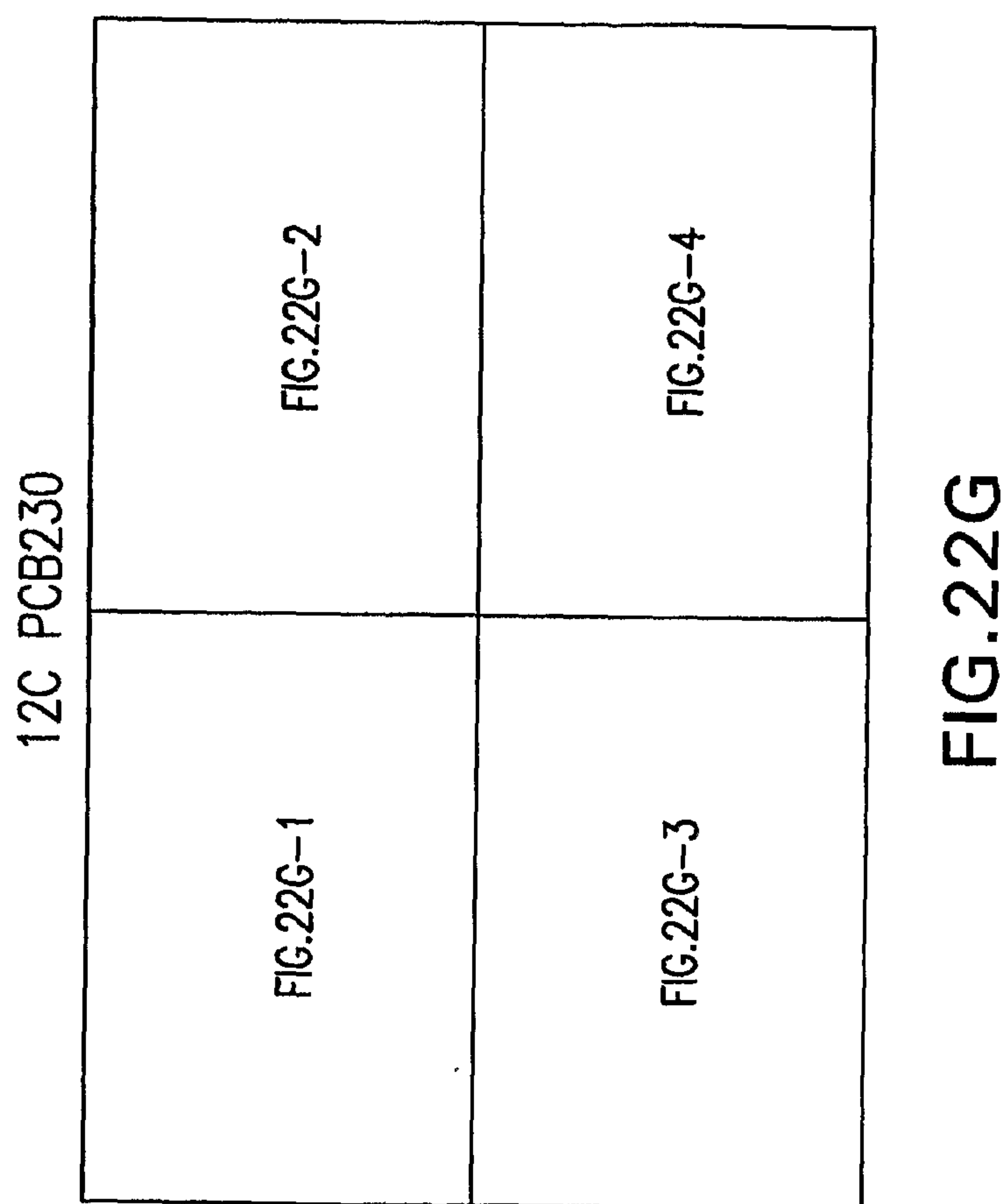


FIG.22F-2

118/227



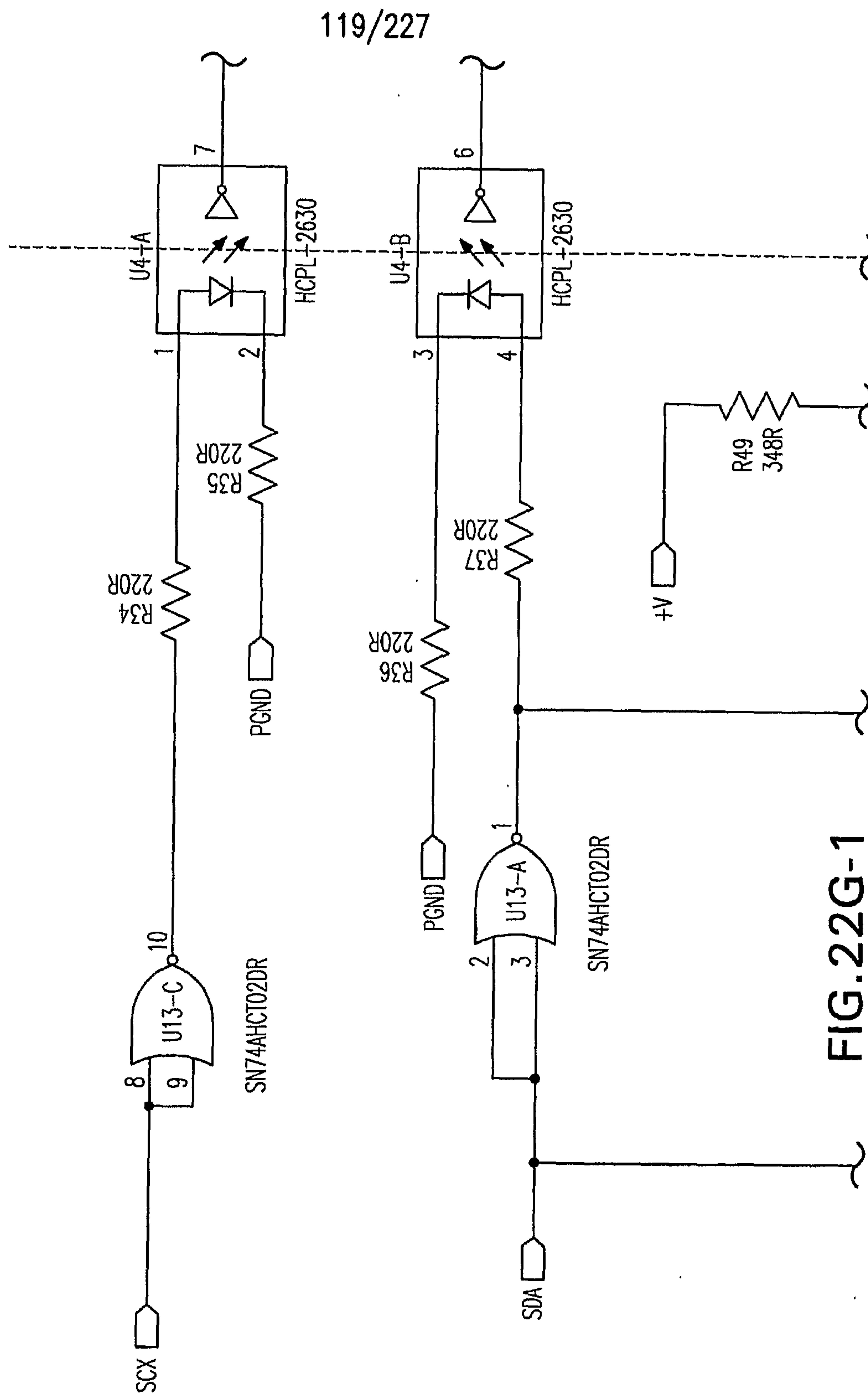


FIG. 22G-1

120/227

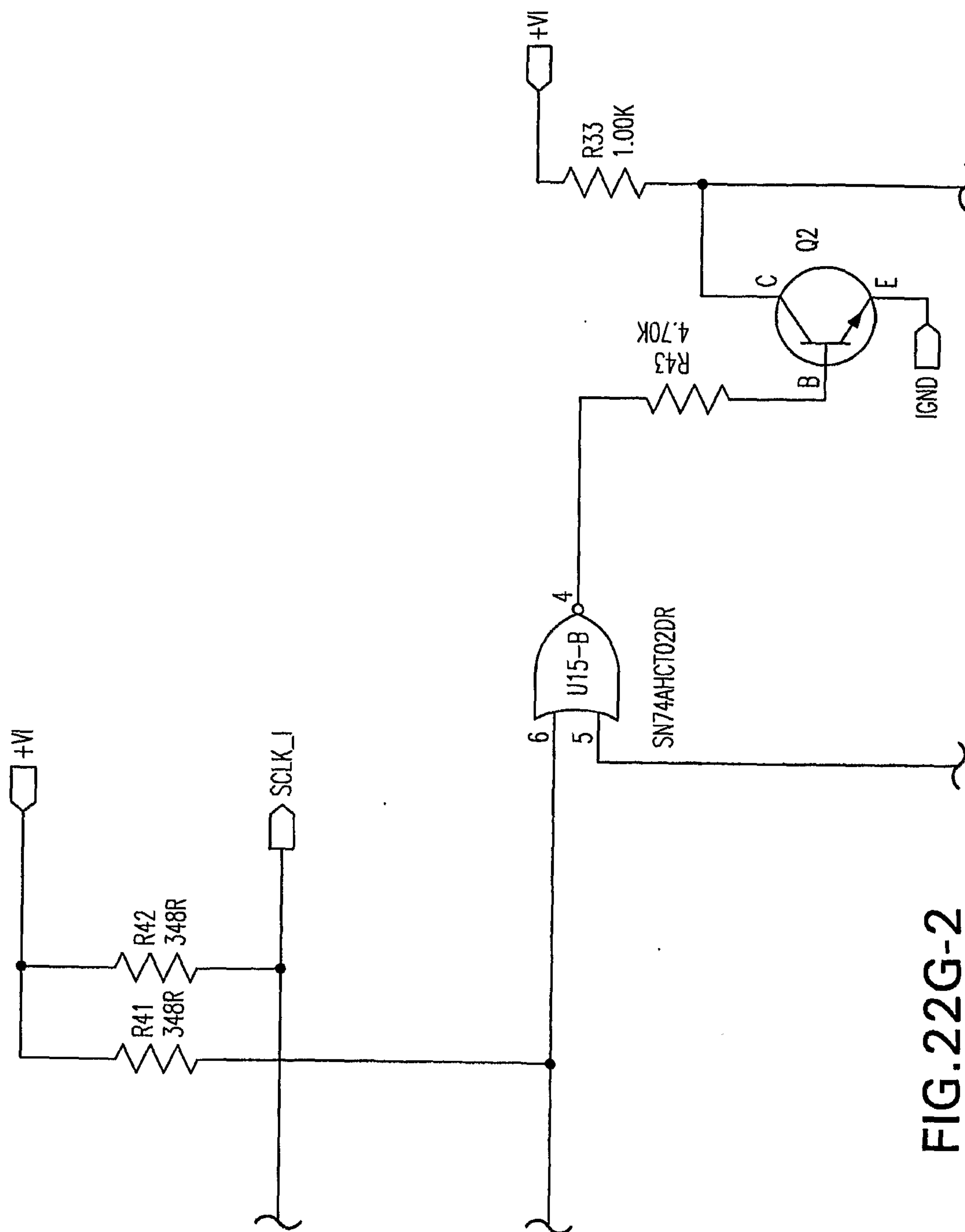


FIG. 22G-2

121/227

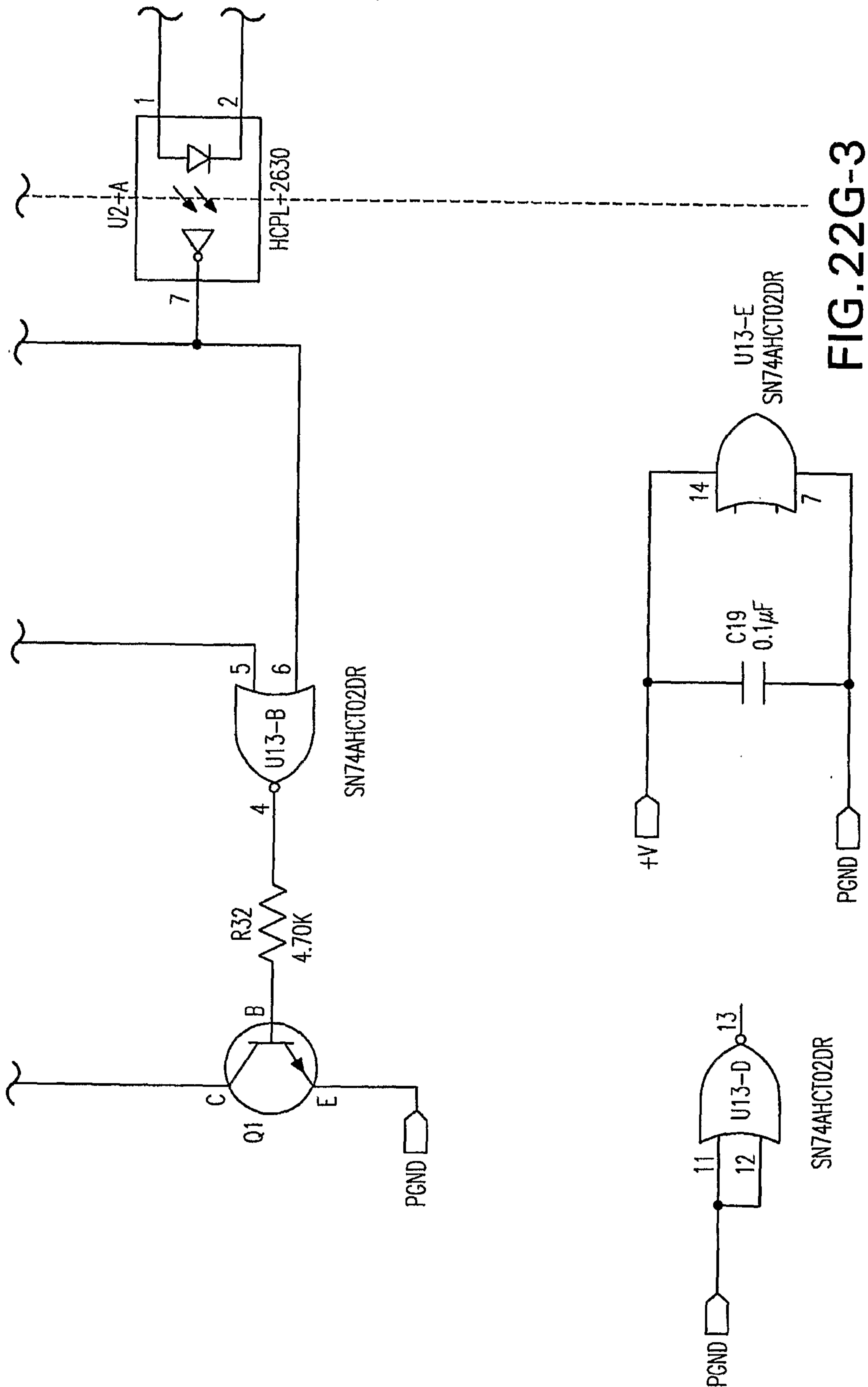


FIG. 22G-3

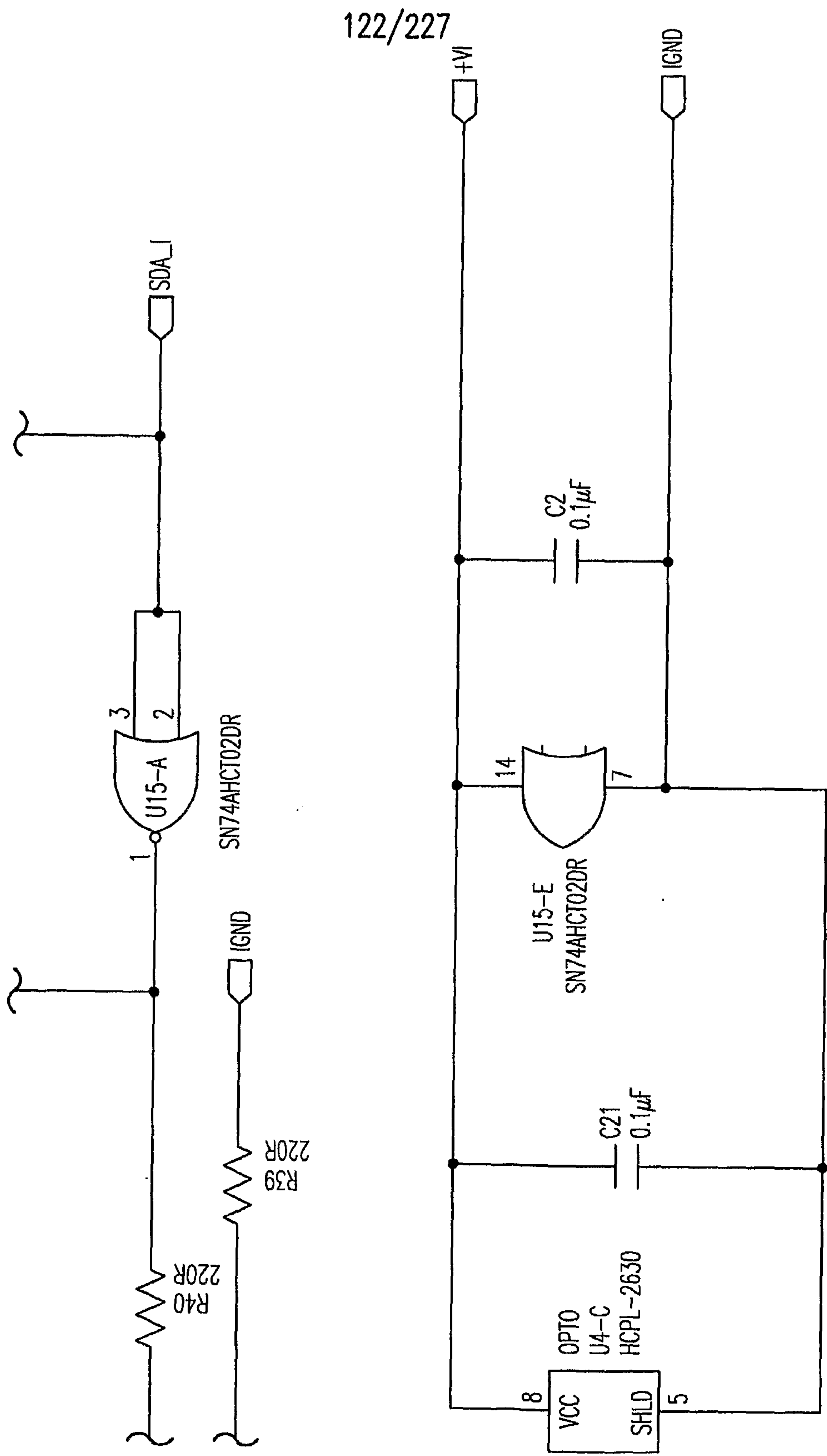
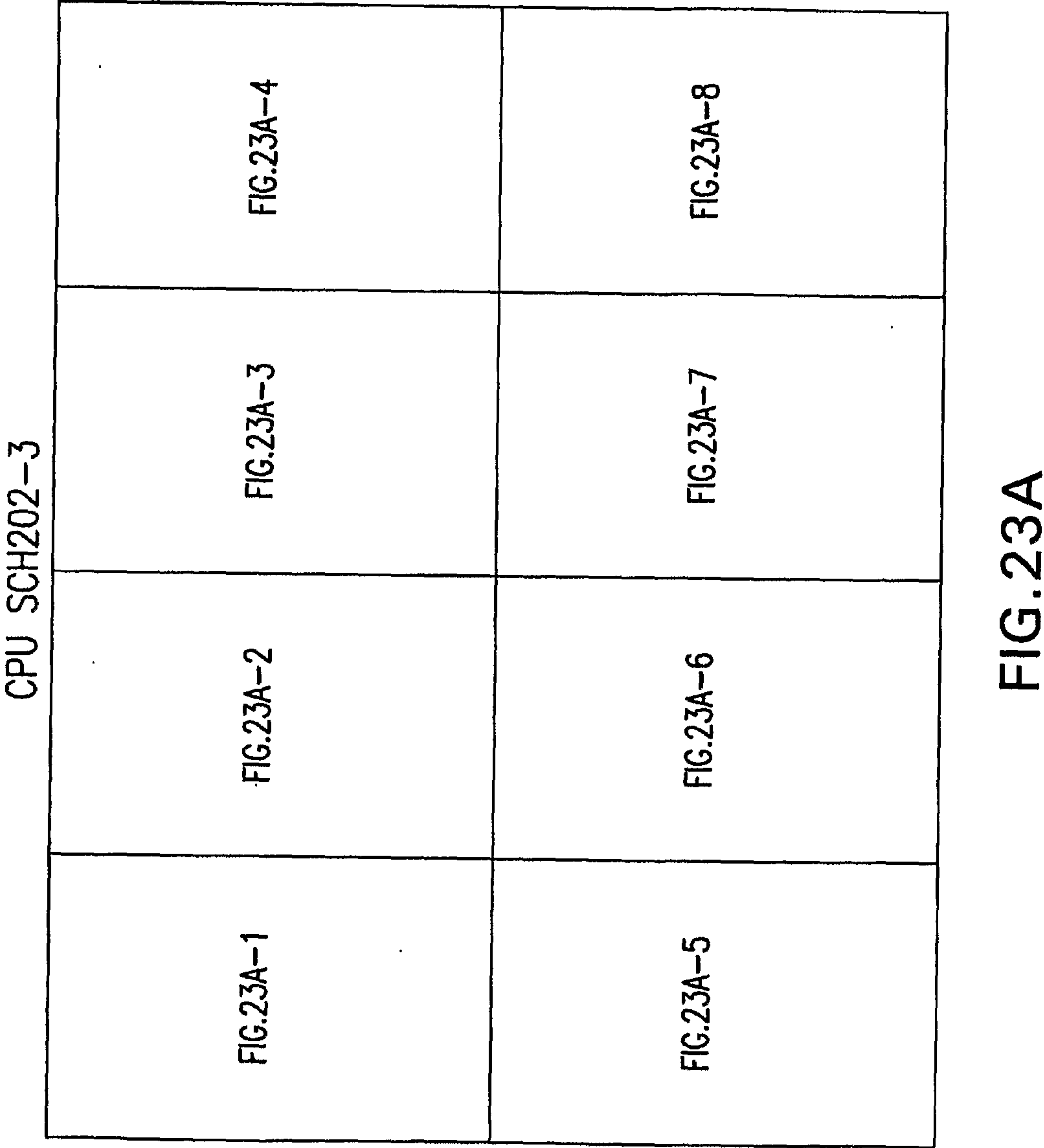


FIG. 22G-4



124/227

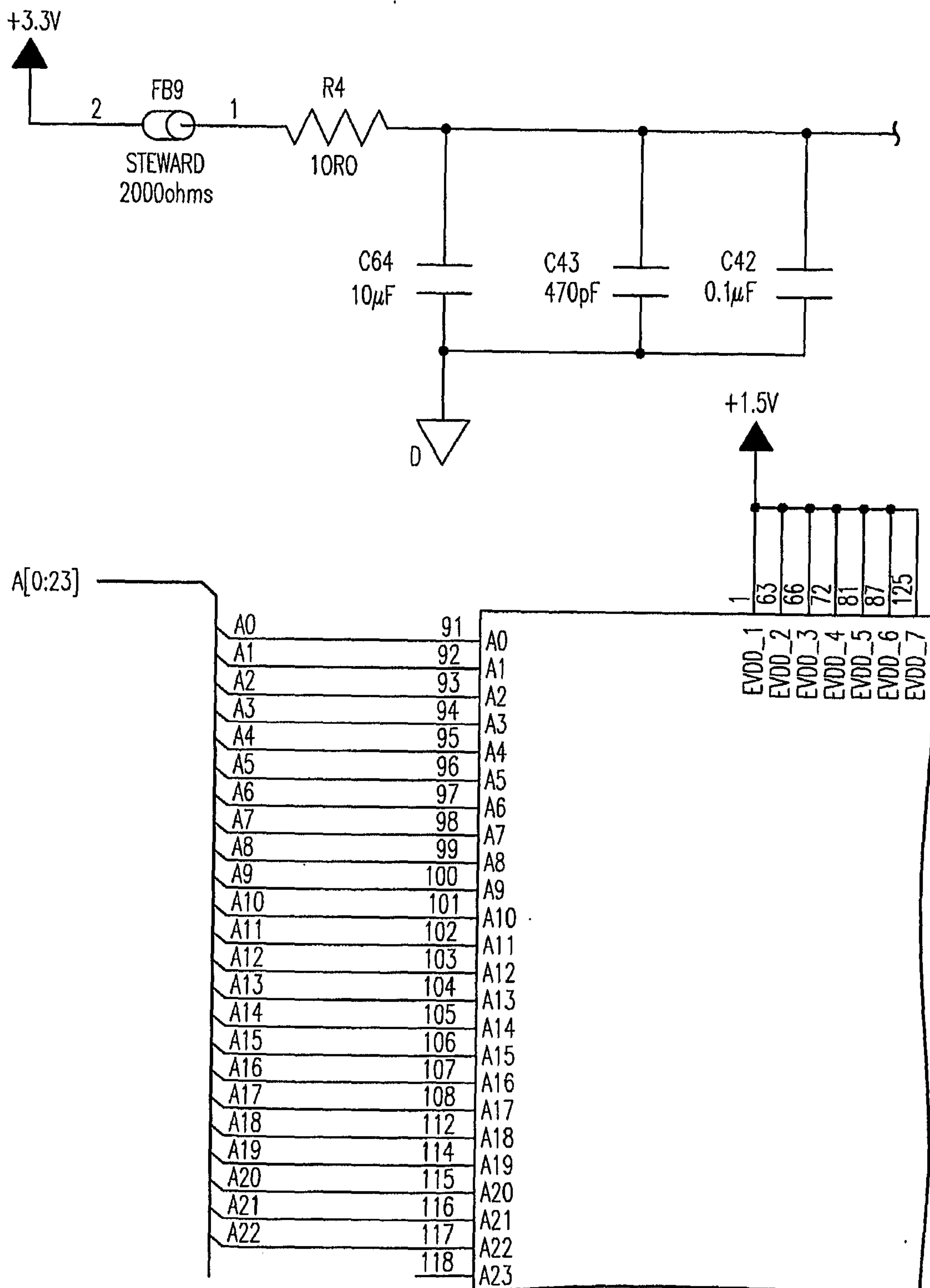


FIG.23A-1

125/227

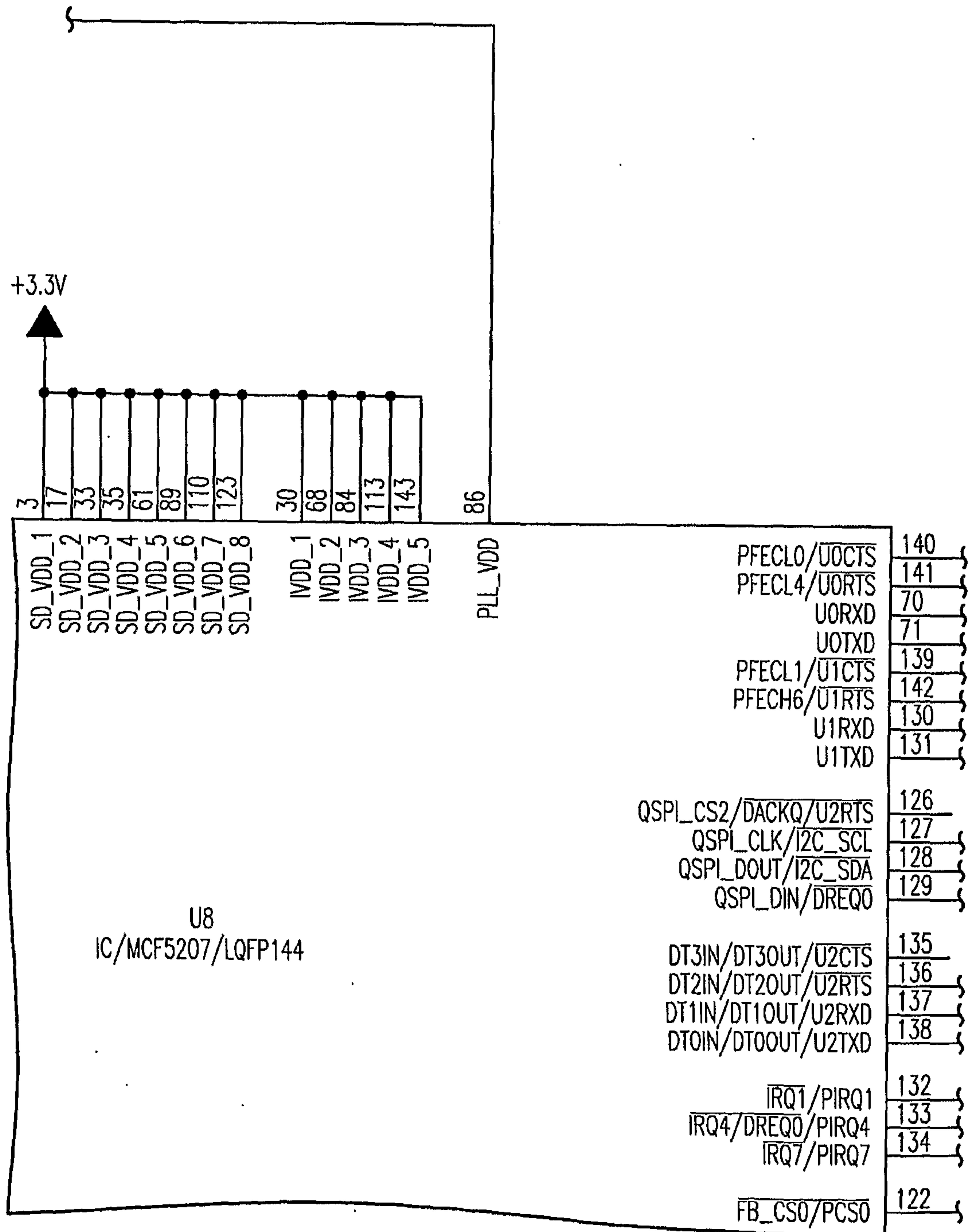


FIG.23A-2

126/227

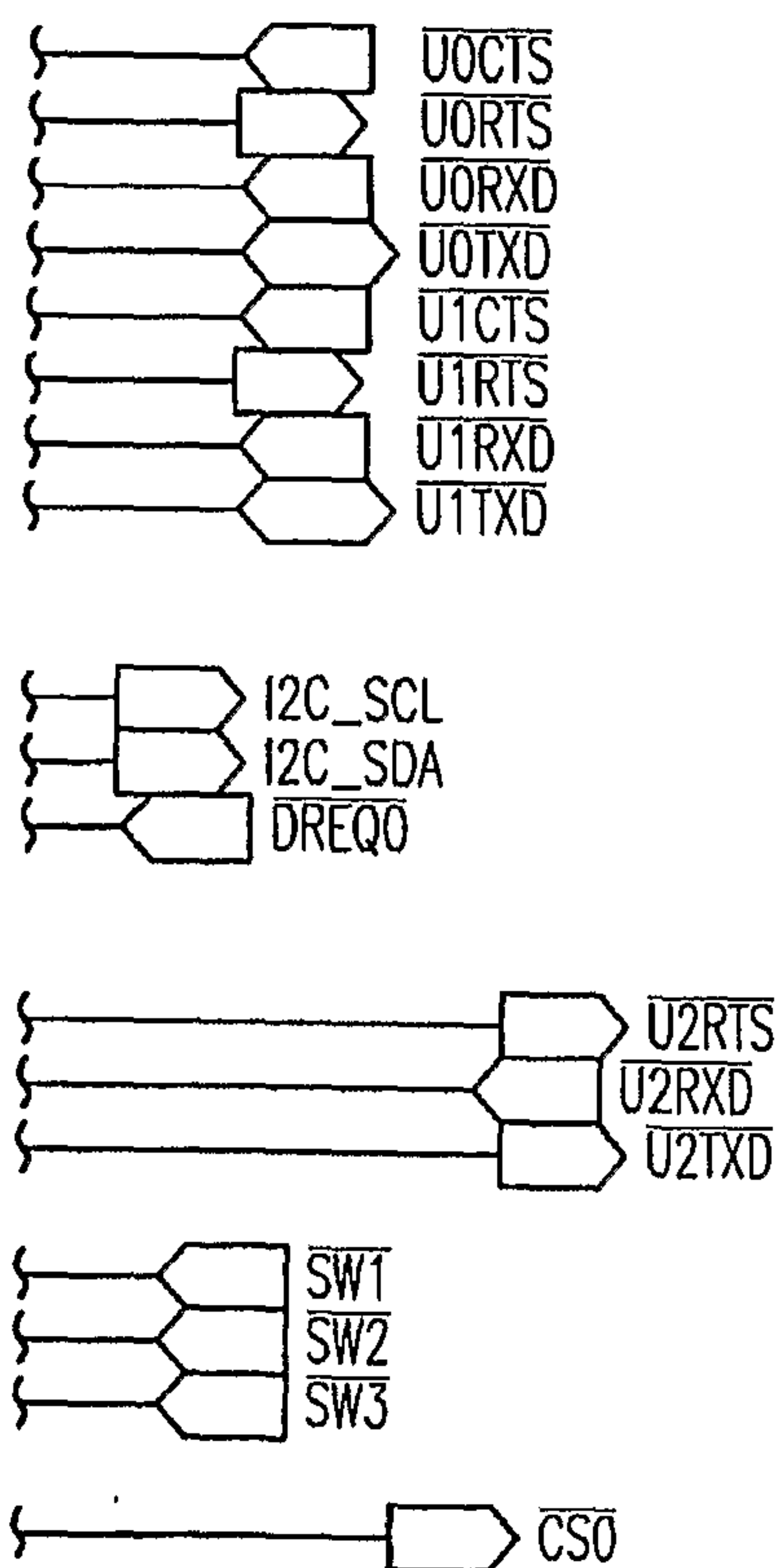


FIG.23A-3

127/227

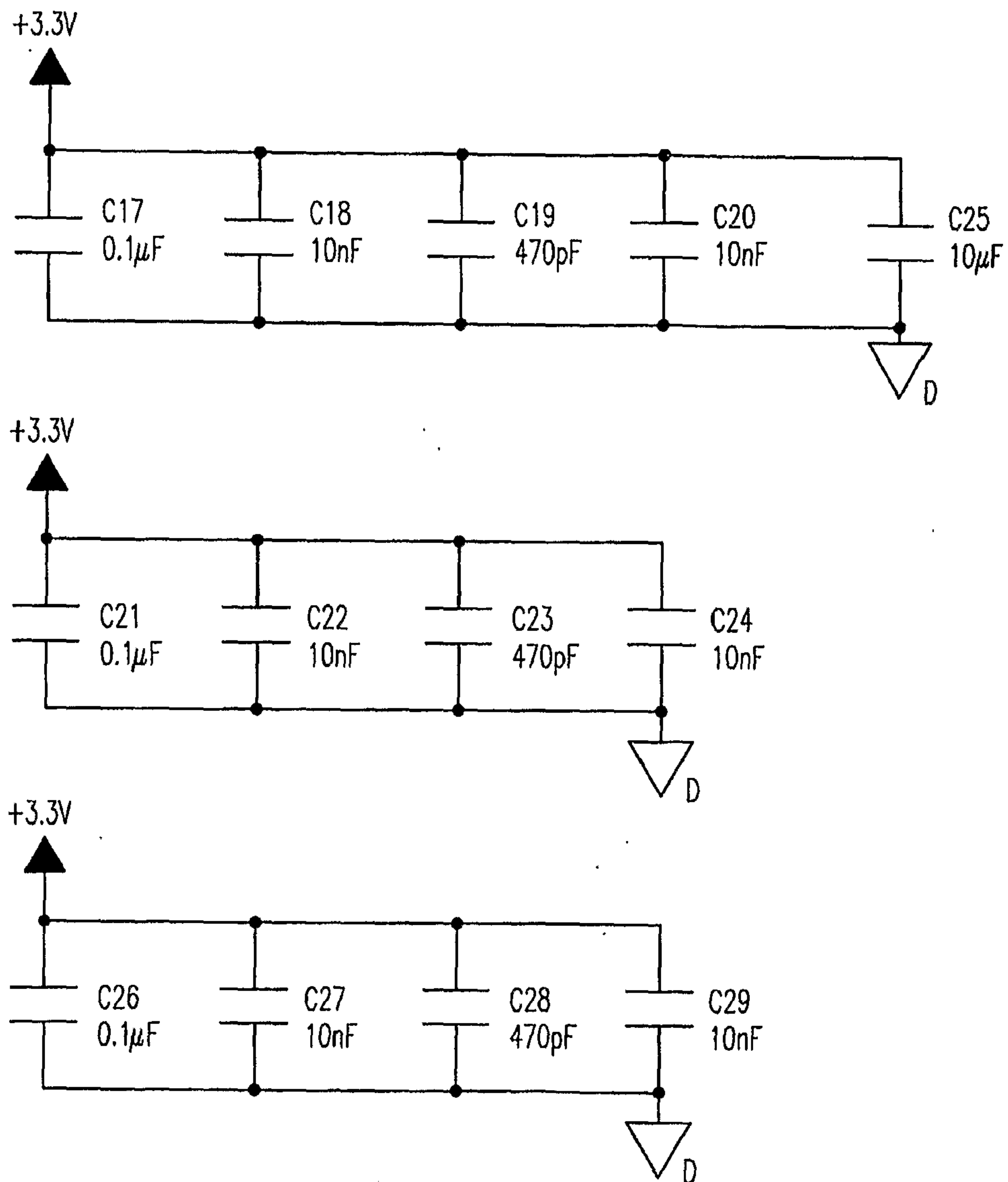


FIG.23A-4

128/227

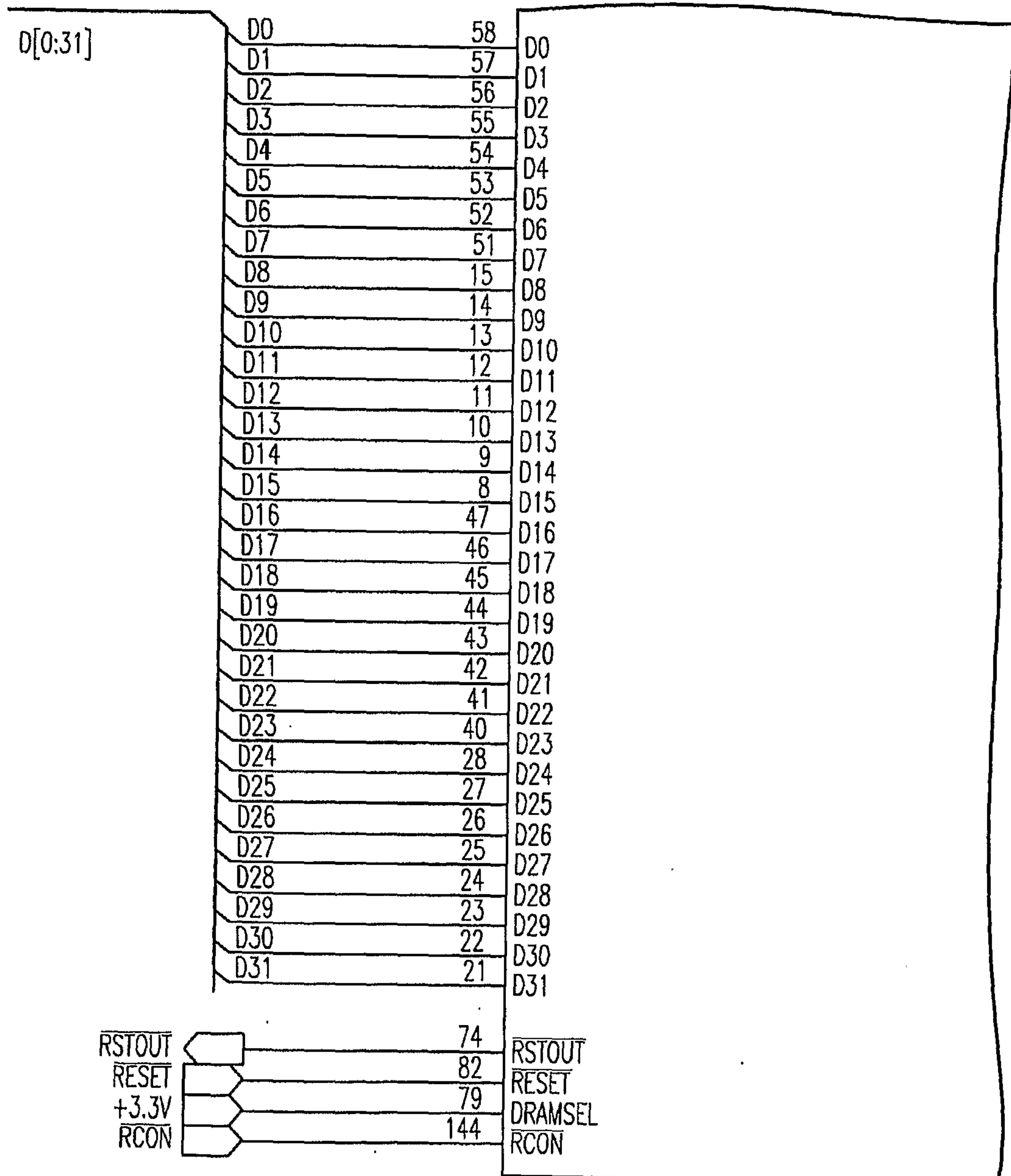


FIG.23A-5

129/227

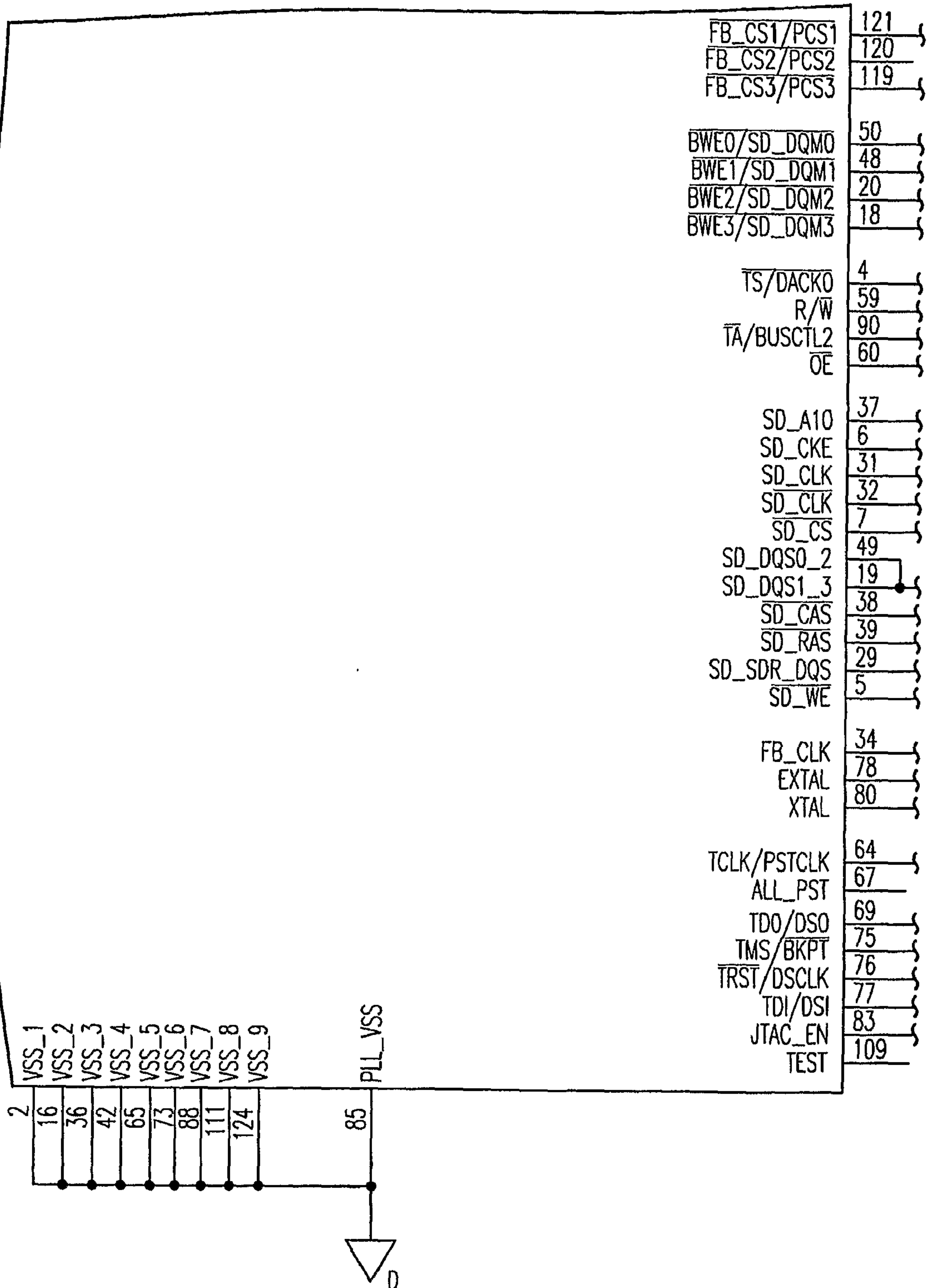


FIG.23A-6

130/227

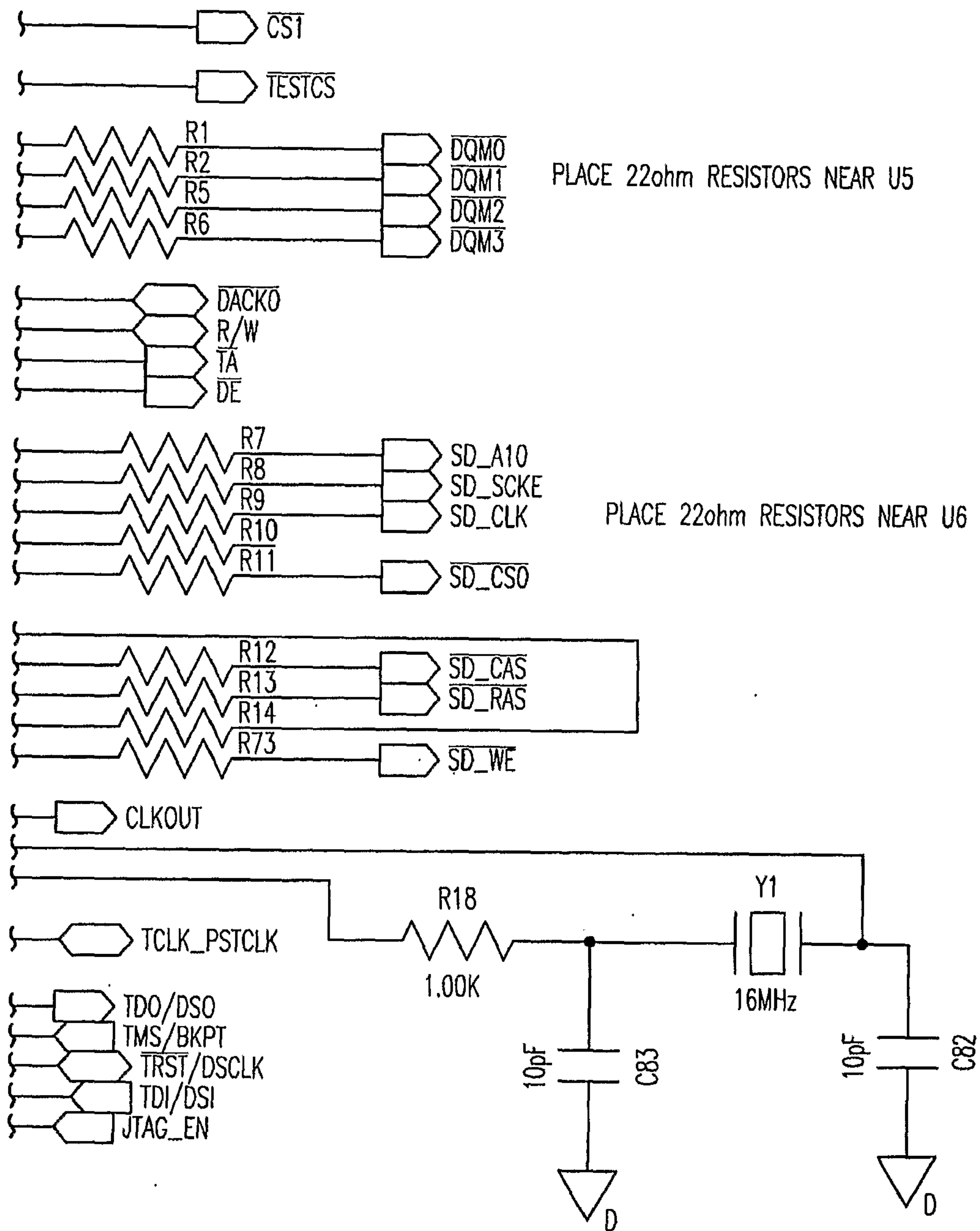


FIG.23A-7

131/227

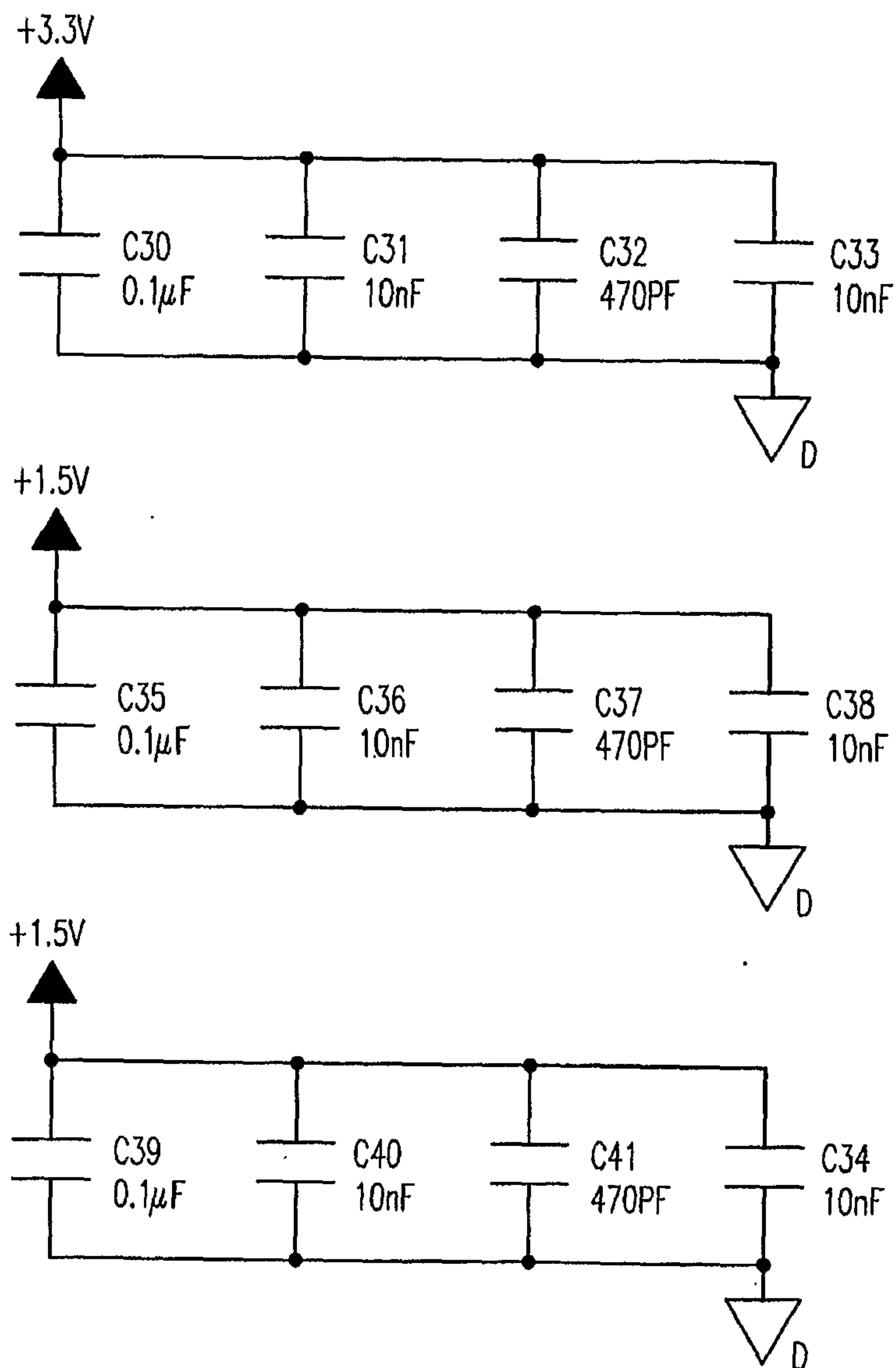


FIG.23A-8

132/227

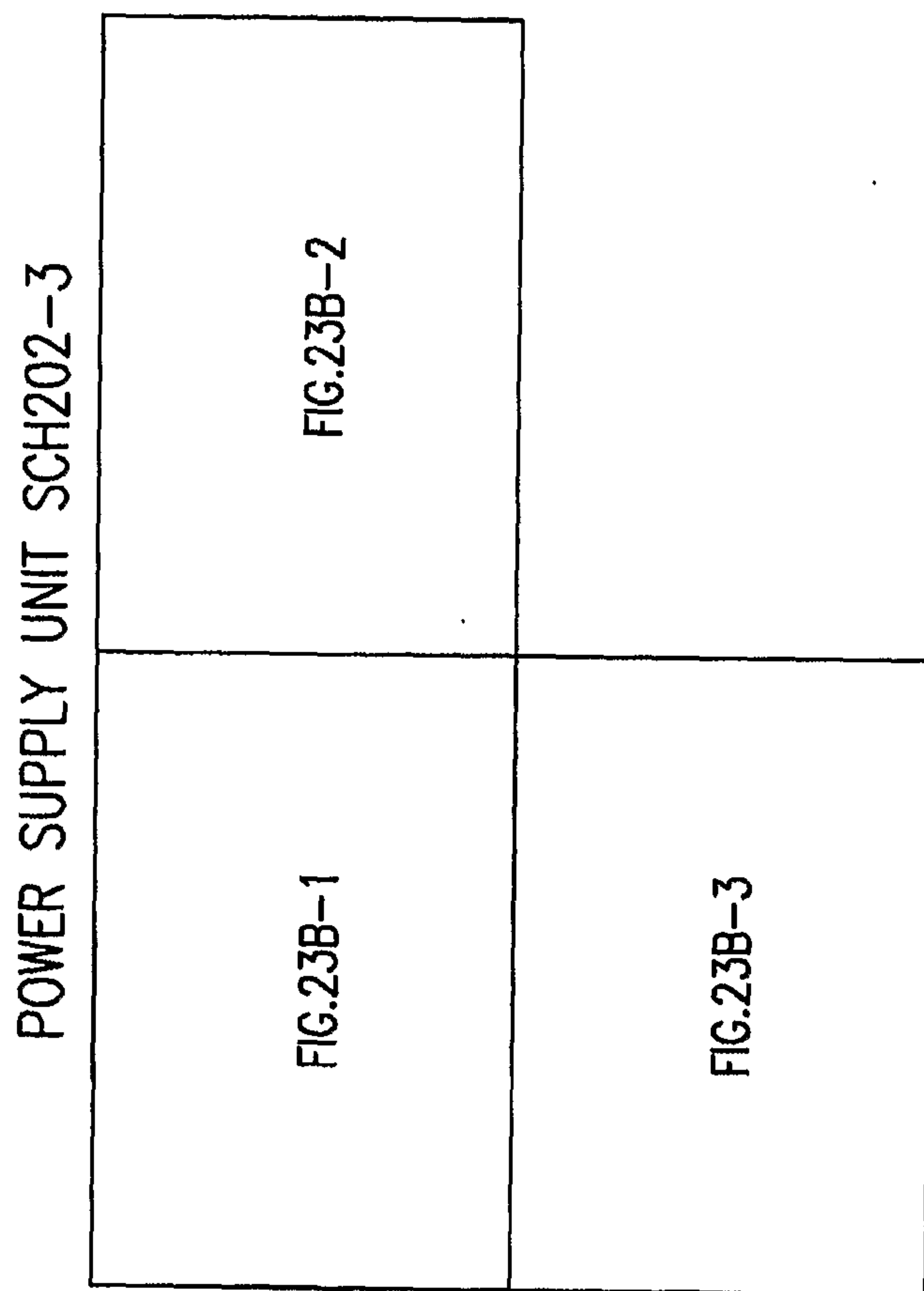


FIG.23B

133/227

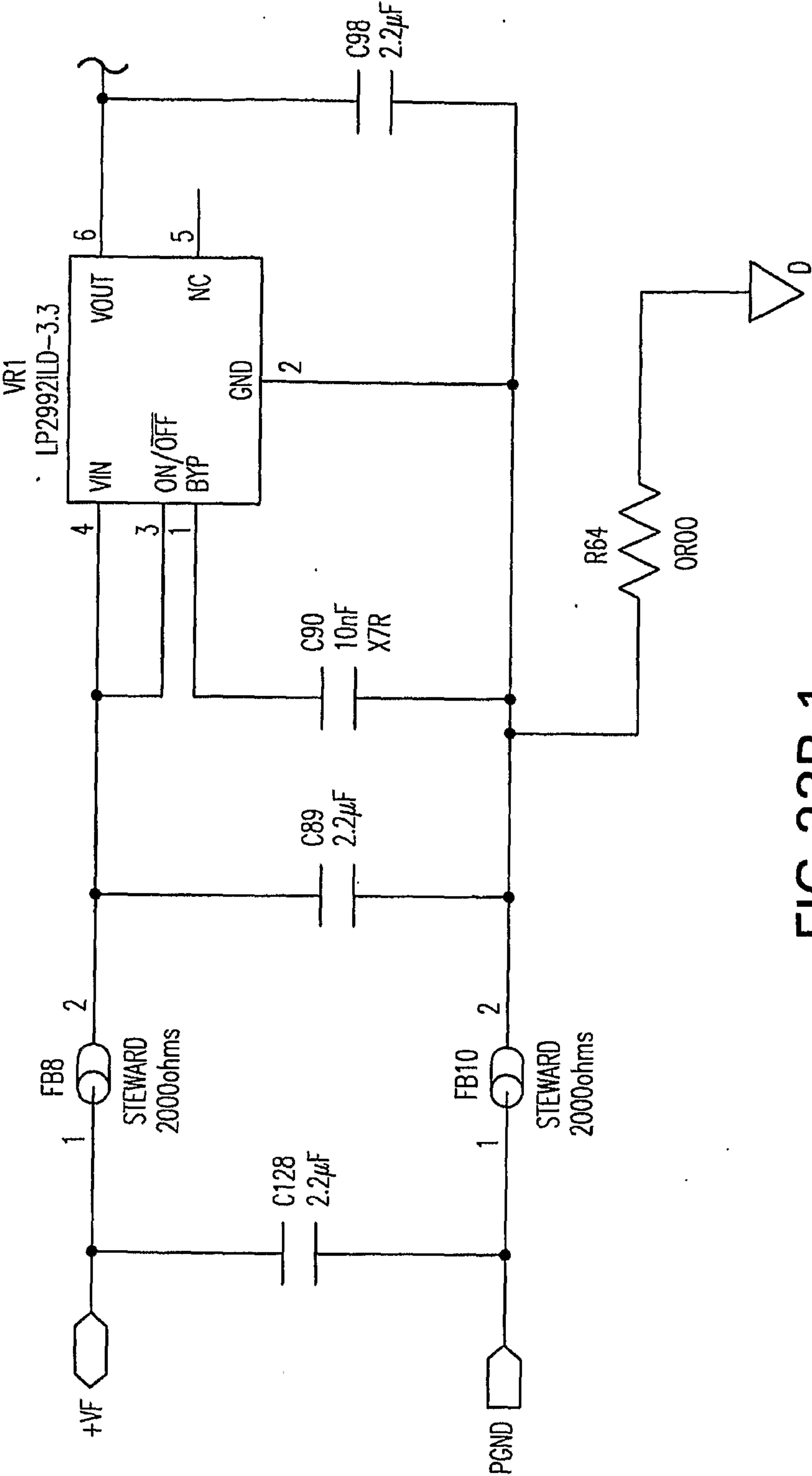


FIG.23B-1

134/227

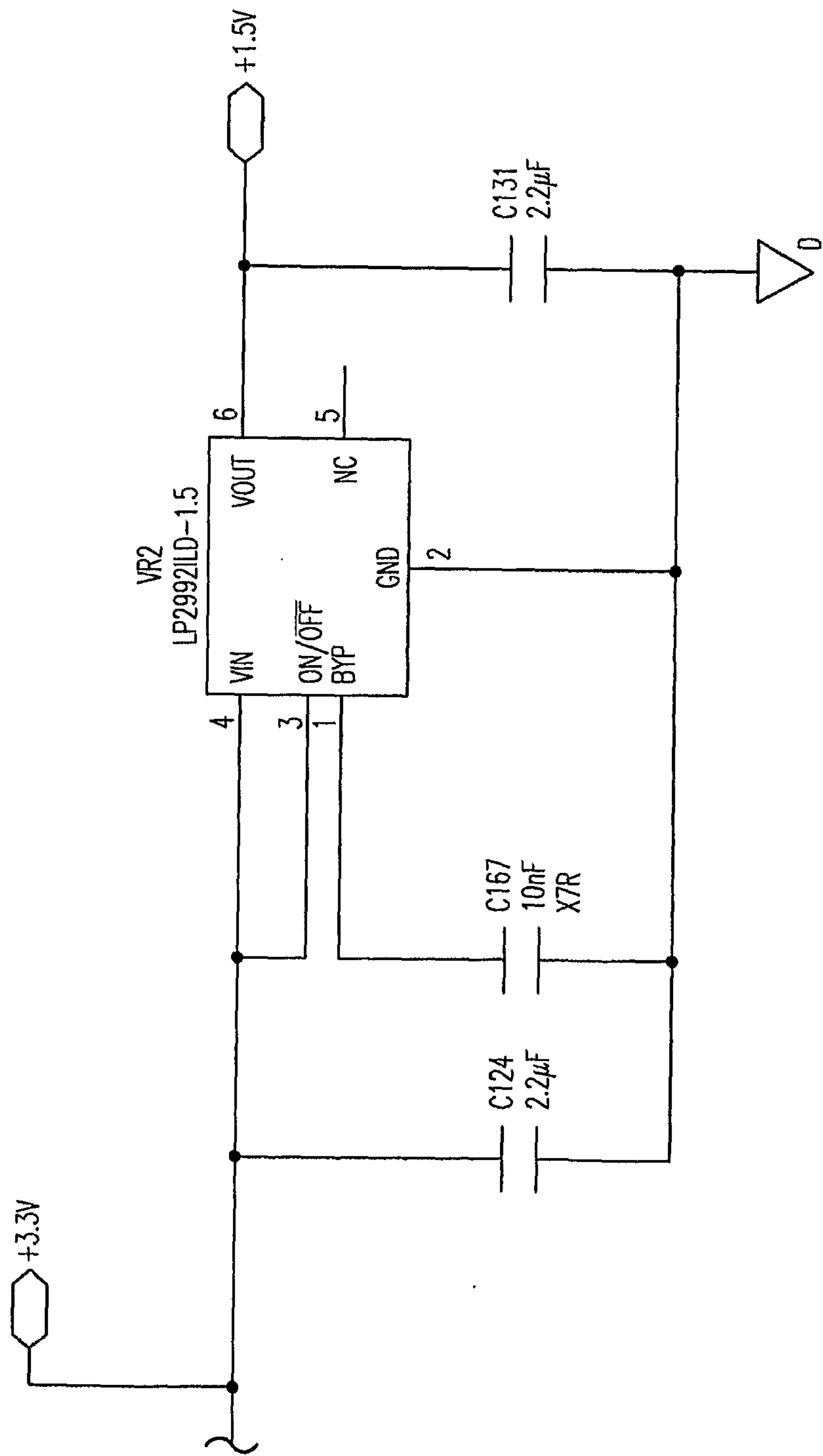


FIG.23B-2

135/227

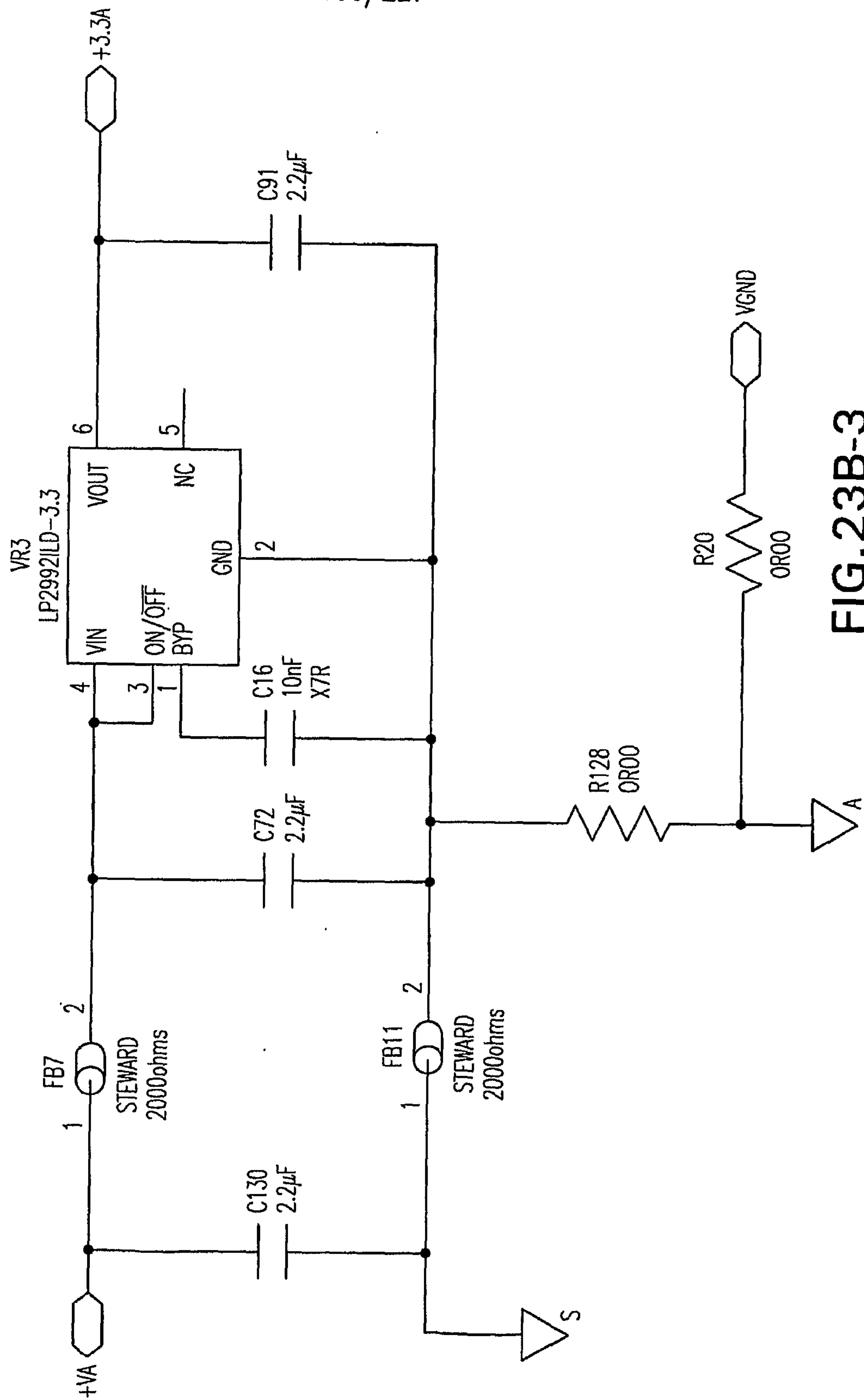


FIG.23B-3

136/227

DMA RCON PCB202-3

FIG.23C-1	FIG.23C-2	FIG.23C-3
FIG.23C-4	FIG.23C-5	FIG.23C-6

FIG.23C

137/227

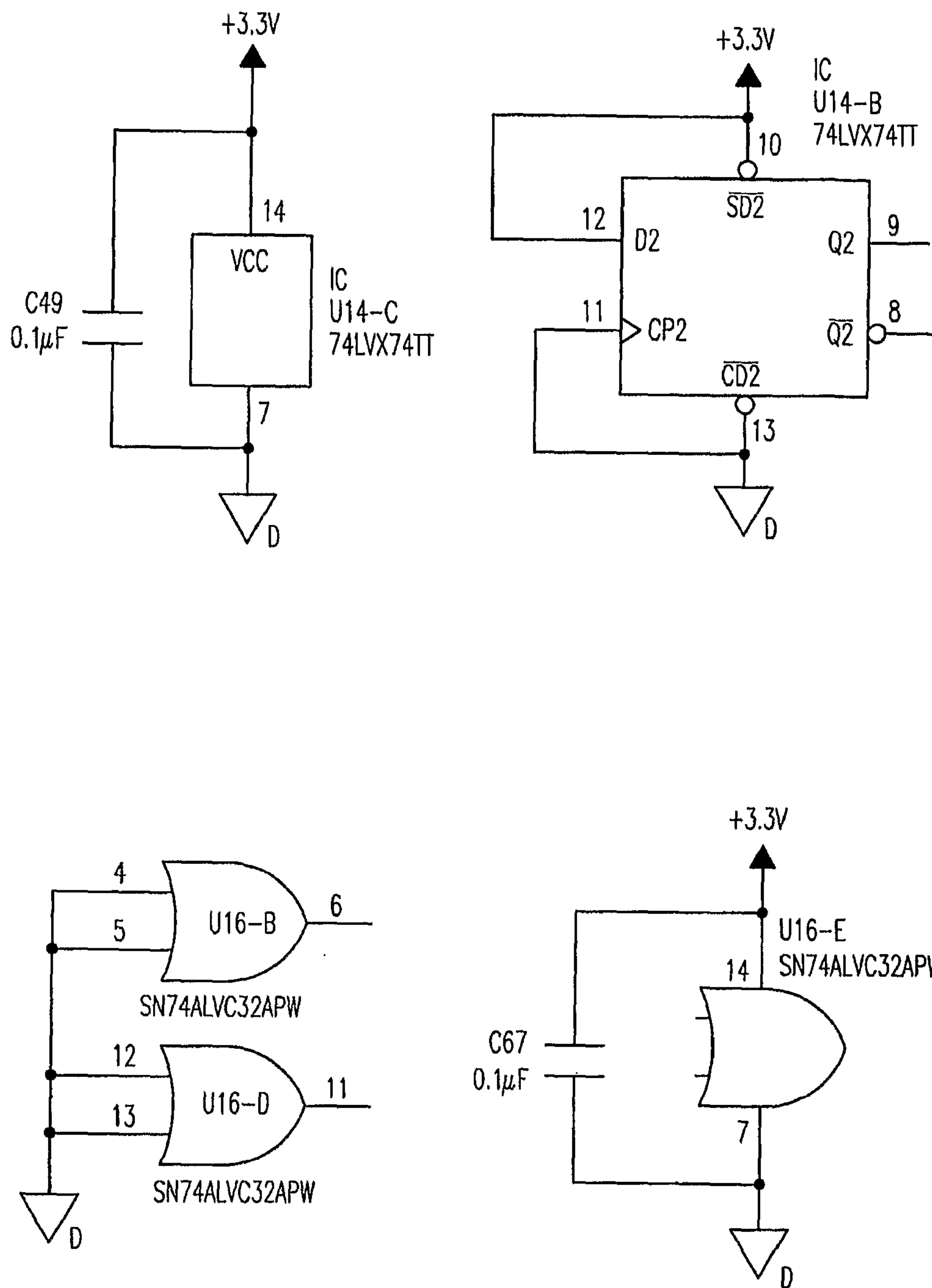


FIG.23C-1

138/227

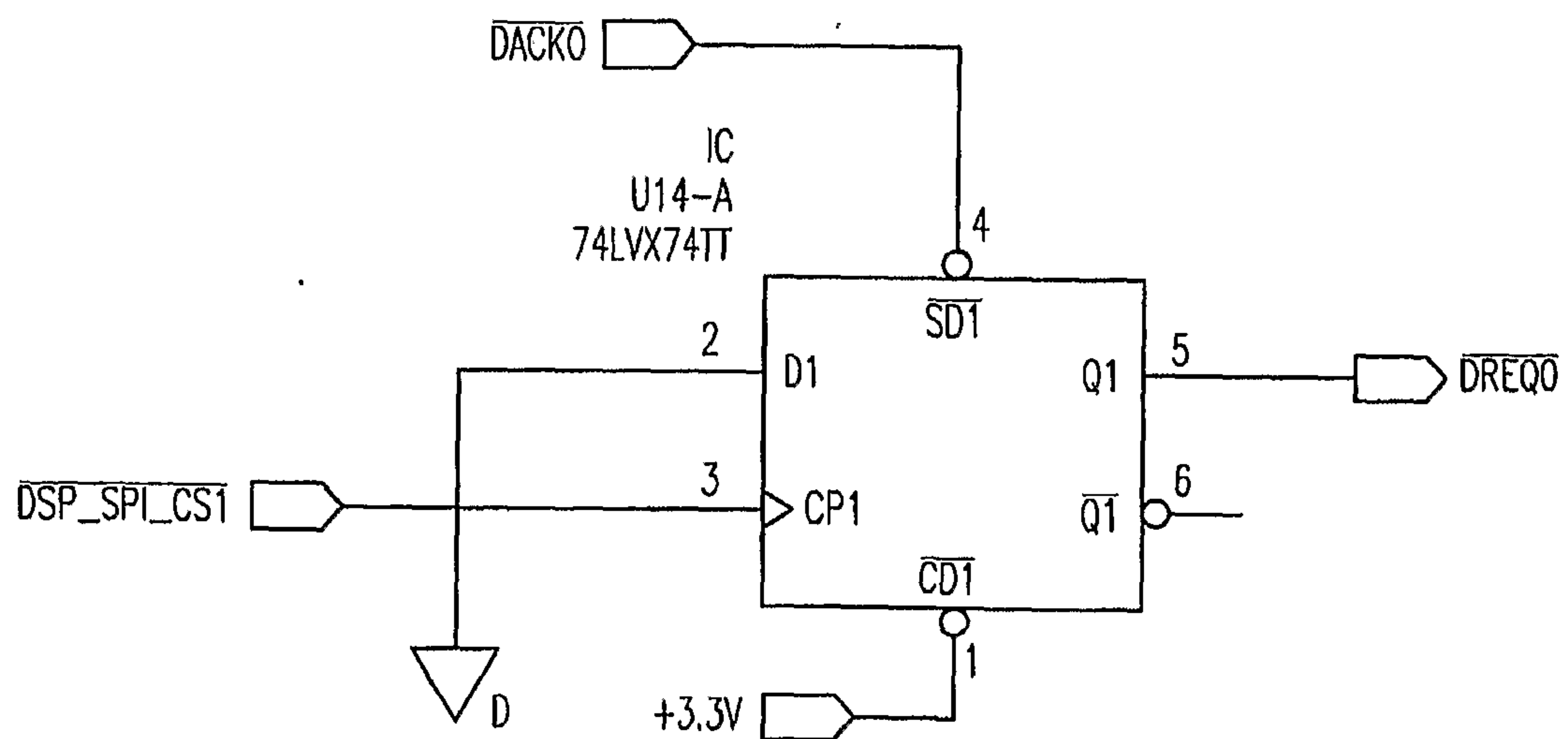
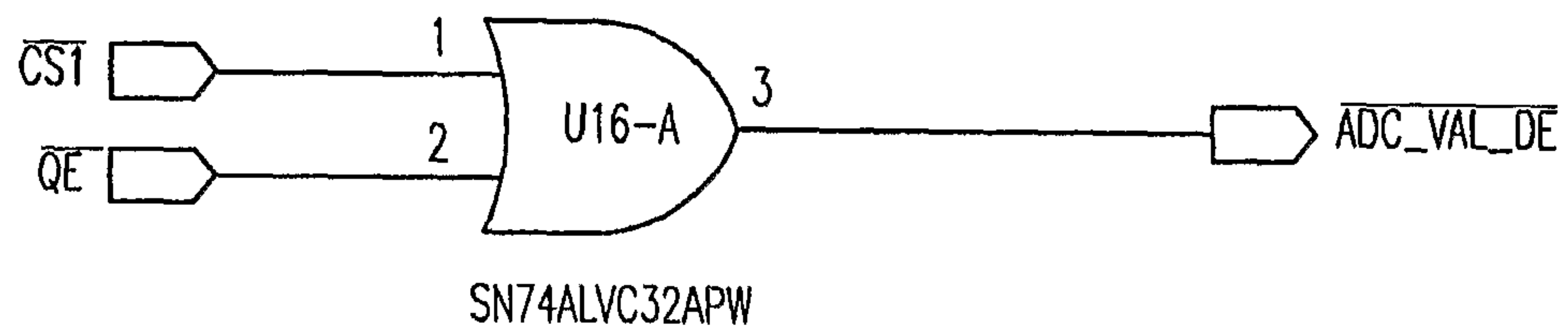
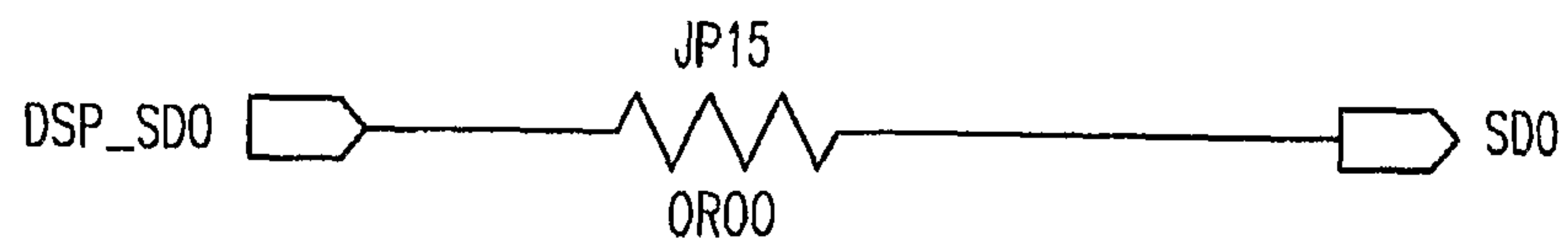


FIG.23C-2

139/227

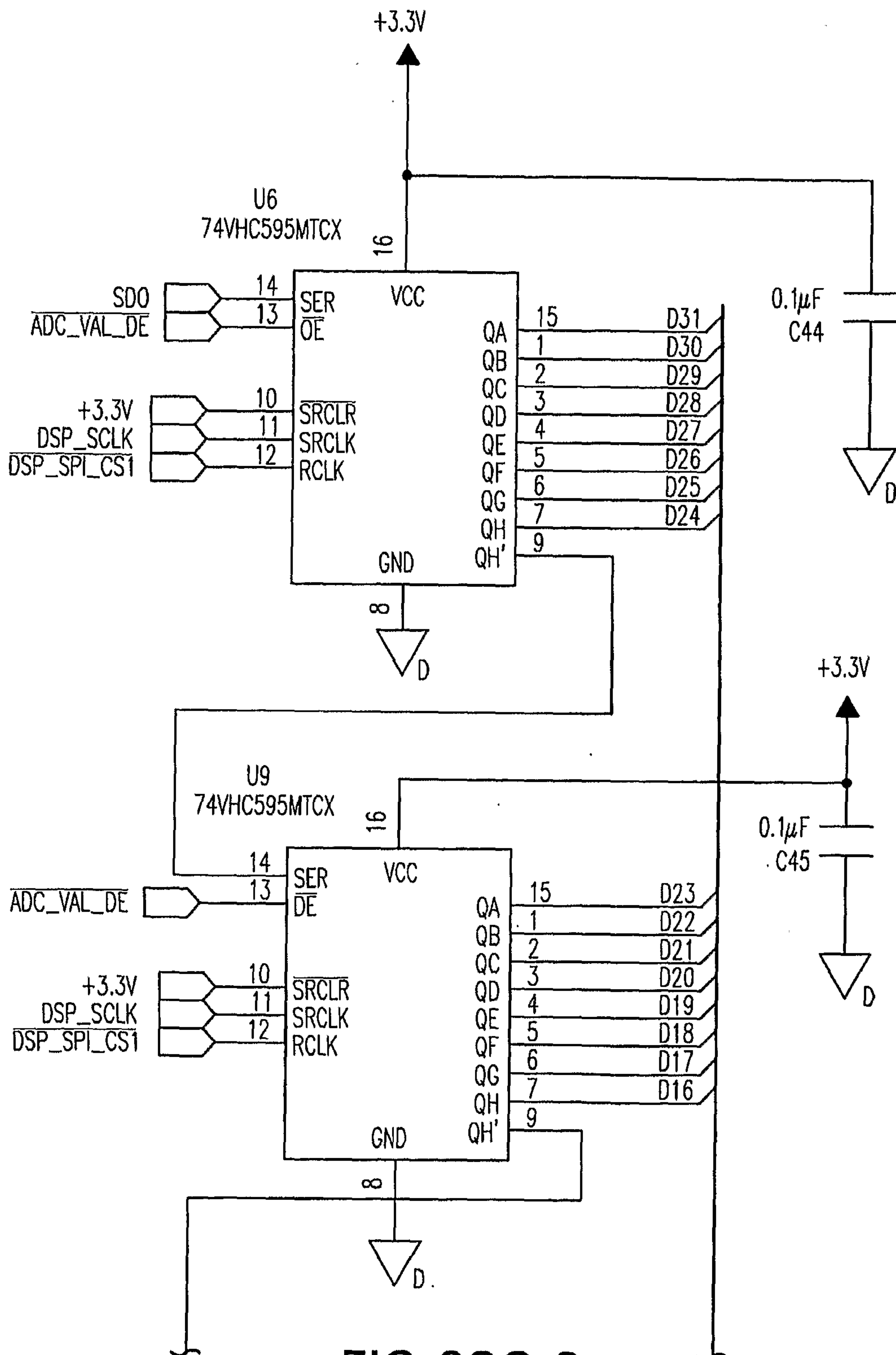


FIG.23C-3

140/227

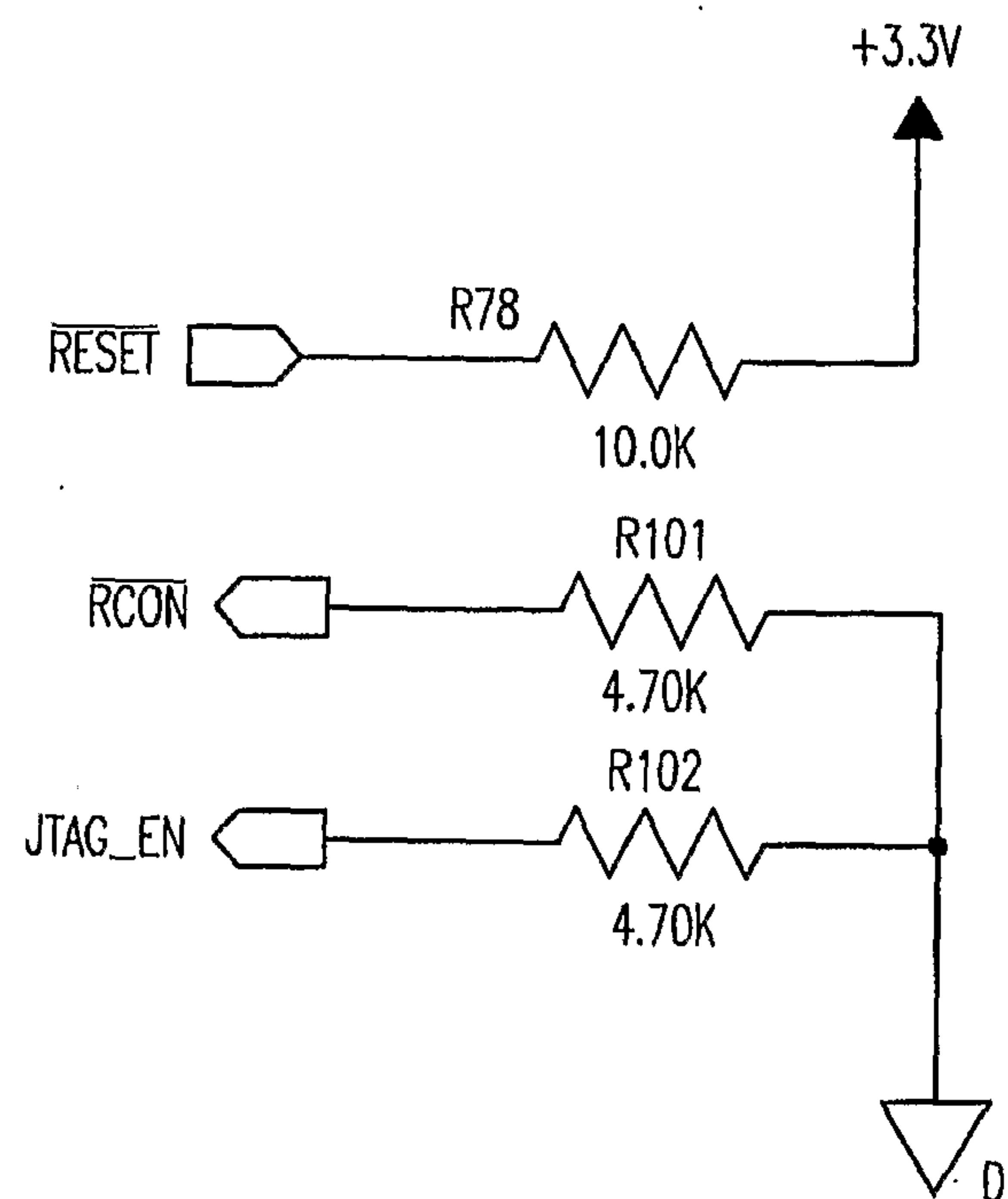


FIG.23C-4

141/227

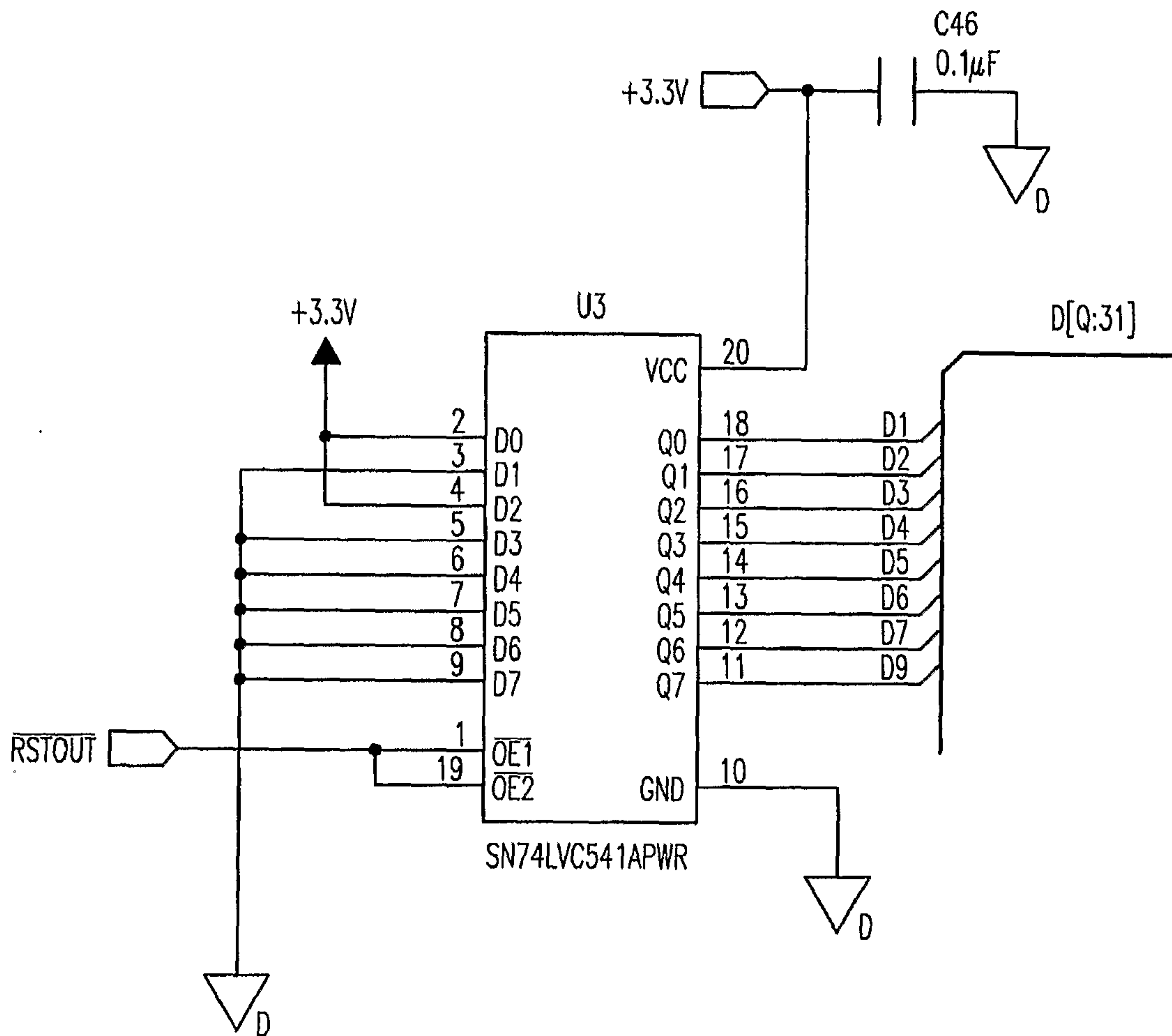


FIG.23C-5

142/227

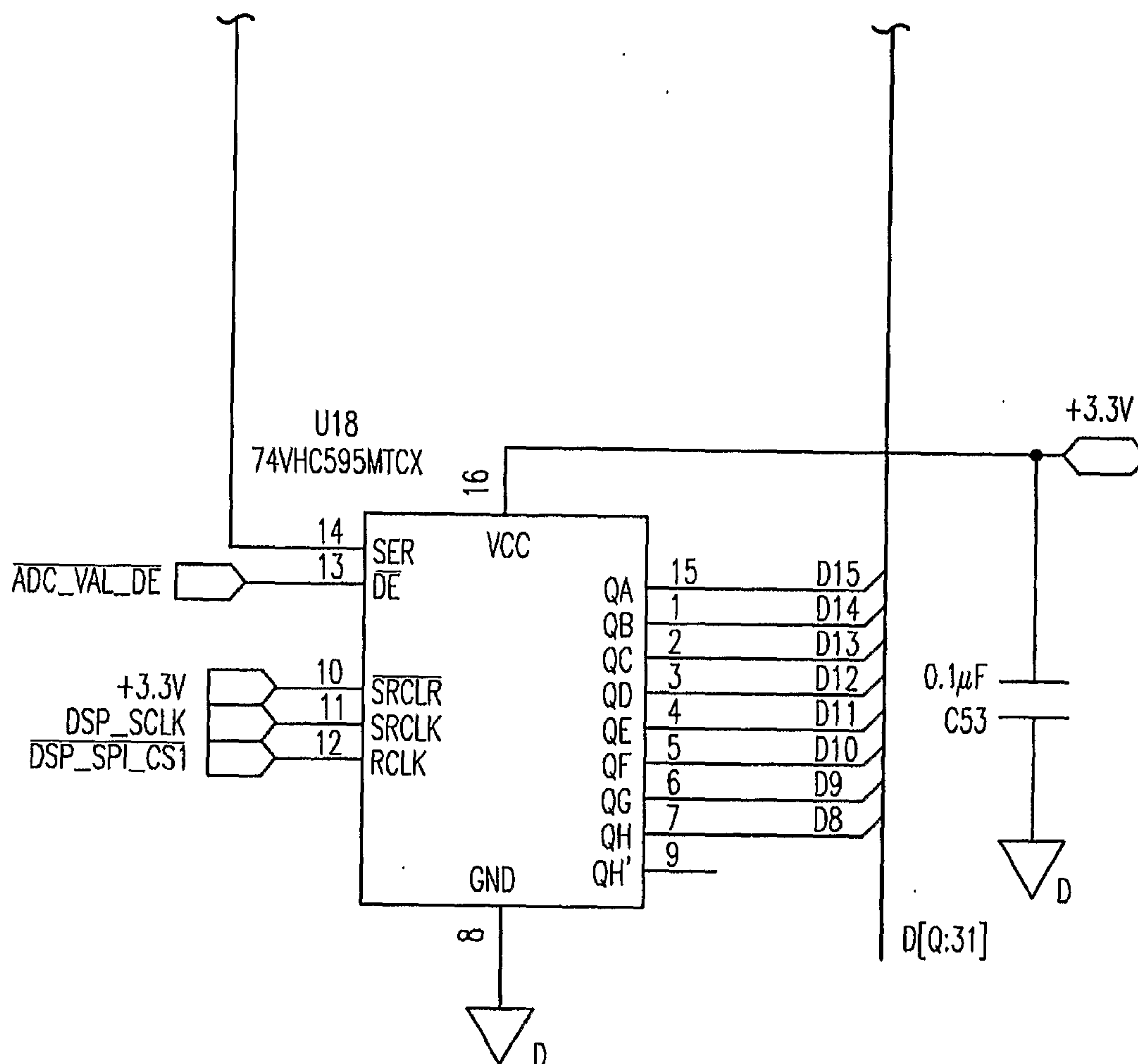
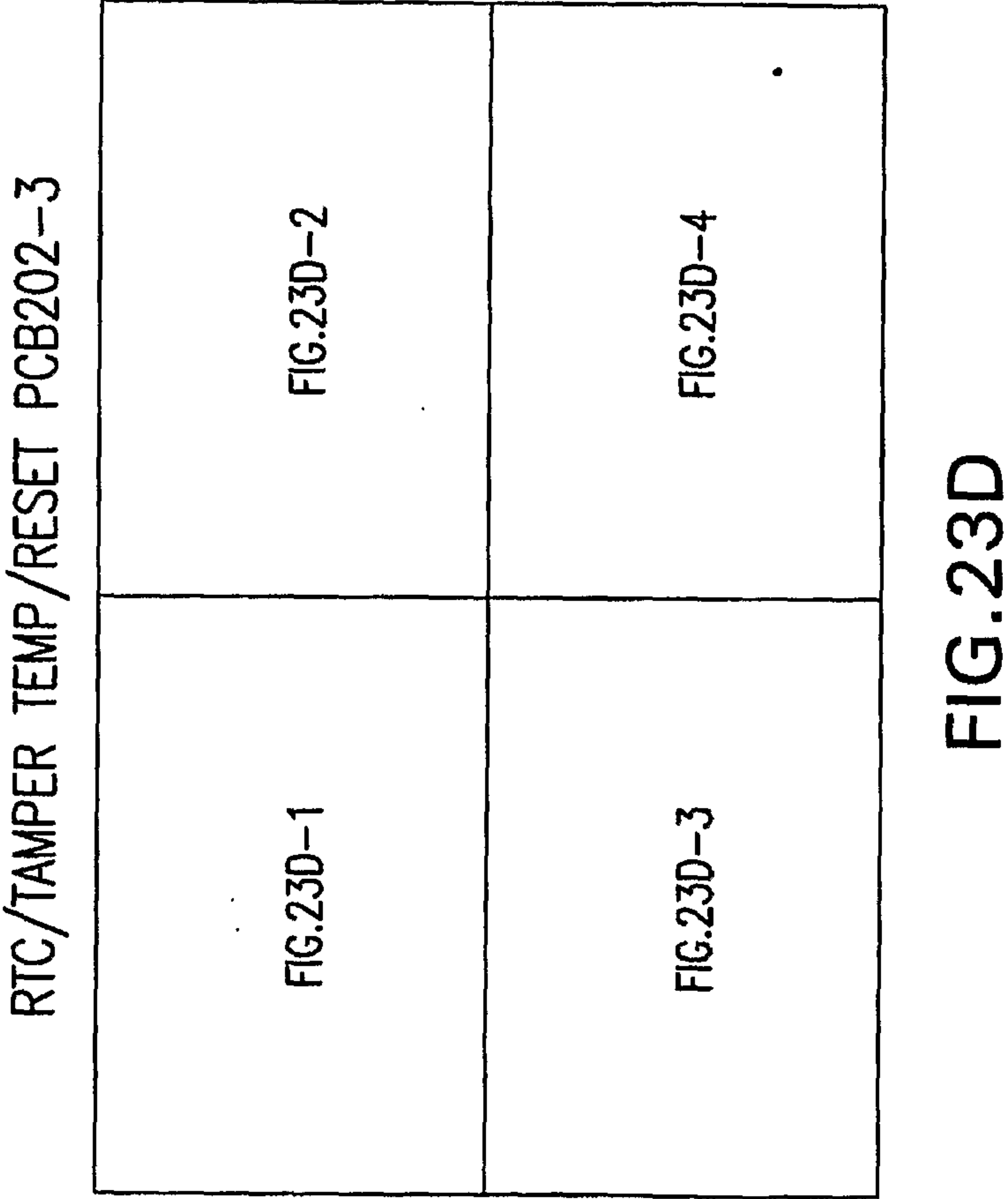


FIG.23C-6



144/227

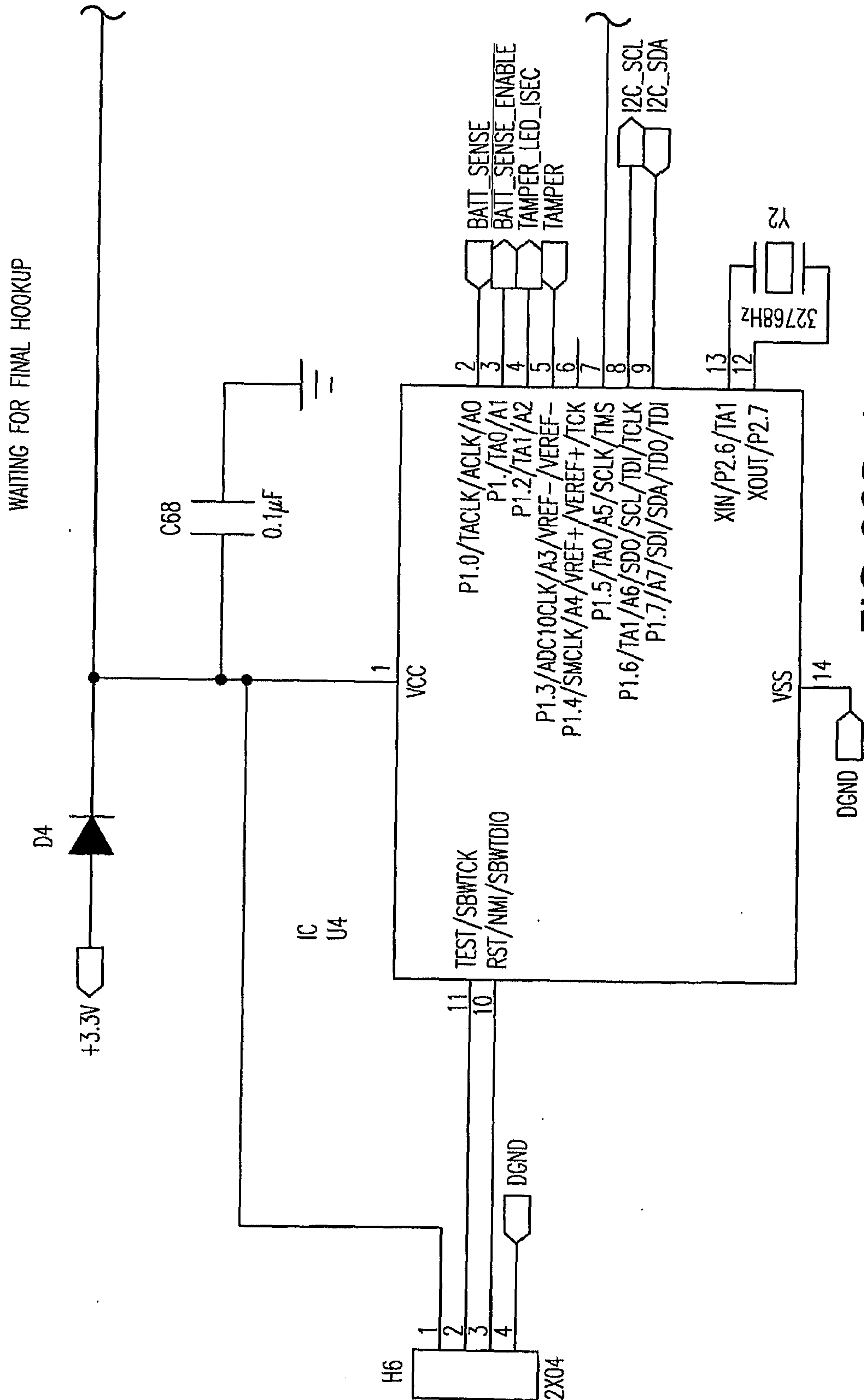


FIG.23D-1

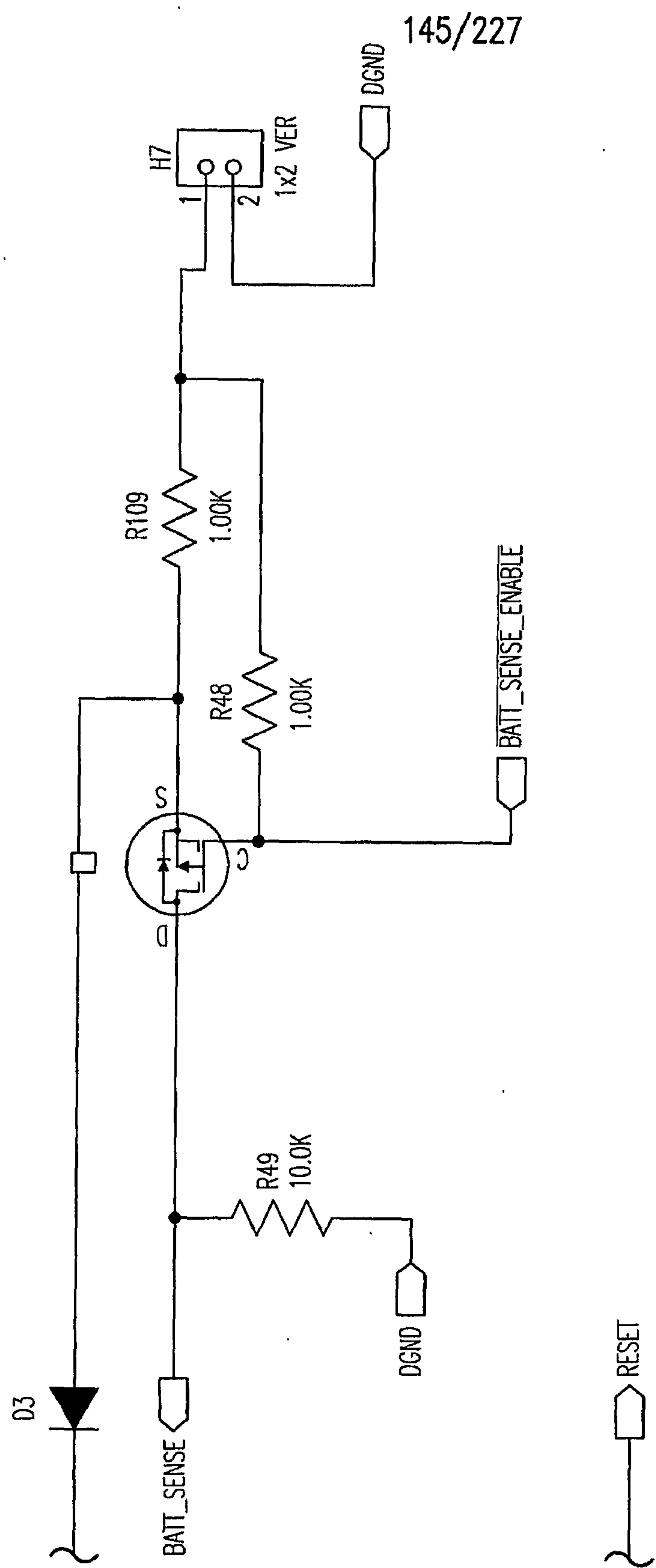
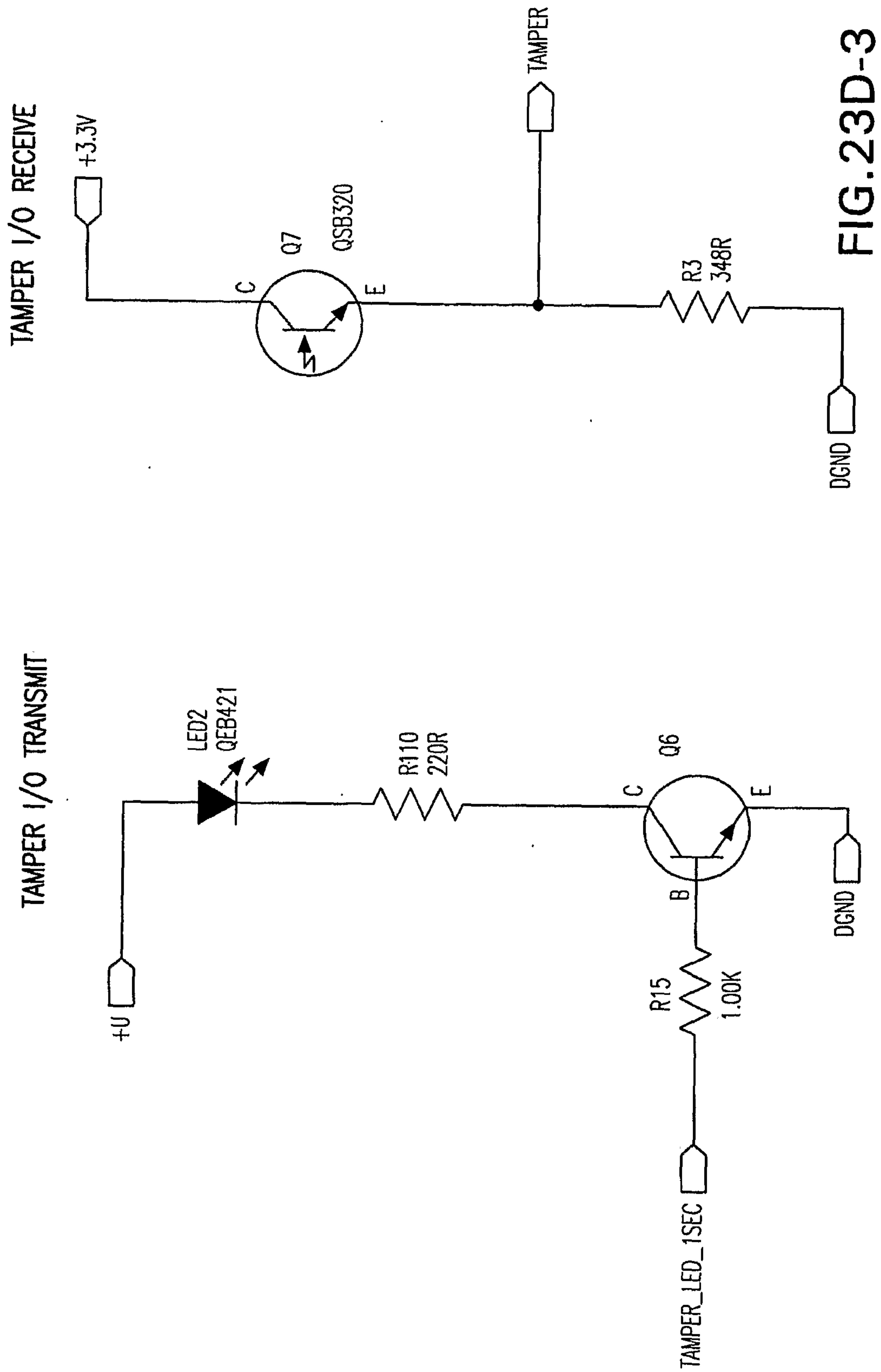


FIG. 23D-2

146/227



147/227

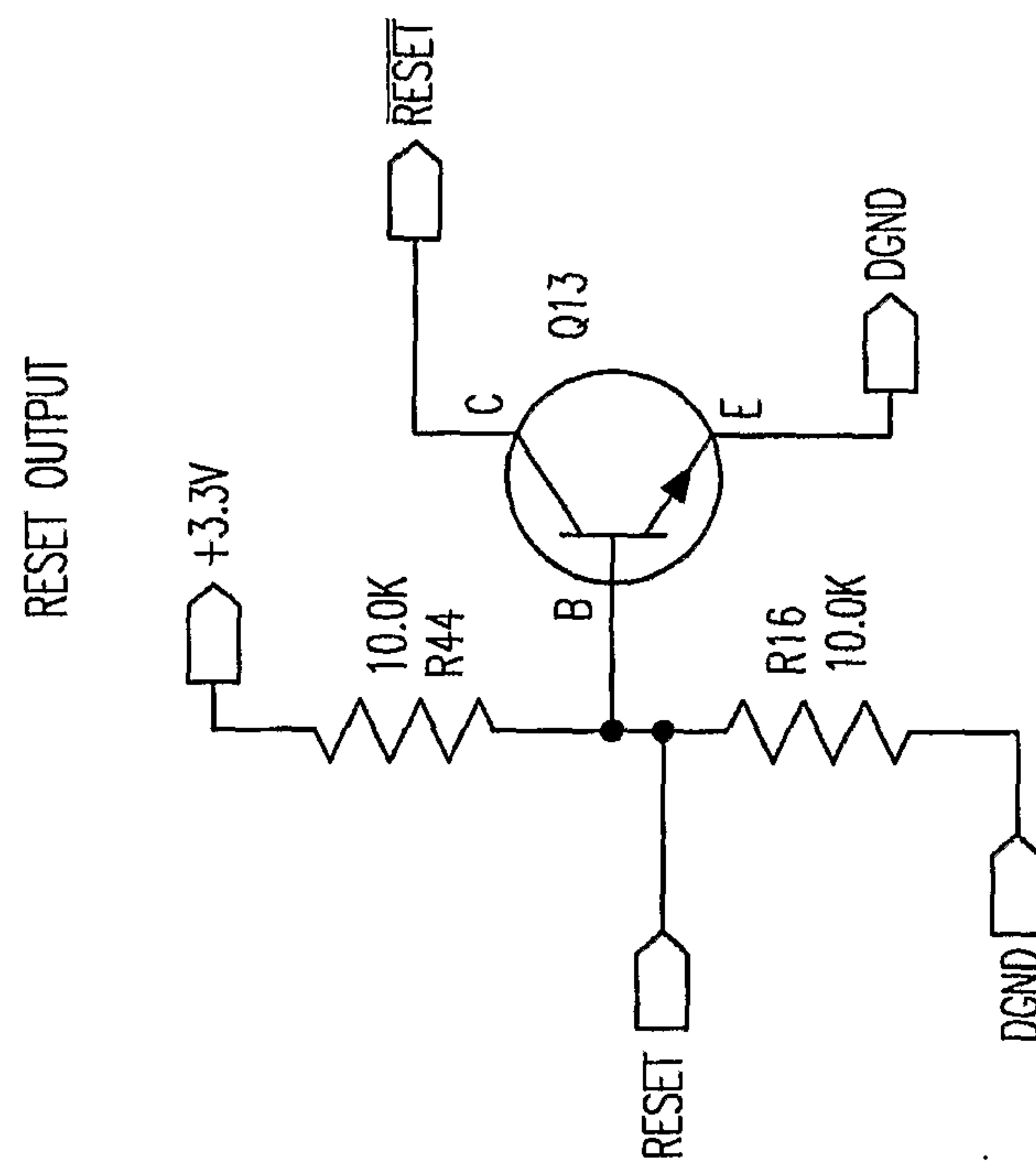


FIG.23D-4

148/227

FLASH MEMORY SCH202-3

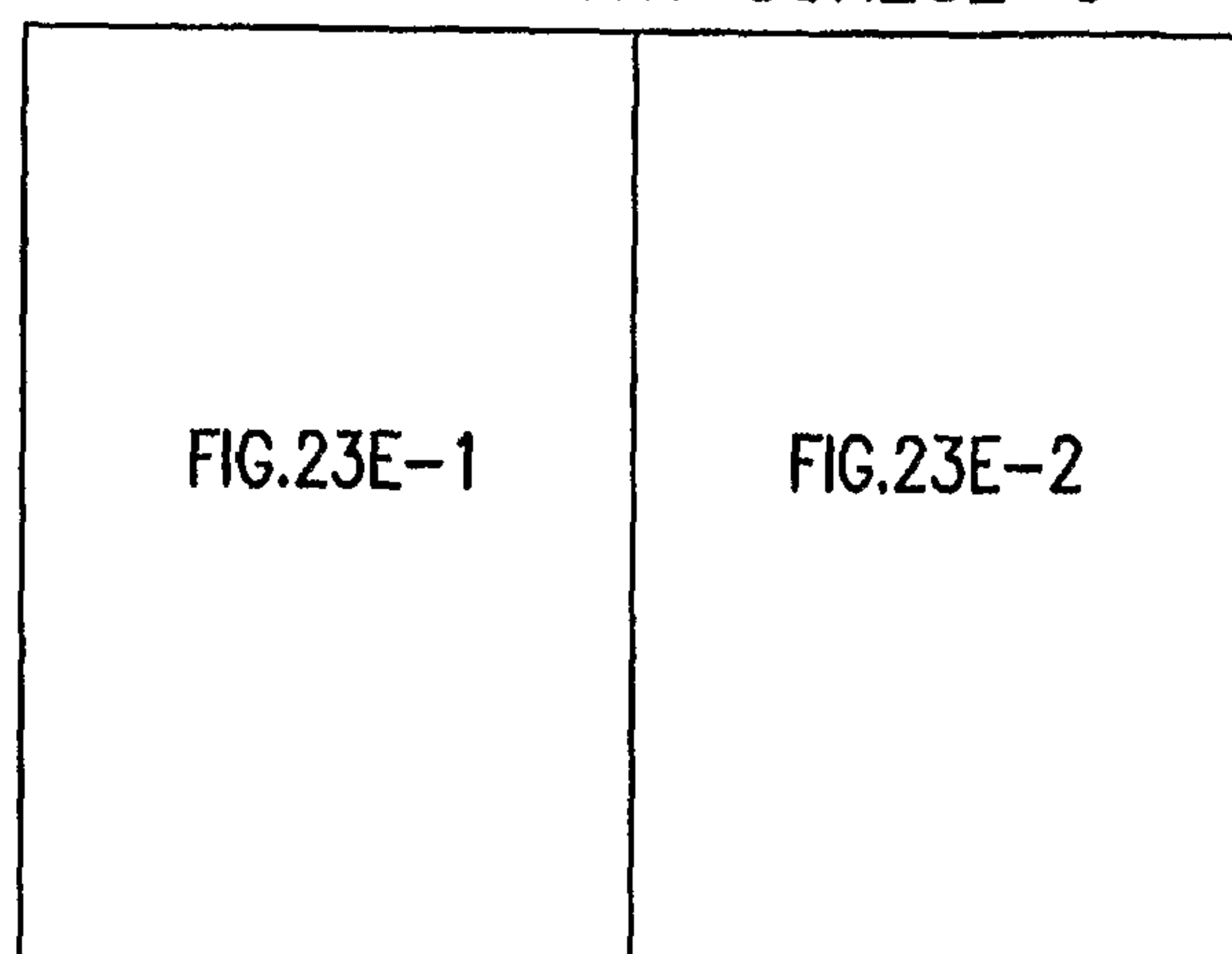


FIG.23E

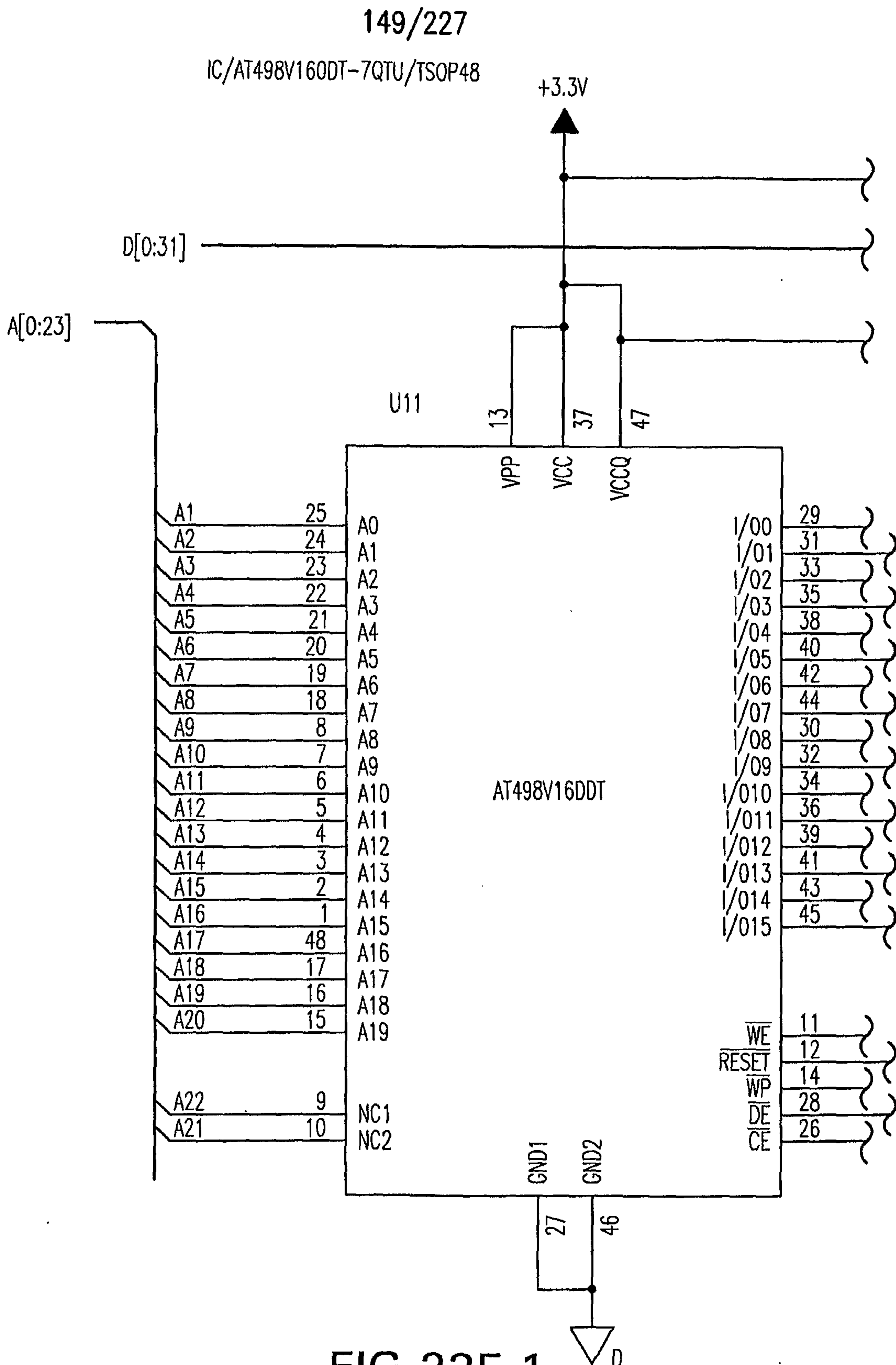


FIG.23E-1

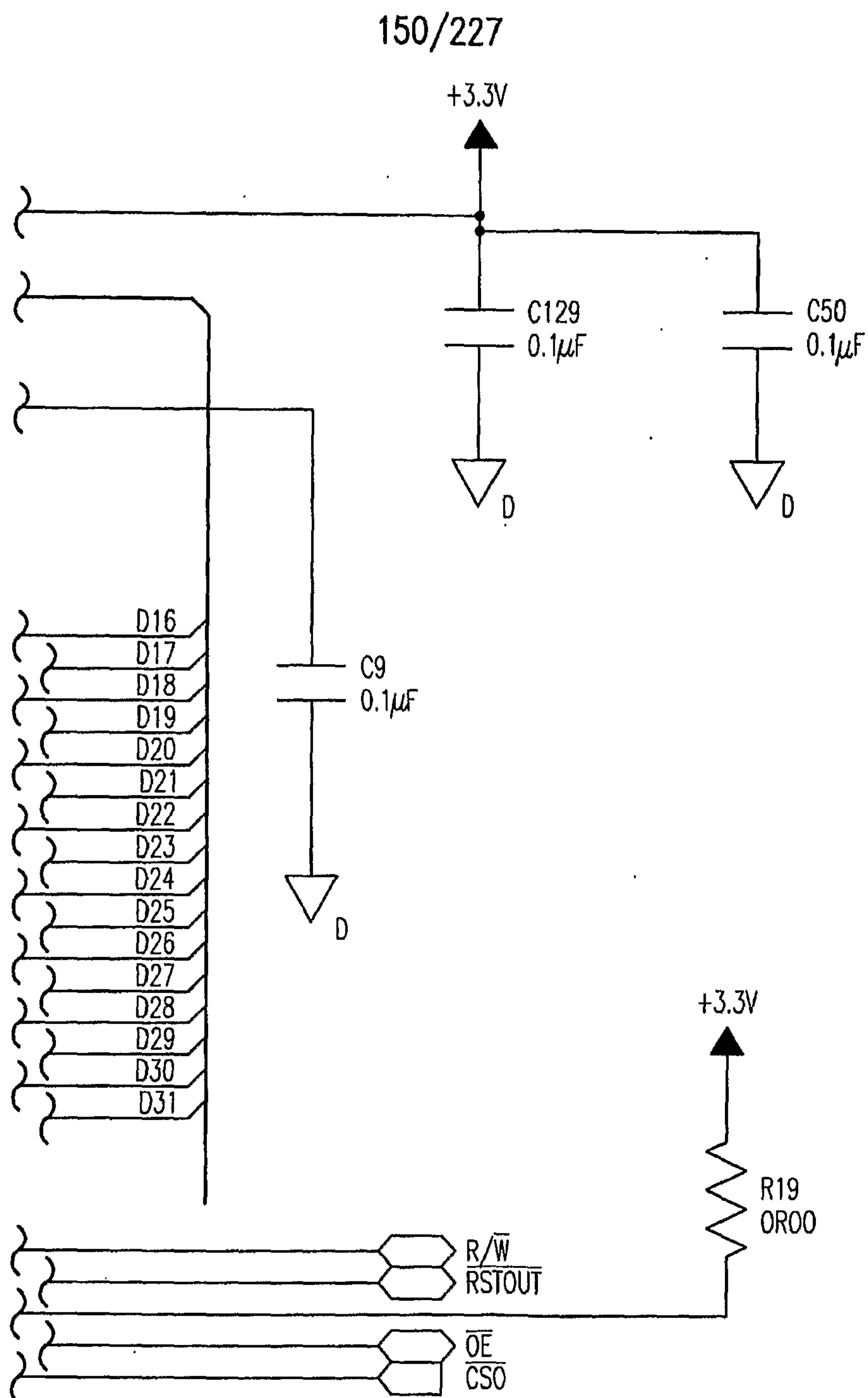


FIG. 23E-2

151/227

SDRAM MEMORY SCH202-3

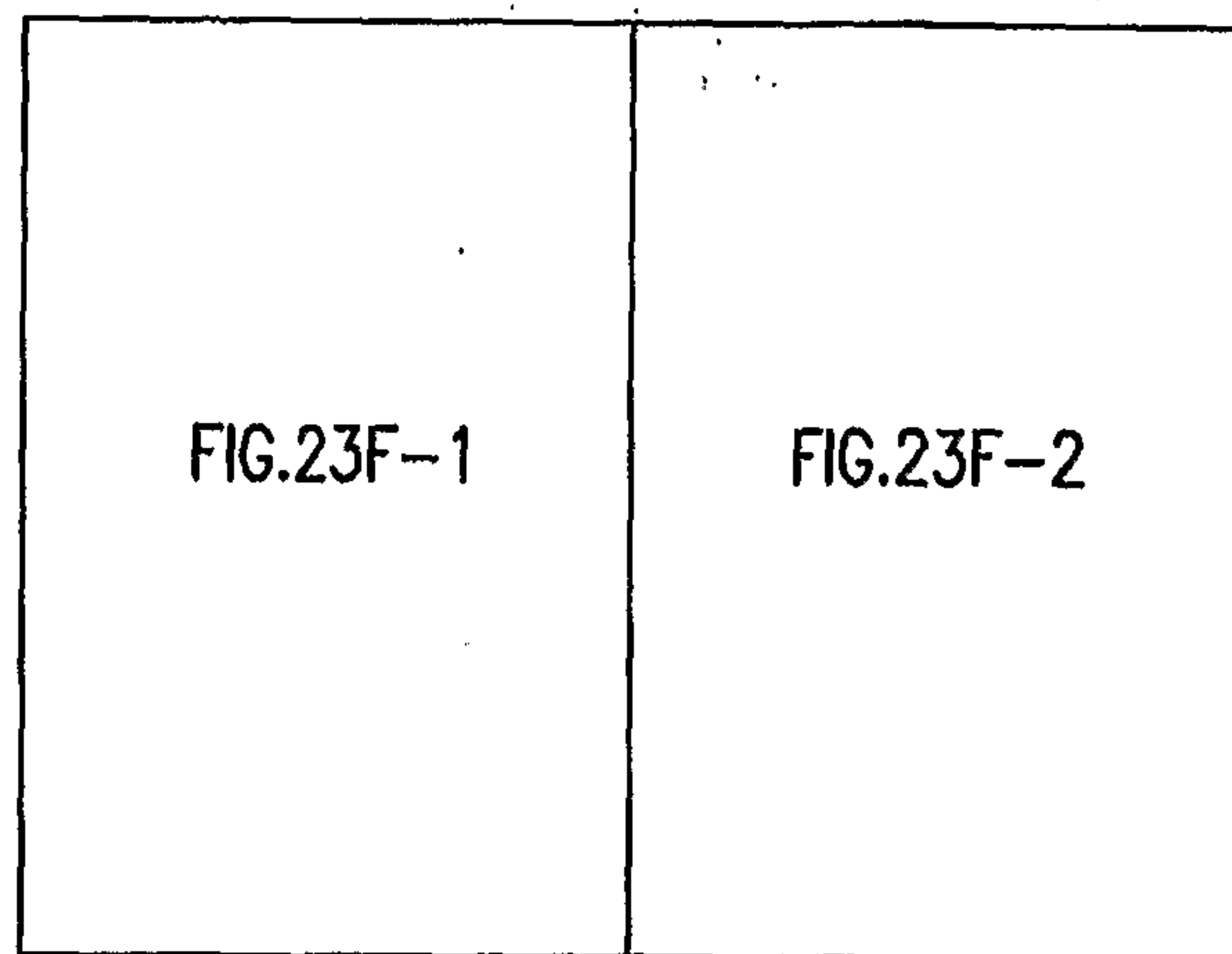


FIG.23F

152/227

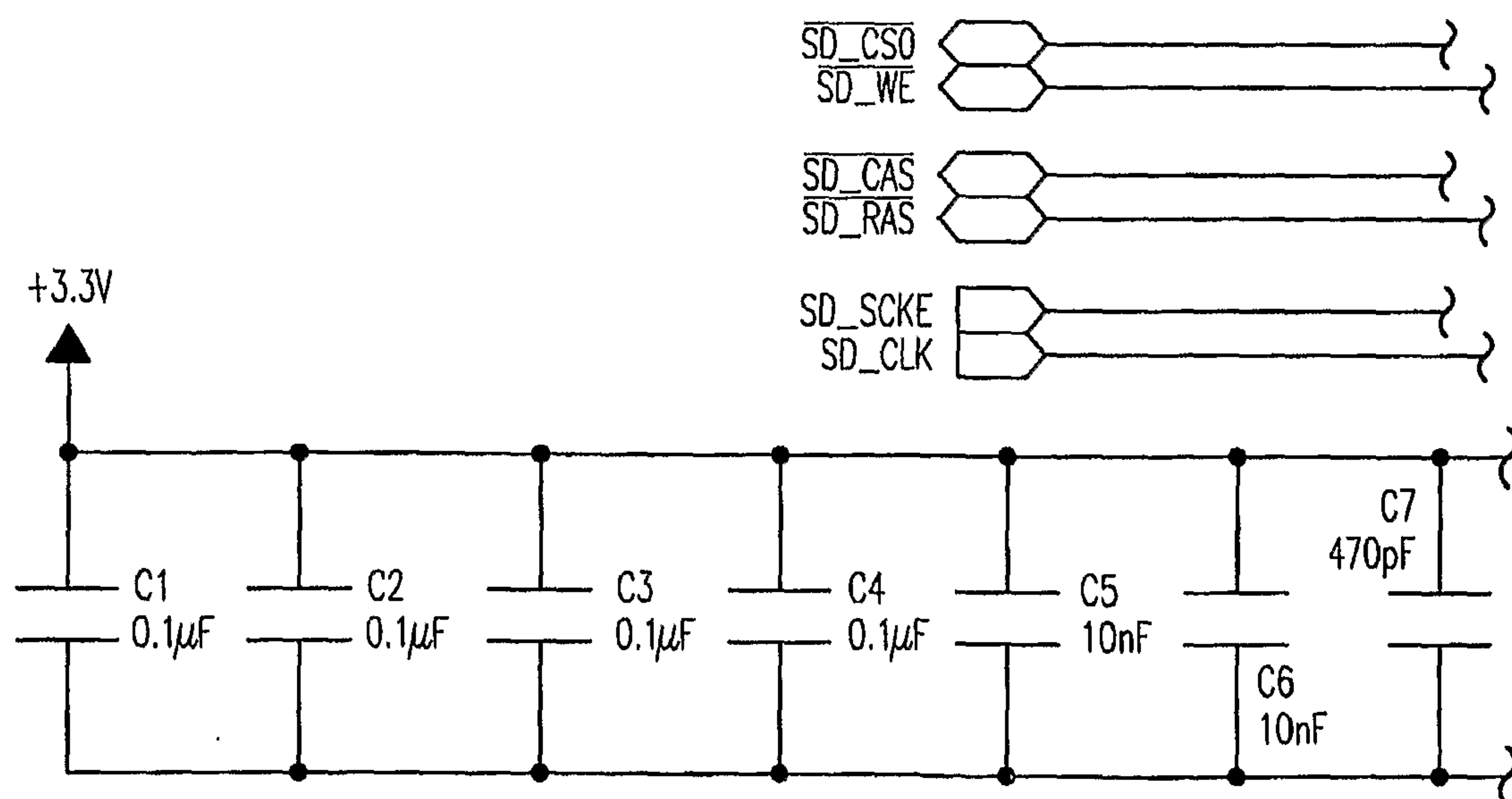
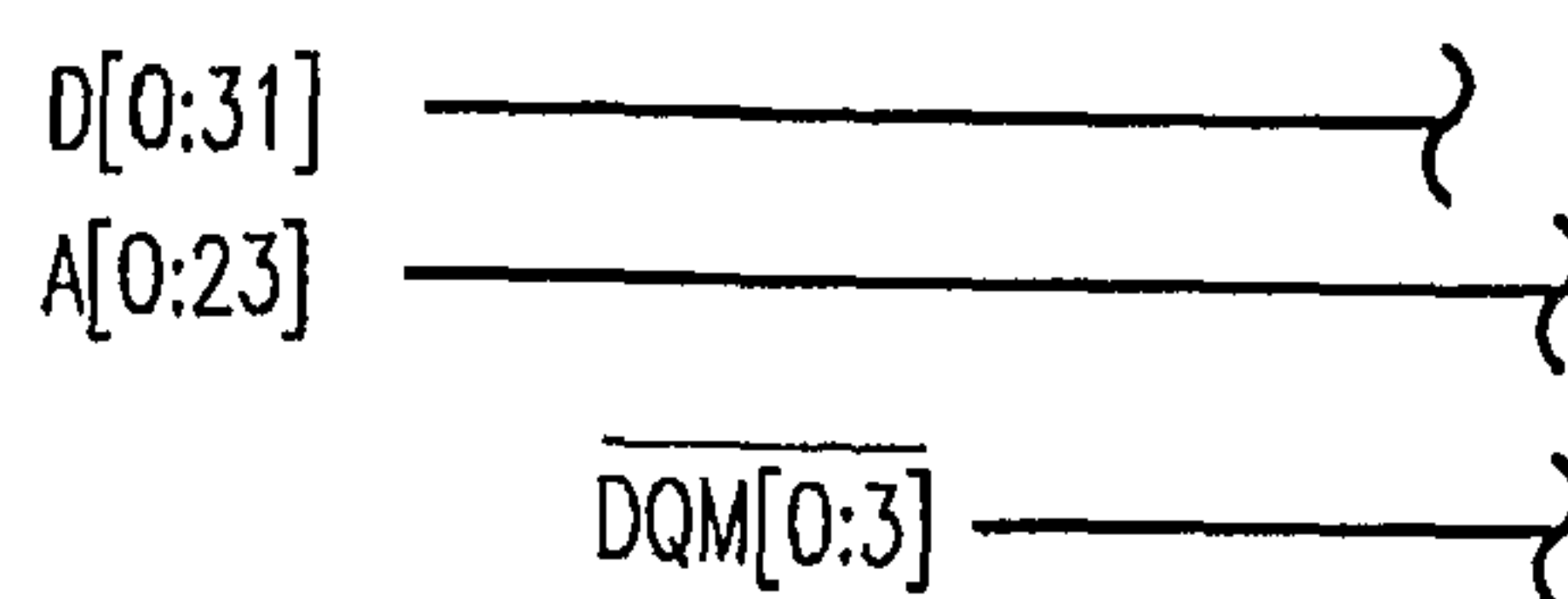
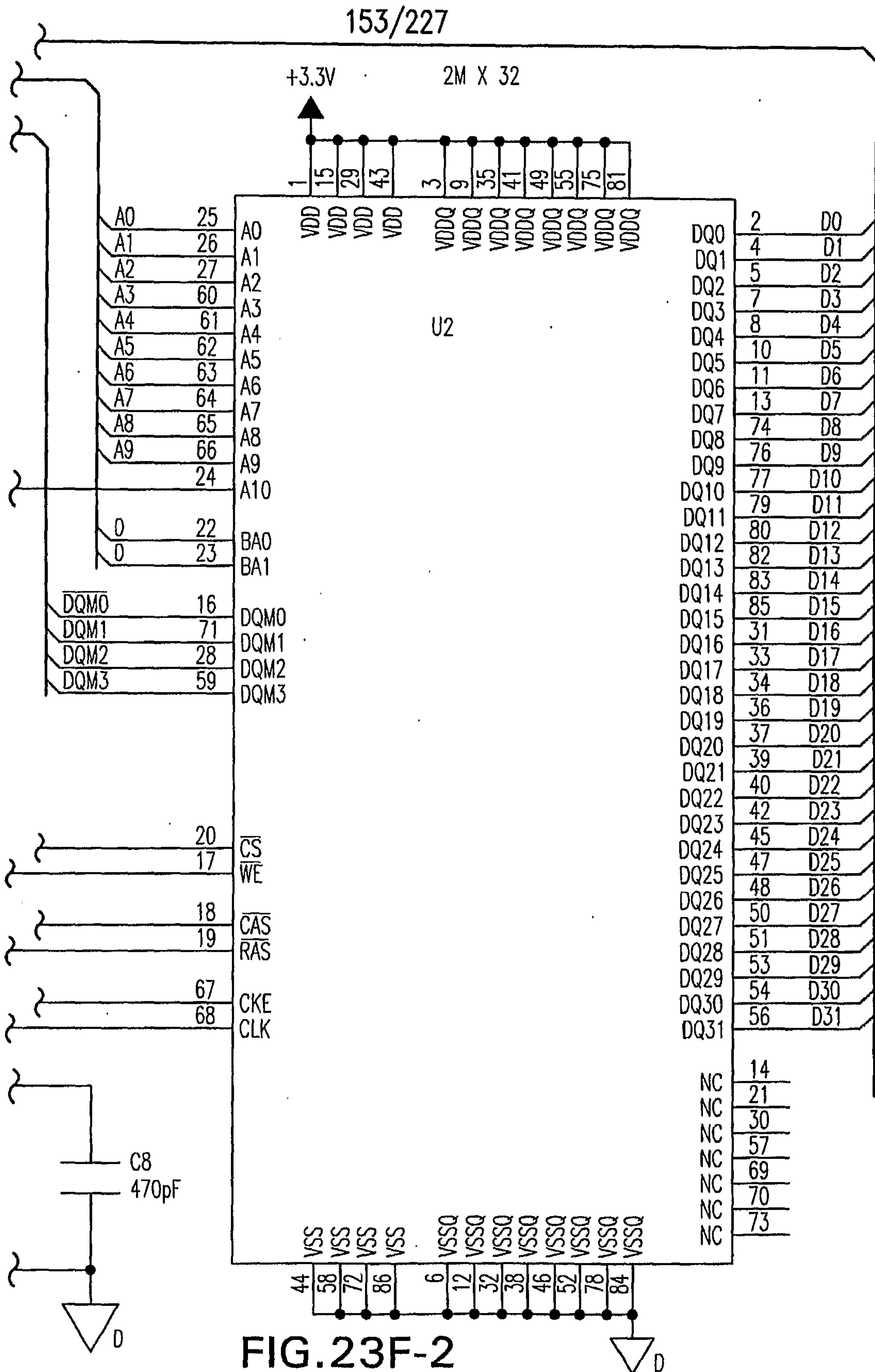
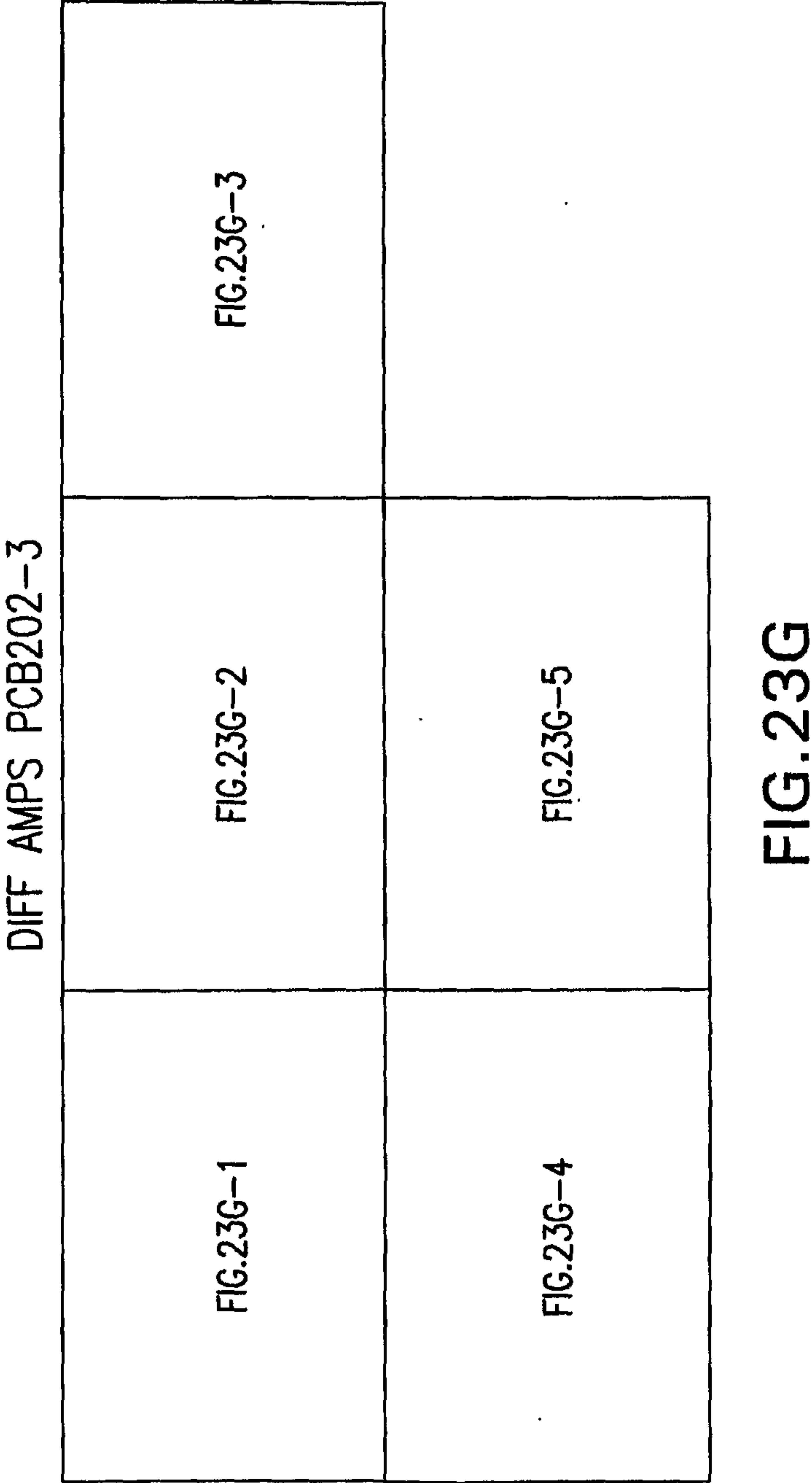


FIG.23F-1





155/227

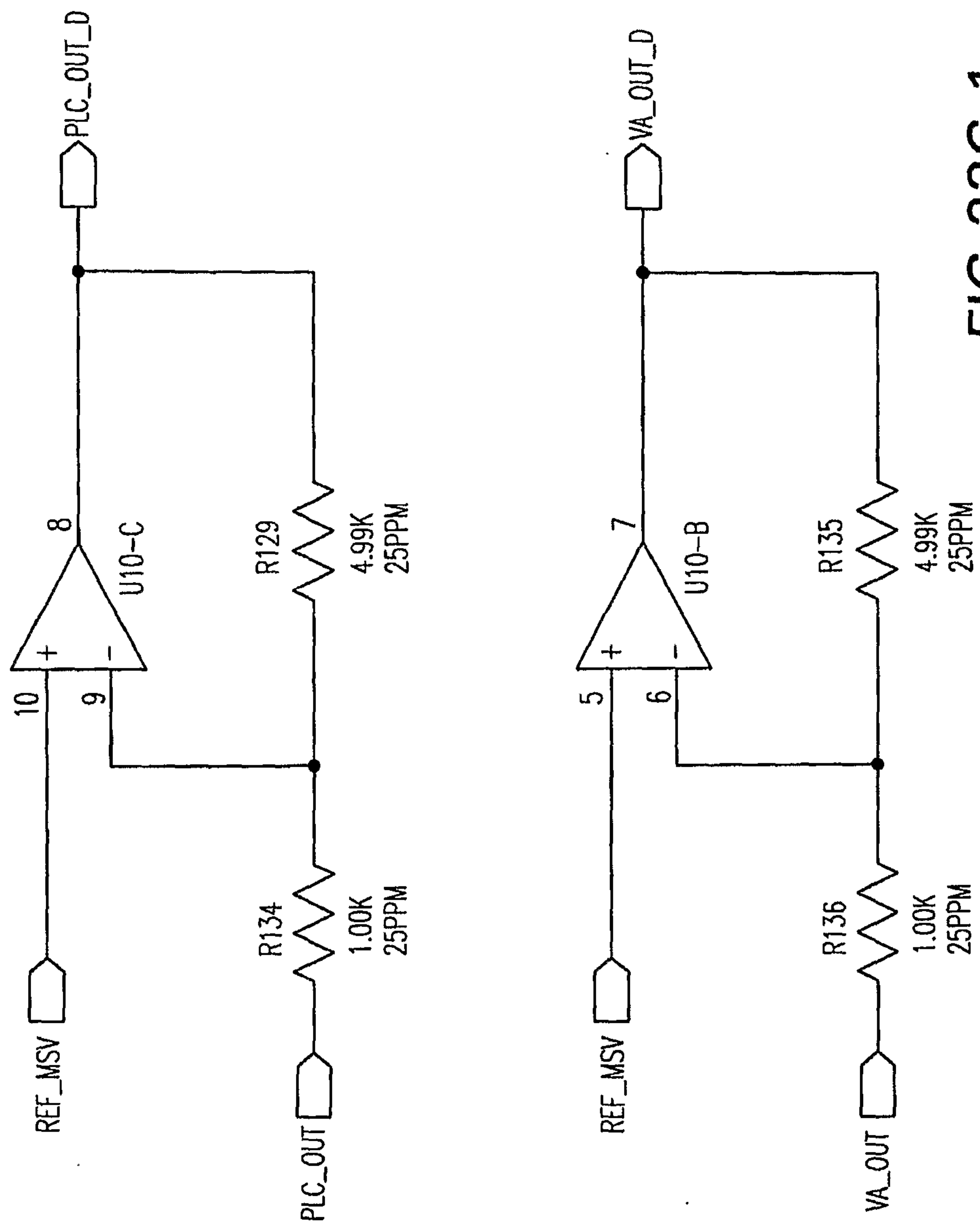


FIG. 23G-1

156/227

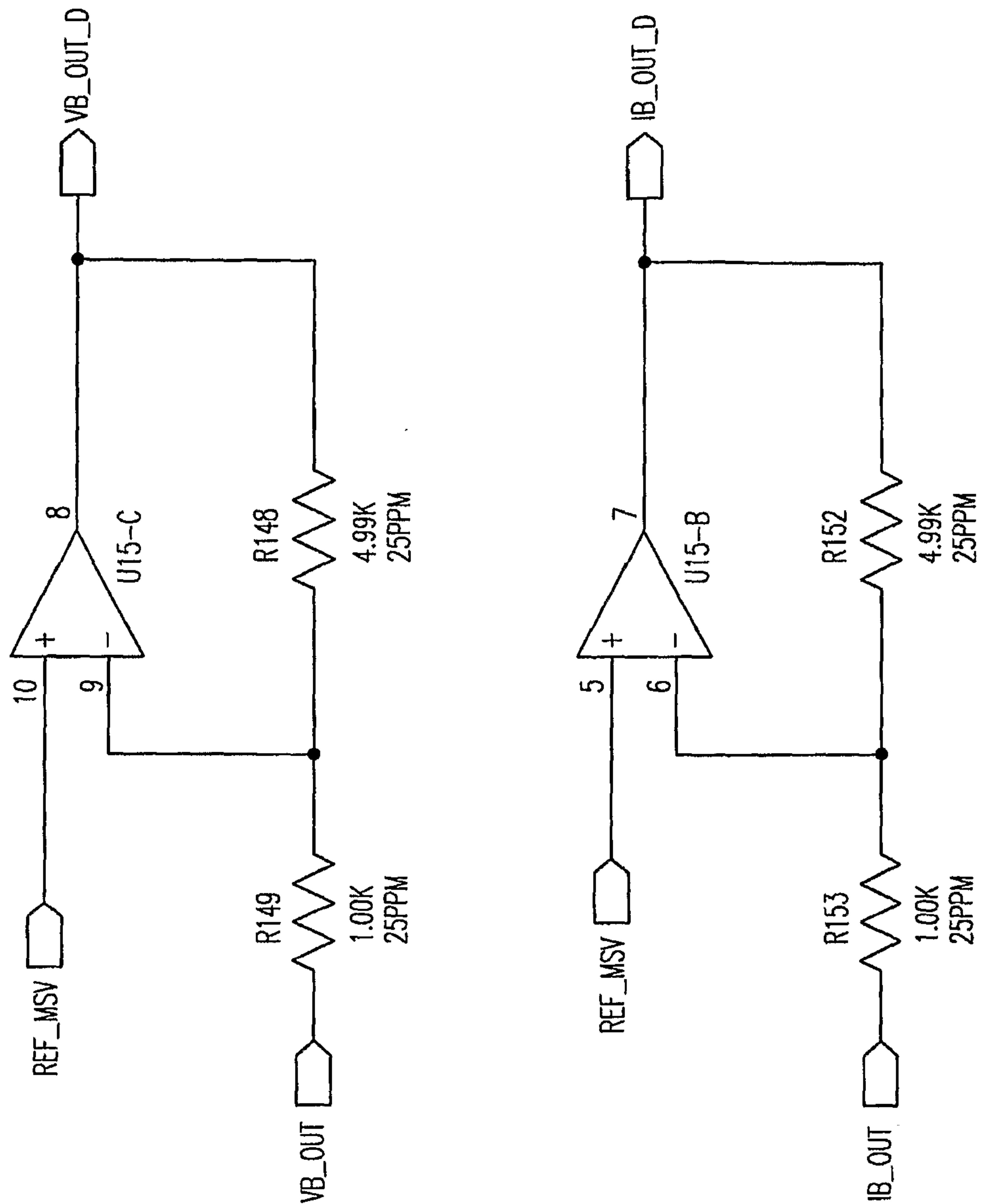
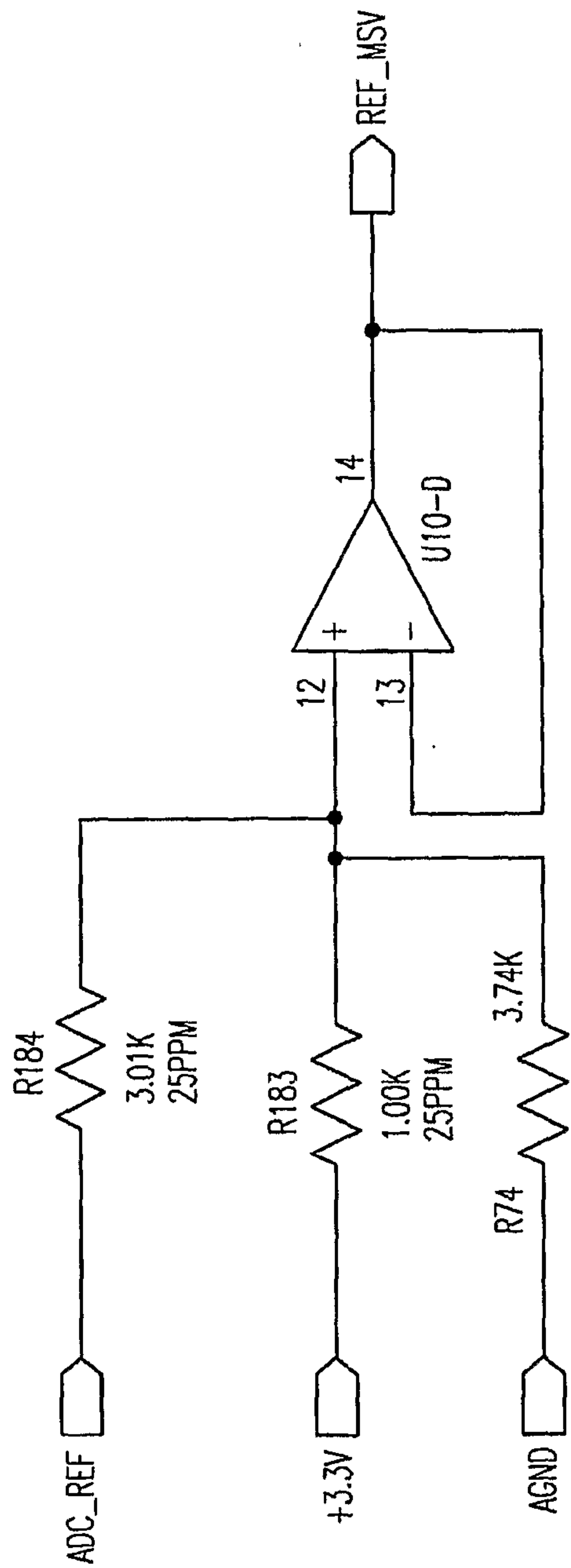


FIG. 23G-2



EXTERNAL REFERENCE

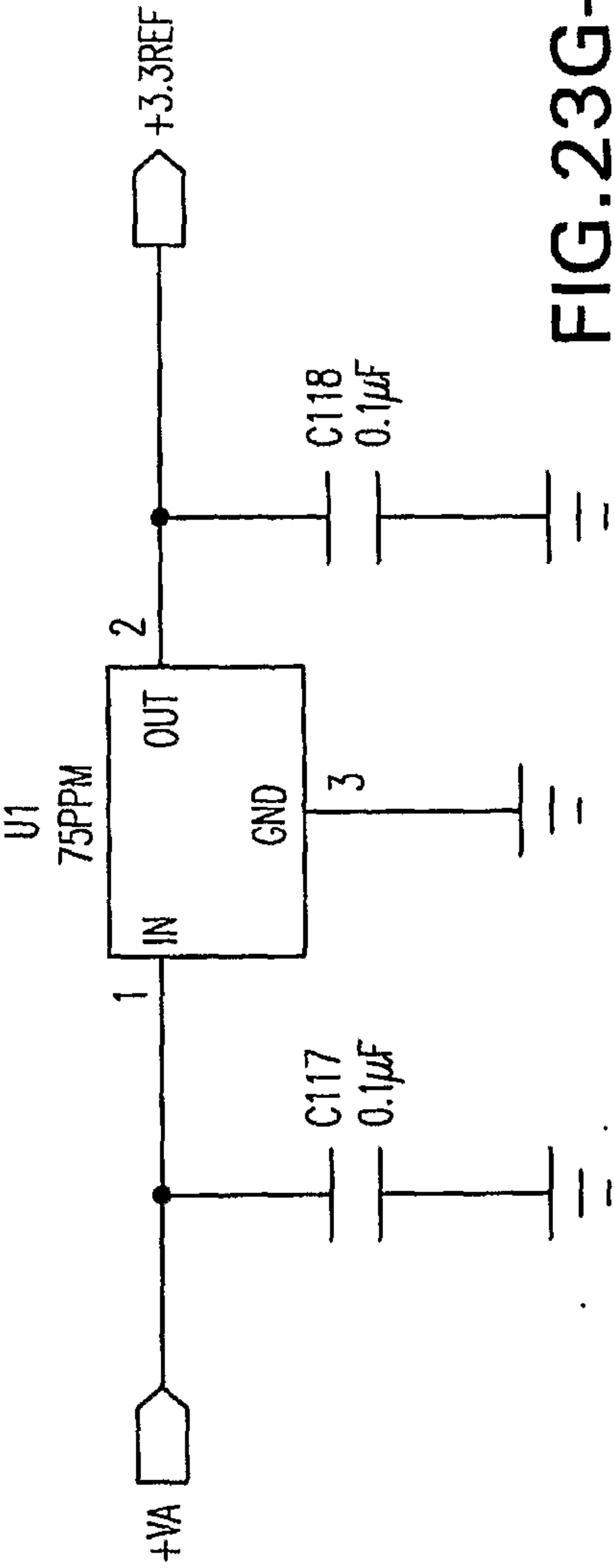


FIG.23G-3

158/227

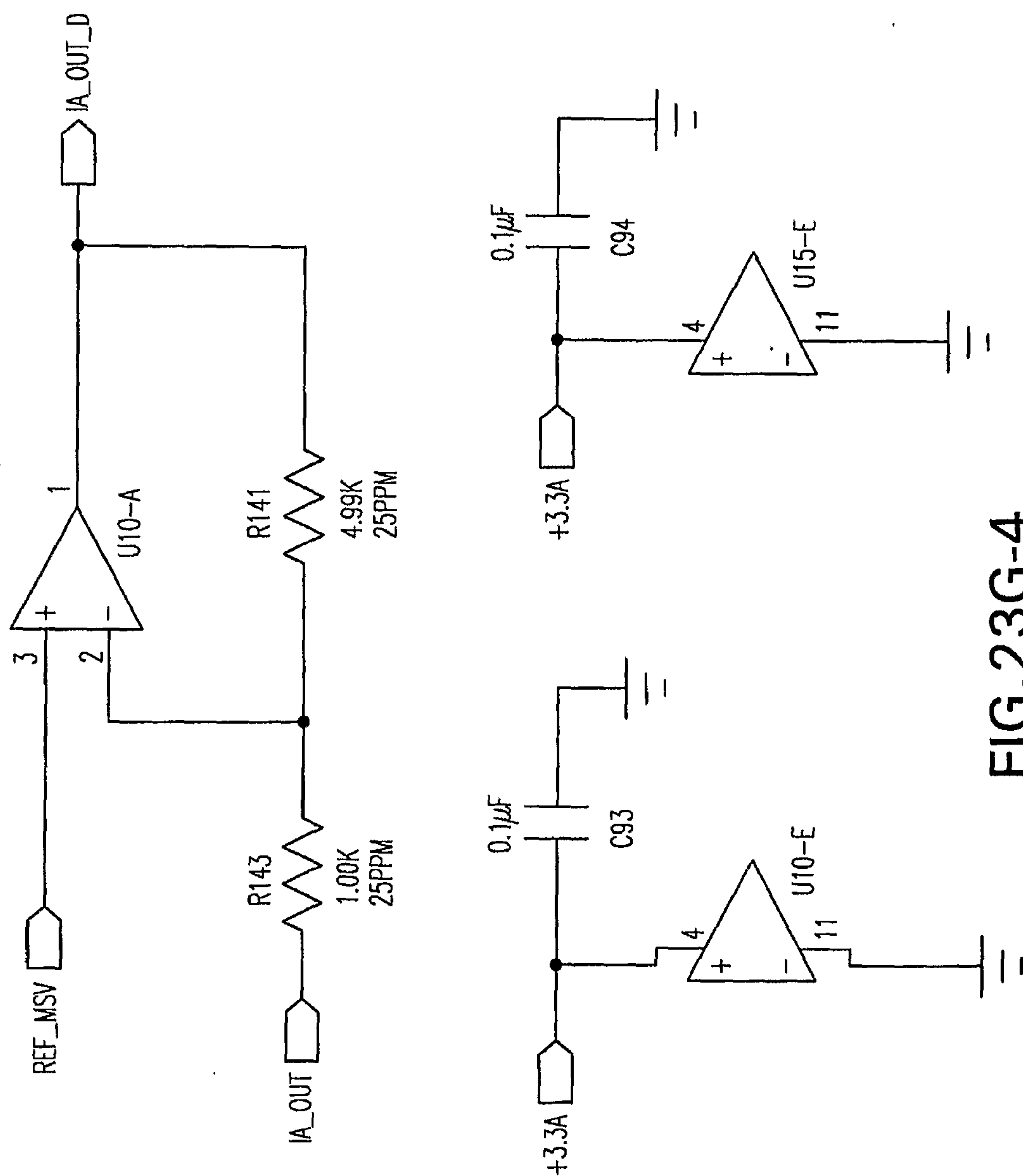


FIG. 23G-4

159/227

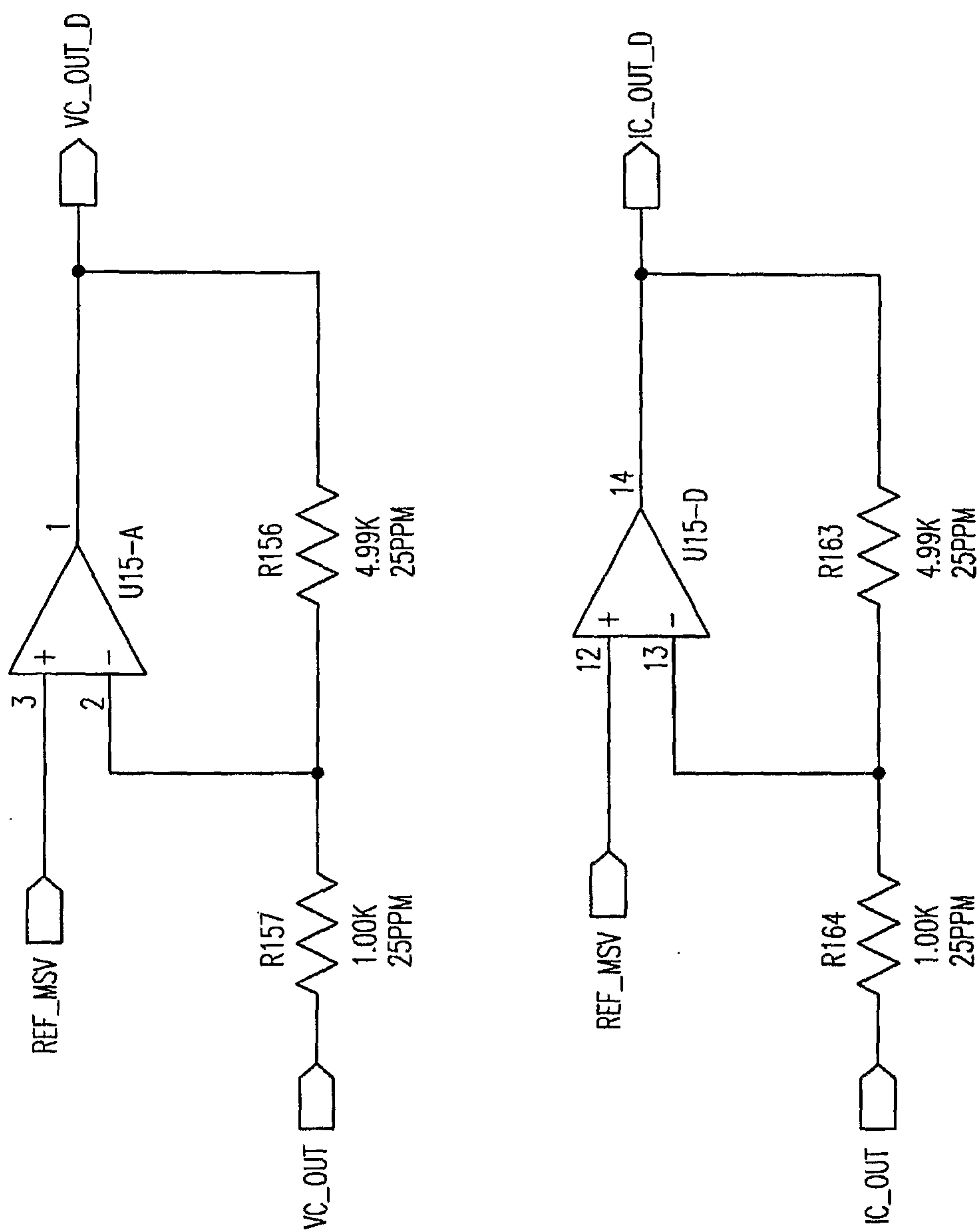


FIG. 23G-5

METER I PCB202-3

FIG.23H-1	FIG.23H-2	FIG.23H-3
FIG.23H-4	FIG.23H-5	FIG.23H-6
FIG.23H-7	FIG.23H-8	FIG.23H-9

FIG. 23H

161/227

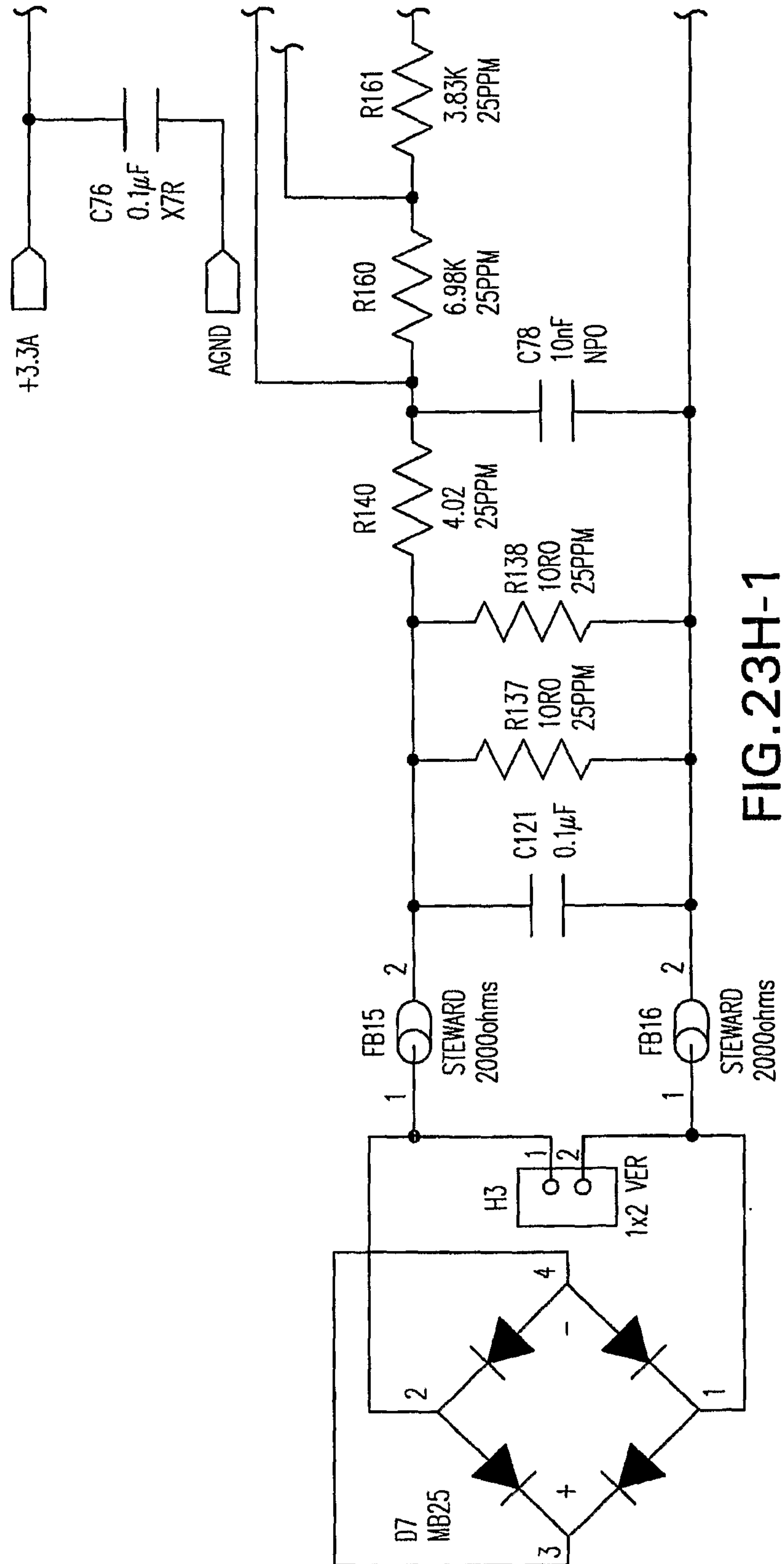
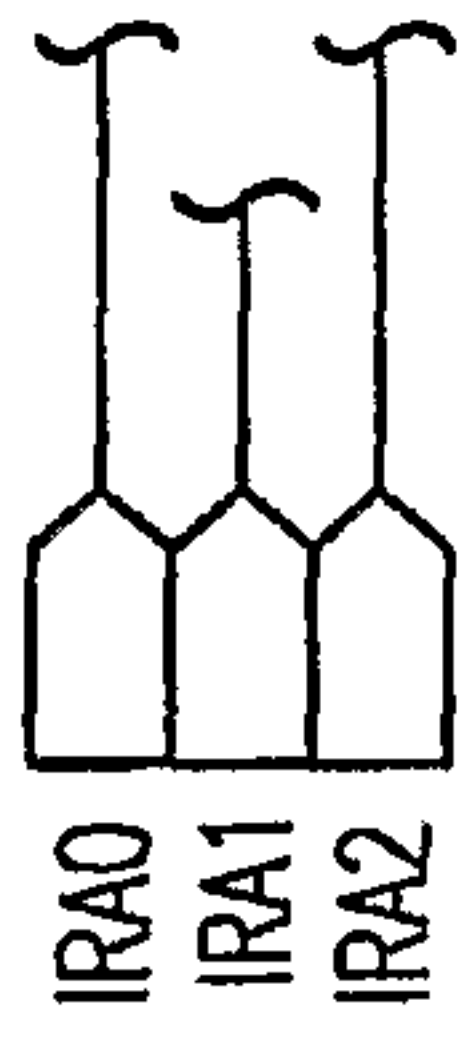


FIG.23H-1

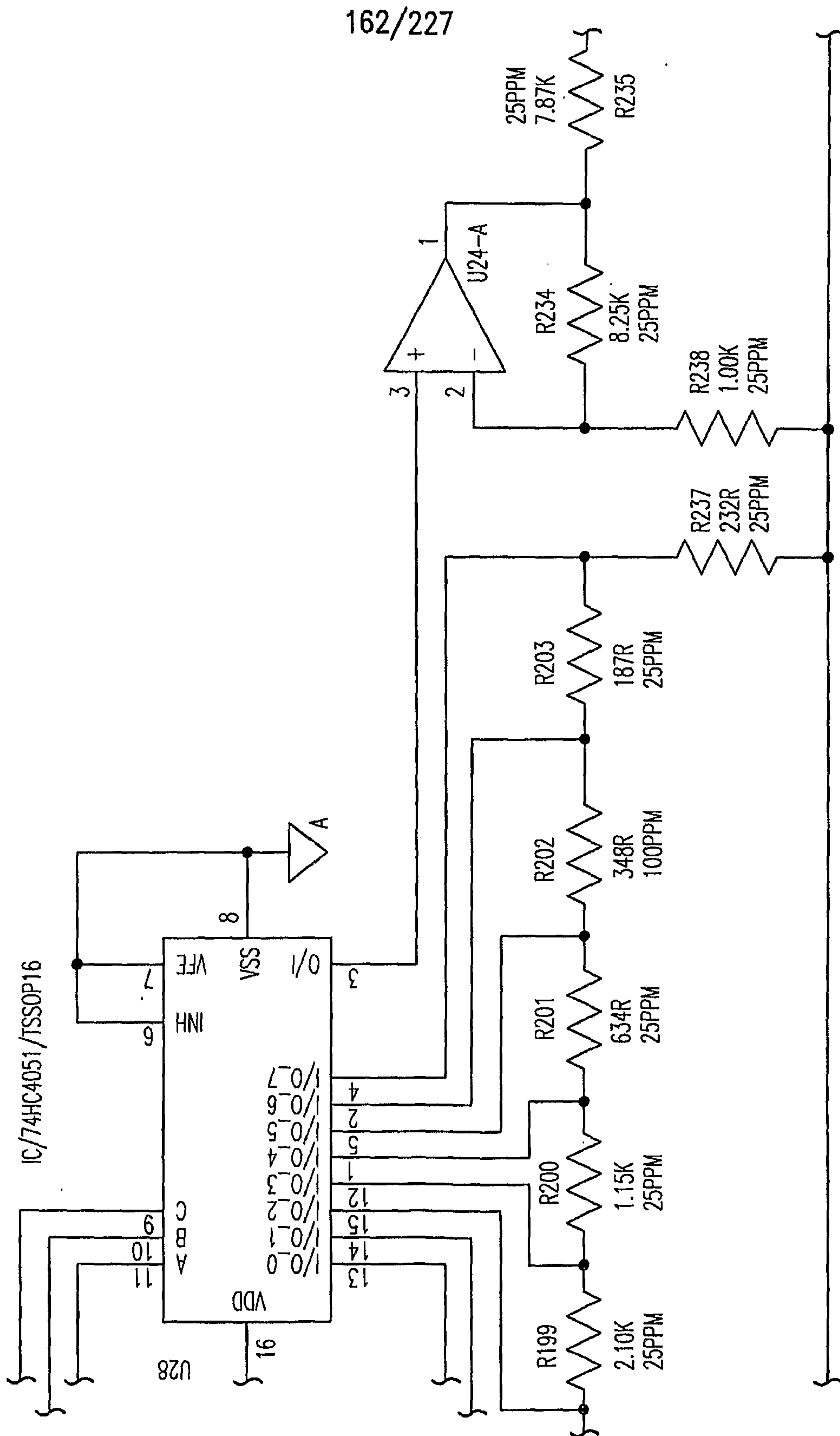


FIG. 23H-2

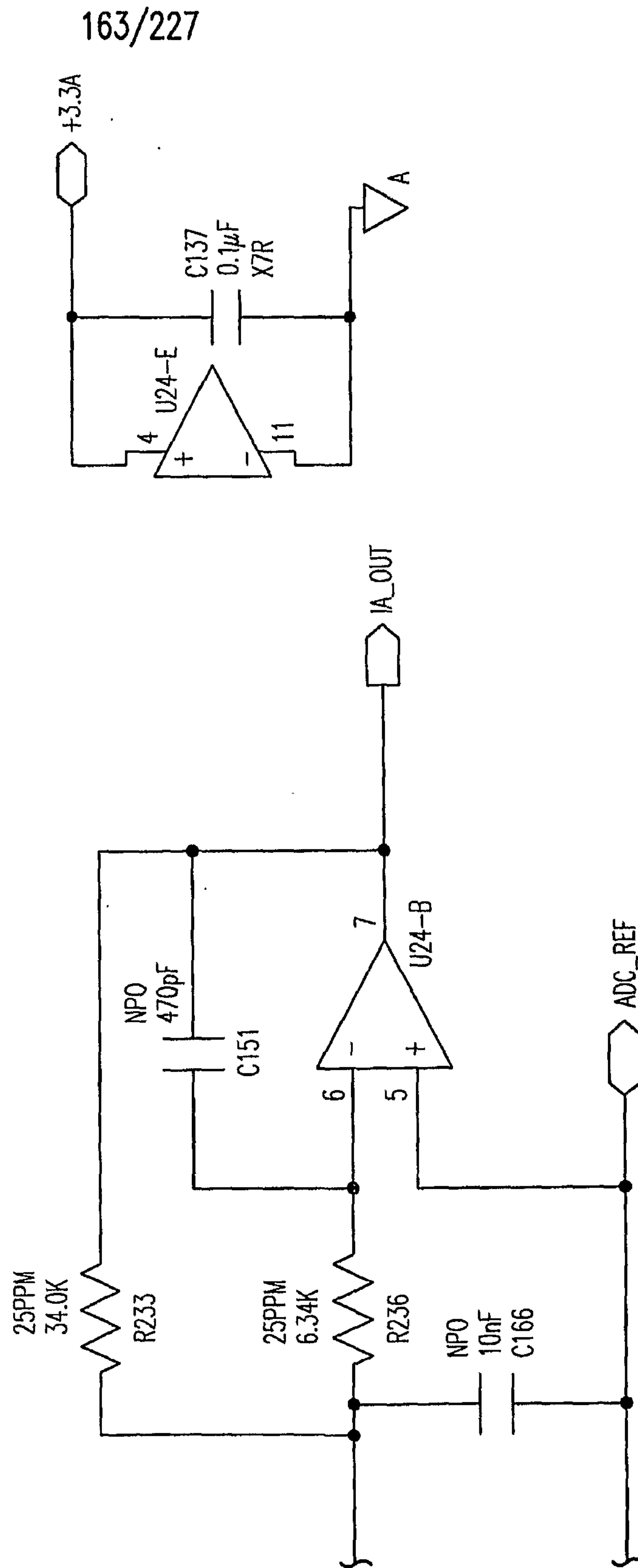
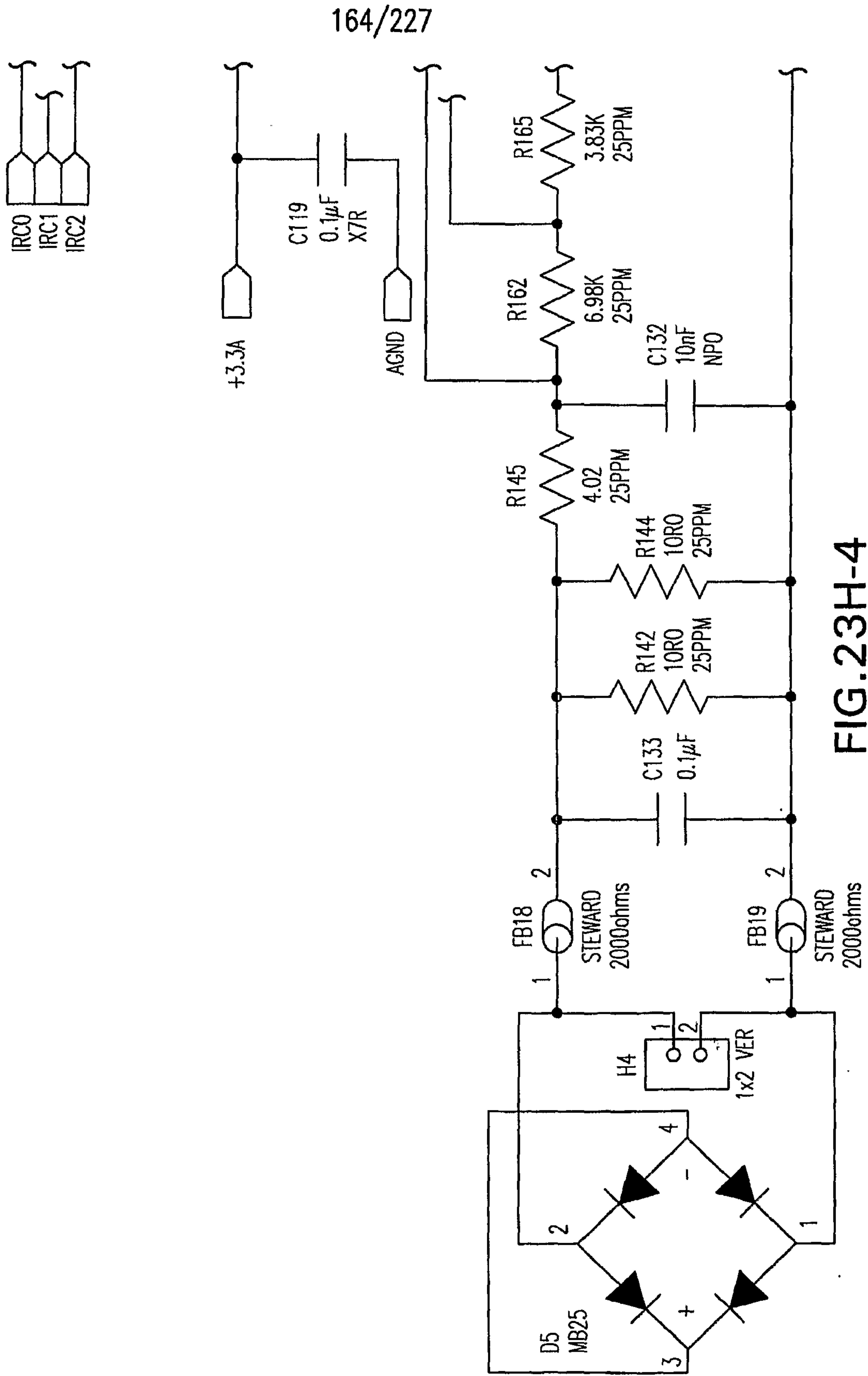


FIG. 23H-3



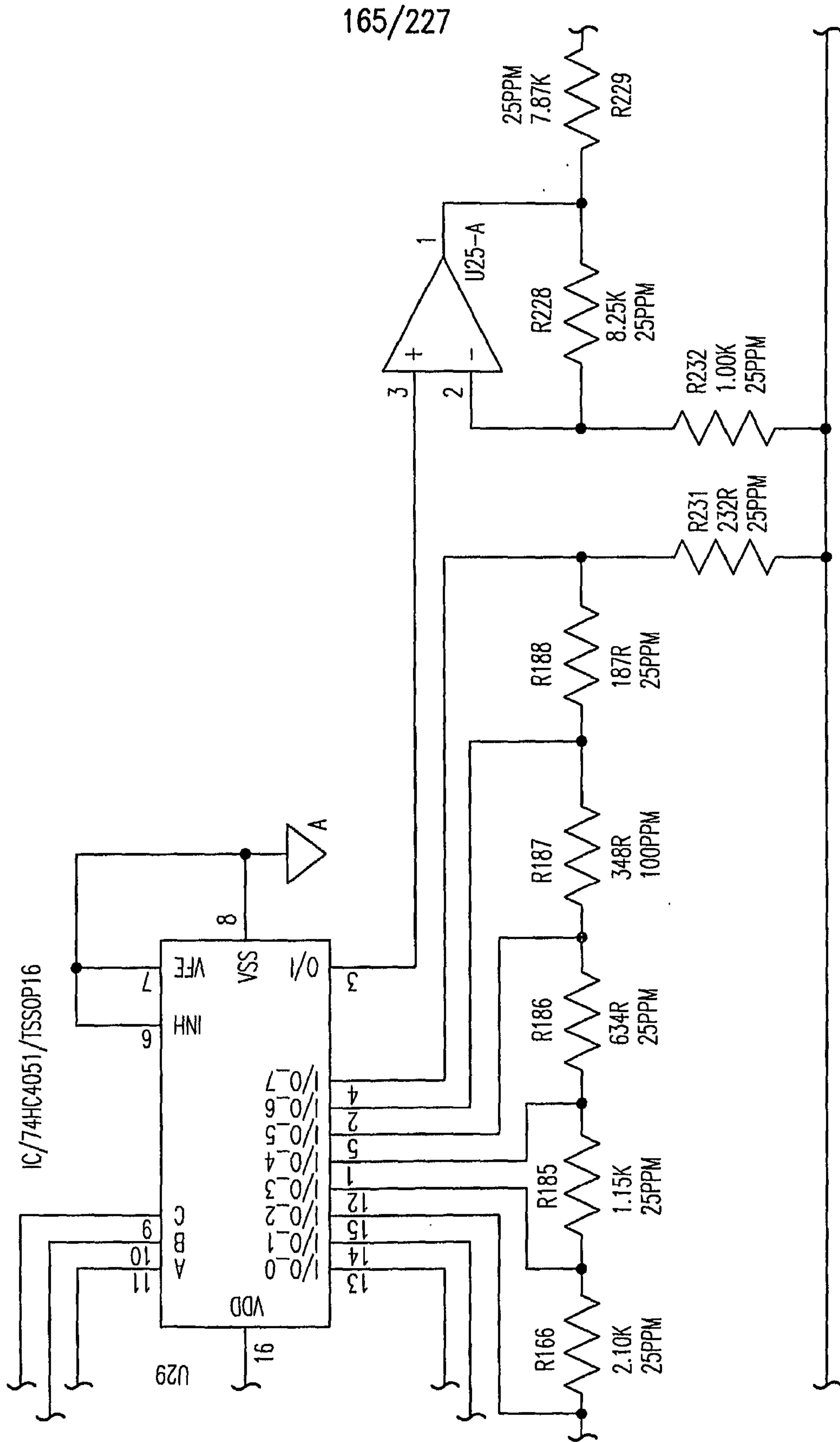


FIG. 23H-5

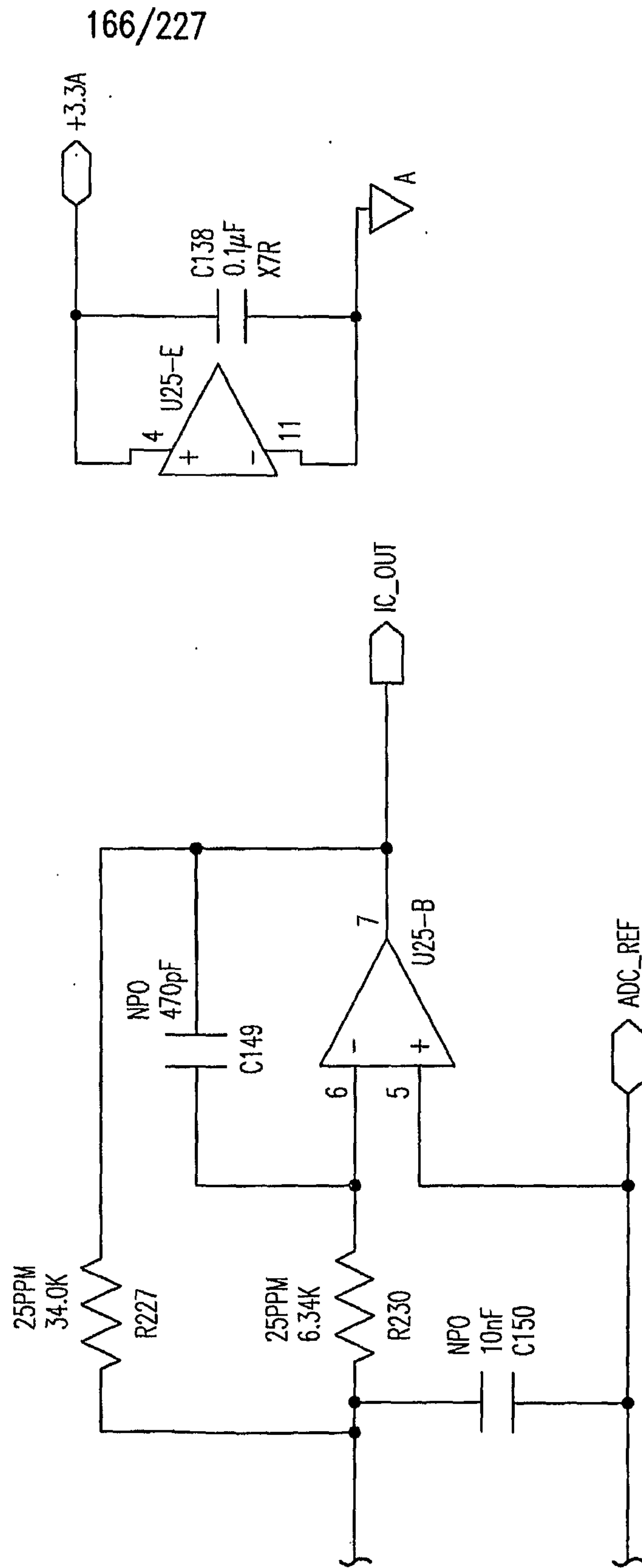
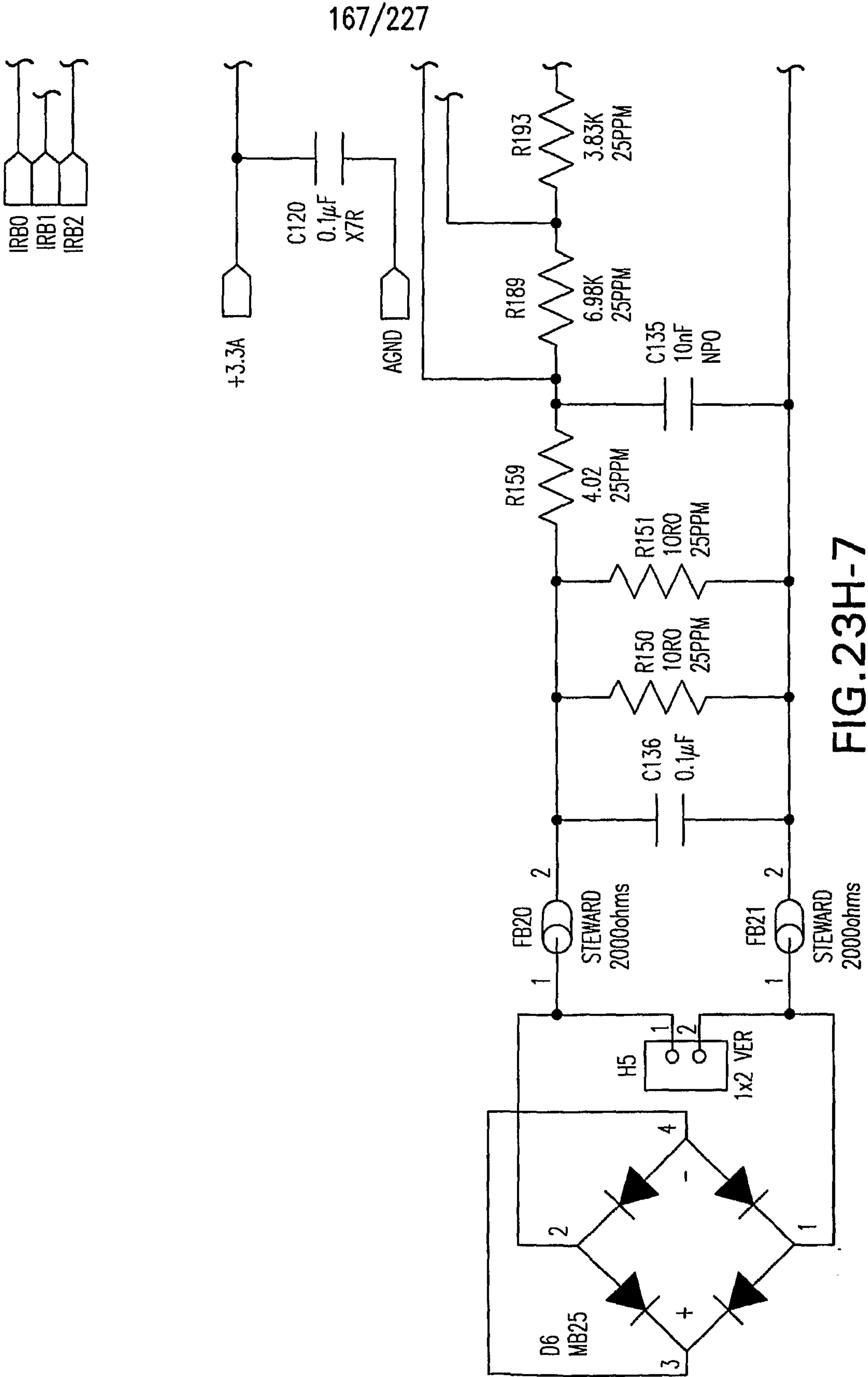


FIG. 23H-6



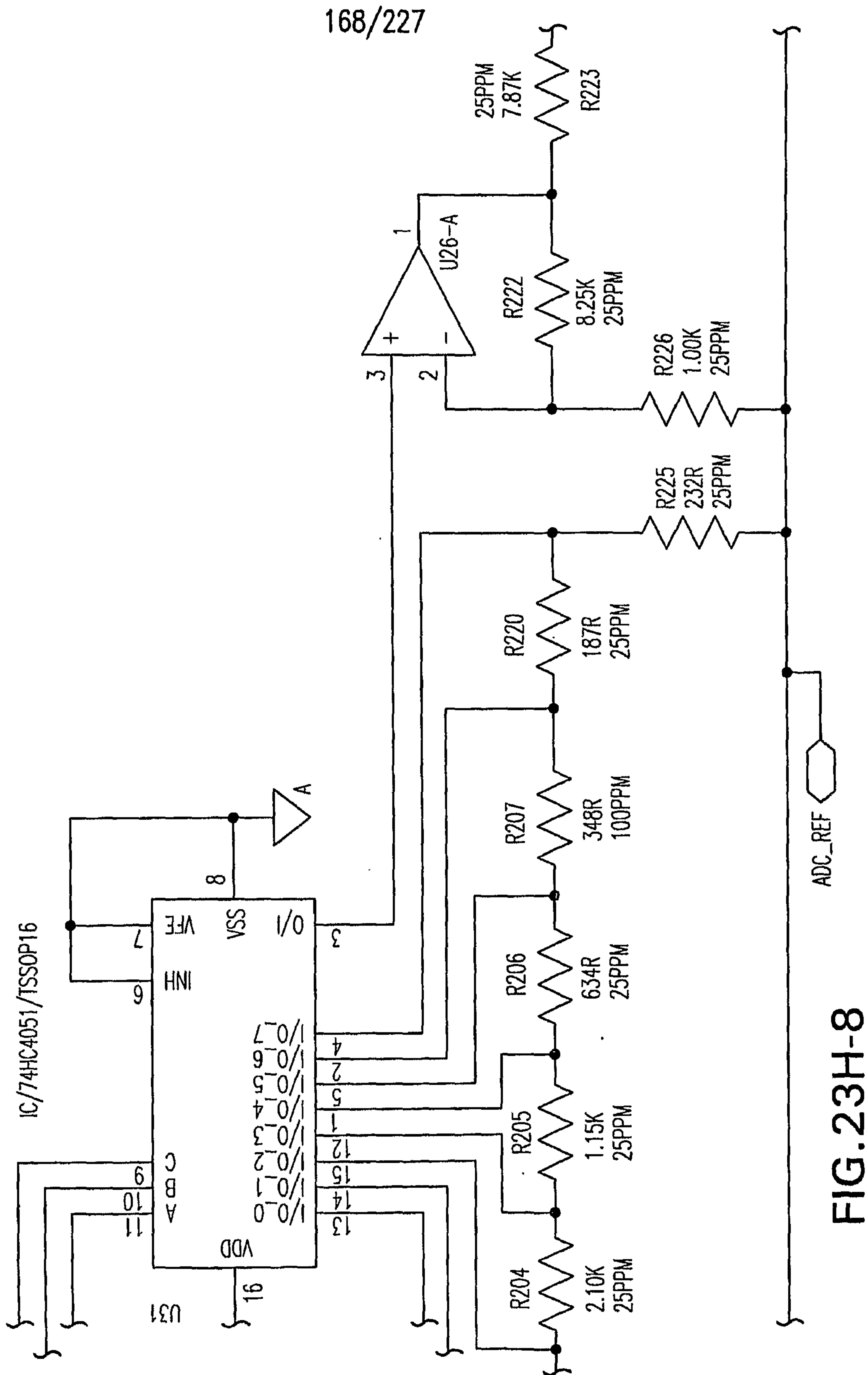


FIG. 23H-8

169/227

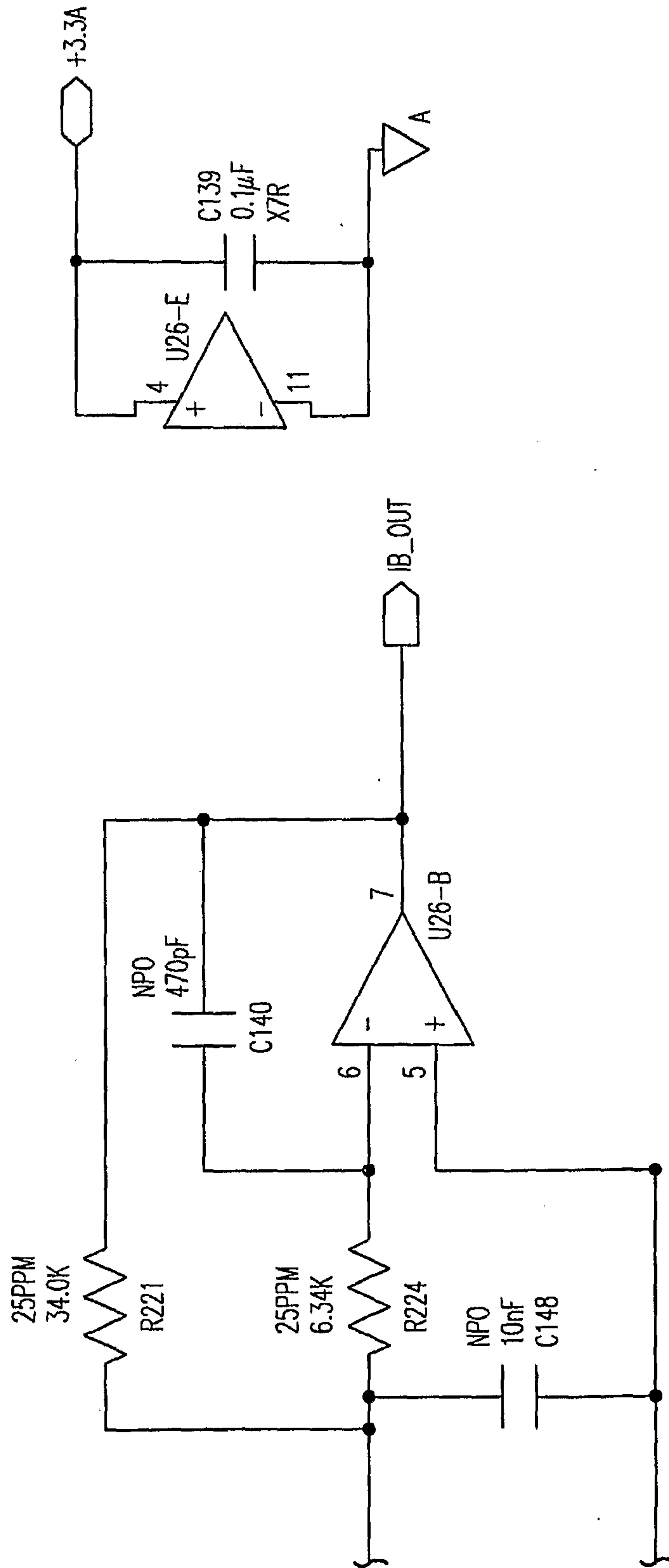
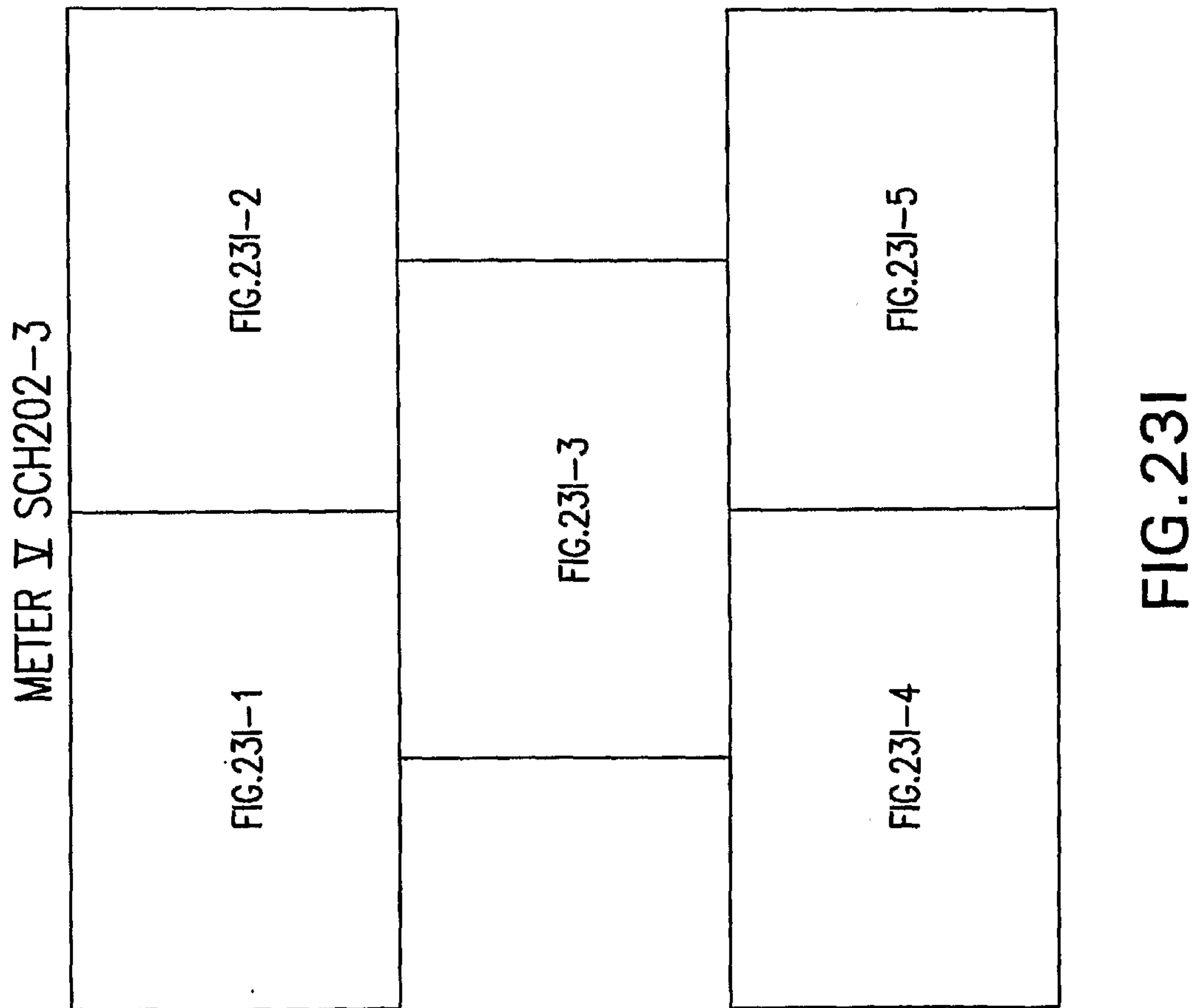


FIG. 23H-9

170/227



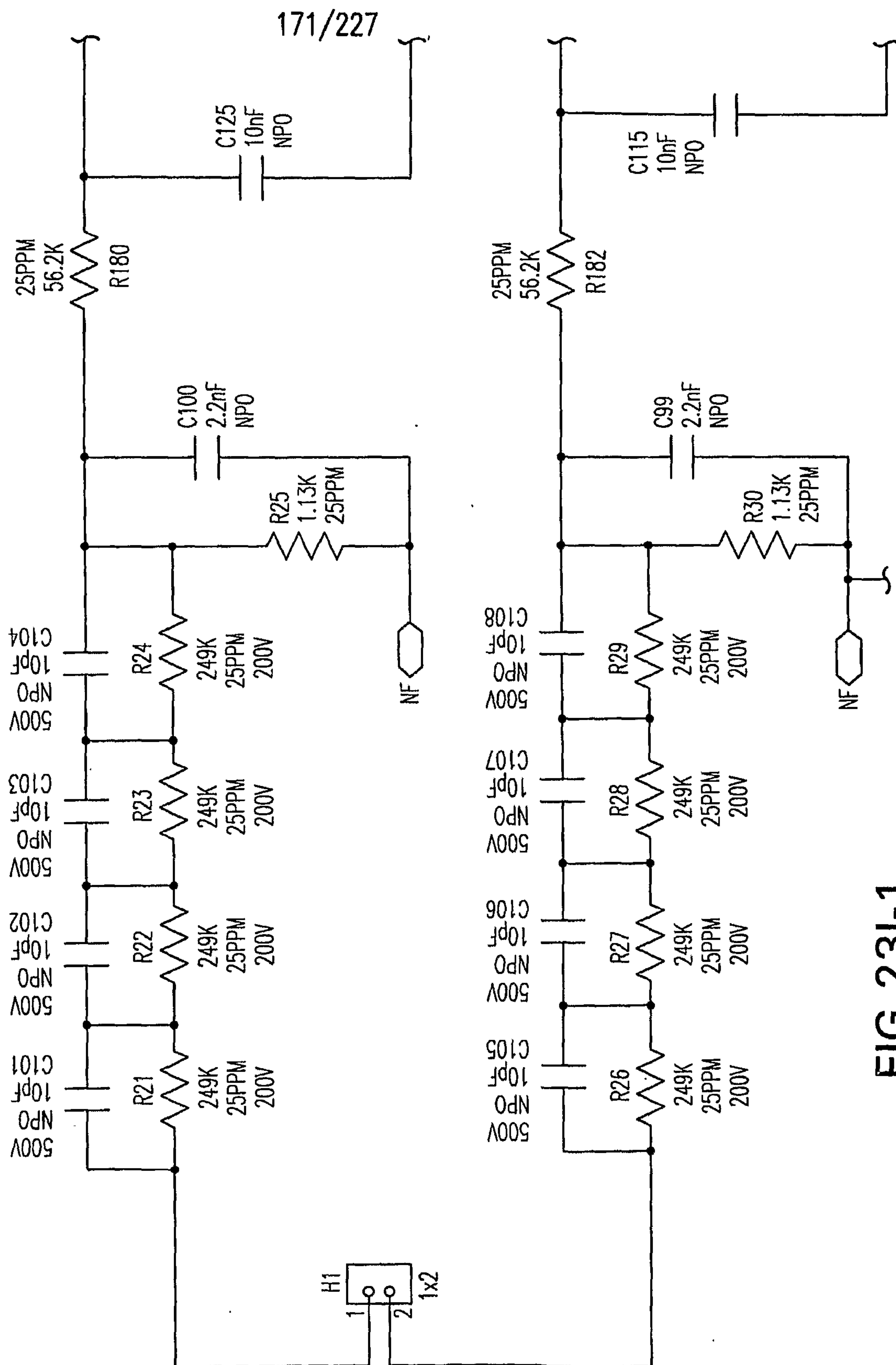


FIG. 23I-1

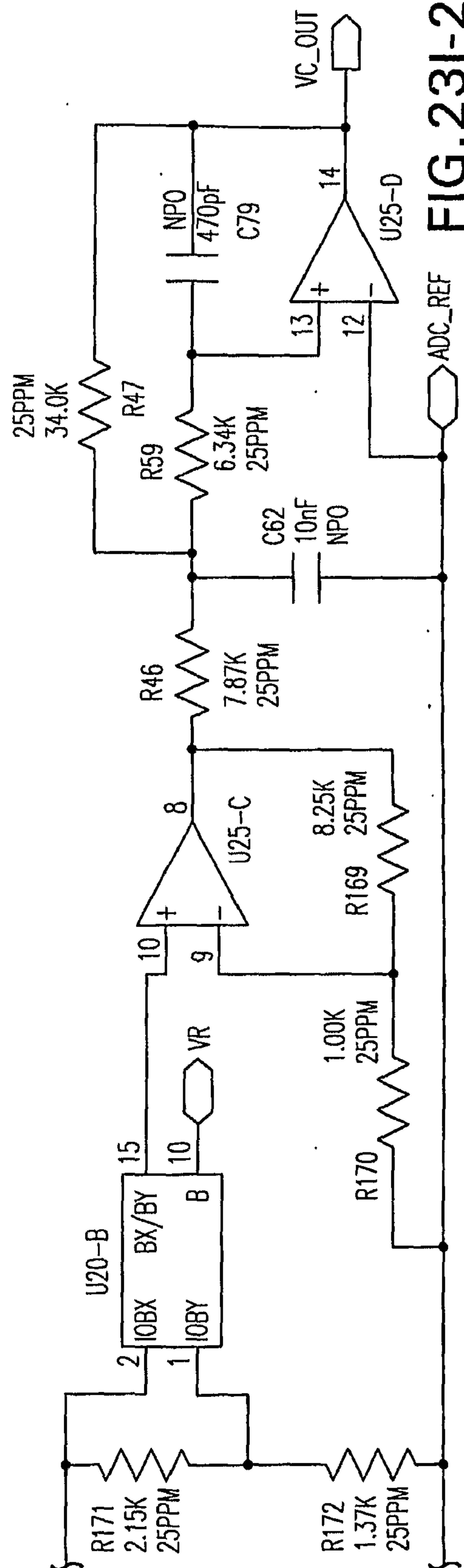
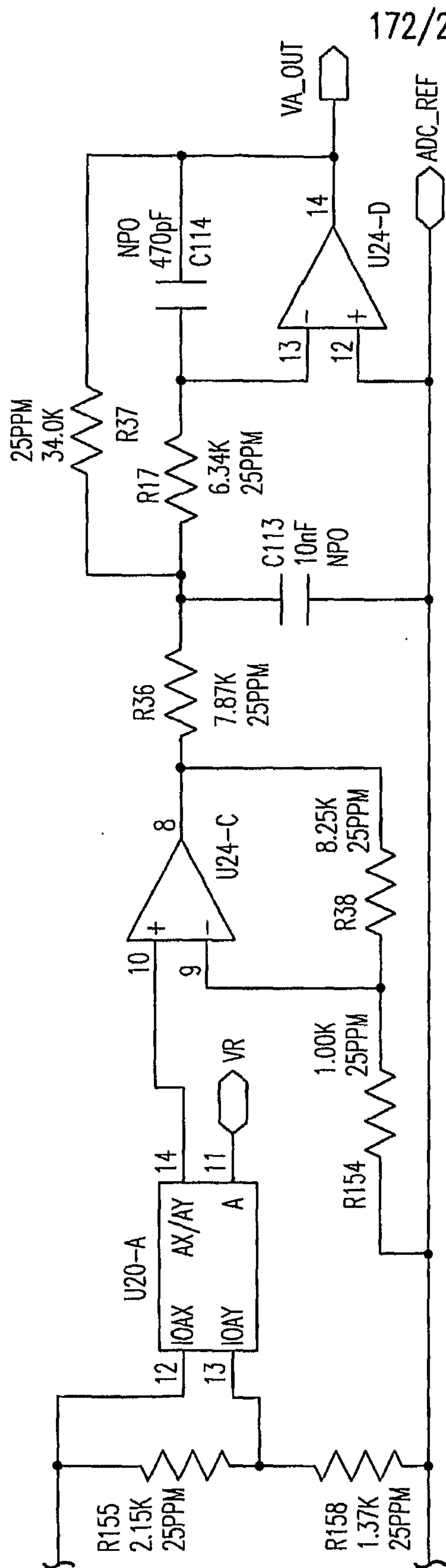


FIG. 231-2

173/227

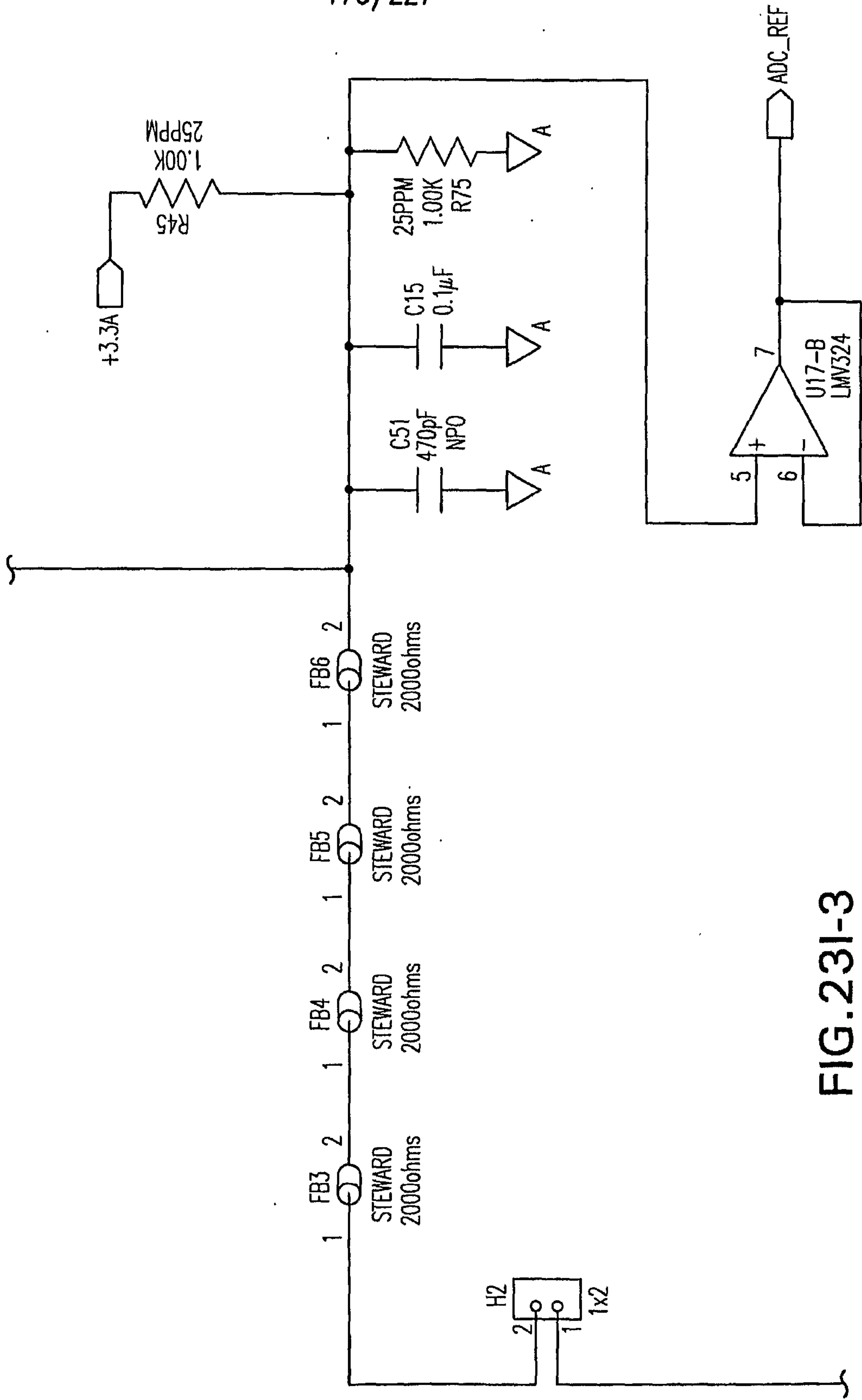


FIG.231-3

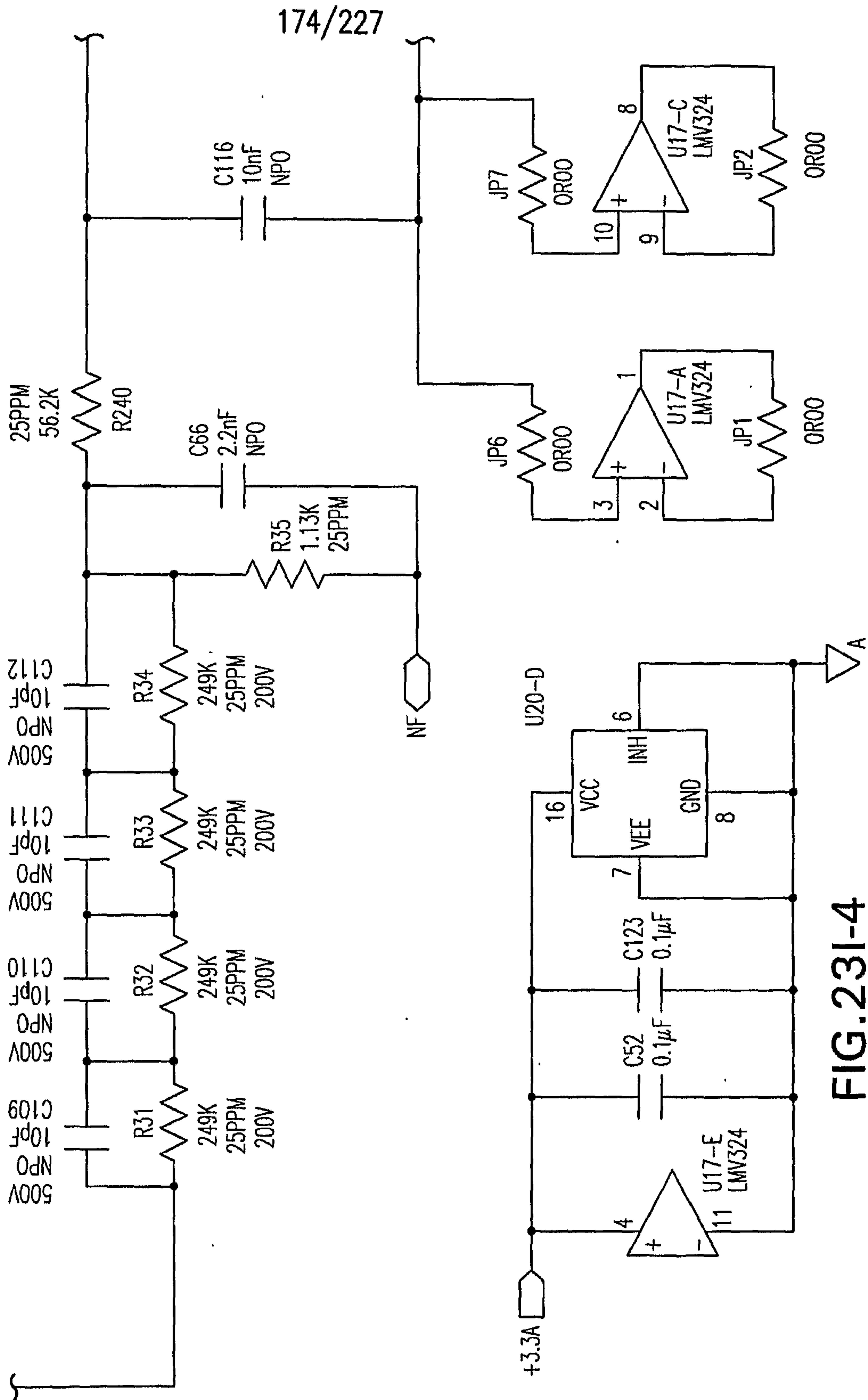


FIG. 231-4

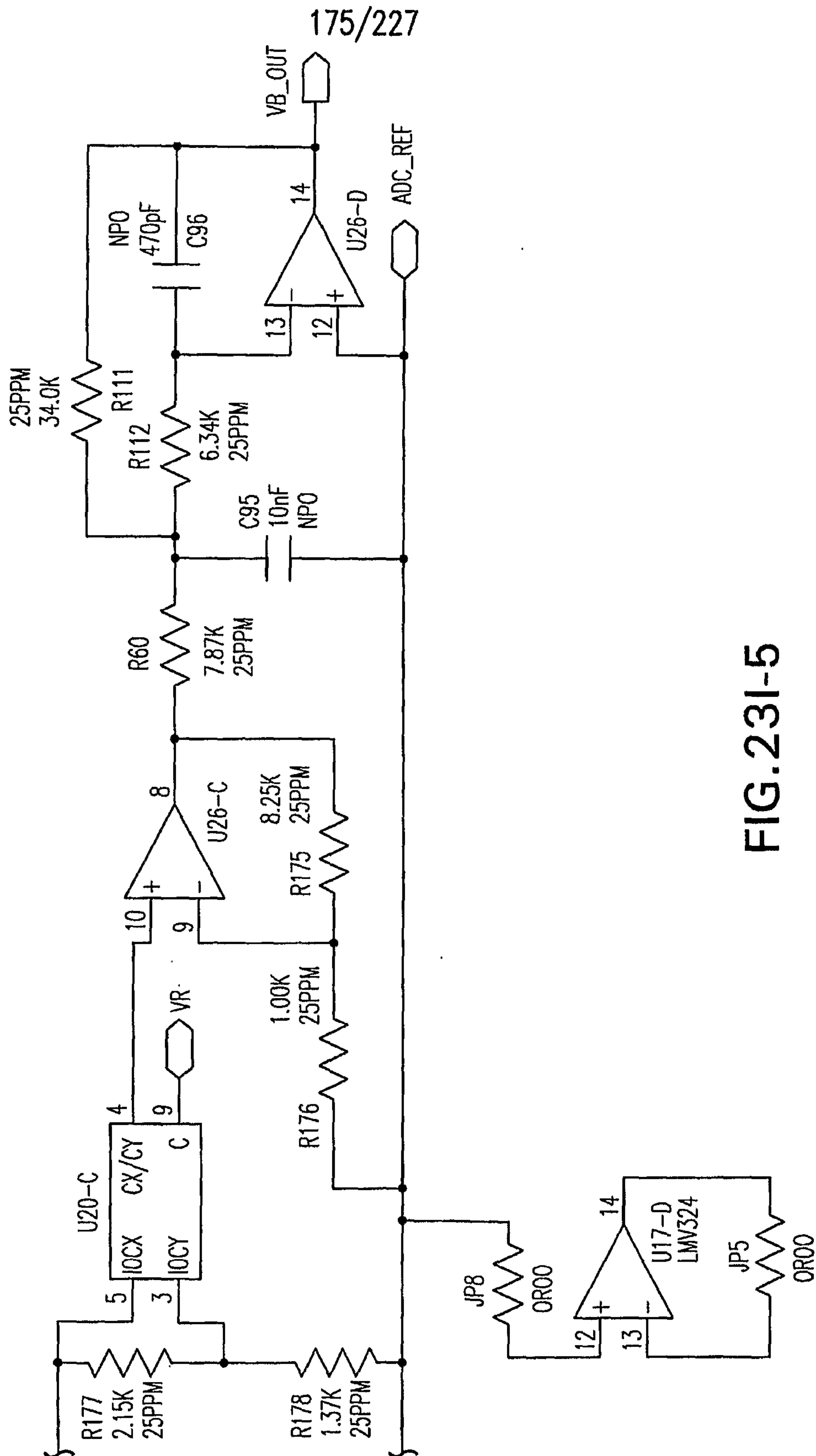


FIG. 231-5

176/227

PLC RCV PCB202-3

FIG.23J-1	FIG.23J-2	FIG.23J-3
FIG.23J-4	FIG.23J-5	FIG.23J-6

FIG.23J

177/227

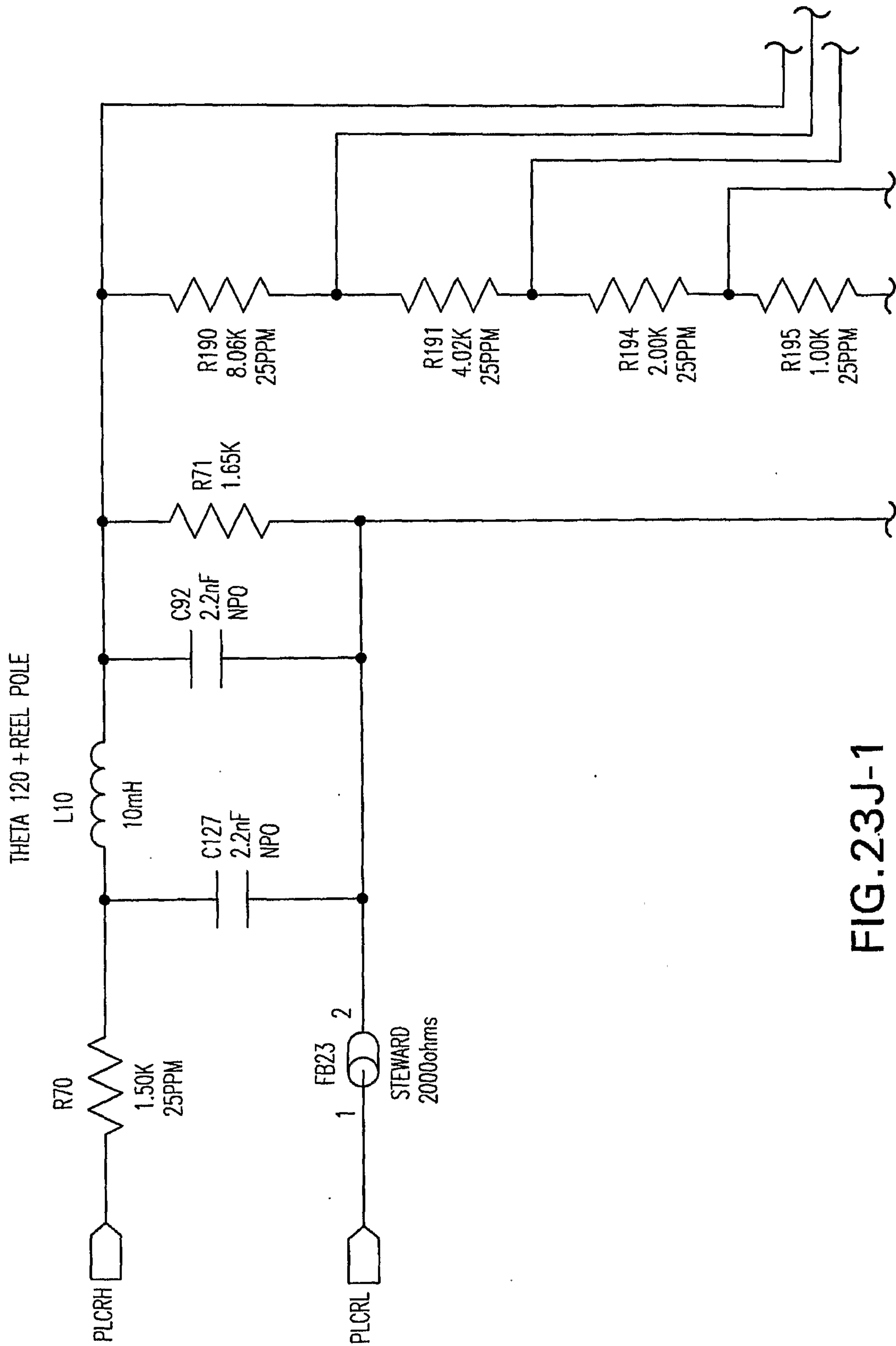
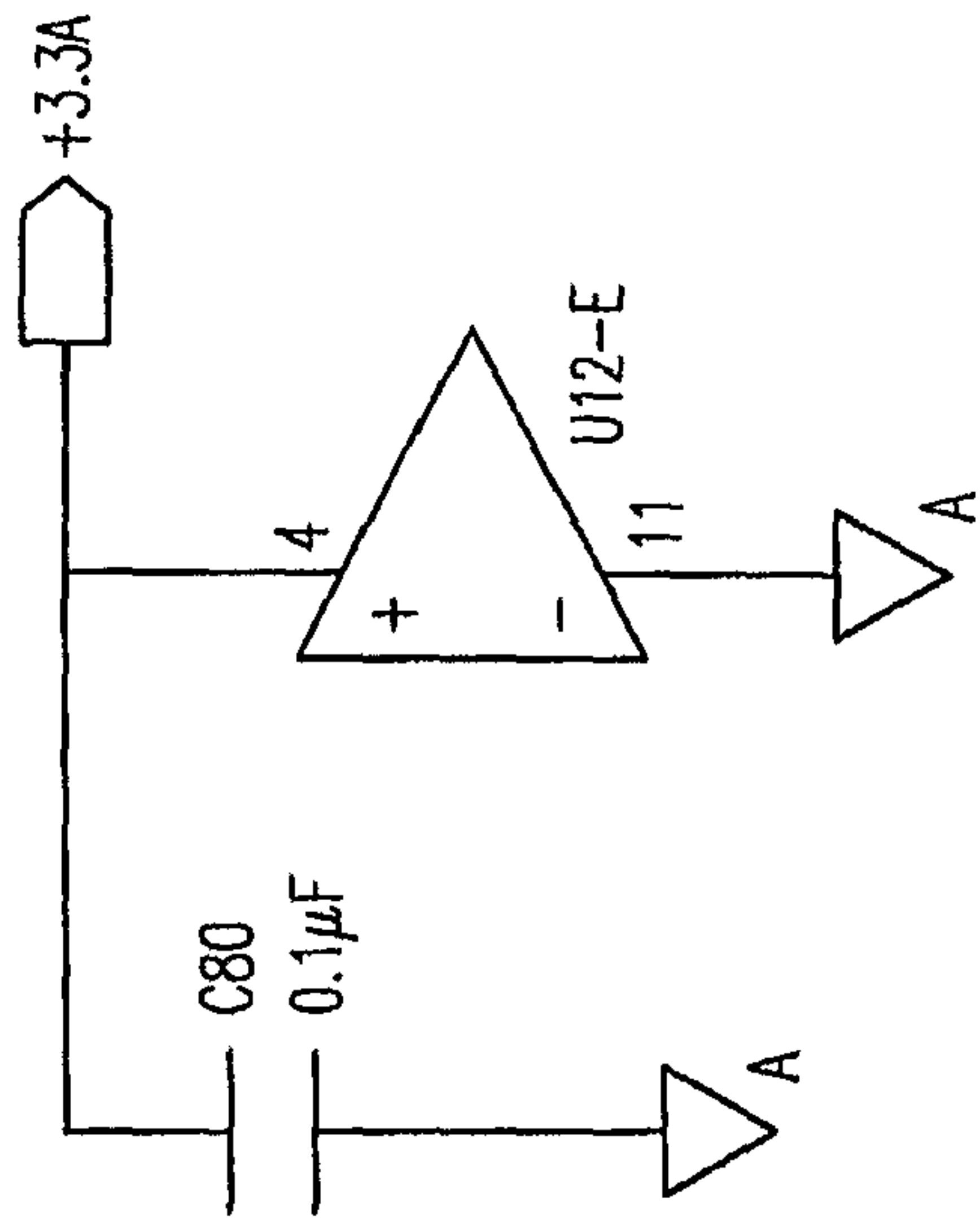


FIG. 23J-1

178/227



IC/74HC4051/TSSQP16

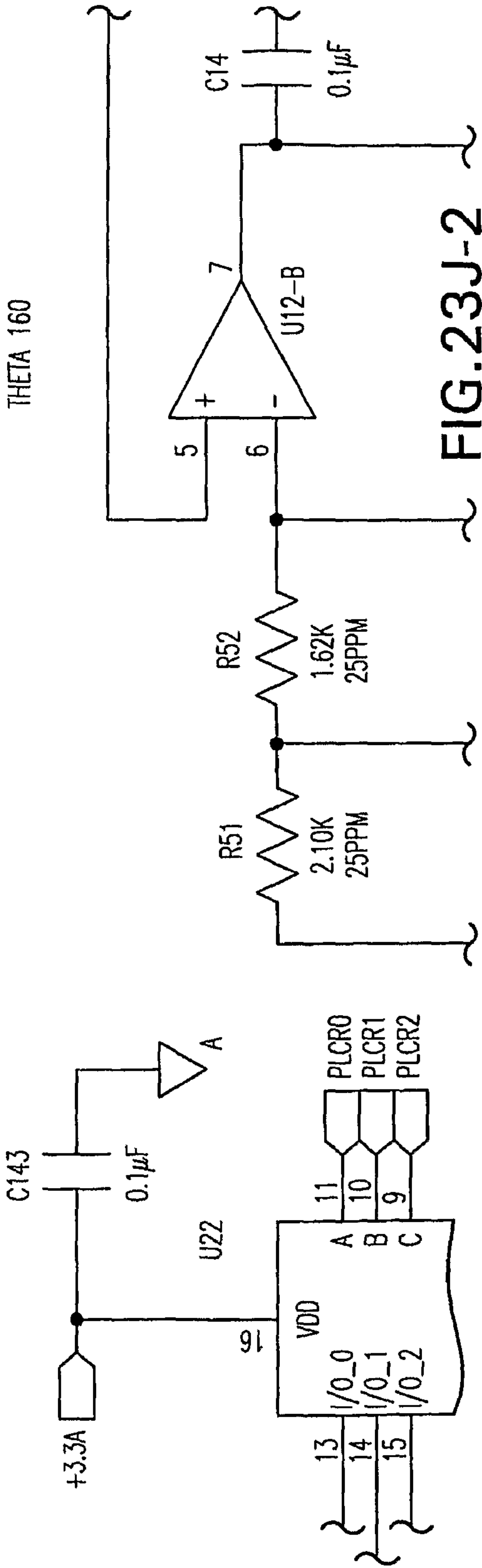
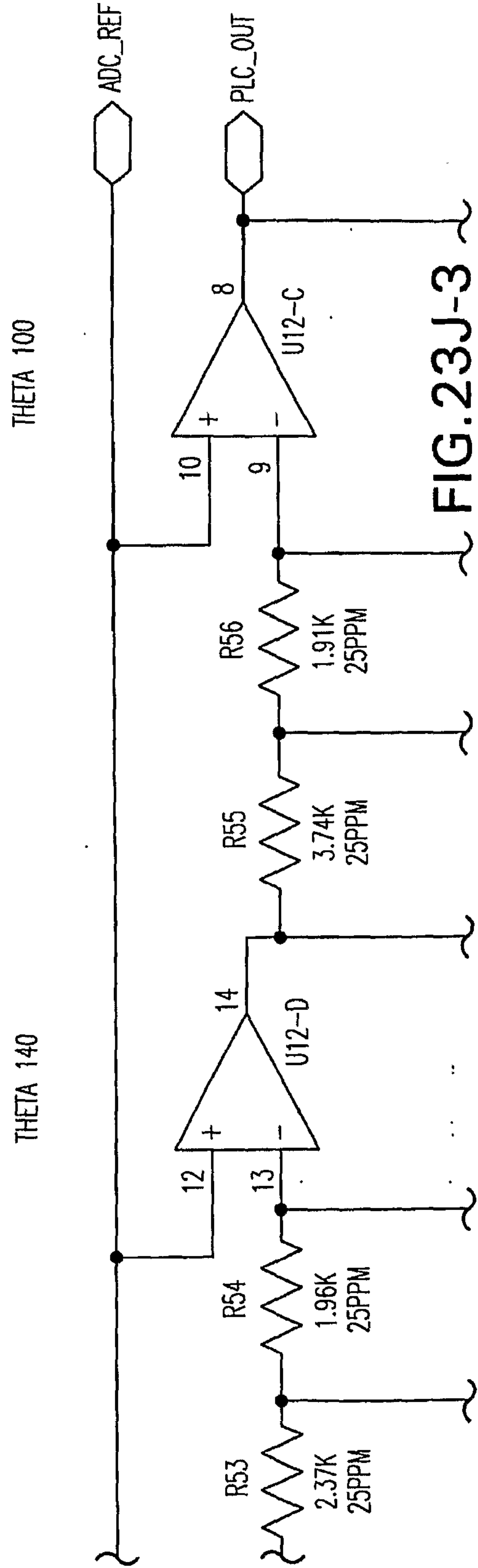


FIG.23J-2

179/227



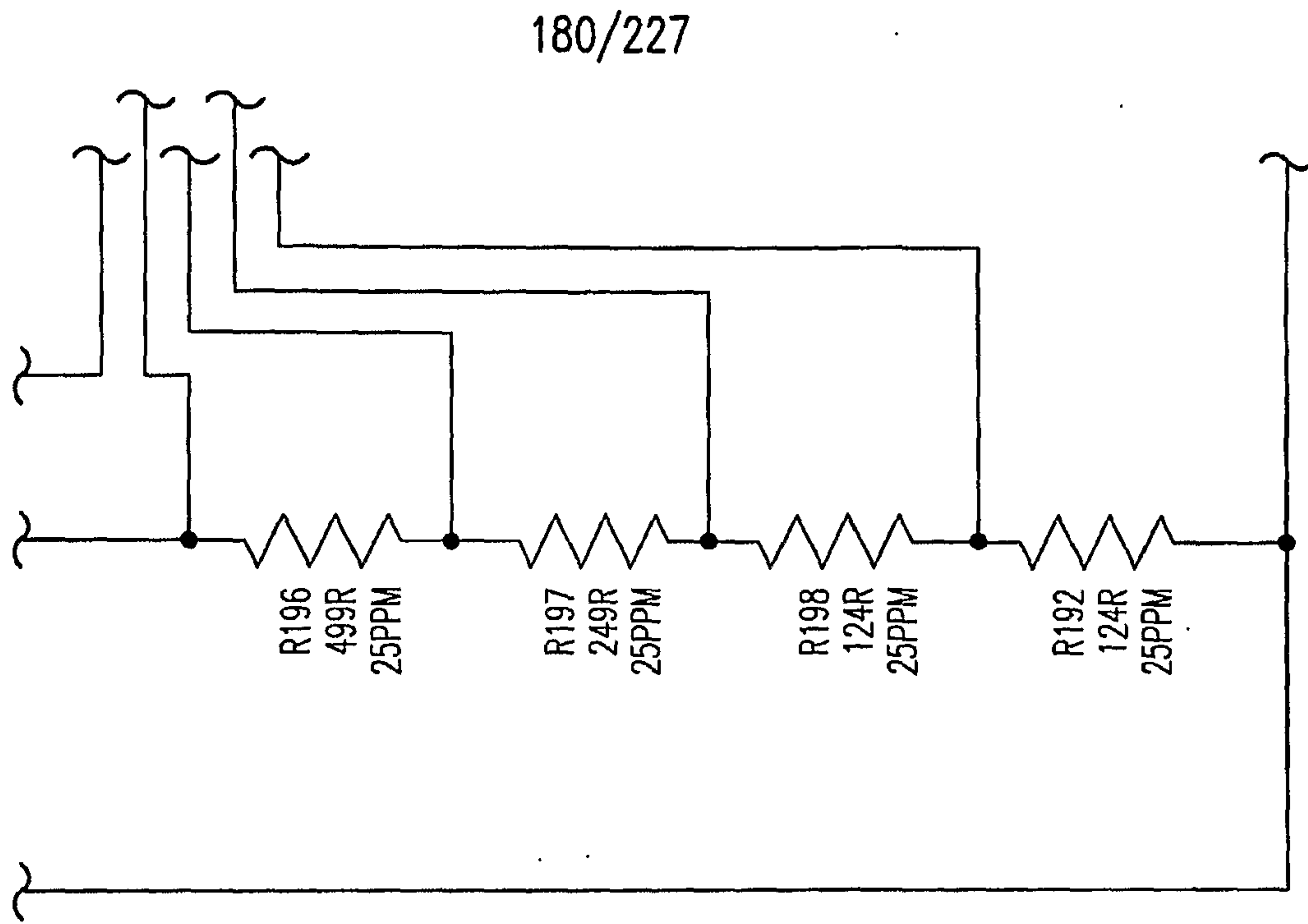


FIG. 23J-4

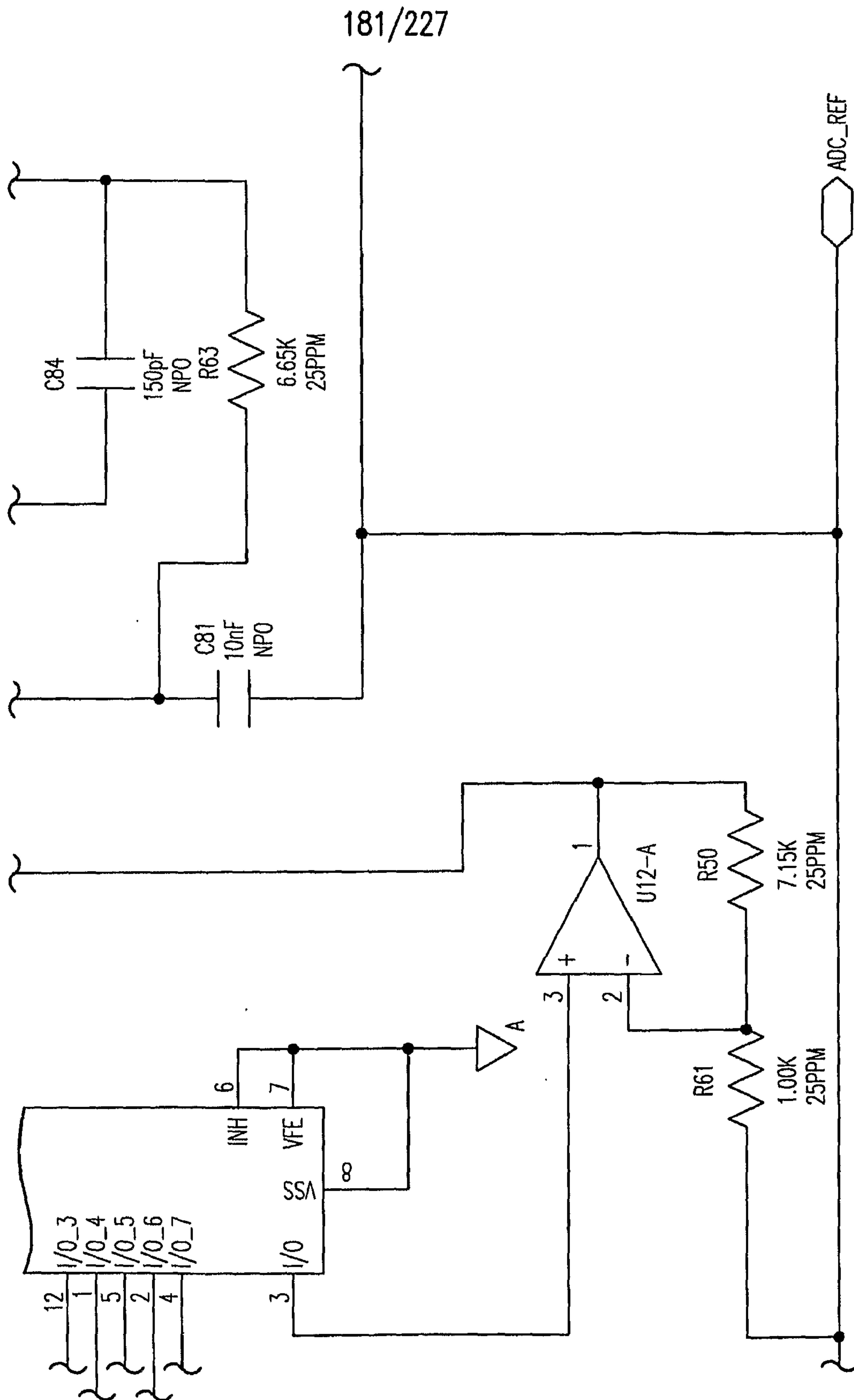


FIG. 23J-5

182/227

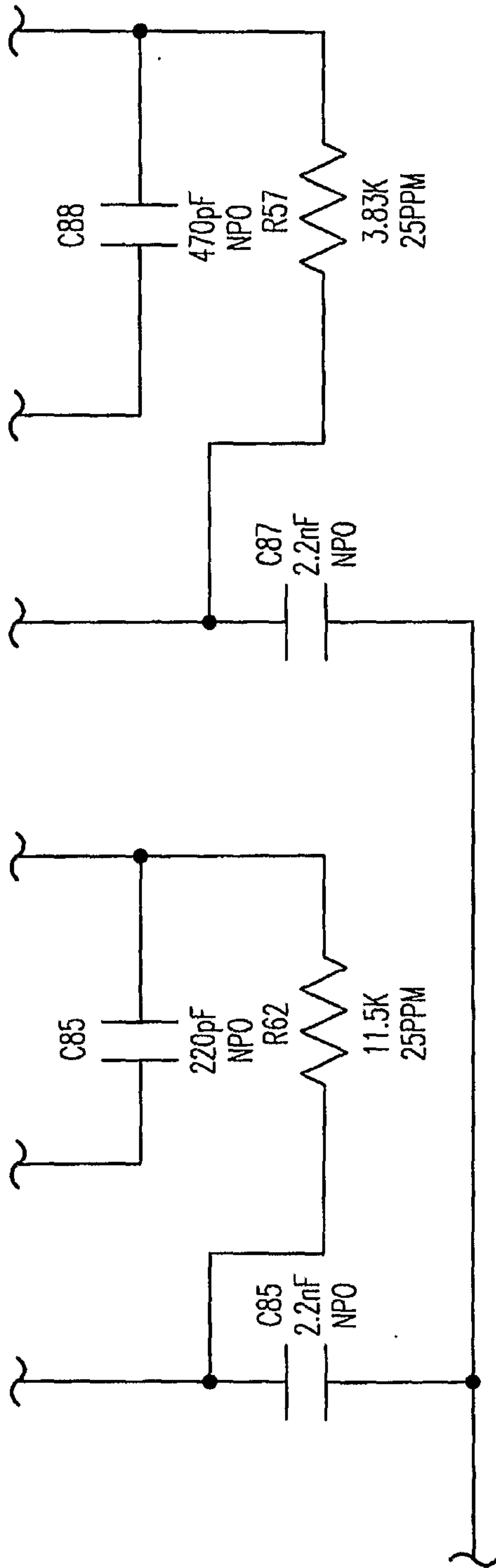


FIG. 23J-6

183/227

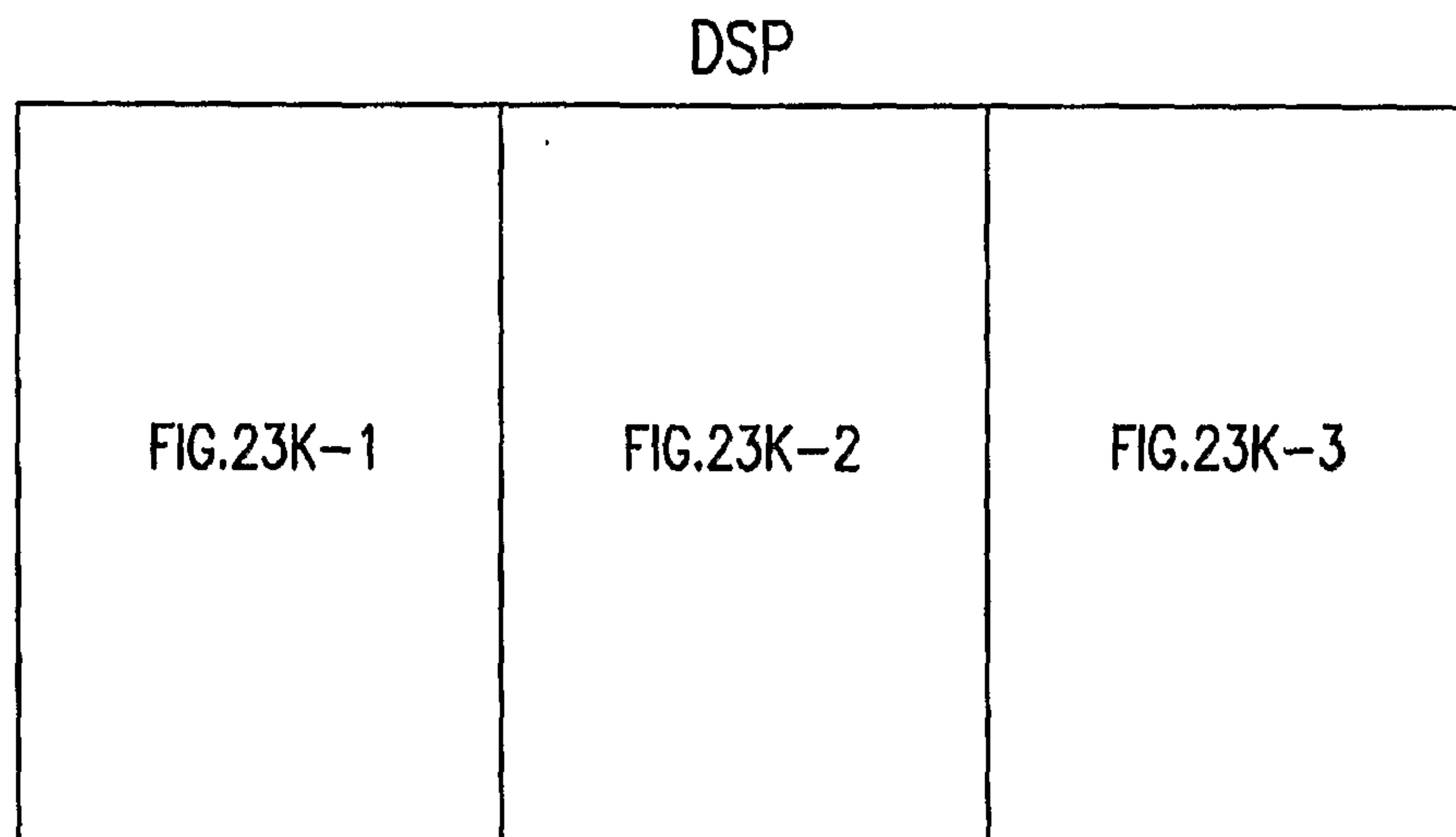


FIG.23K

184/227

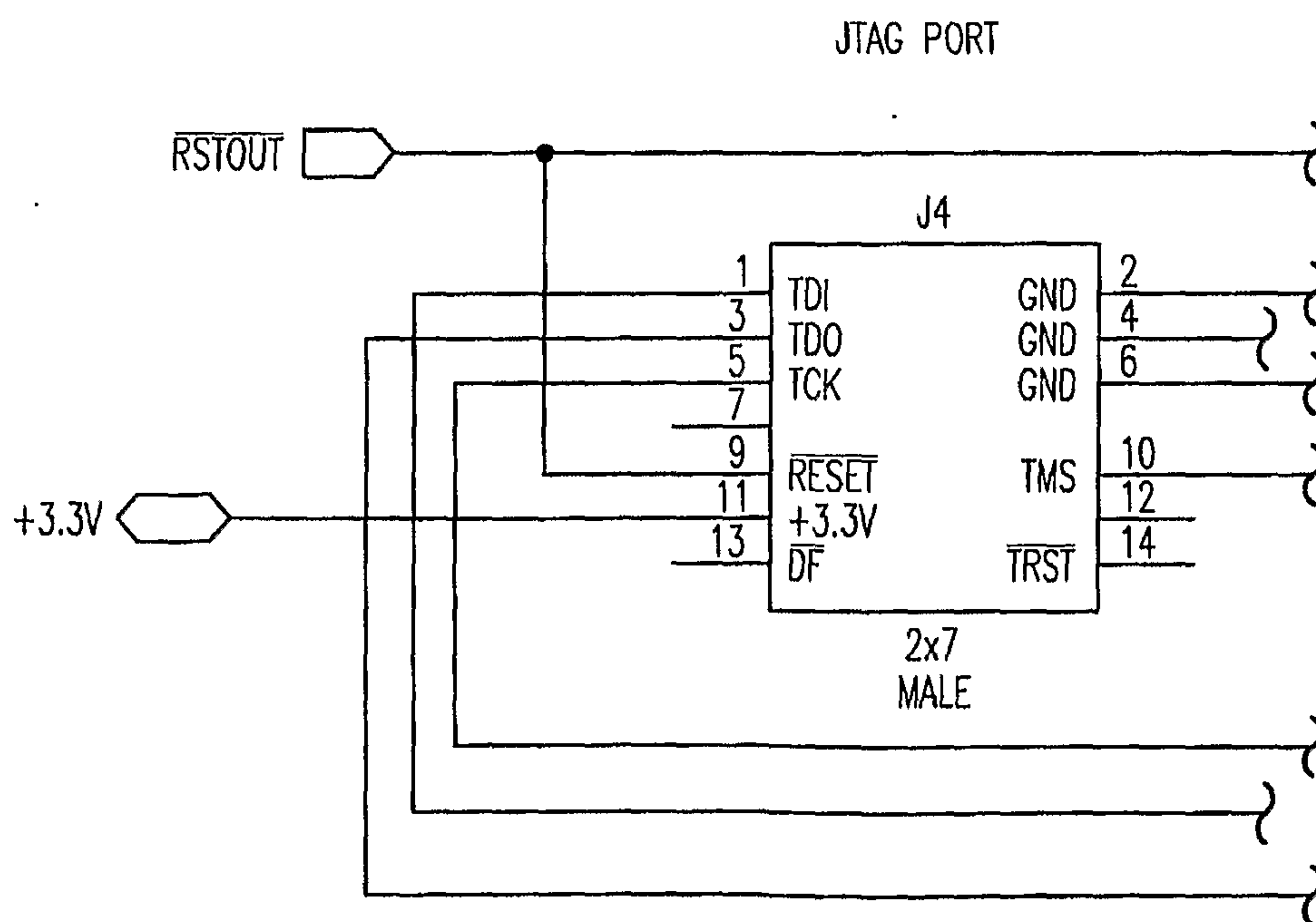
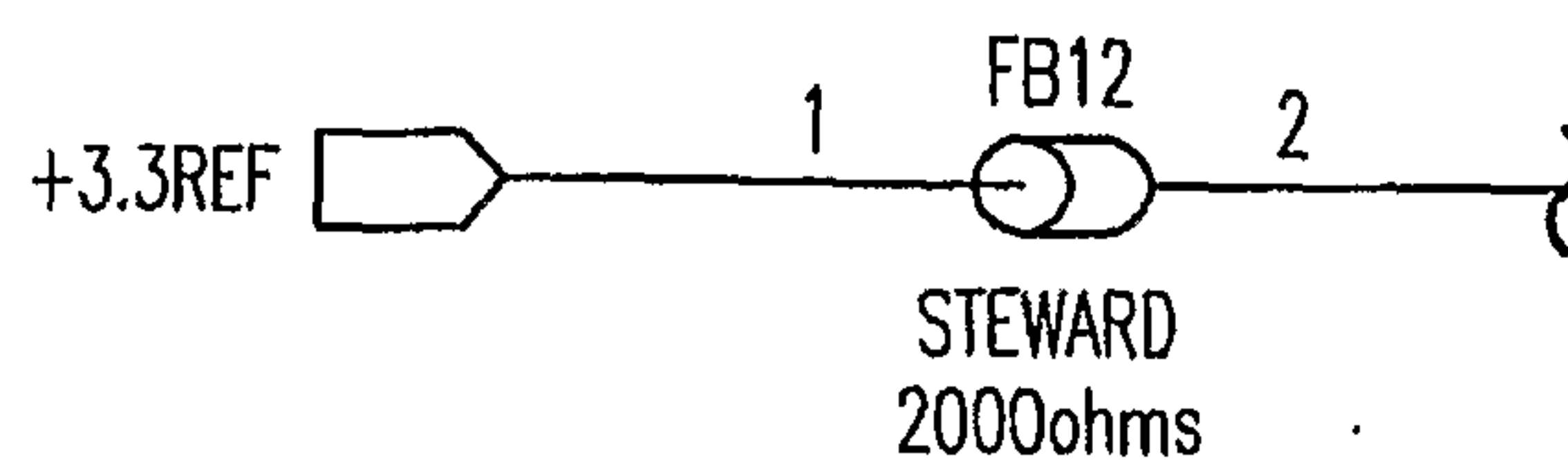


FIG.23K-1

185/227

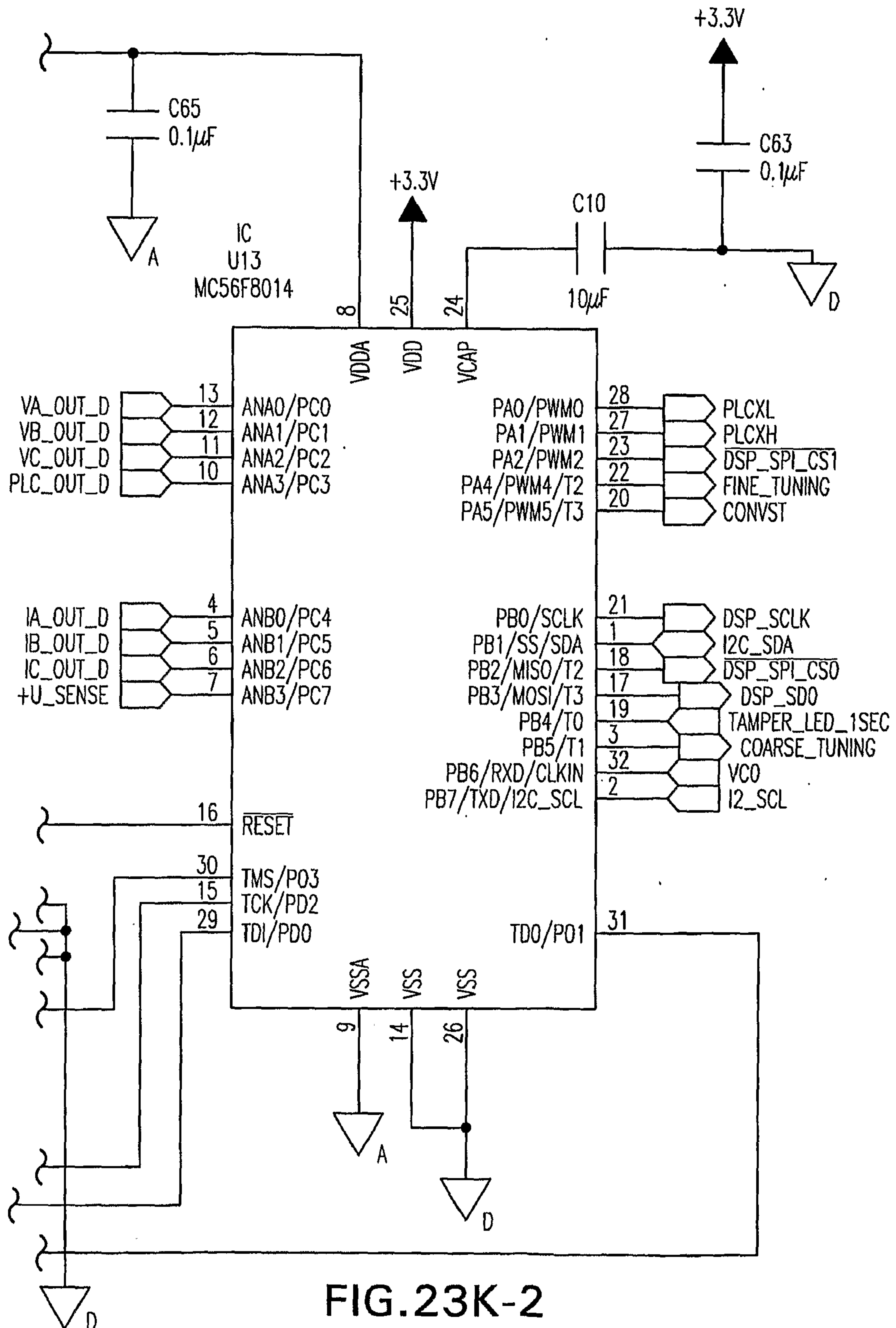


FIG.23K-2

186/227

CONNECTED TO DSP, ADC
FOR SENSING INPUT POWER SUPPLY LEVELS

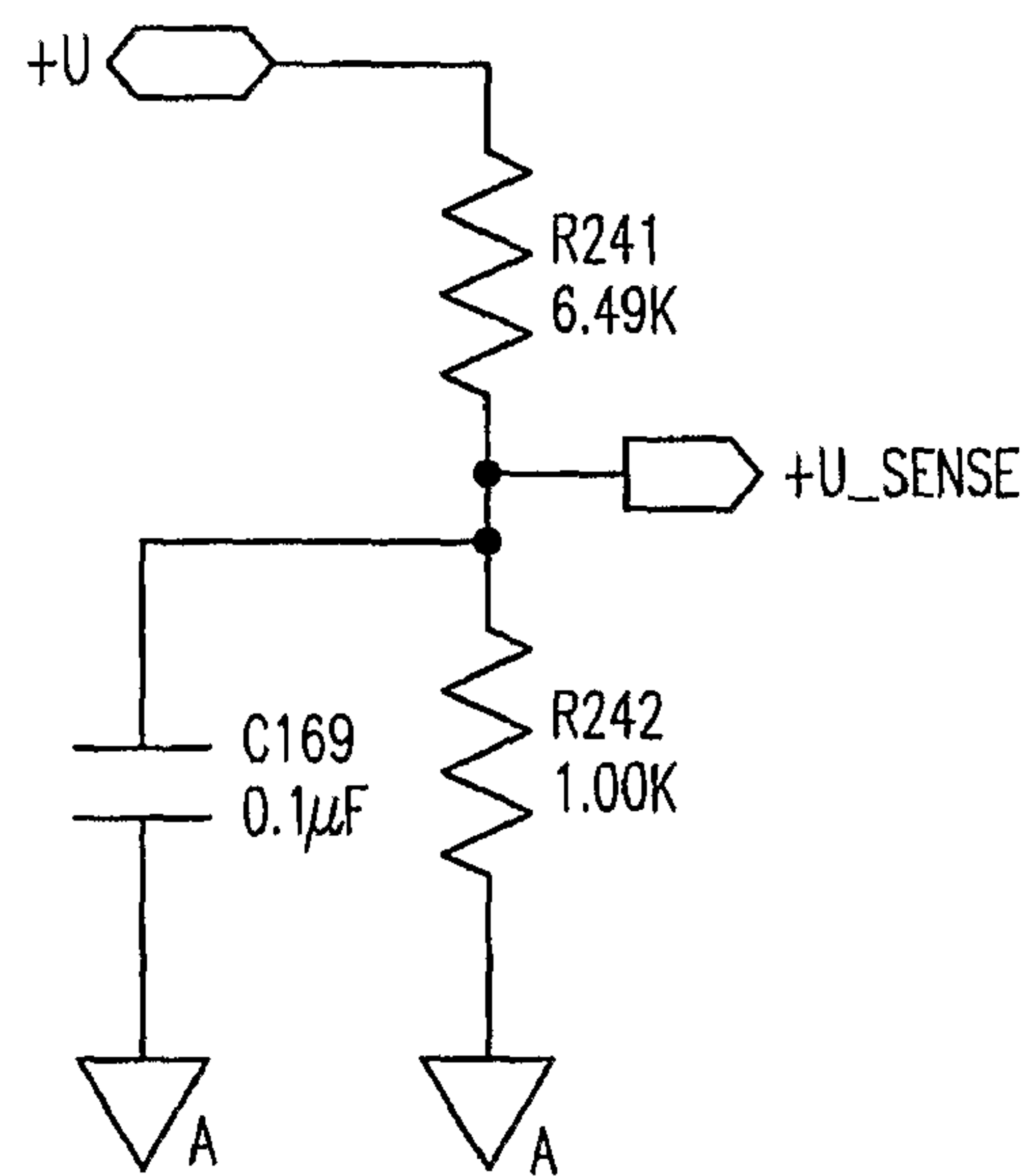


FIG.23K-3

187/227

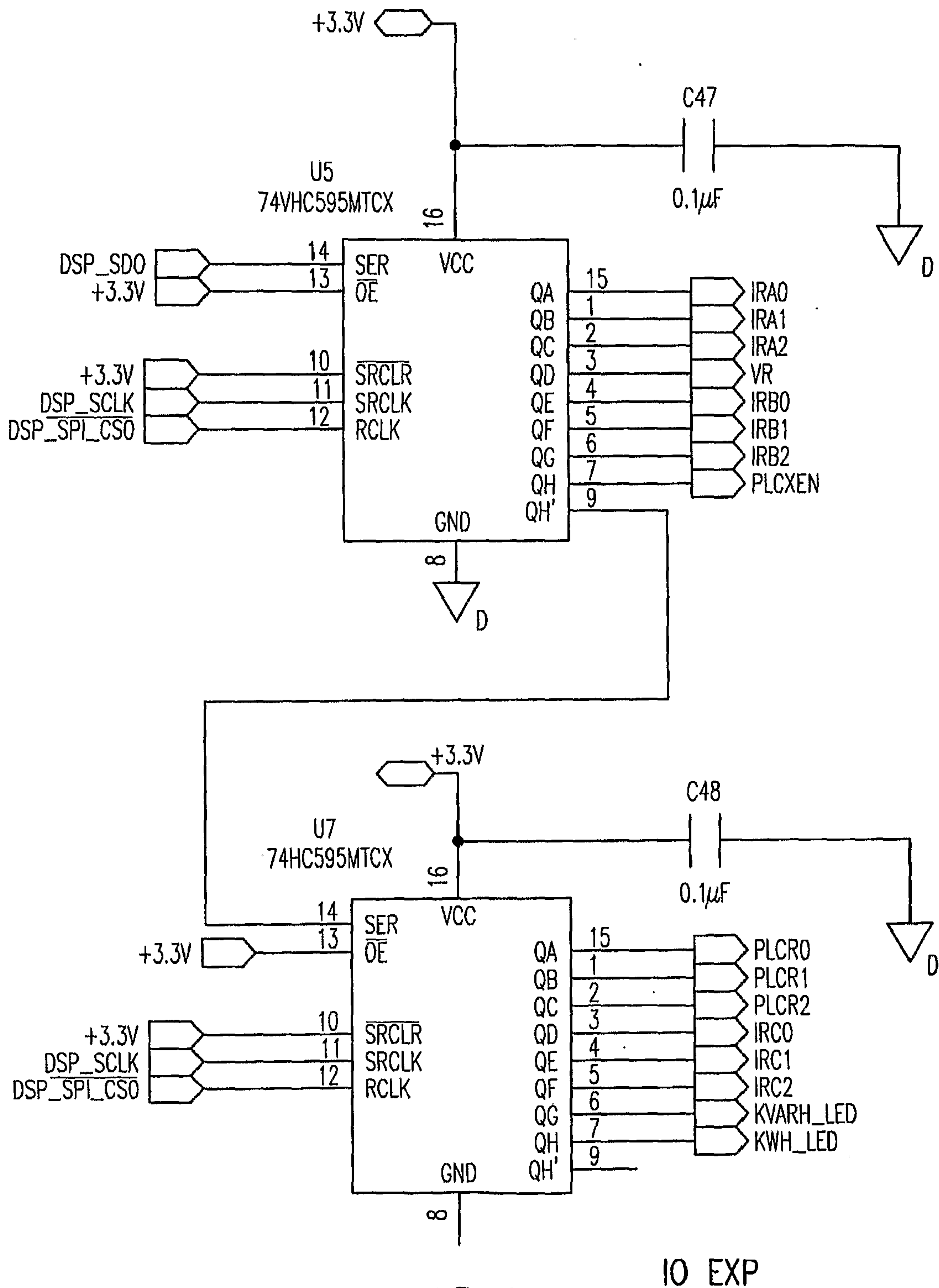
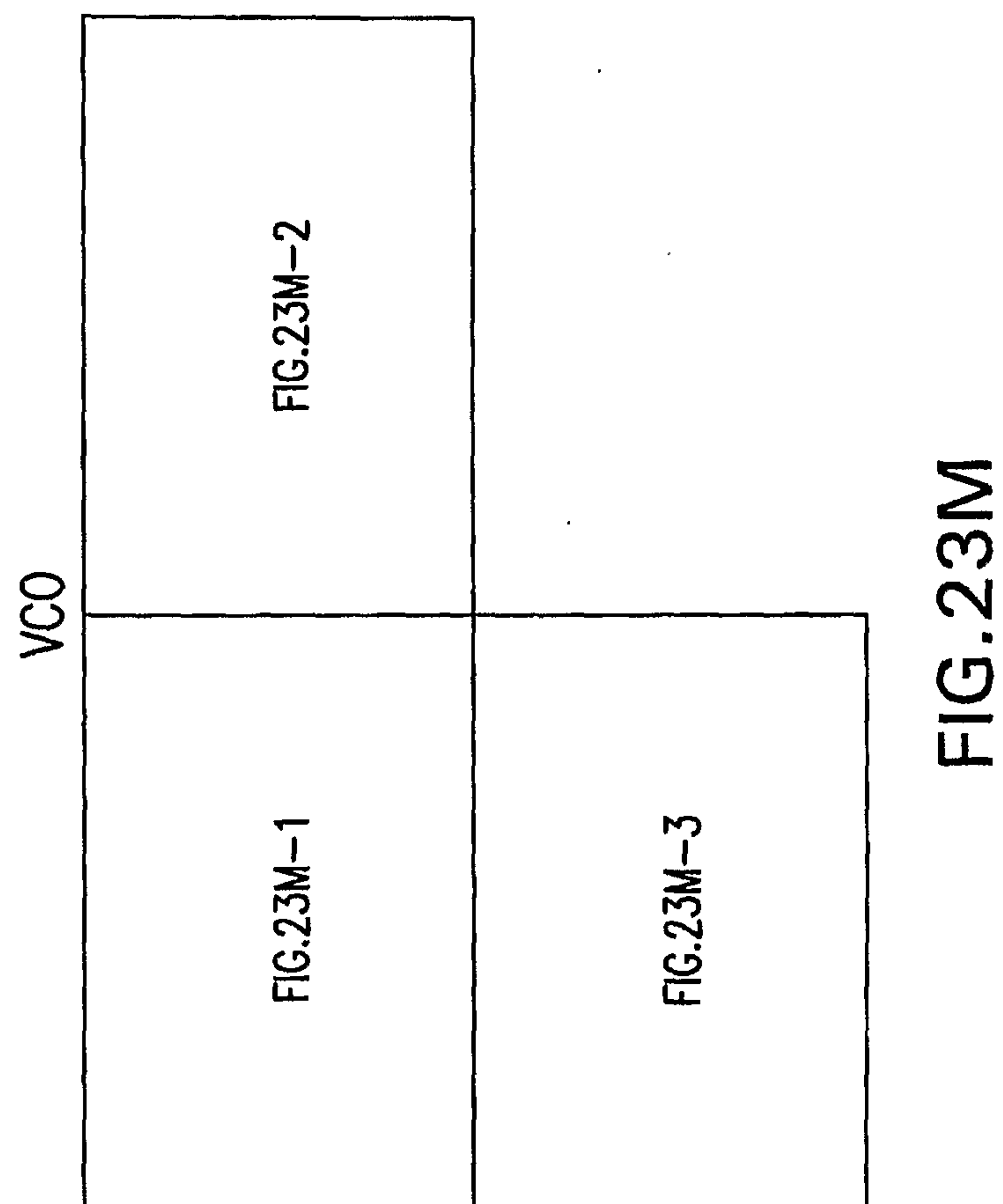


FIG.23L

188/227



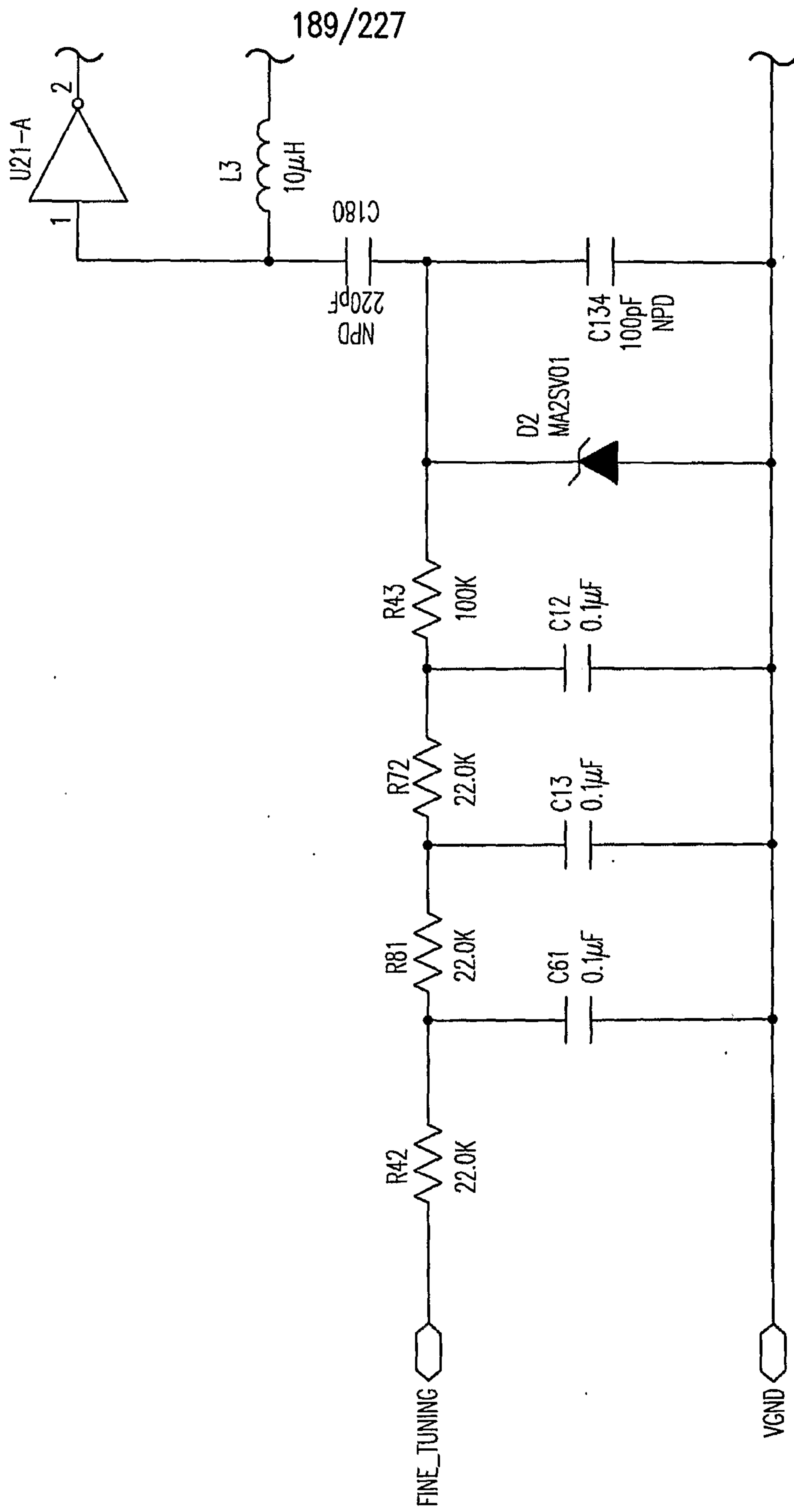


FIG. 23M-1

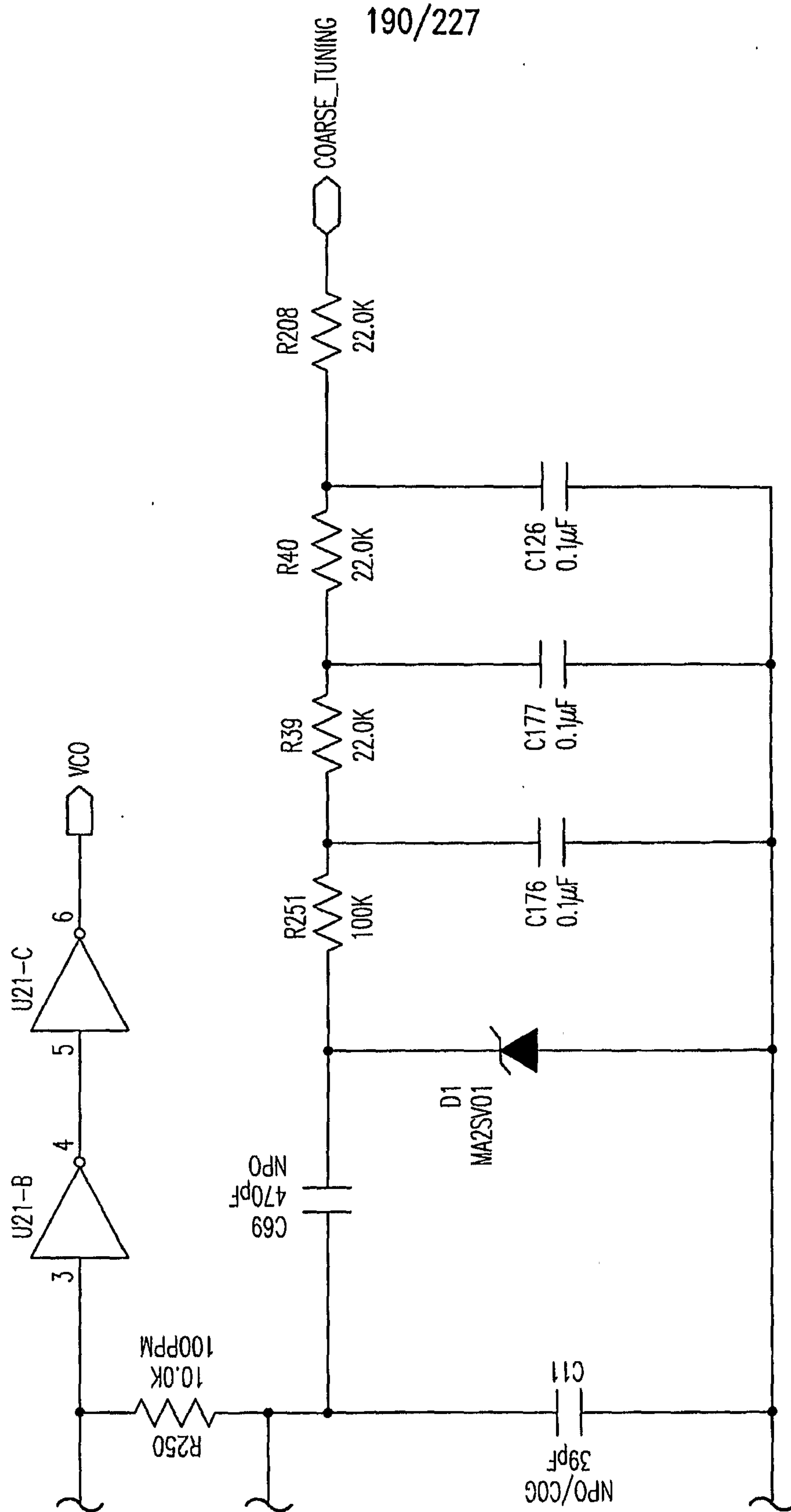


FIG. 23M-2

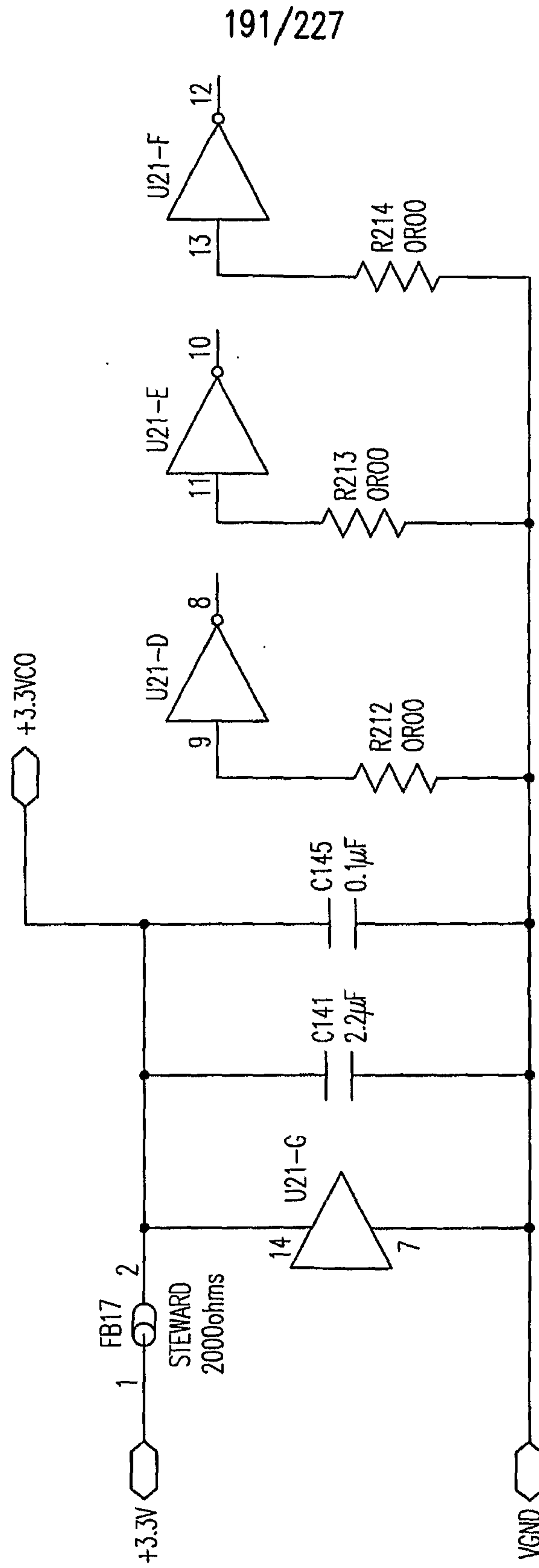


FIG.23M-3

192/227

PLC SCH202-3

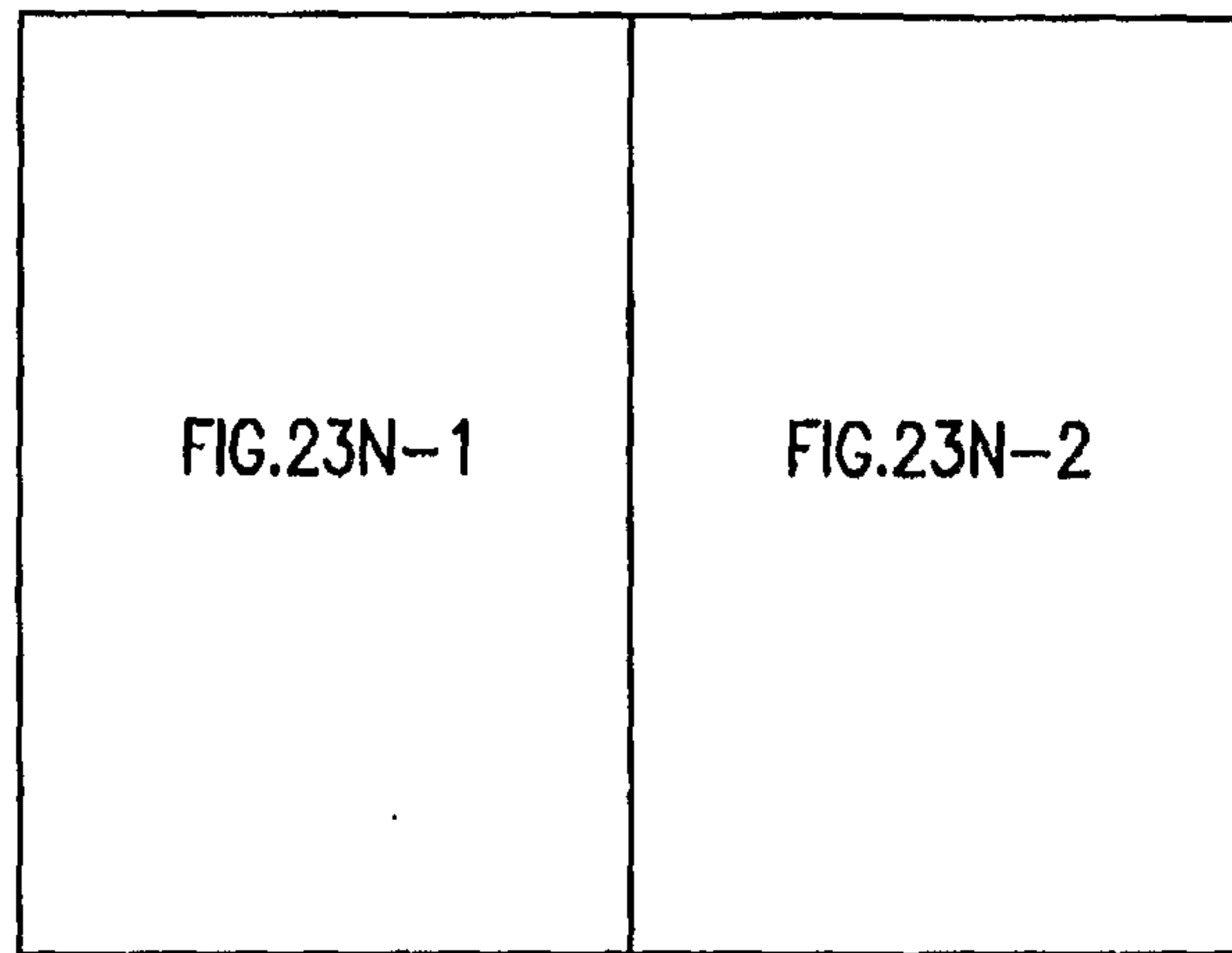


FIG.23N

193/227

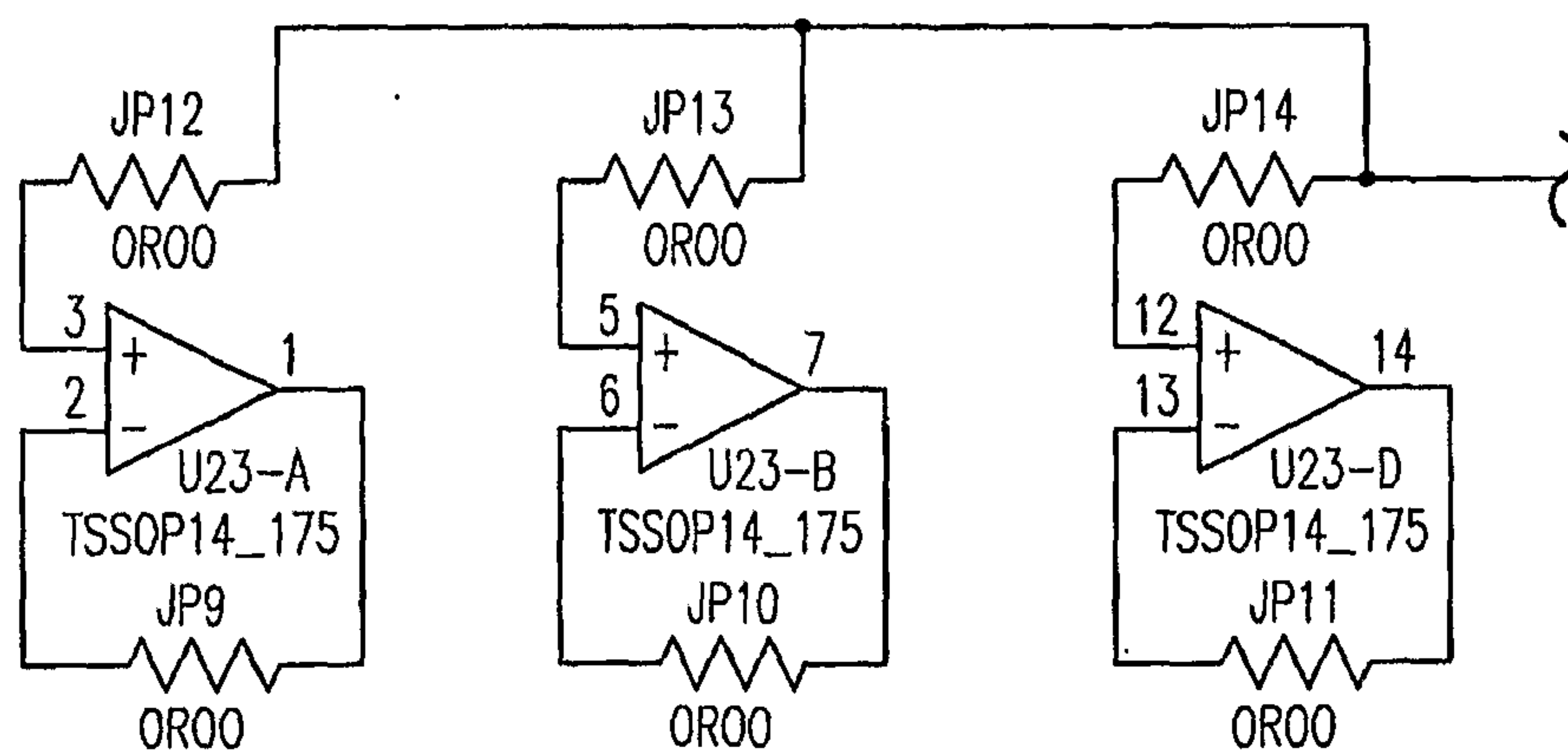
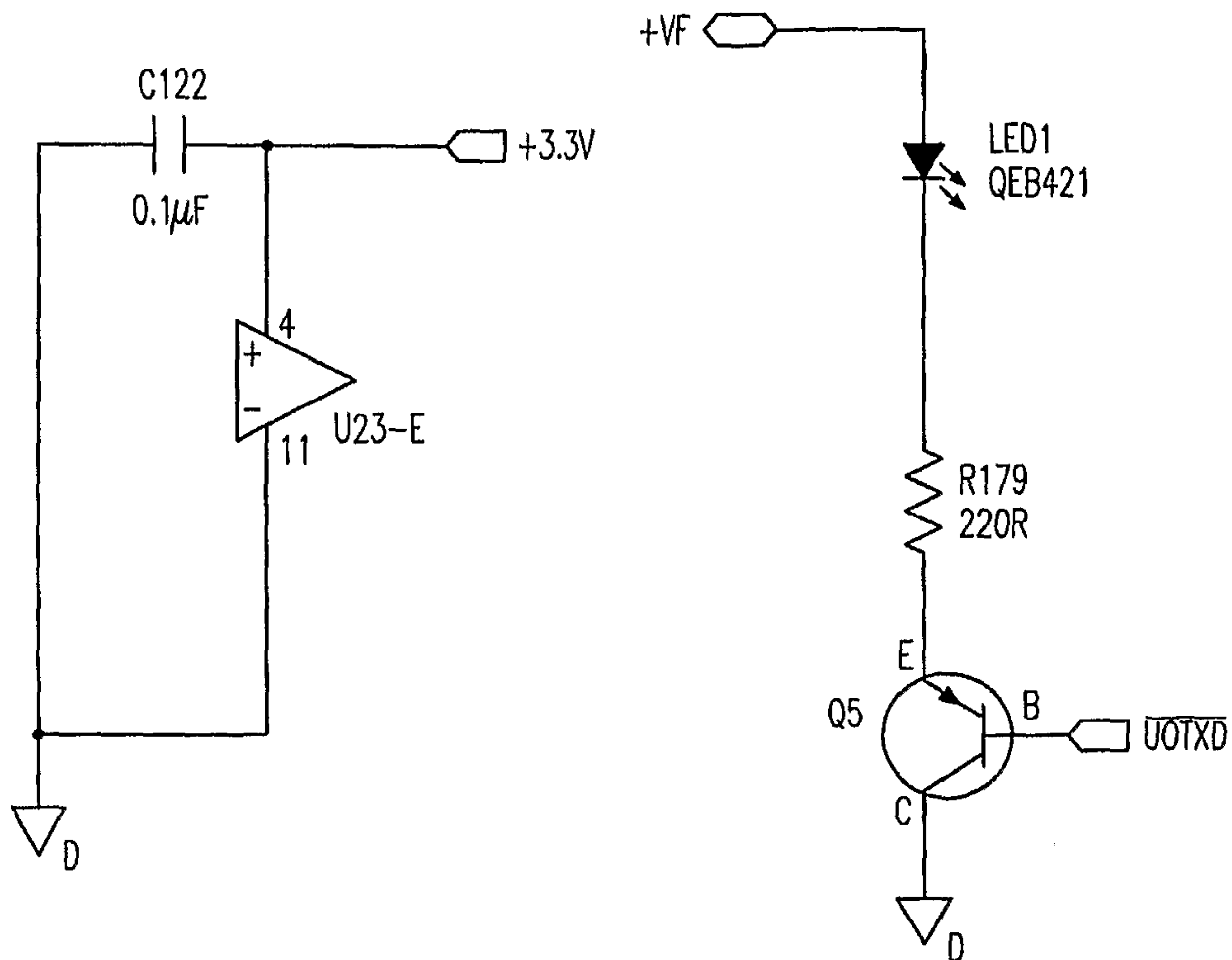


FIG.23N-1

194/227

OPTICAL PORT

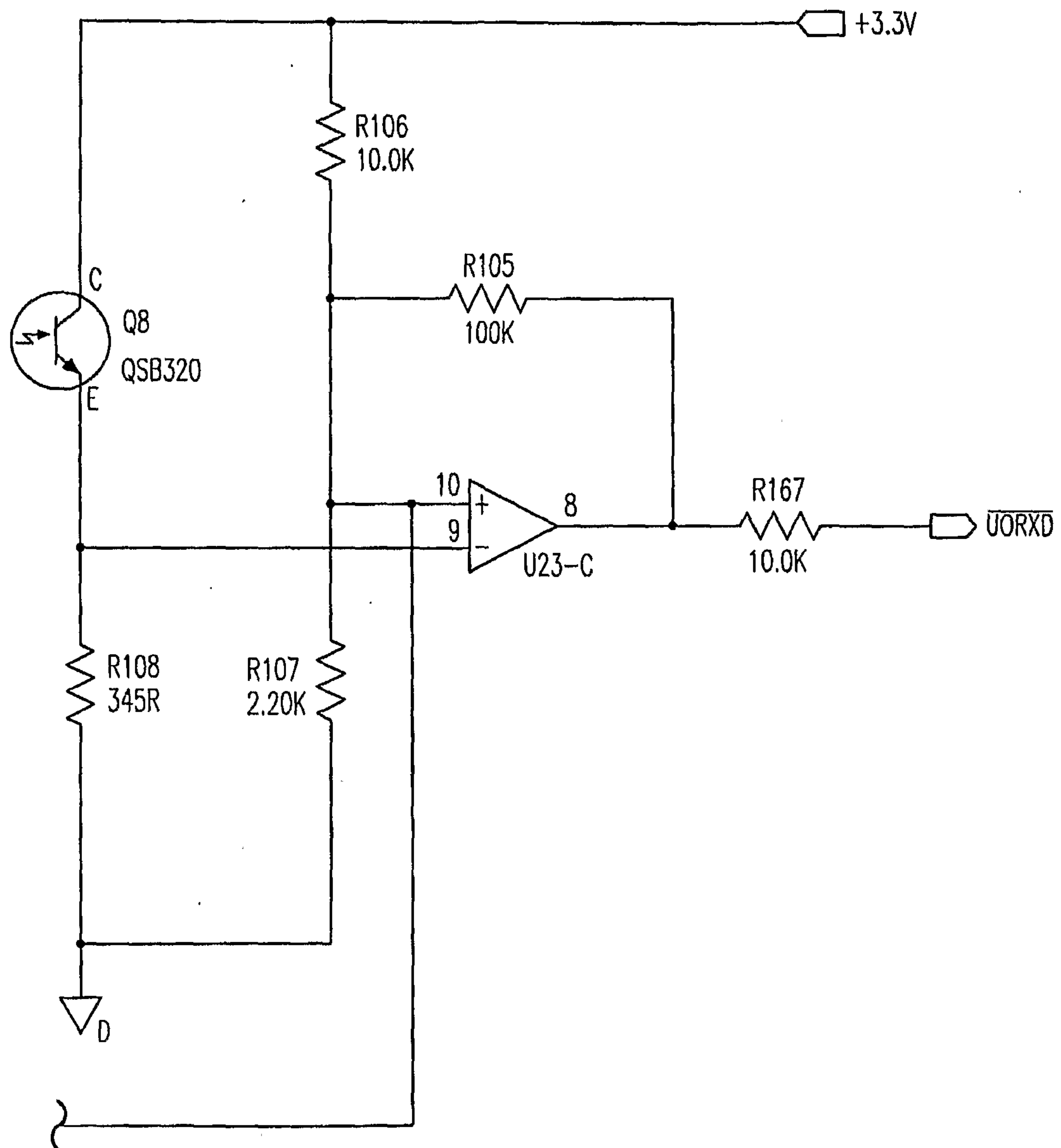


FIG.23N-2

195/227

SERIAL I/O SCH202-3

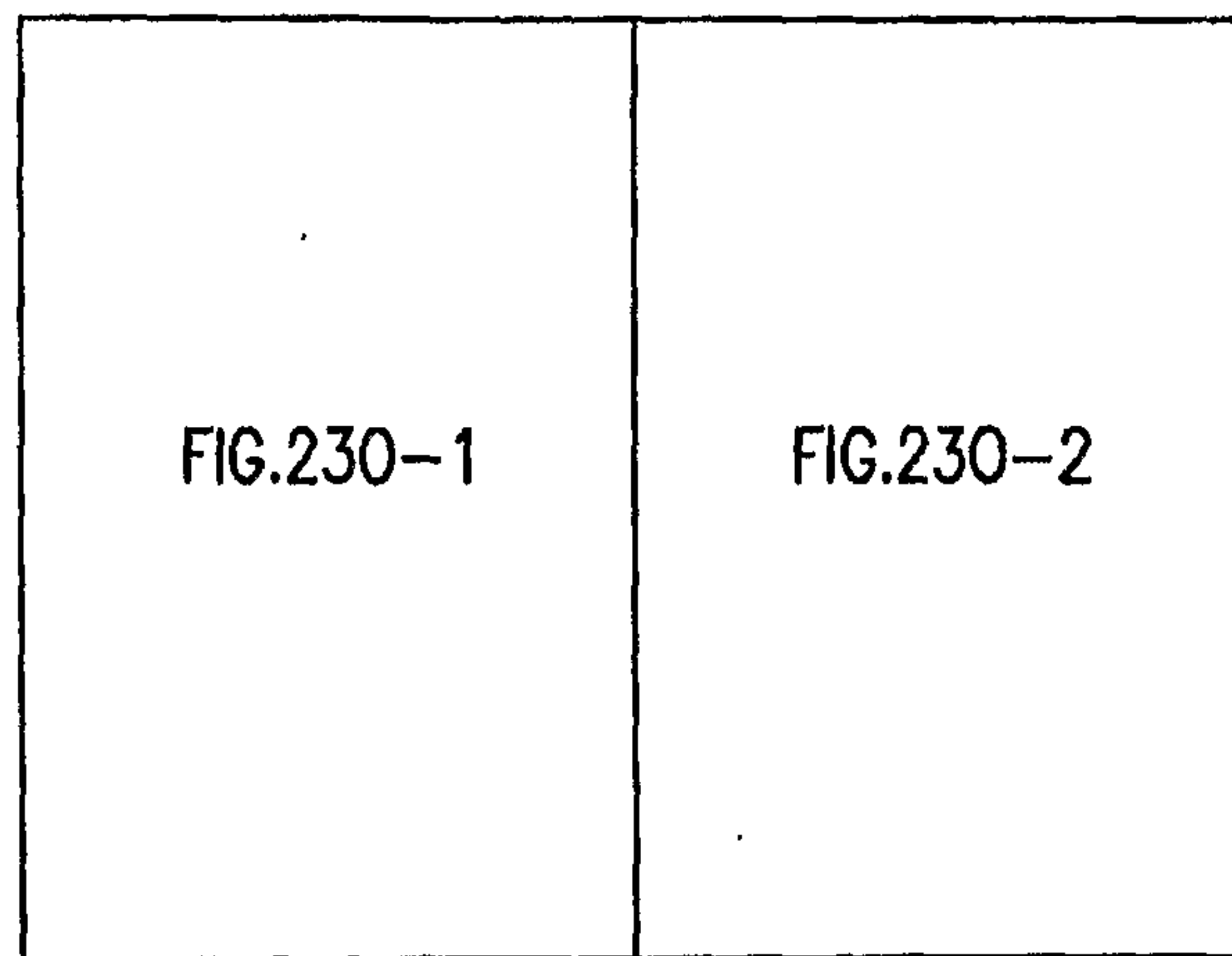
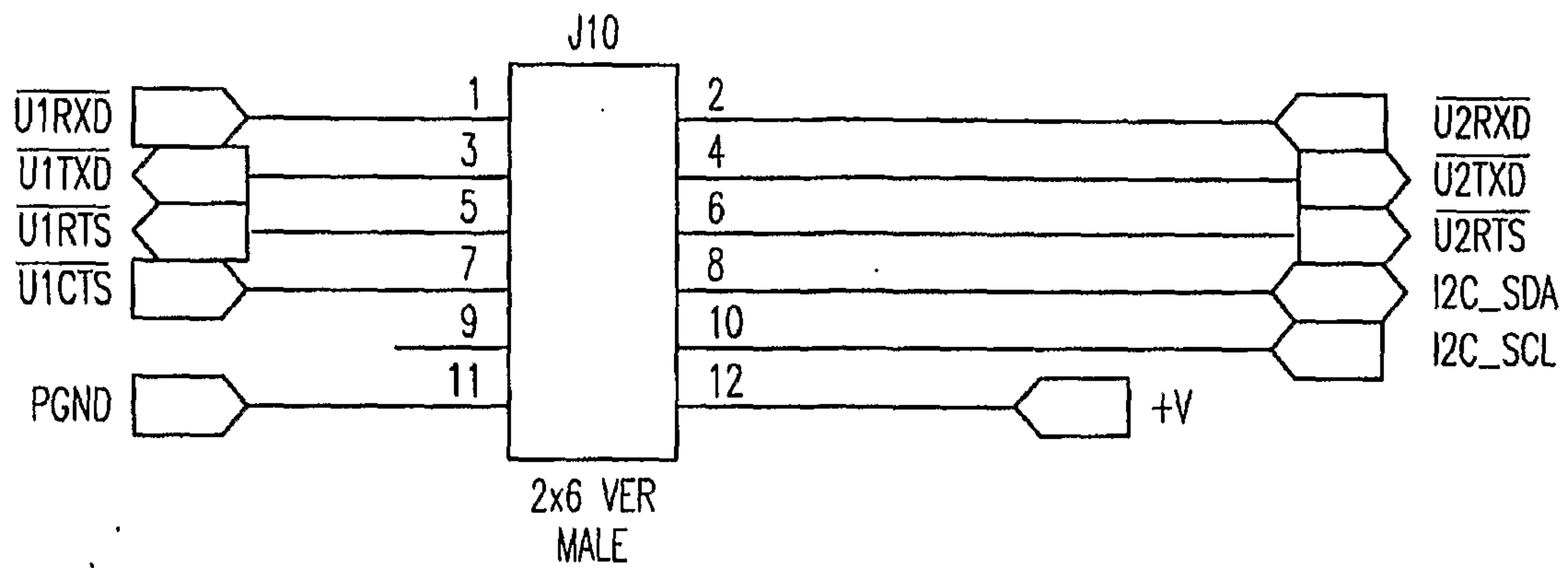


FIG.230

196/227

HEADER TO IO BOARD



HEADER TO POWER BOARD

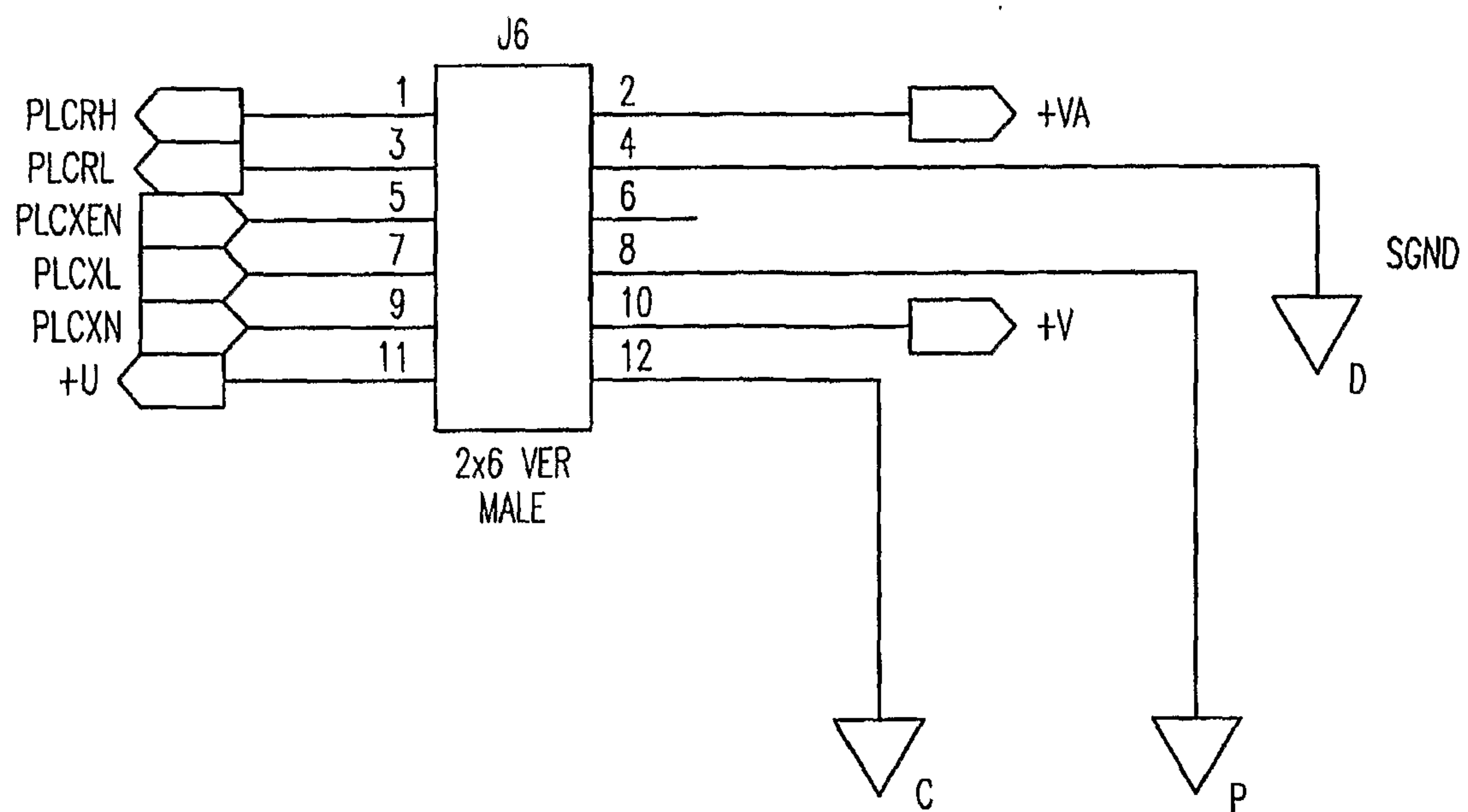
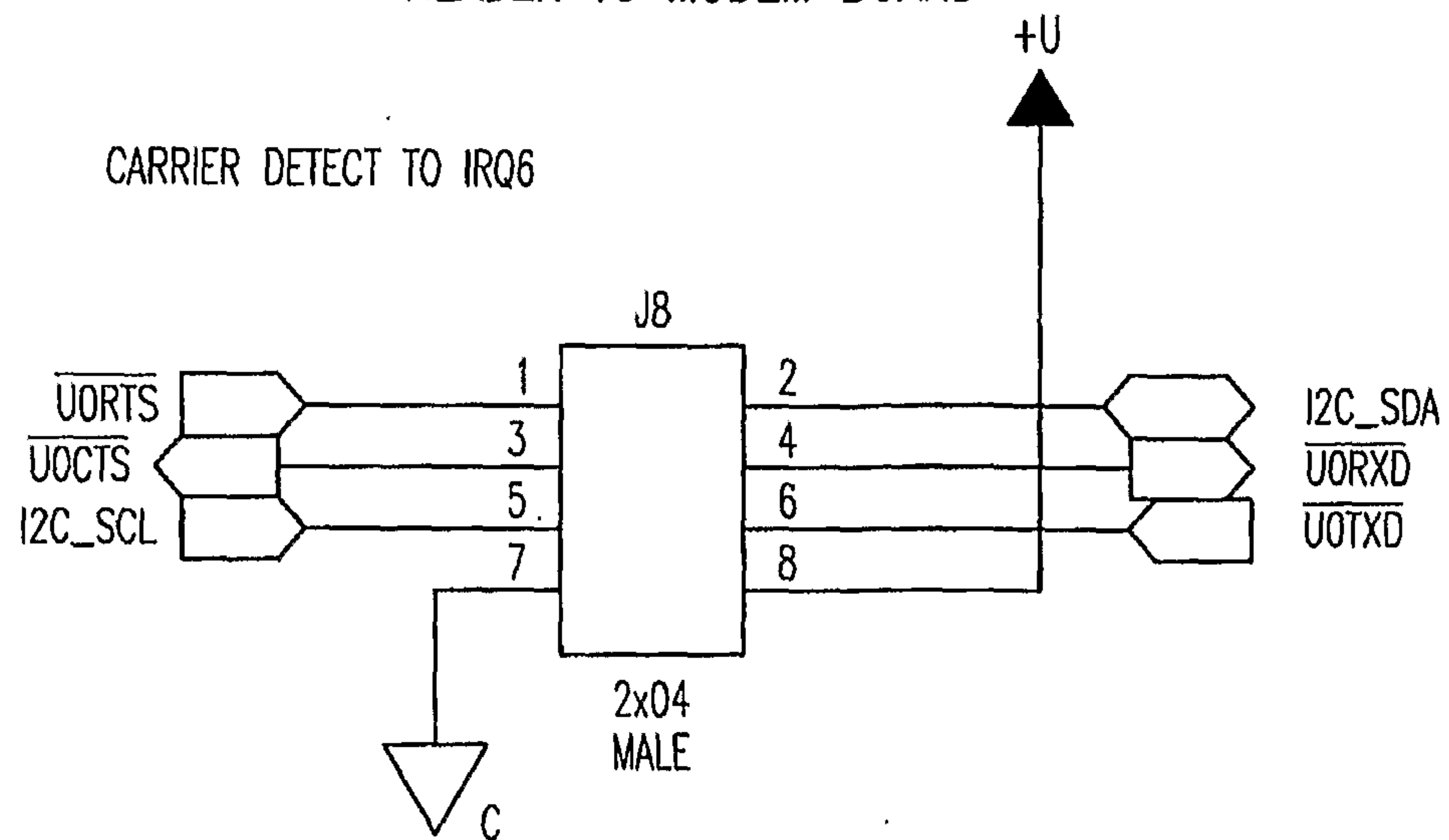


FIG.230-1

197/227

HEADER TO MODEM BOARD



HEADER TO DISPLAY BOARD

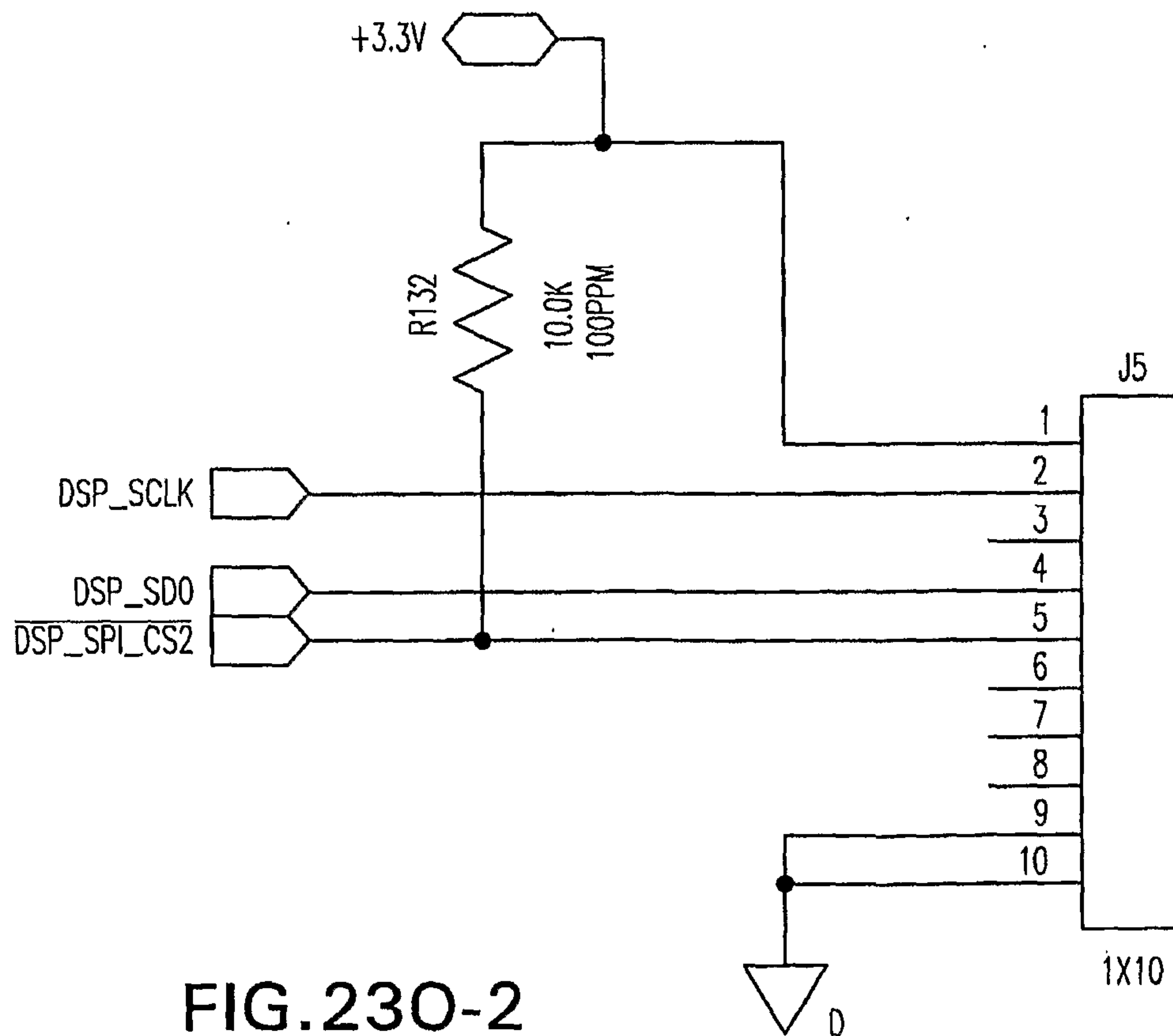
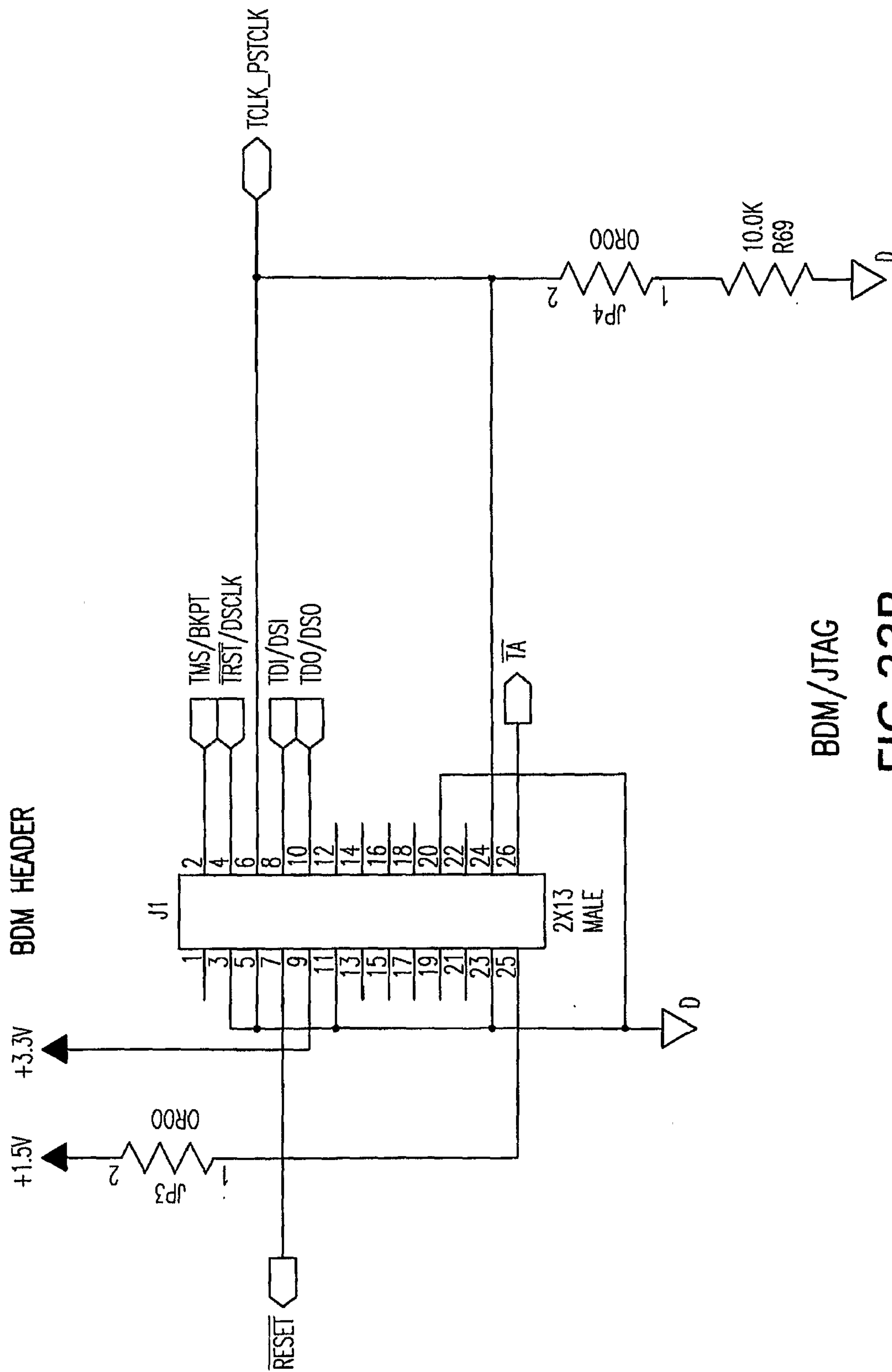


FIG.230-2

198/227

BDM/JTAG
FIG. 23P

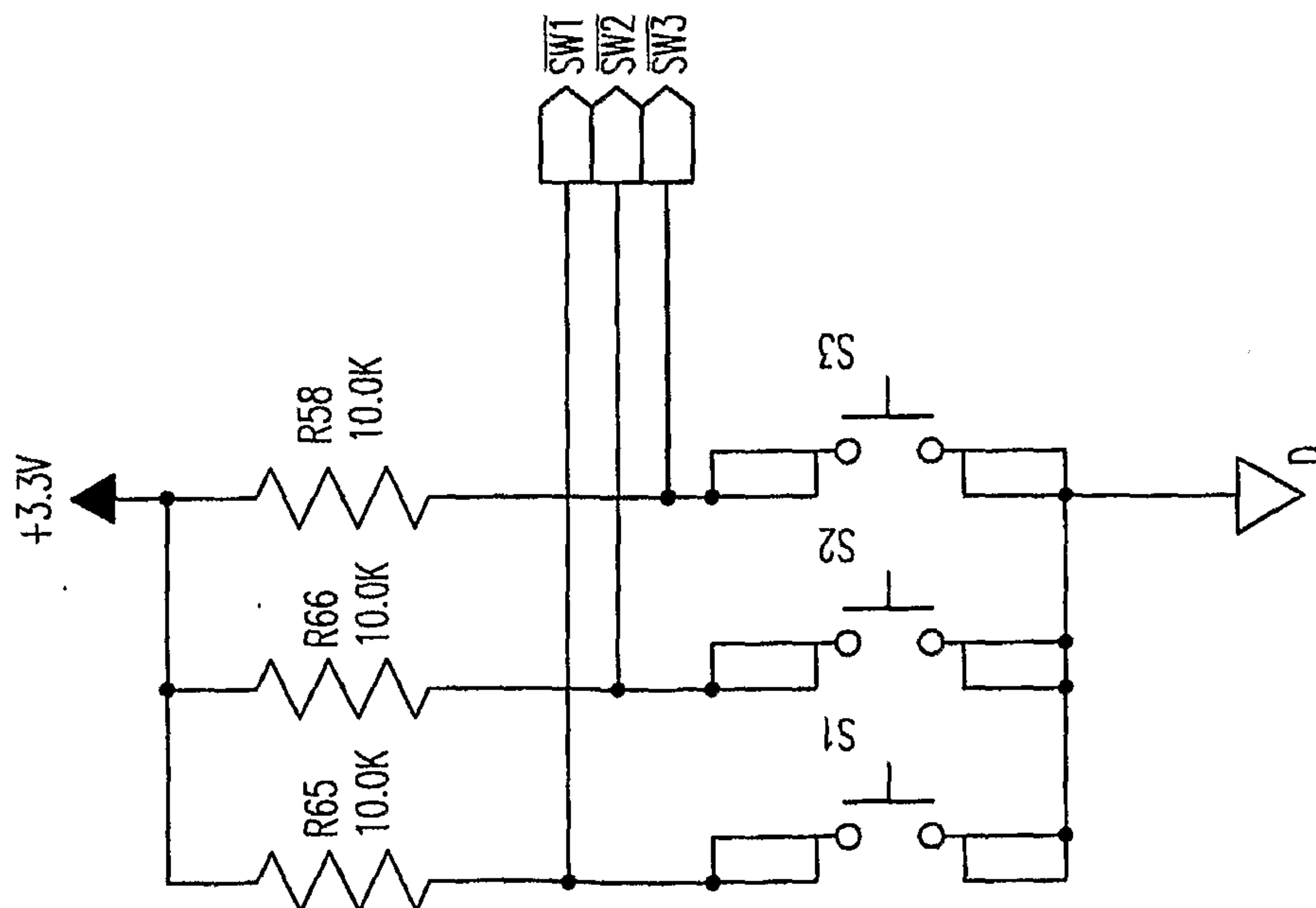
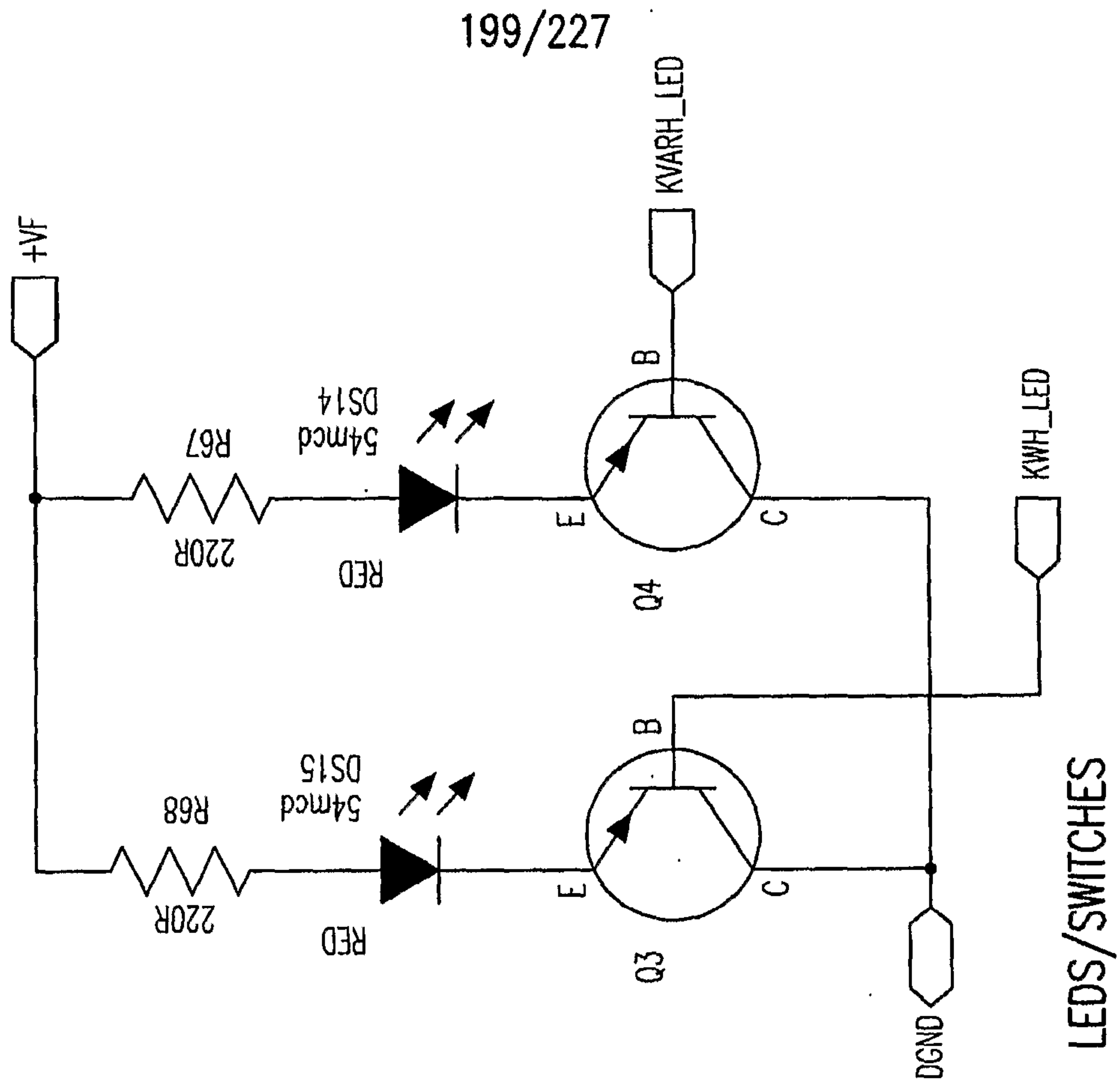
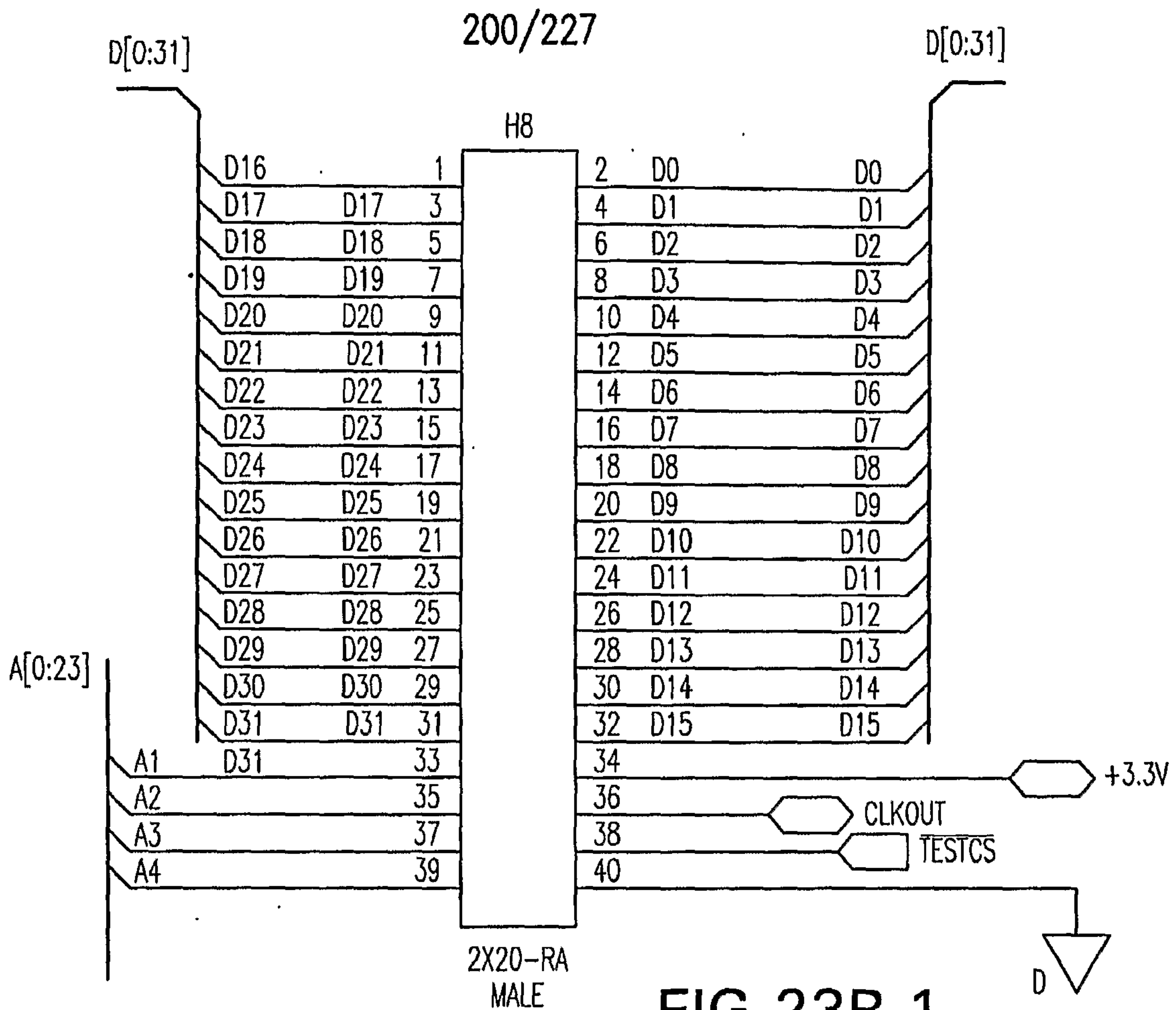
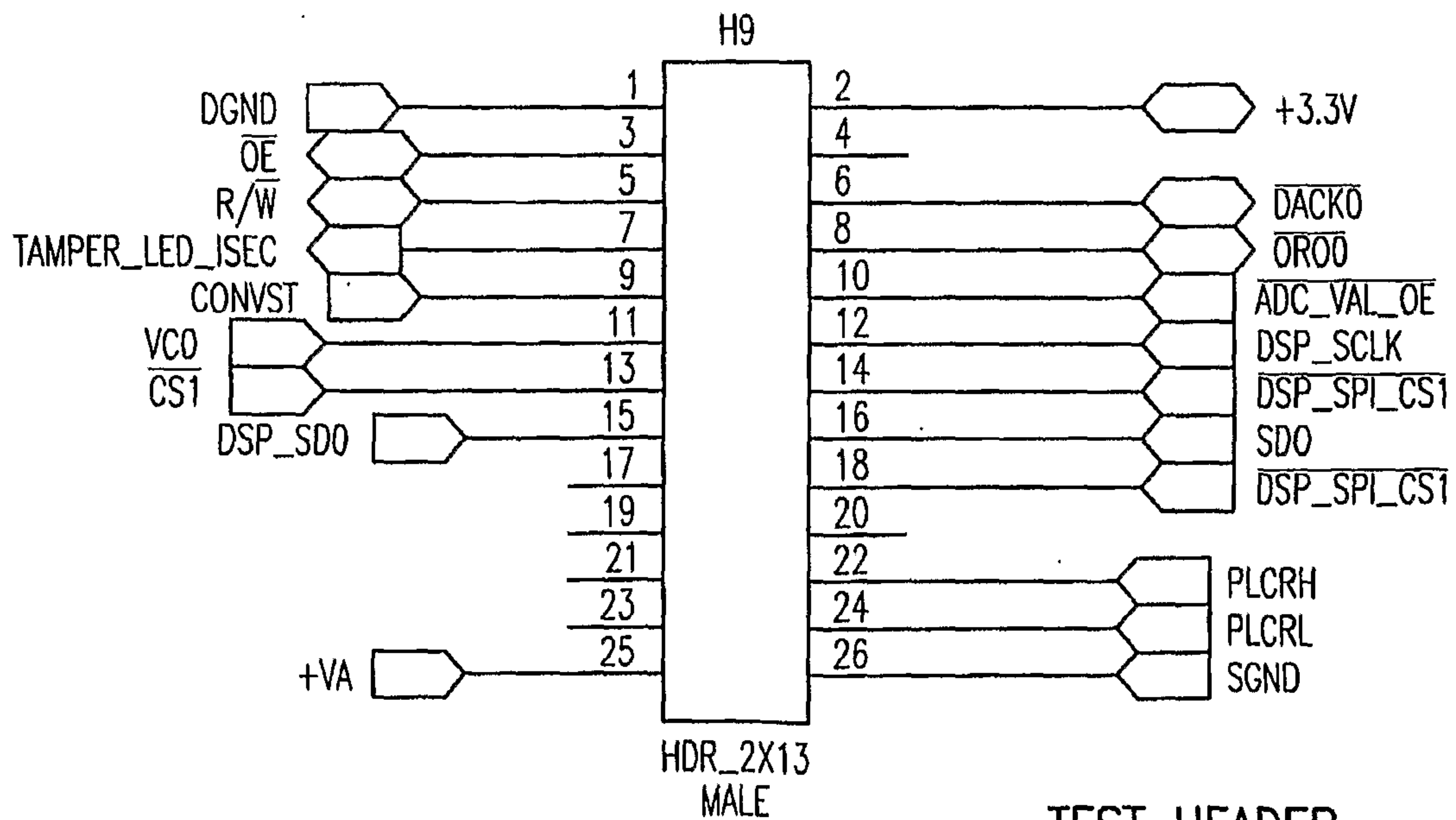


FIG. 230

**FIG.23R-1****FIG.23R-2**

201/227

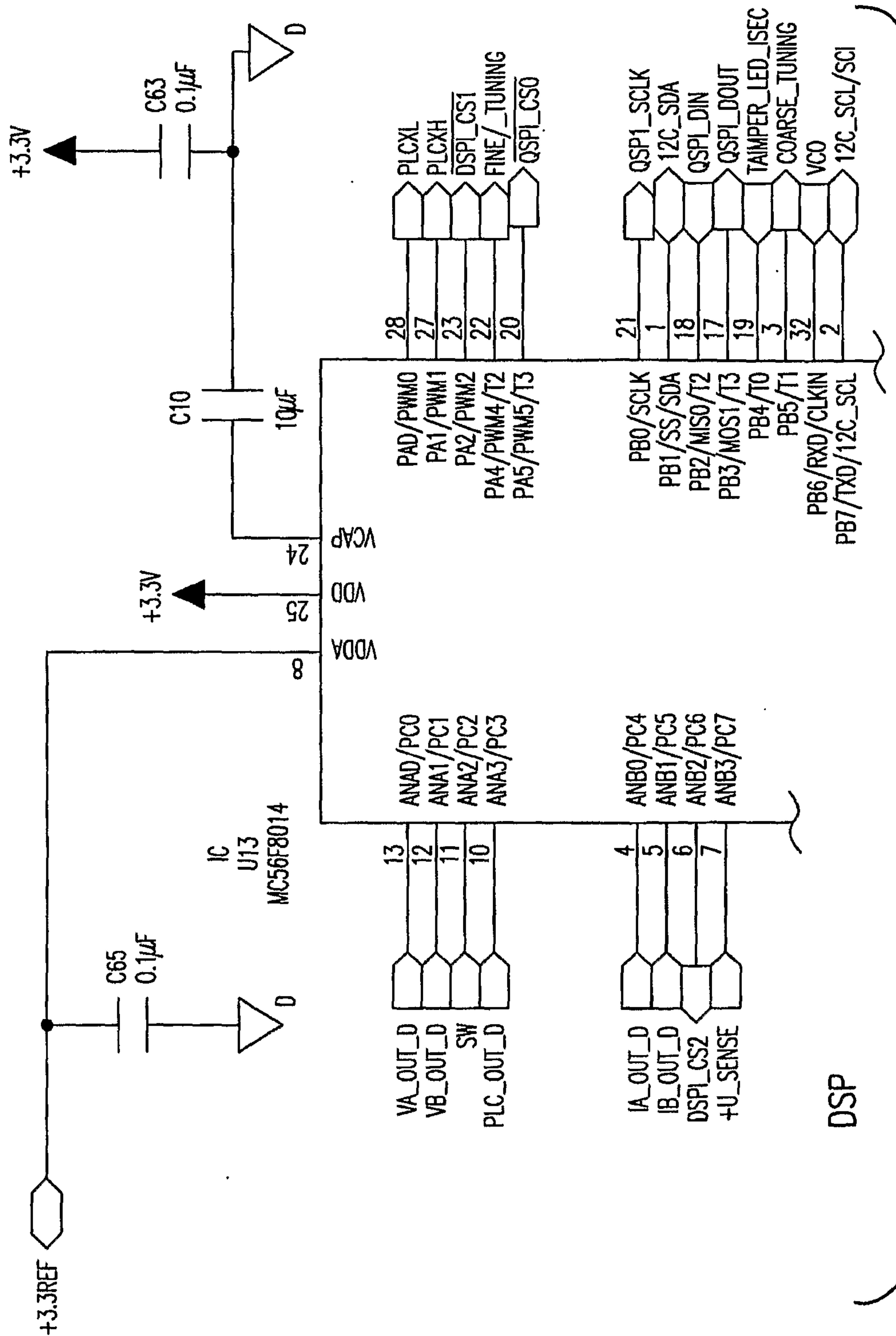
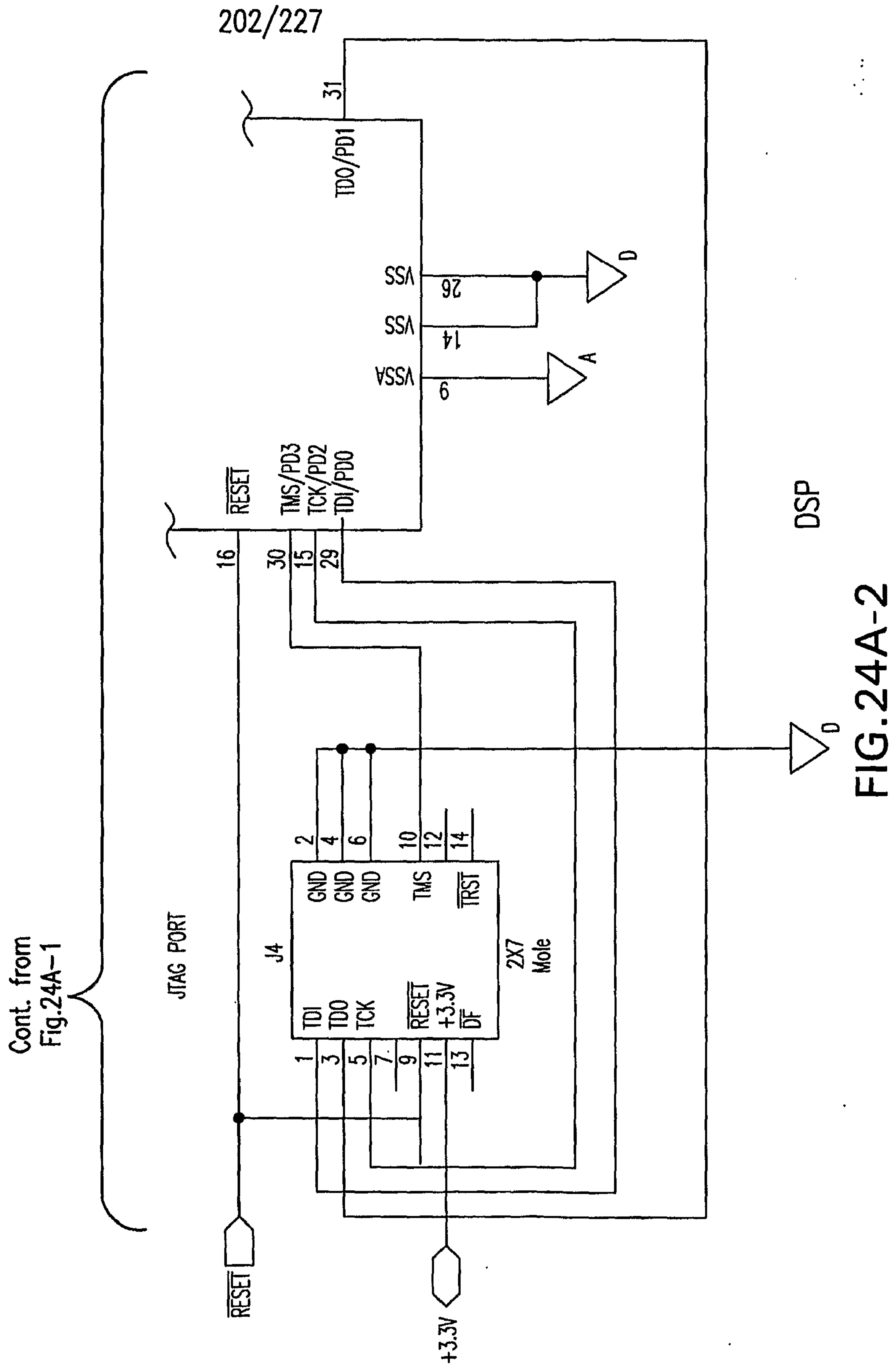


FIG. 24A-1

Cont. on
Fig. 24A-2



203/227

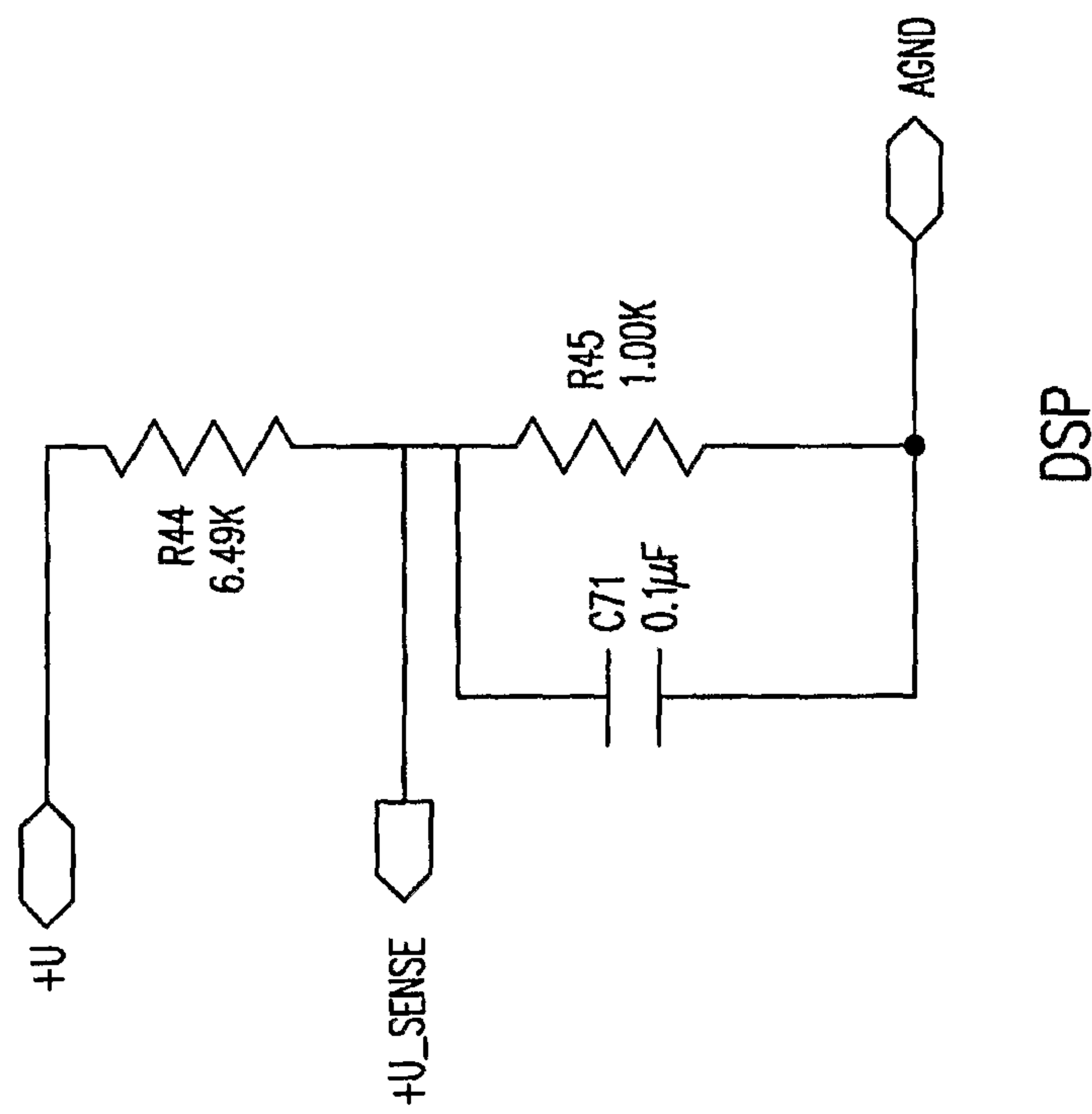


FIG.24A-3

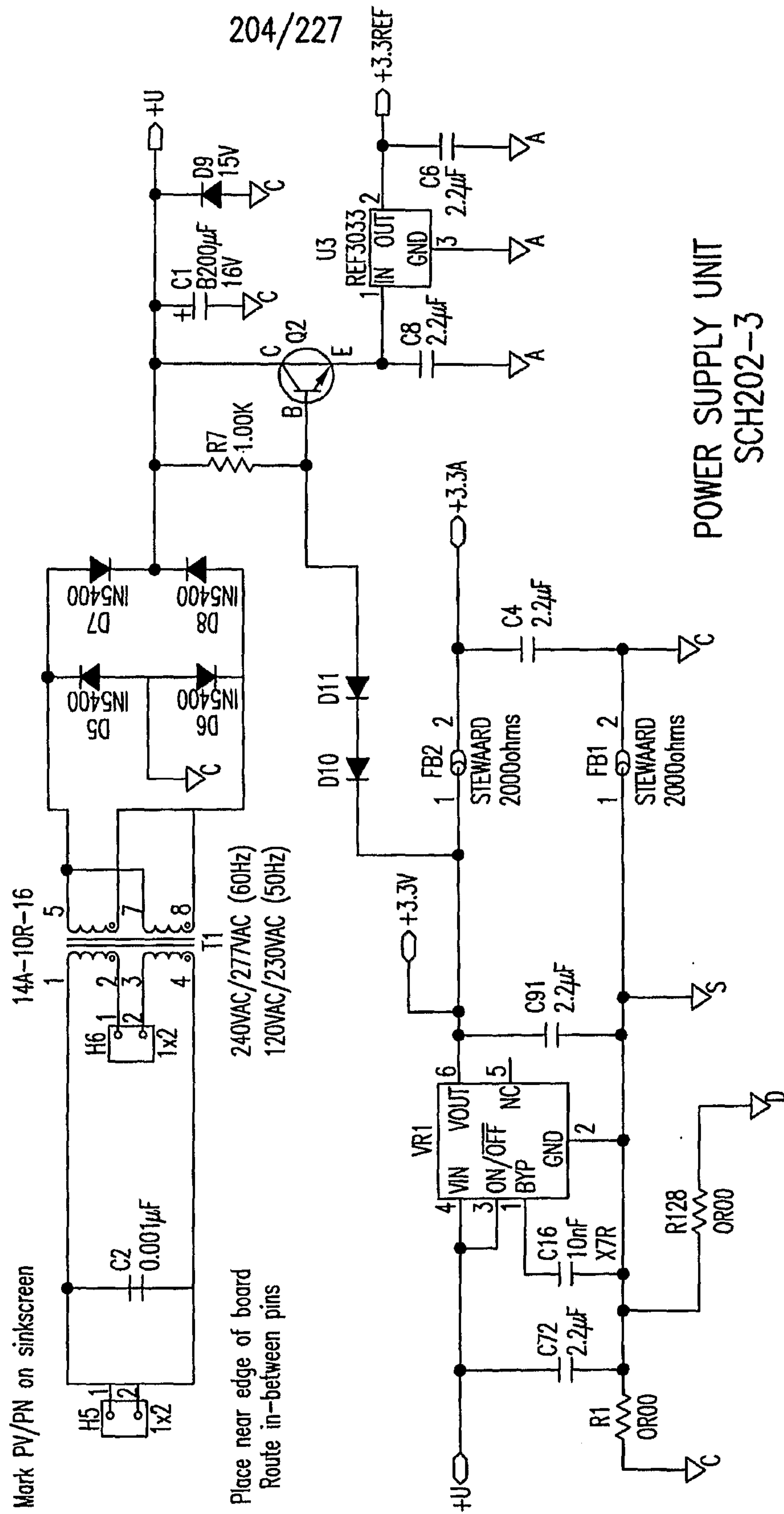
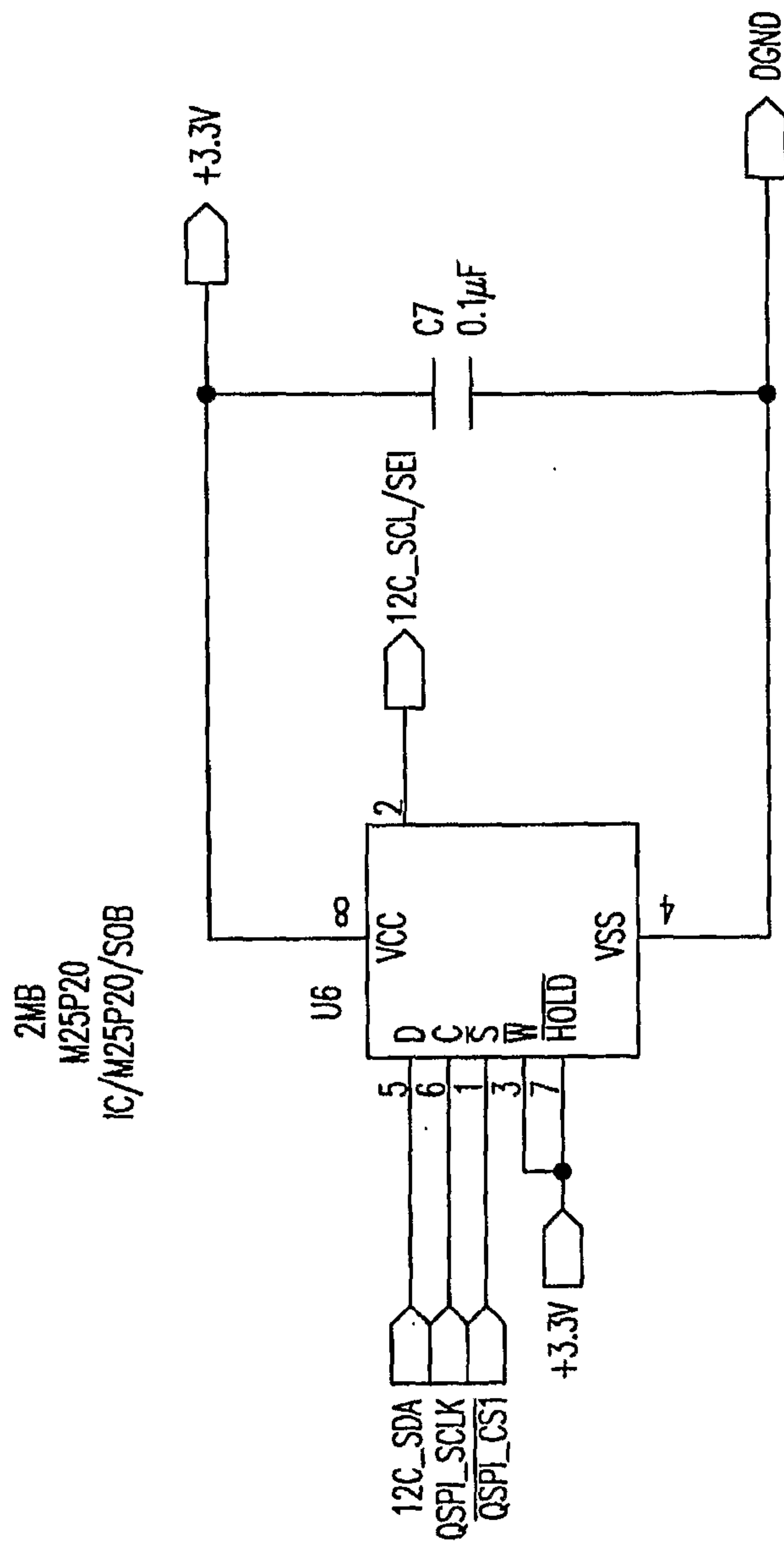


FIG. 24B

205/227



FLASH MEMORY

FIG.24C

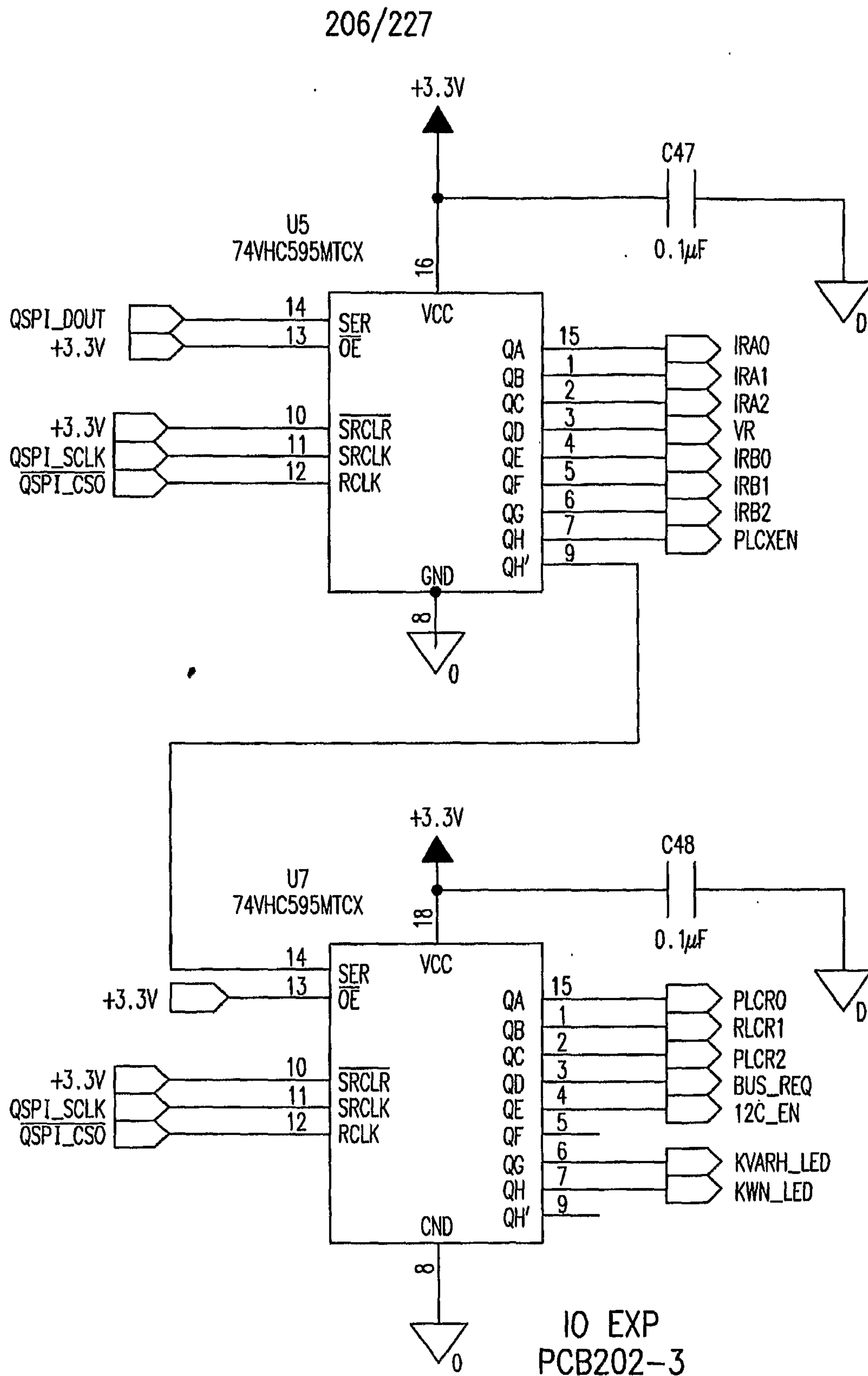


FIG.24D

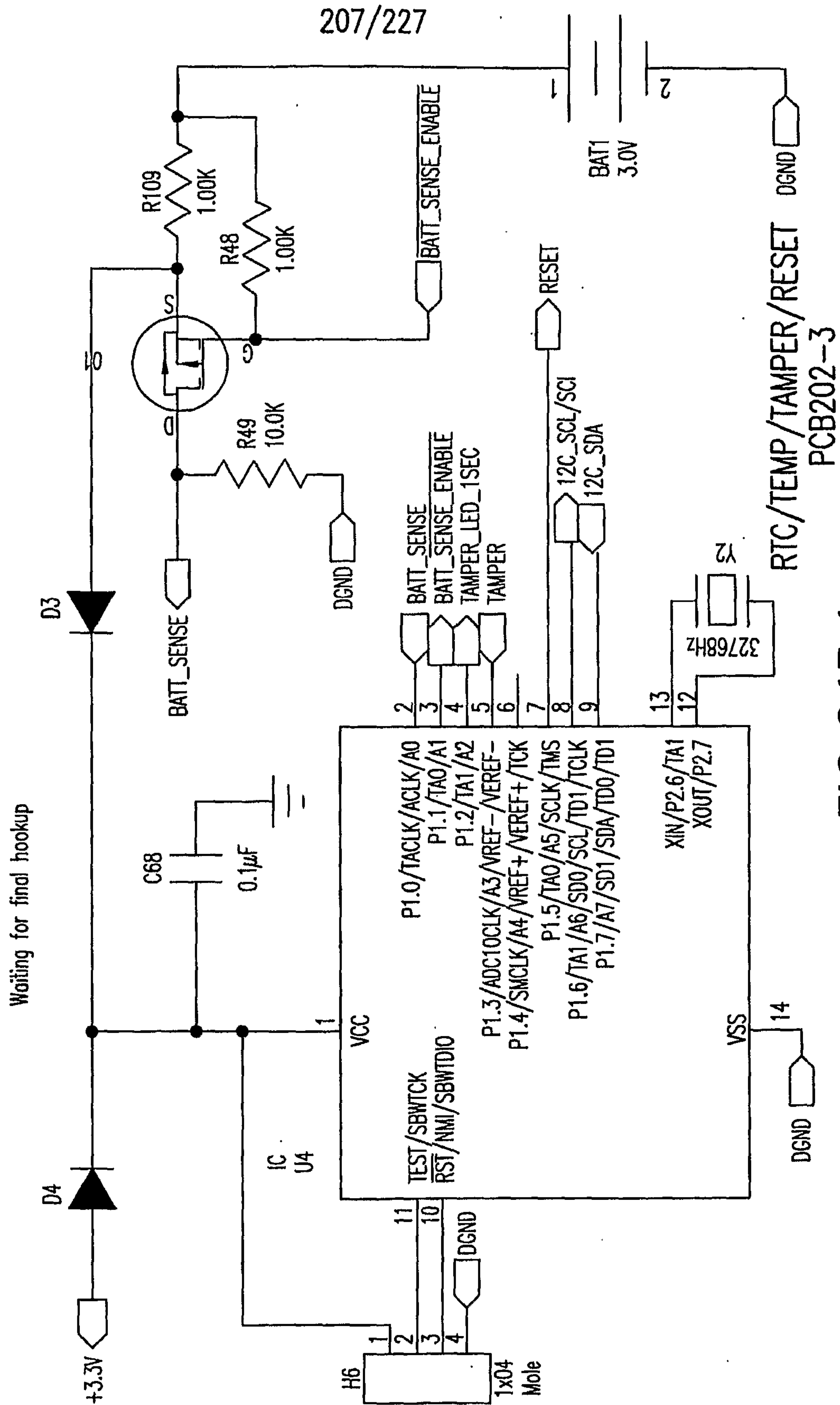


FIG. 24E-1

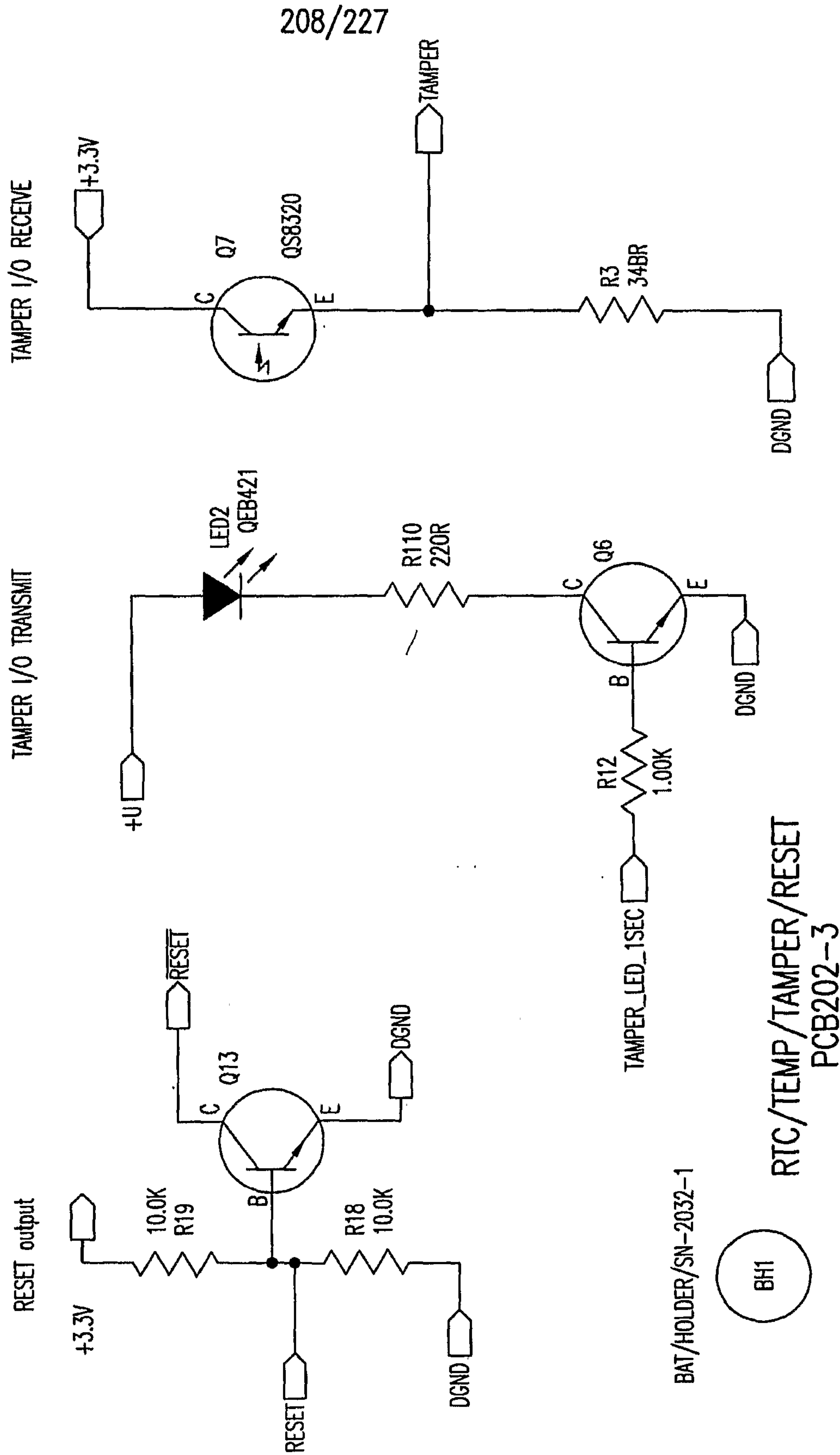


FIG. 24E-2

8 differential amplifiers
Shifts 1.0 – 2.0 (1.0 VPP around 1.5 REF)
To 5VPP around 2.5V (MSV)

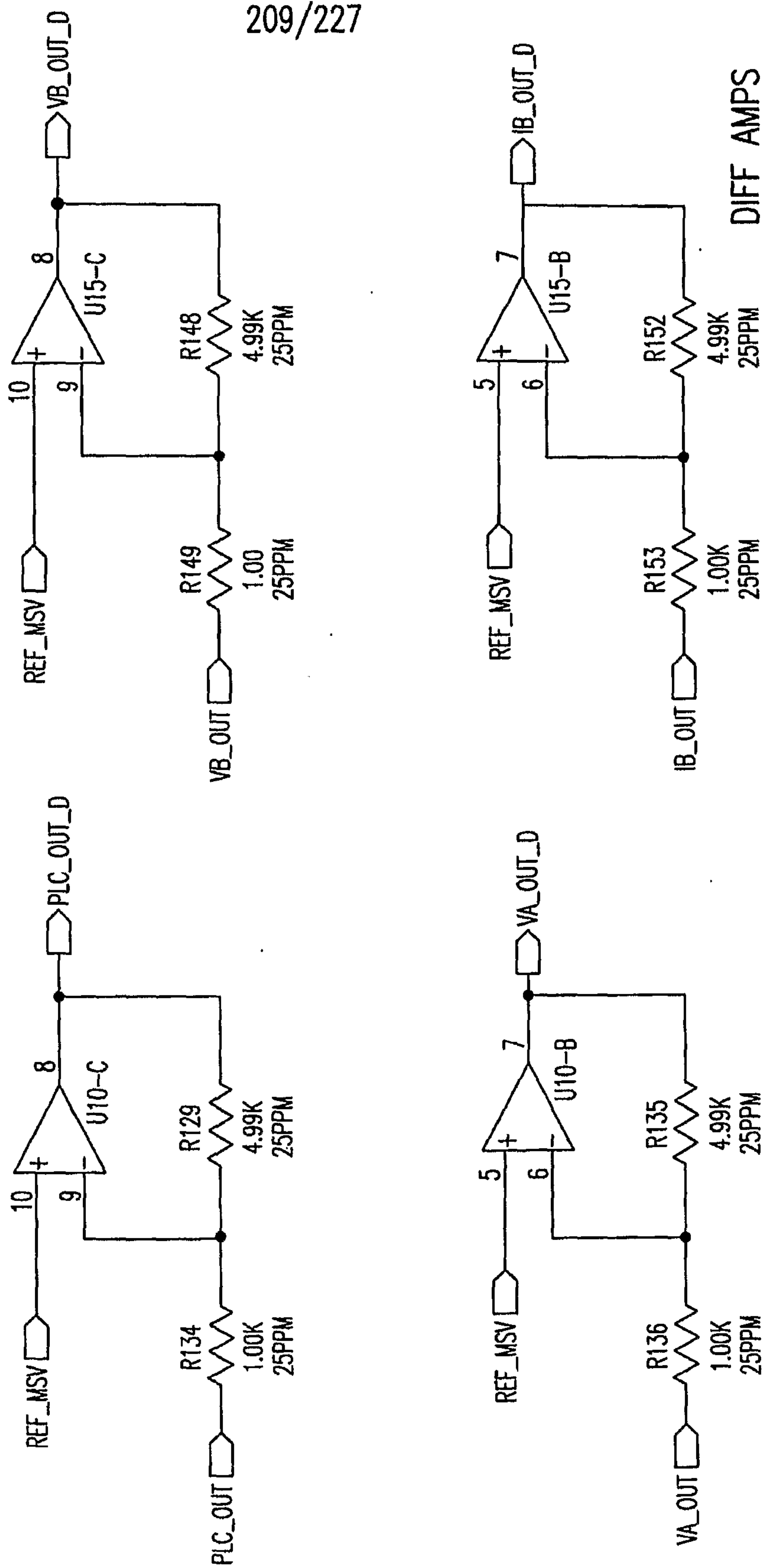


FIG.24F-1

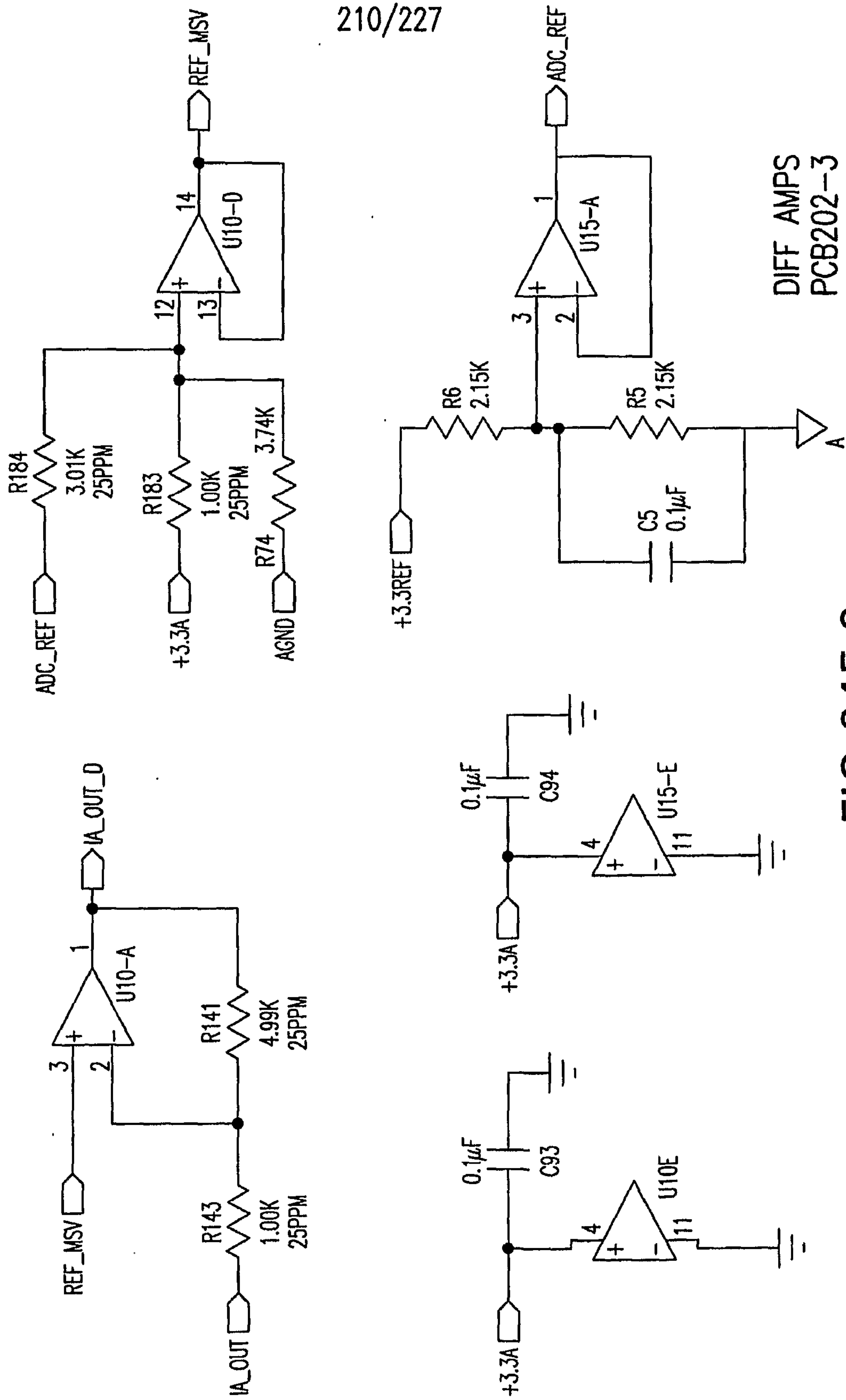


FIG.24F-2

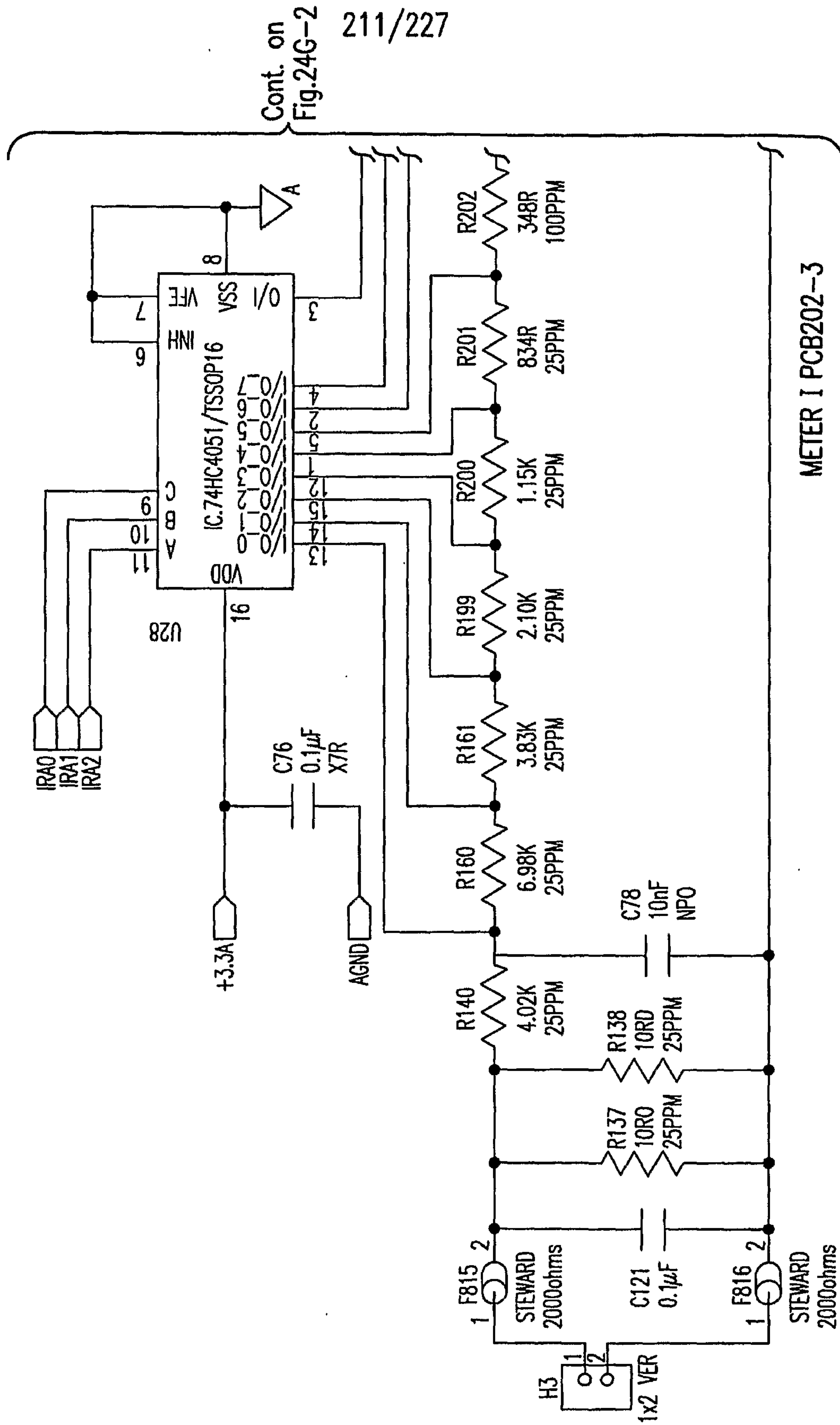
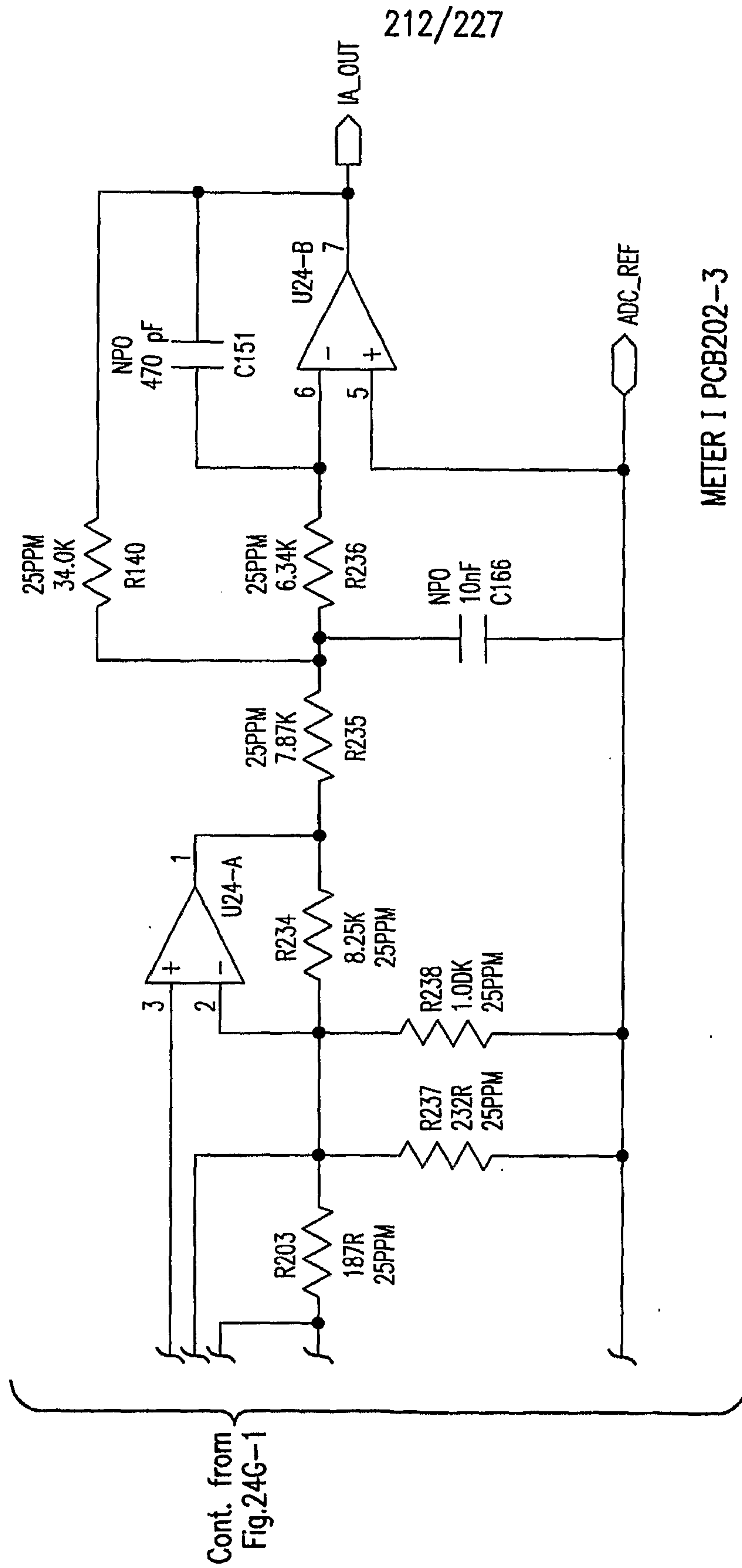
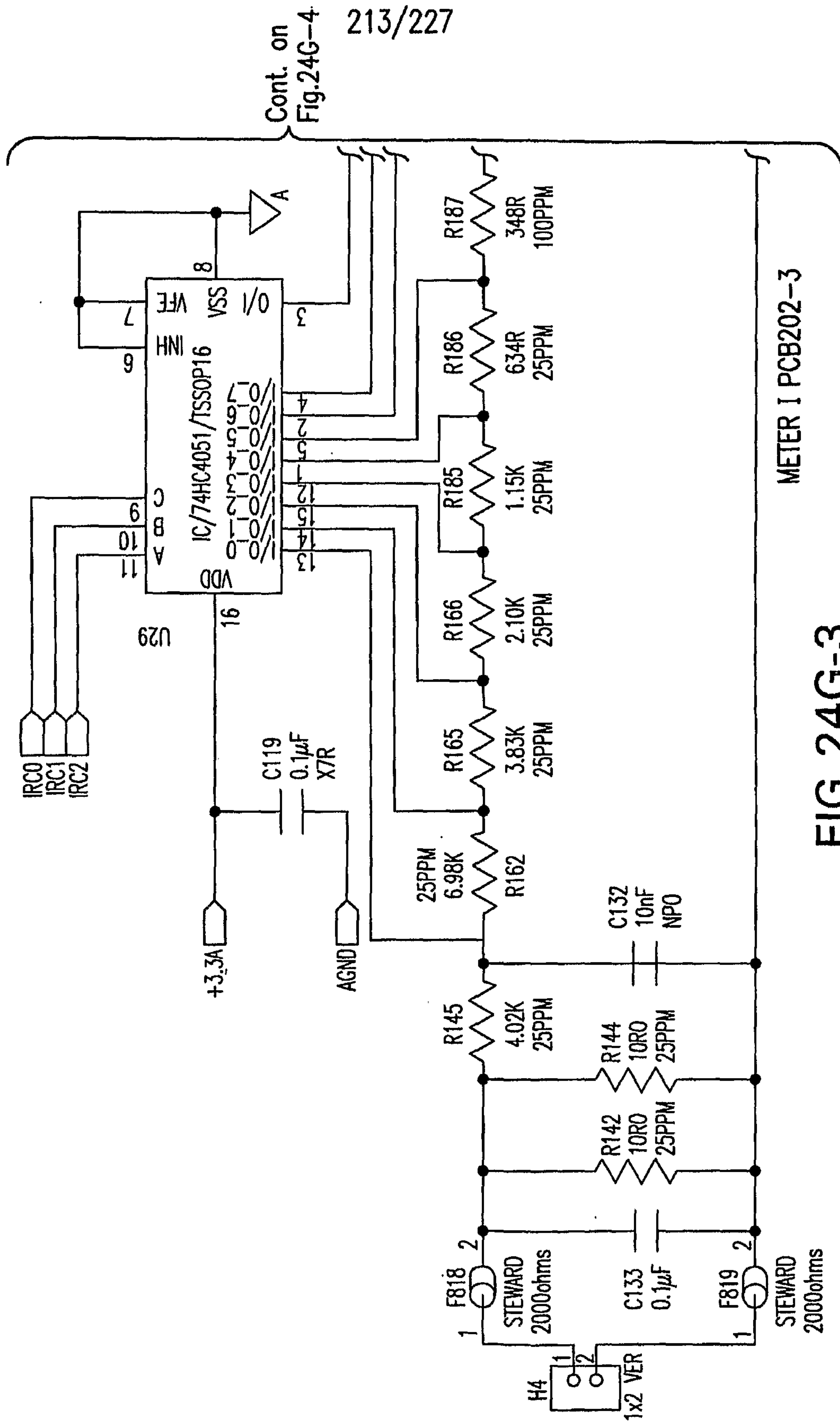
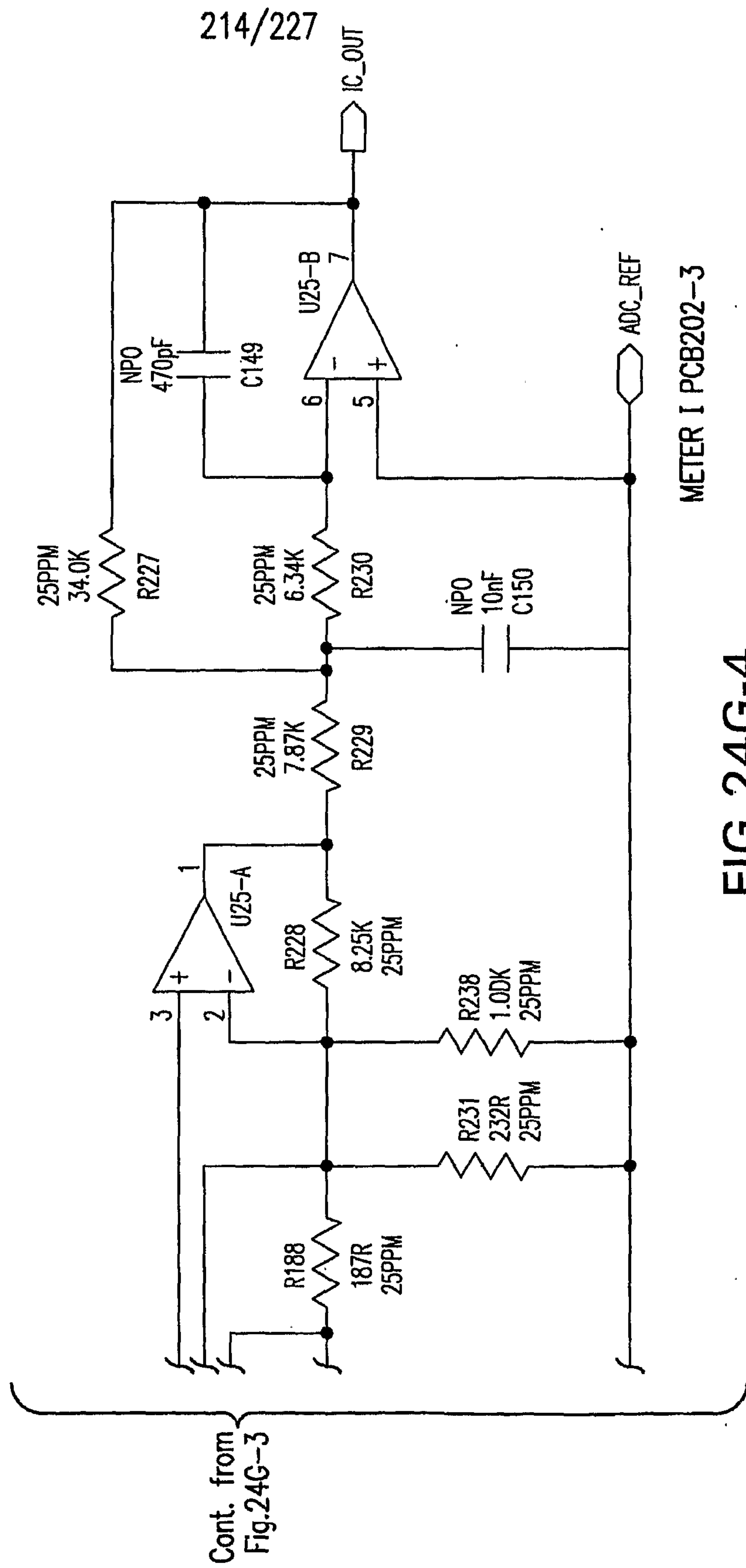


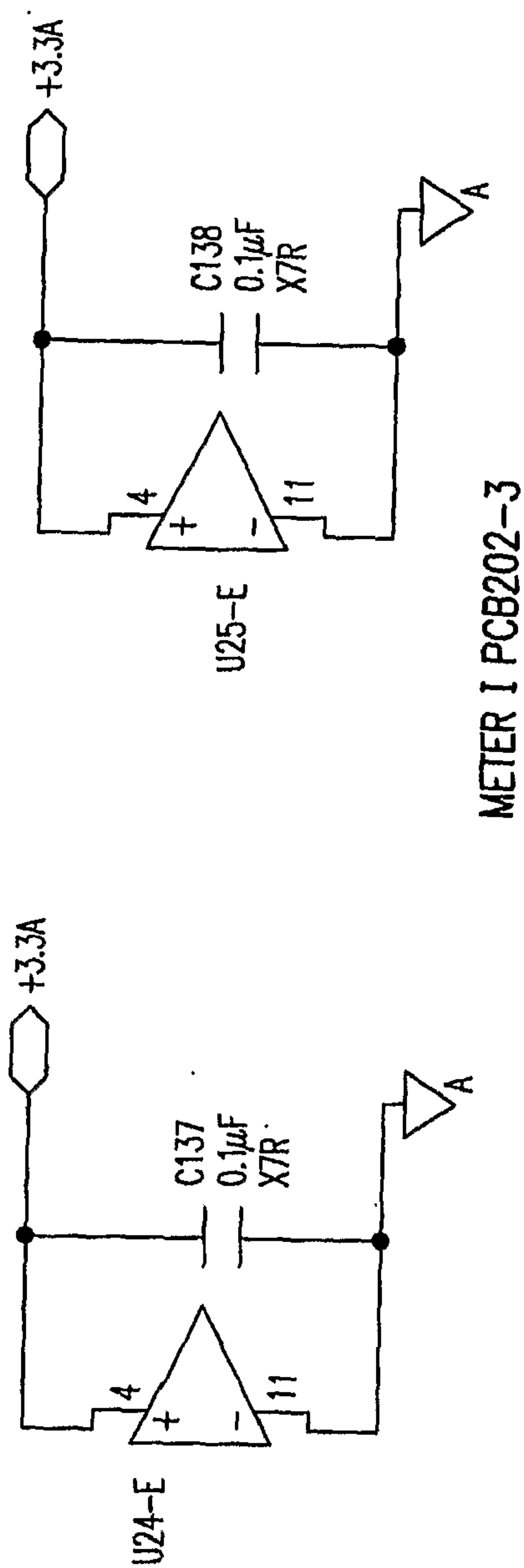
FIG. 24G-1







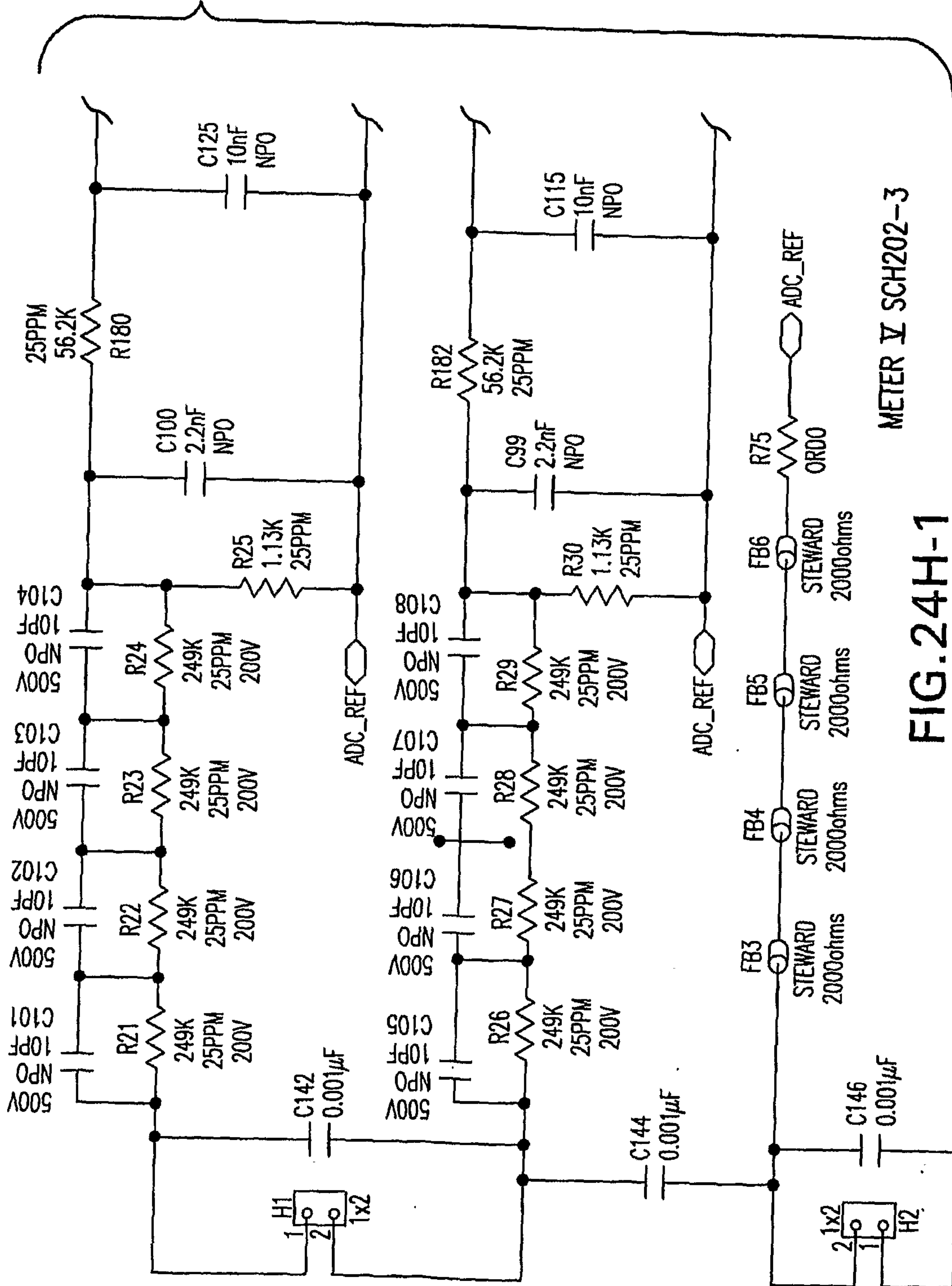
215/227



METER I PCB202-3

FIG. 24G-5

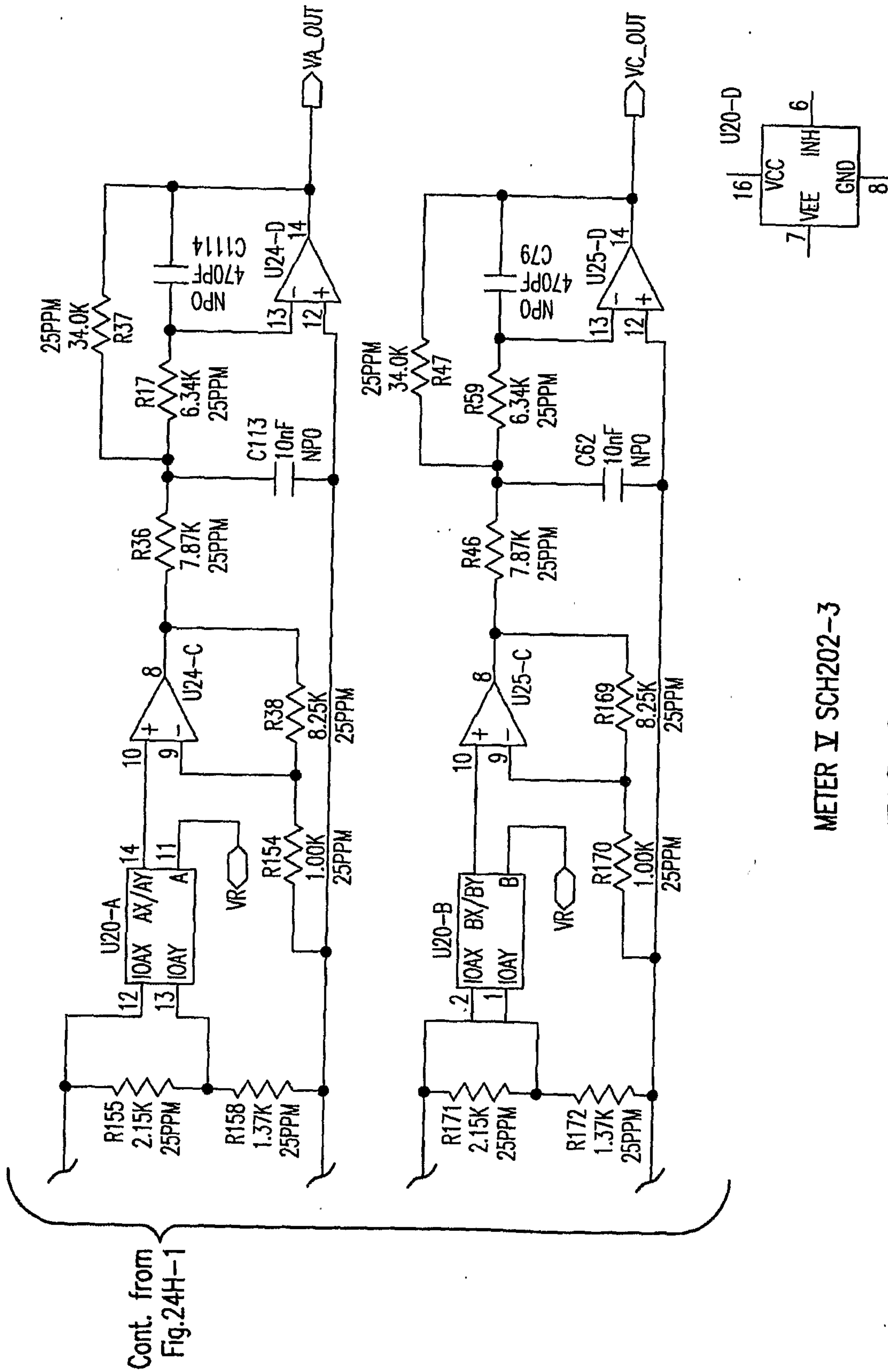
216/227

Cont. on
Fig. 24H-2

METER V SCH202-3

FIG. 24H-1

217/227



METER V SCH202-3

FIG.24H-2

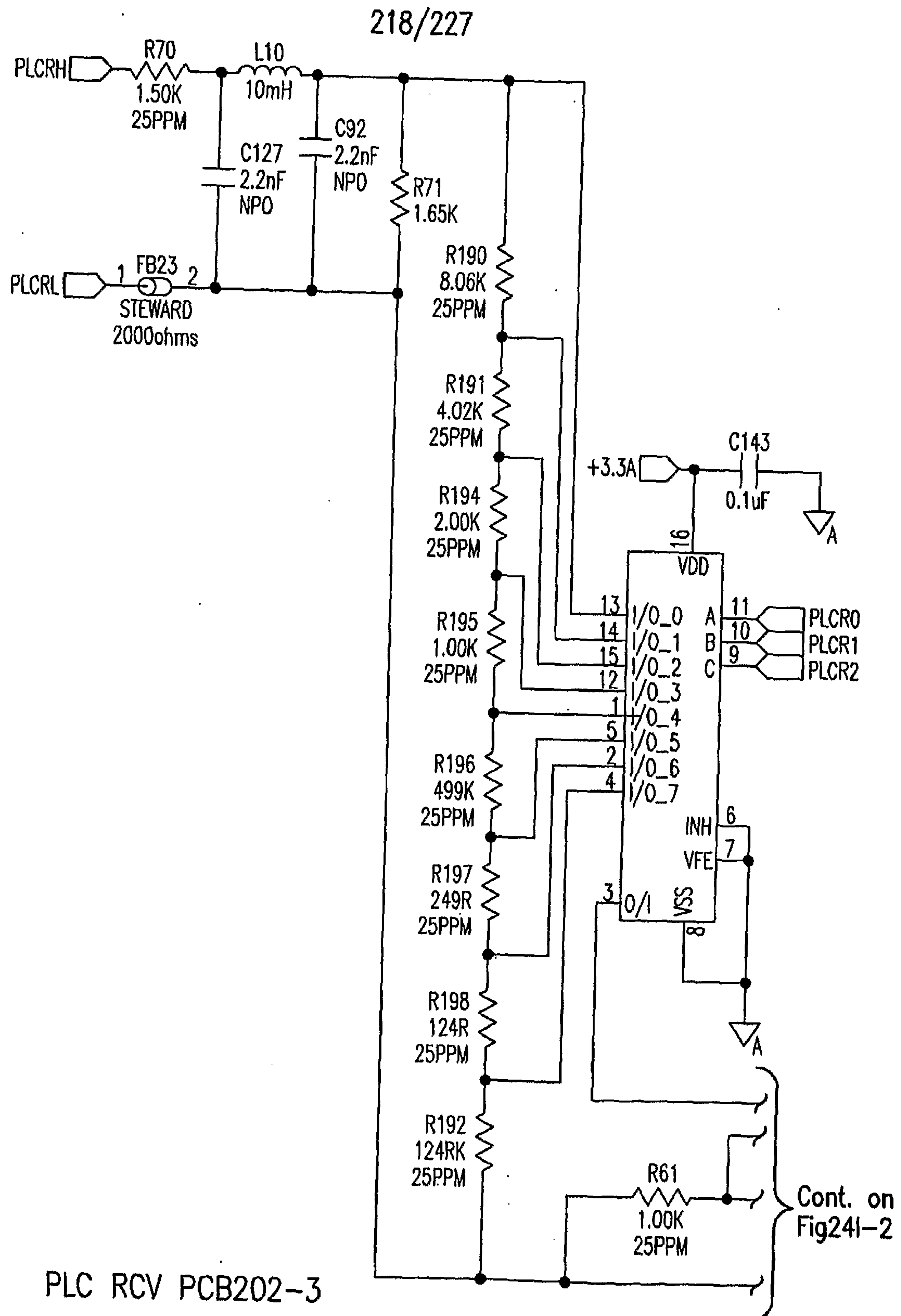
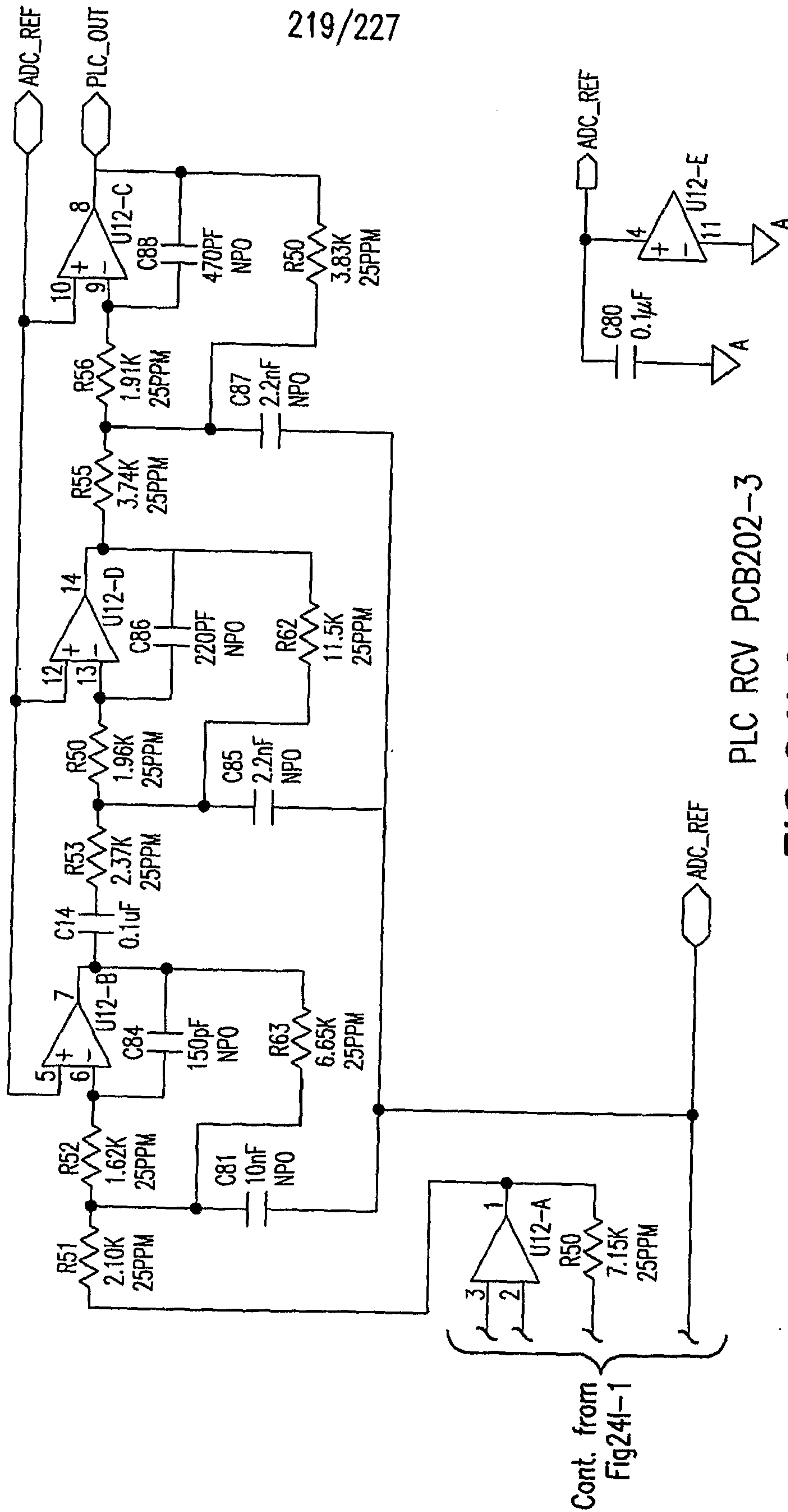


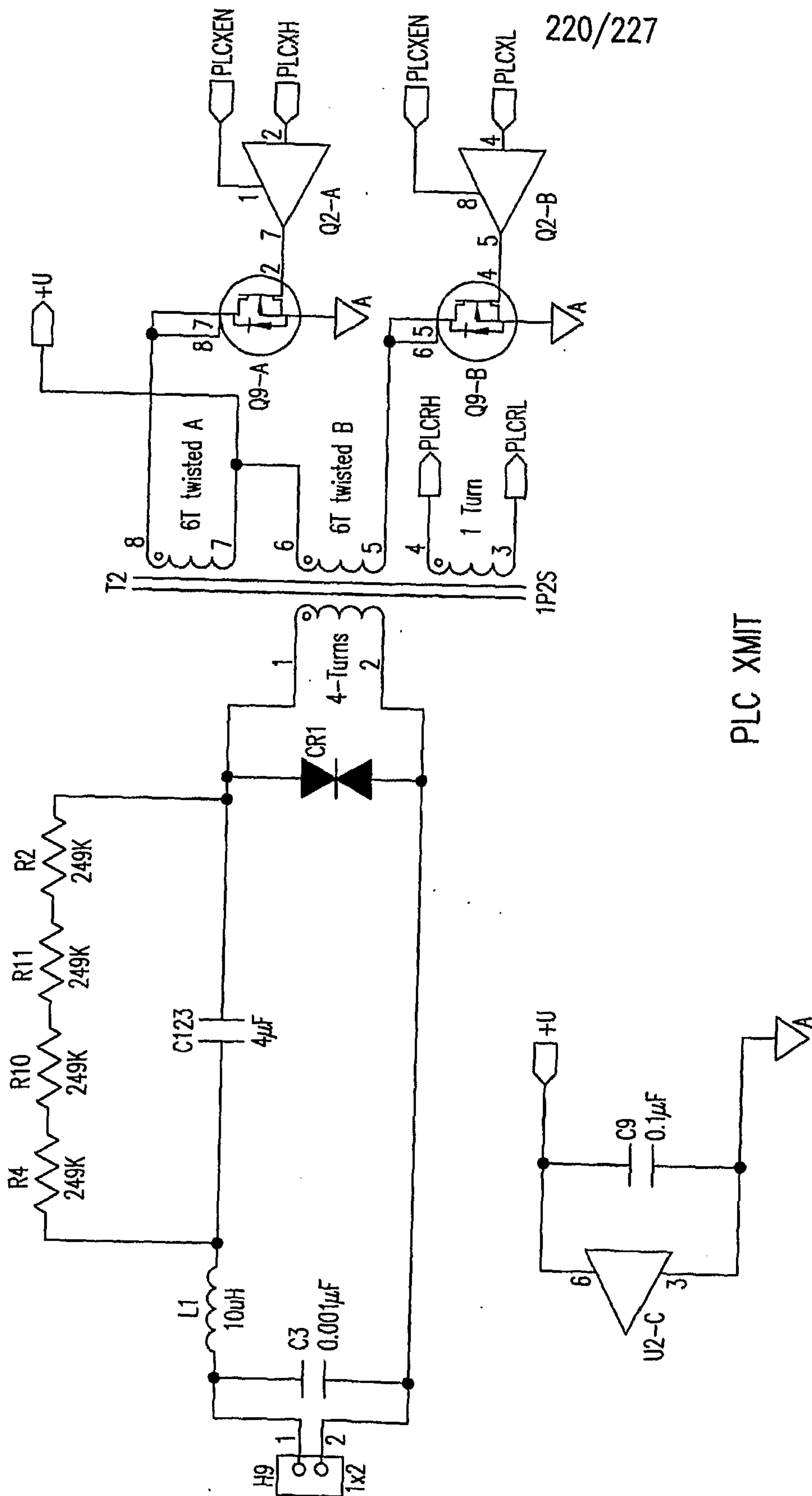
FIG.24I-1

219/227

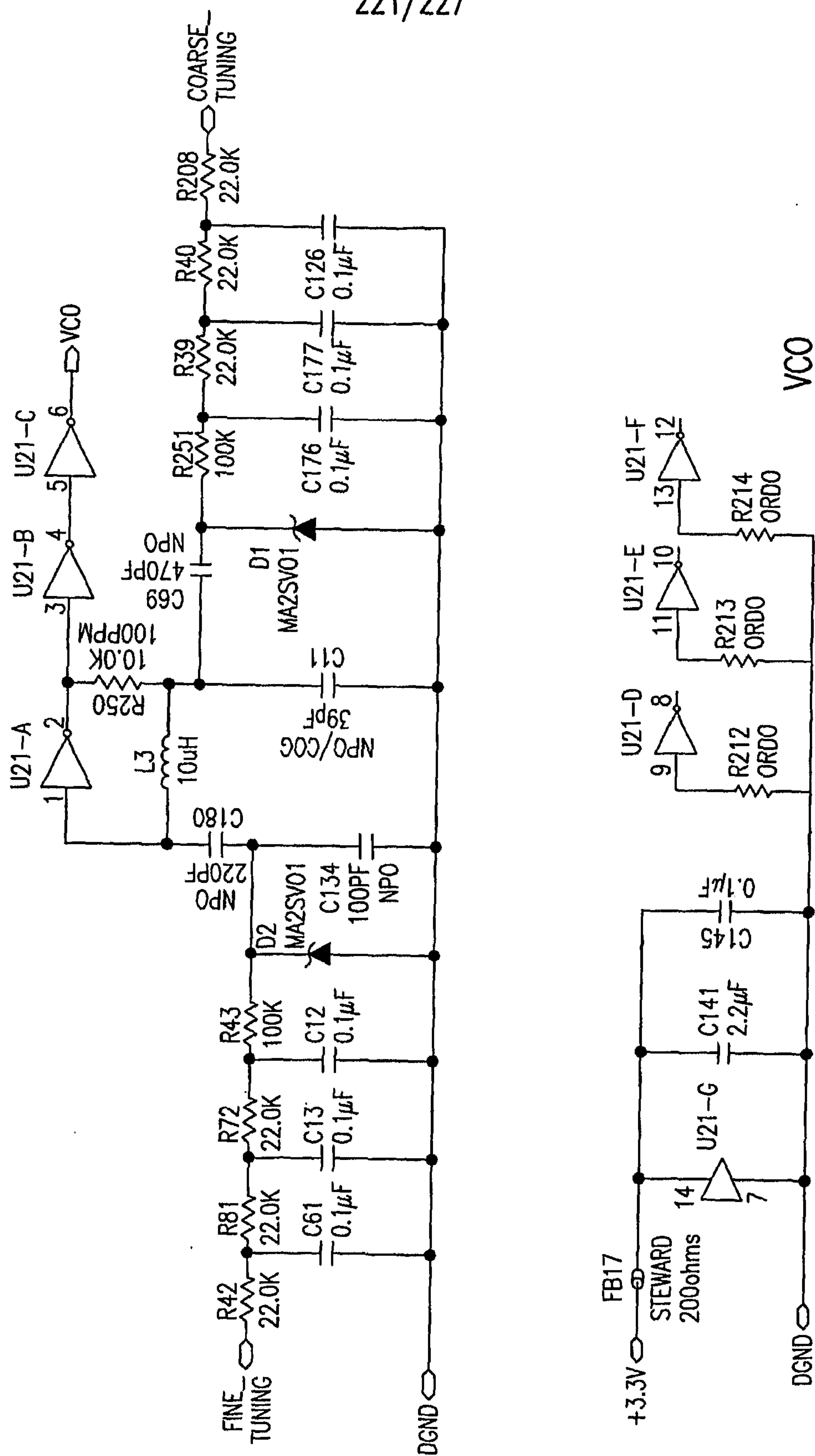


PLC RCV PCB202-3

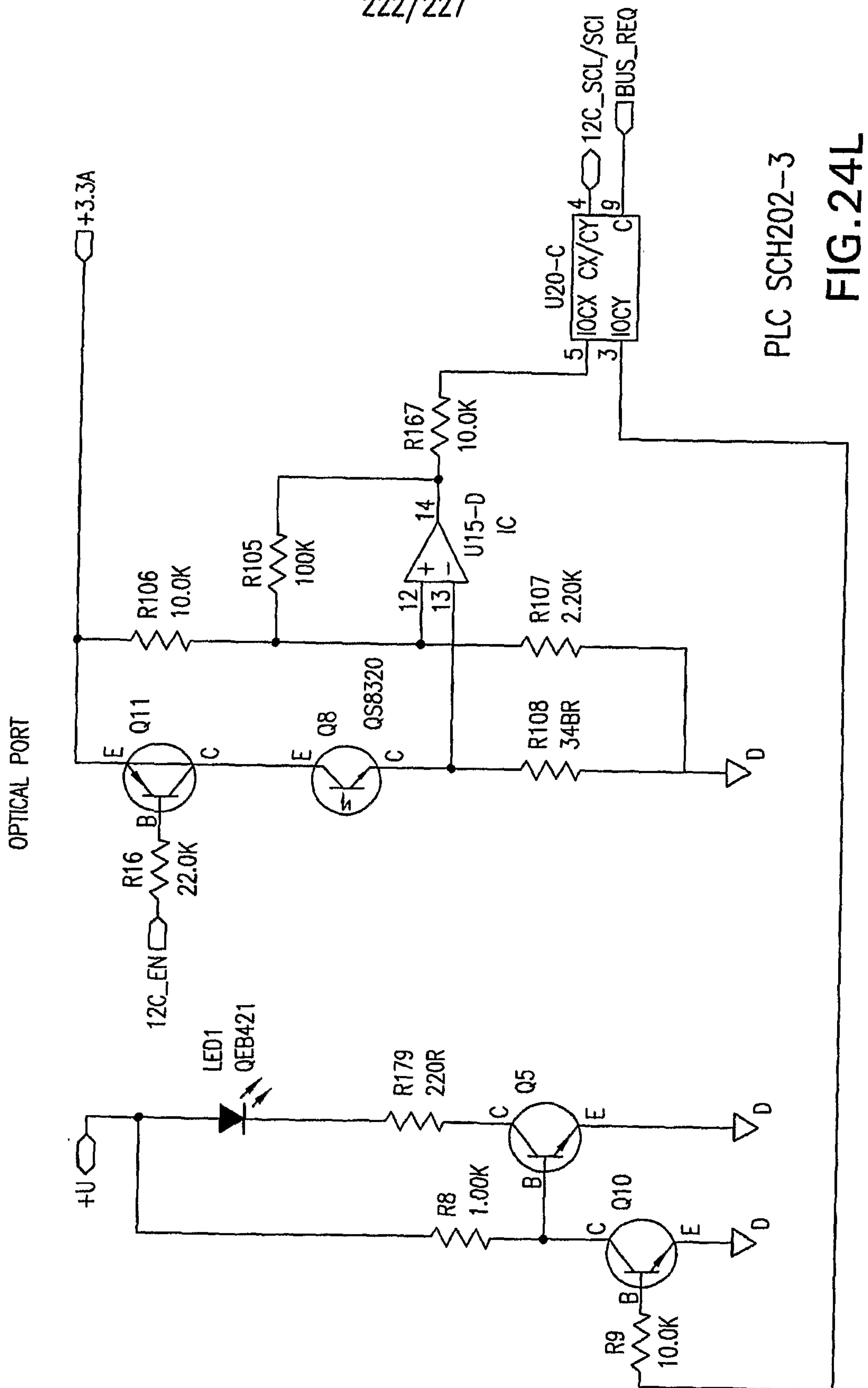
FIG.241-2



221/227



222/227



223/227

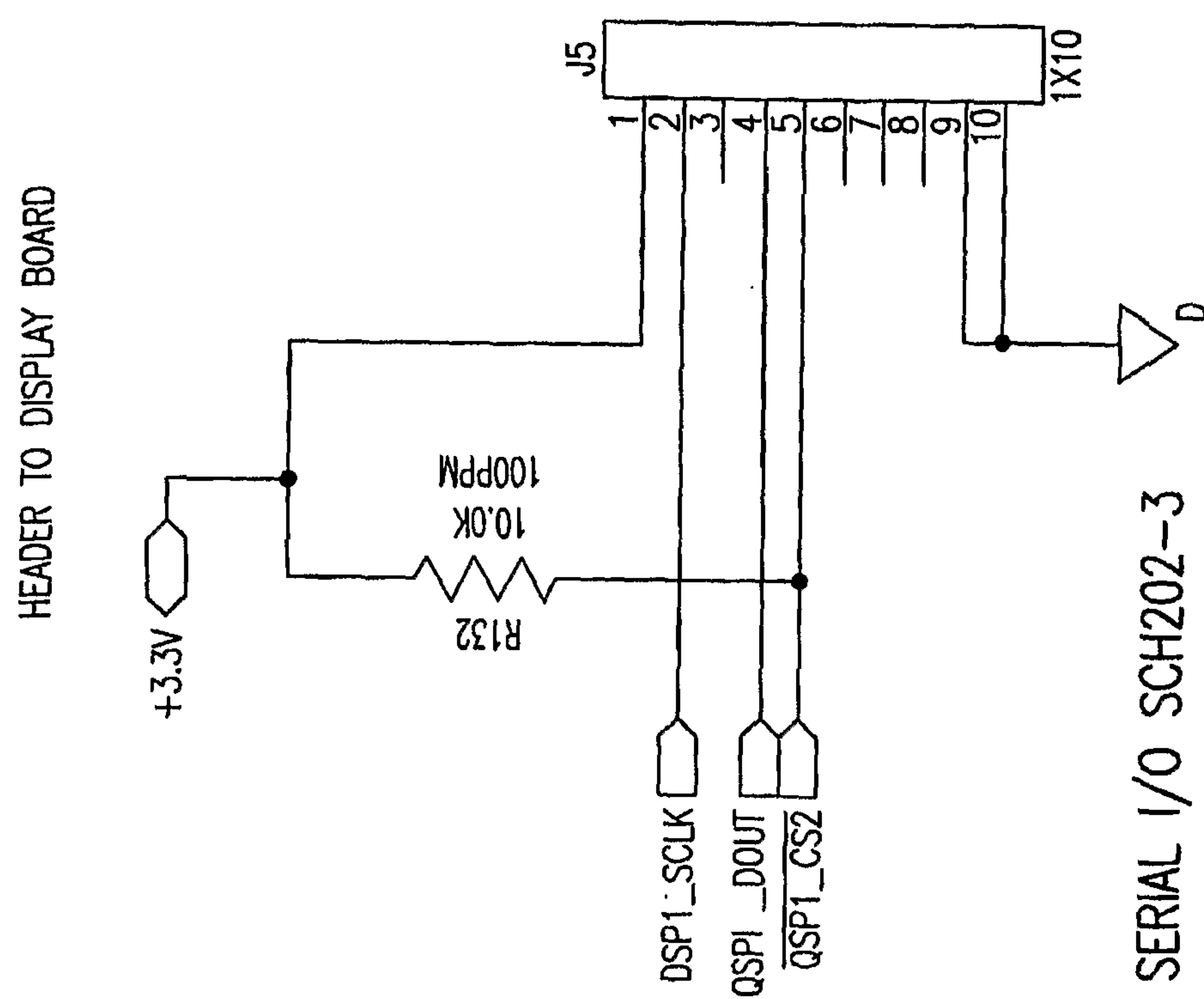


FIG. 24M

224/227

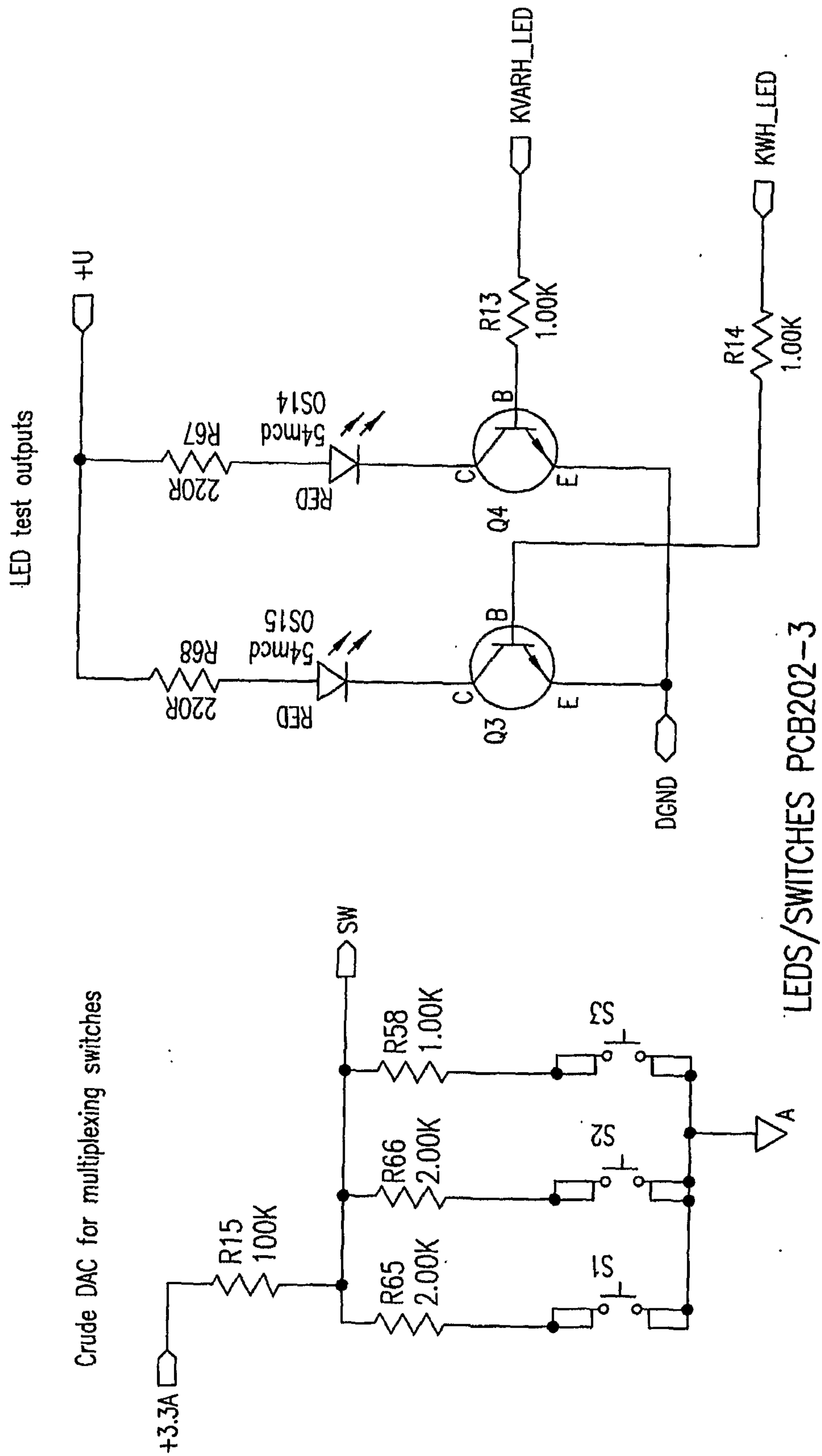
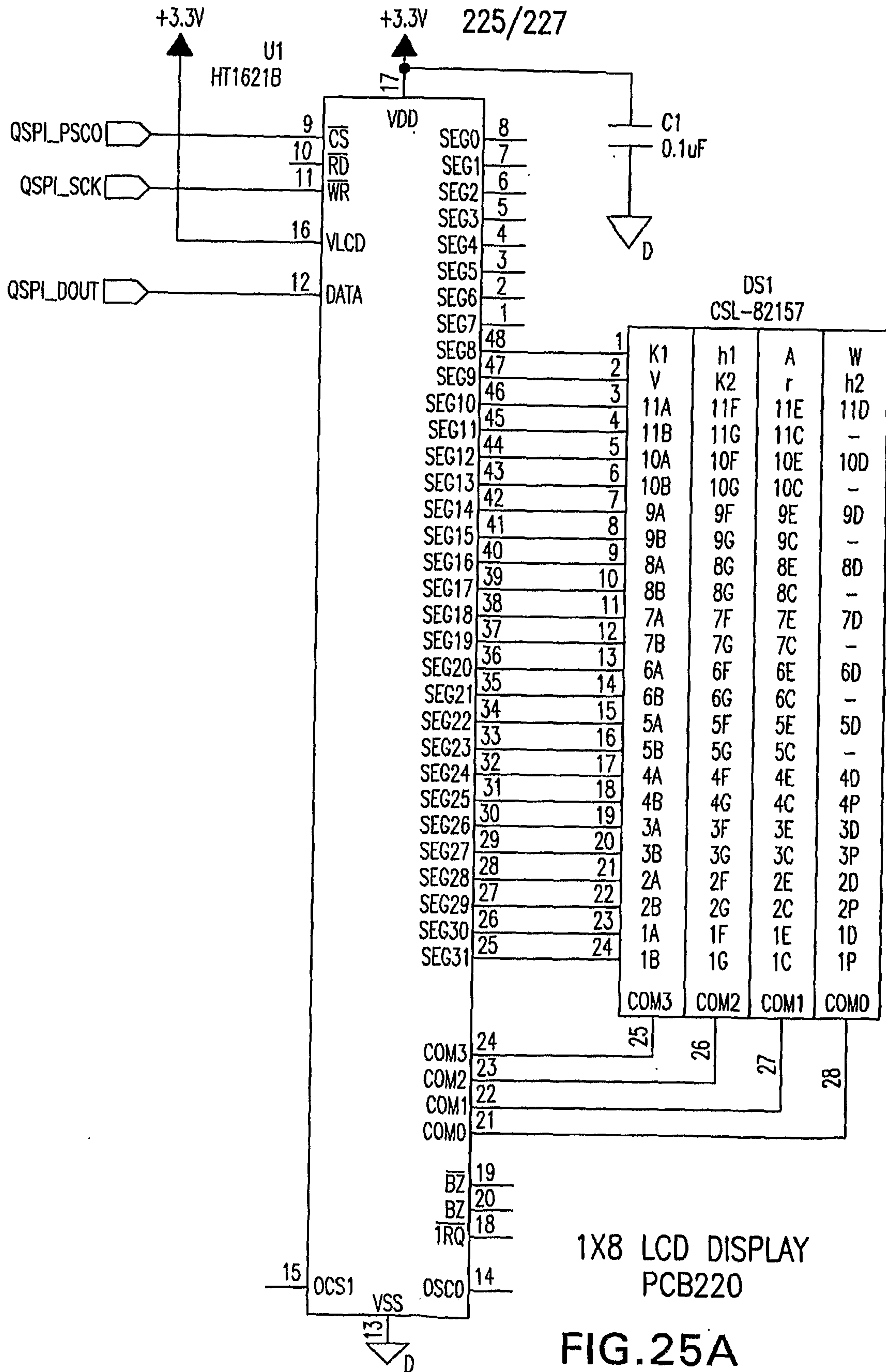


FIG. 24N



226/227

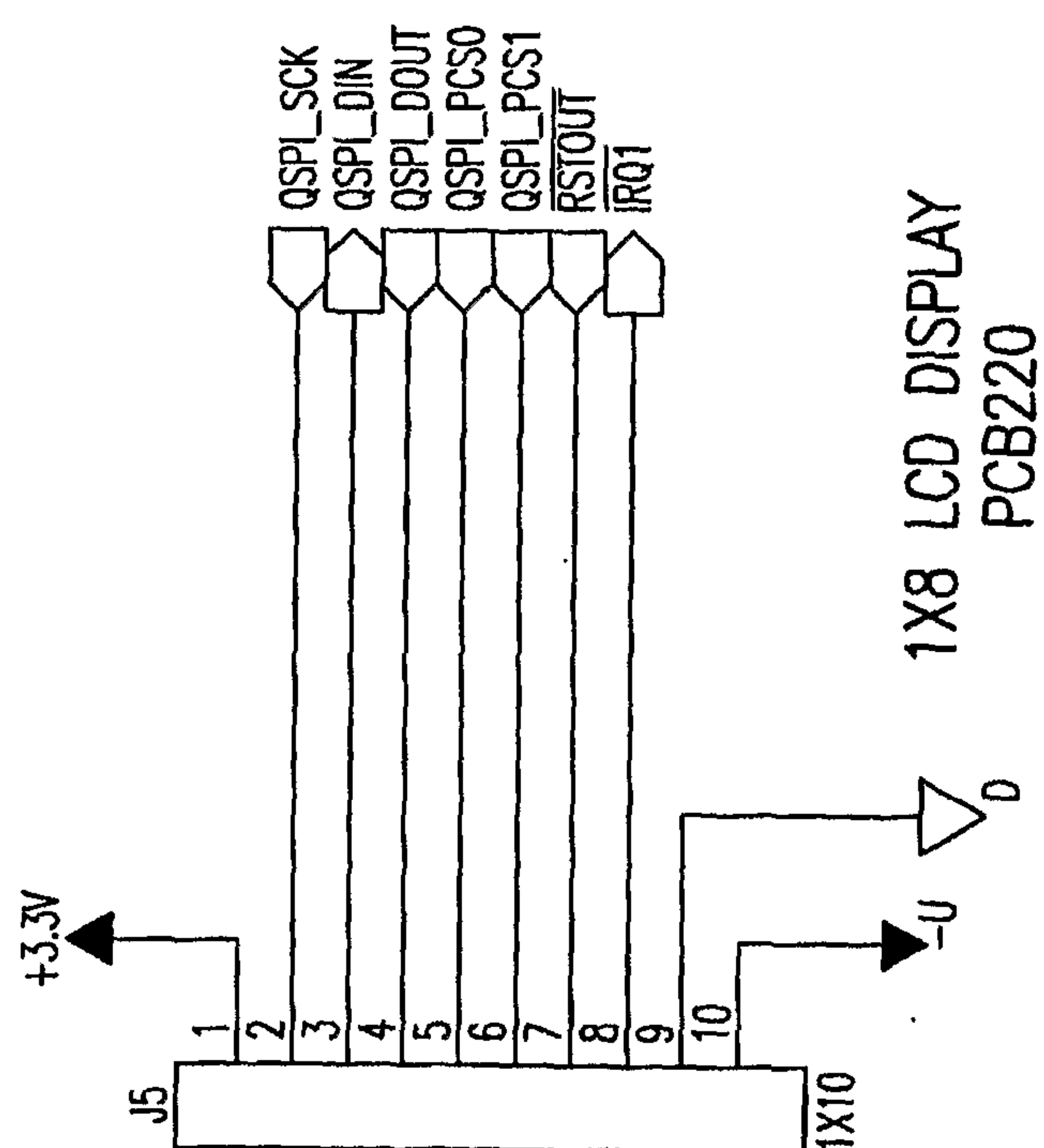


FIG. 25B

227/227

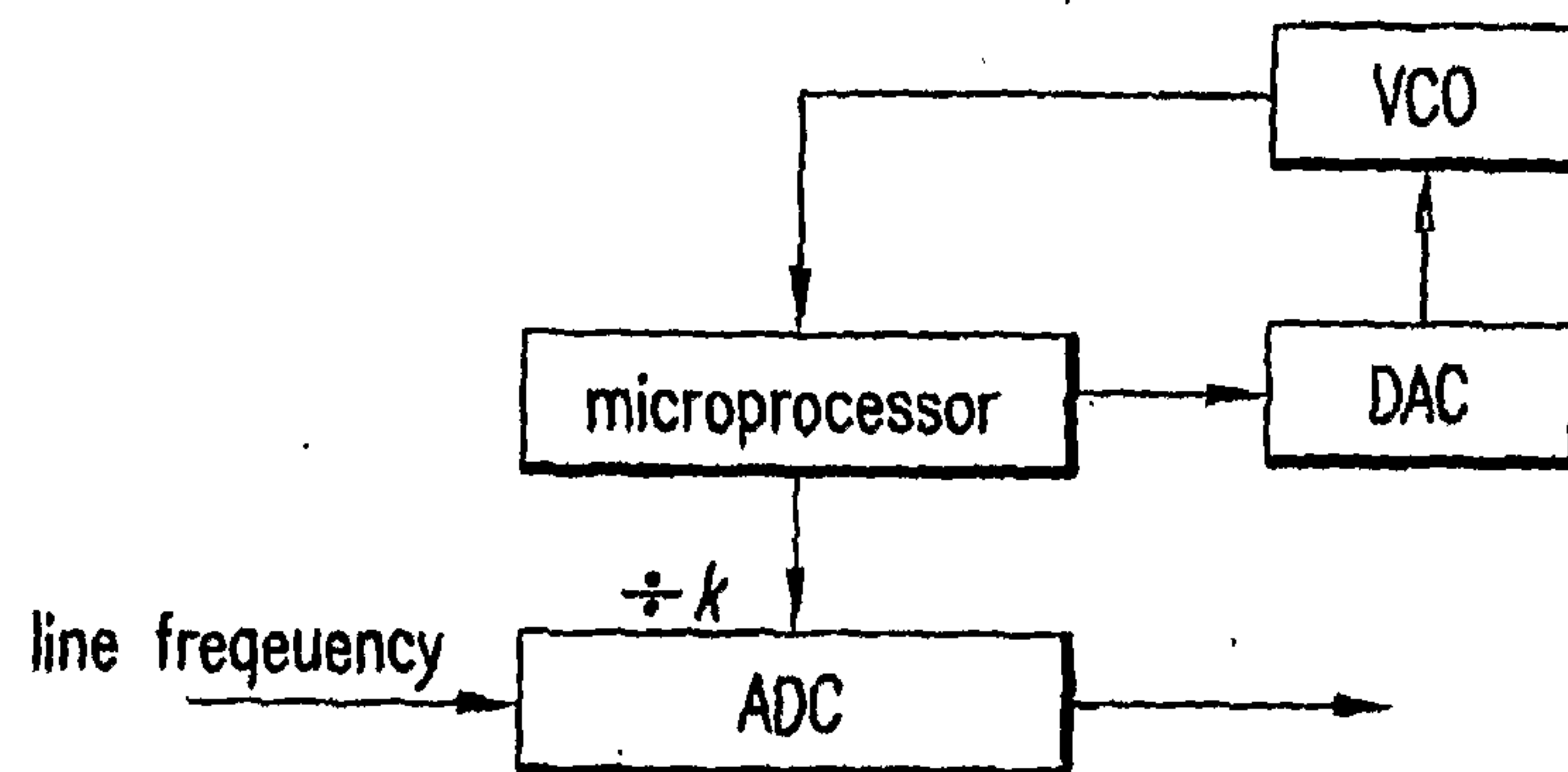


FIG.26

