República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(11) PI 0806143-2 B1

(22) Data do Depósito: 27/06/2008

(45) Data de Concessão: 30/04/2019

(54) Título: APARELHOS PARA MEDIÇÃO DE MULTI-CANAL DE ELETRICIDADE

(51) Int.Cl.: G01D 4/02; G01R 22/04; H04Q 9/02; H02J 13/00.

(73) Titular(es): QUADLOGIC CONTROLS CORPORATION.

(72) Inventor(es): SAYRE A. SWARZTRAUBER; DORON SHAFRIR; SIDDHARTH MALIK; ROBERT HAYWARD.

(57) Resumo: APARELHOS PARA MEDIÇÃO DE MULTI-CANAL DE ELETRICIDADE. Em um aspecto, a invenção inclui um dispositivo para medir uso de eletricidade, incluindo: meio para desconexão remota por comunicação de linha de potência; meio para detecção de roubo de eletricidade; meio para detecção de fraude; e meio para detecção de tensão inversa. Em outro aspecto, a invenção inclui um aparelho para medição de multi-canal de eletricidade, incluindo: (a) uma cabeça de medidor operável para medir uso de eletricidade para uma pluralidade de linhas de consumidor de eletricidade; (b) um transponder operável para transmitir dados recebidos da cabeça de medidor por comunicação de linha de potência para um computador localizado remotamente, e para transmitir dados recebidos por comunicação de linha de potência do computador localizado remotamente para a cabeça de medidor; e (c) um módulo de controle de carga operável para atuar conexão e desconexão de cada um de uma pluralidade de relés, cada relé da pluralidade de relés correspondendo a uma da pluralidade de linhas de consumidor de eletricidade.

"APARELHOS PARA MEDIÇÃO DE MULTI-CANAL DE ELETRICIDADE"

Referência Cruzada a Pedidos Relacionados

[0001] Este pedido reivindica o benefício de Pedido de Patente Provisório US N° 60/737.580, depositado em 15 de novembro de 2005, Pedido de Patente Provisório US N° 60/739.375, depositado em 23 de novembro de 2005, e Pedido Provisório N° 60/813.901, depositado em 15 de junho de 2006, e é uma continuação em parte de Pedido de Patente US N° 11/431.849, depositado em 9 de maio de 2006, que é uma parte de Pedido de Patente US N° 11/030.417, depositado em 6 de janeiro de 2005 (agora Patente No. 7.054.770), que é uma parte de Pedido de Patente US N° 09/795.838, depositado em 28 de fevereiro de 2001 (agora Patente US N° 6.947.854). Os conteúdos inteiros de cada um desses pedidos estão incorporados aqui por referência.

Fundamento e Resumo

[0002] Uma concretização da presente invenção inclui um dispositivo de medição que está relacionado à família de medidores baseada em ASIC Quadlogic (veja Patente US N° 6.947.854, e Publicação de Pedido de Patente US N° 20060036388, os conteúdos inteiros de quais estão incorporados aqui por referência). Especificamente, esta concretização (referida aqui para conveniência como "Guarda de Energia") é um medidor de multi-canal que preferivelmente é capaz de prover muita da funcionalidade da família supracitada de medidores, e ademais provê as melhorias, características e componentes listados abaixo.

[0003] Usado em pelo menos uma concretização, um 'MiniCloset' é um dispositivo de medição de 24 canais que pode medir uso elétrico para até 24 clientes monofásicos, 12 clientes bifásicos, ou 8 clientes trifásicos. Preferivelmente conectado ao MiniCloset estão um ou mais Módulos de Controle de Carga (LCMs), discutidos abaixo.

[0004] Guarda de energia inclui preferivelmente um módulo de cabeça de medidor de MiniCloset e dois LCMs montados em uma caixa de aço. Relés que permitem para um cliente de eletricidade ser remotamente desconectado e reconectado, junto com transformadores de corrente, também estão montados na

caixa. Veja a Figura 1.

[0005] Na instalação, a linha de provisão de eletricidade de um cliente de eletricidade é derivada do alimentador elétrico principal, passado pelo aparelho de Guarda de Energia, e corre diretamente à casa do cliente. A construção e uso do Guarda de Energia serão aparentes àqueles qualificados na arte na revisão da descrição abaixo e figuras relacionadas. Código fonte é provido no Apêndice anexo.

[0006] Medidores de Guarda de Energia preferivelmente são operáveis para prover:

(A) Desconexão/Reconexão Remota: O medidor suporta comunicação duplex total (bidirecional) por comunicação de linha de potência ("PLC") e pode ser equipado com relés operados remotamente (60 A, 100 A, ou 200 A) que permitem desconexão e reconexão de usuários elétricos remotamente.

(B) Prevenção de Roubo: O sistema é projetado com três características específicas para prevenir roubo. Primeira, um aparelho de Guarda de Energia preferivelmente é instalado em um poste de utilidade acima das linhas de média tensão, fazendo difícil para clientes alcançarem e fraudarem. Segunda, porque não há nenhum fio de sinal adicional com o sistema (isto é, toda a comunicação é pela linha de potência), quaisquer fios de comunicação cortados são imediatamente detectáveis. Quer dizer, se um fio de comunicação for cortado, o serviço é cortado, que é prontamente aparente. Uma terceira característica de prevenção de roubo é que o medidor pode ser usado para medir a energia de transformador a fim de validar os totais medidos de clientes individuais. Discrepâncias podem indicar roubo de energia.

(C) Detecção de Fraude: O Guarda de Energia preferivelmente provê dois modos de detecção de fraude óptica. Cada unidade contém uma luz que reflete contra um pequeno adesivo como espelho. A ausência desta luz reflexiva indica que a caixa foi aberta. Esta detecção desconectará automaticamente todos os clientes medidos por aquela unidade de Guarda de Energia. Além disso, se o invólucro de Guarda de Energia for aberto e luz ambiente entrar, isto também desconectará automaticamente todos os clientes medidos por aquela unidade de

Guarda de Energia. Estes dois modos de detecção de fraude são comprometidos continuamente e múltiplas vezes alternadas por segundo para segurança máxima.

(D) Detecção de Tensão Inversa: Em alguns casos, uma companhia de utilidade pode desconectar energia a um cliente individual e esse cliente é capaz de obter energia por uma alimentação alternativa. Se a utilidade fosse reconectar energia sob estas condições, dano poderia ocorrer ao equipamento de medição e/ou ao sistema de distribuição. Guarda de energia preferivelmente é capaz de detectar esta condição de falha. O Guarda de Energia pode detectar qualquer tensão que realimenta na desconexão aberta pelas linhas que conectam às instalações dos clientes. Se tensão for detectada, o firmware do Guarda de Energia prevenirá automaticamente a reconexão.

(E) Pré-pagamento: Pré-pagamento por energia pode ser feito por telefone, transação eletrônica, ou pessoalmente. A quantidade de kWh comprada é transmitida ao medidor e armazenada em sua memória. O medidor contará abaixo, mostrando quanta energia ainda está disponível antes de alcançar zero e desconectar. Contanto que o cliente continue comprando energia, não haverá nenhuma interrupção no serviço, e a companhia de utilidade terá um relatório de atividade diário.

(F) Limitação de Carga: Como uma alternativa à desconexão por não pagamento ou parte de um sistema de pré-pagamento, medidores de Guarda de Energia podem permitir a utilidade limitar remotamente a potência entregue a um nível fixo, desconectando quando essa carga é excedida. Se o cliente exceder essa carga e for desconectado, o cliente pode reajustar um botão na unidade de exibição remota opcional para restabelecer carga contanto que a carga conectada seja menos do que o limite prefixado. Alternativamente, clientes podem chamar uma linha de serviço de utilidade elétrica por telefone para ter o serviço restabelecido. Esta característica permite a utilidades elétricas proverem eletricidade para sistemas críticos até mesmo, por exemplo, no caso de um cliente não pagante.

(G) Limitação de Consumo Mensal: Alguns clientes se beneficiam de taxas subsidiadas e são dados um consumo total máximo por mês. O firmware de

Guarda de Energia é capaz de interromper energia quando um certo nível de consumo é alcançado. Porém, este tipo de programa é implementado melhor quando notificação avançada para clientes é provida. Isto pode ser alcançado tanto com uma exibição na casa, por meio de que uma mensagem ou série de mensagens notifica os clientes que sua taxa de consumo está chegando ao consumo projetado para o mês. Alternativamente (ou em conjunto), interrupções de serviço temporizadas podem ser programadas de forma que quando o limite está se aproximando, energia seja desconectada para períodos de tempo com incrementos cada vez mais longos para notificar os residentes. Estas interrupções planejadas em serviço atuam como uma advertência para clientes que seu limite está se aproximando de forma que eles tenham tempo para alterar seus padrões de consumo.

(H) Validação de Medidor: O módulo integrado do sistema preferivelmente é removível. Isto permite re-validação de laboratório fácil de precisão de medidor no evento de disputas de fatura de cliente.

(I) Benefícios Operacionais para Utilidade: O Guarda de Energia tem registros de evento e funções de diagnóstico a bordo, provendo os técnicos de campo com uma riqueza de dados para comissionar e reparar os sistemas elétrico e de comunicação. Parâmetros de não faturamento incluem: ampères, volts, temperatura, distorção harmônica total, frequência, valores instantâneos de Watts, VARs e volt-ampères, V2 horas, I2 horas, fator de potência, e ângulo de fase.

[0007] Estas características e outras serão aparentes àqueles qualificados na arte depois de revisar as descrições anexas, código de software e esquemas.

[0008] Em um aspecto, a invenção inclui um dispositivo para medir uso de eletricidade, incluindo: meio para desconexão remota por comunicação de linha de potência; meio para detecção de roubo de eletricidade; meio para detecção de fraude; e meio para detecção de tensão inversa.

[0009] Em outro aspecto, a invenção inclui um aparelho para medição de multi-canal de eletricidade, incluindo: (a) uma cabeça de medidor operável para medir uso de eletricidade para uma pluralidade de linhas de consumidor de eletricidade; (b) um

transponder em comunicação com a cabeça de medidor e operável para transmitir dados recebidos da cabeça de medidor por comunicação de linha de potência a um computador localizado remotamente, e para transmitir dados recebidos por comunicação de linha de potência do computador localizado remotamente à cabeça de medidor; e (c) um módulo de controle de carga em comunicação com a cabeça de medidor e operável para atuar conexão e desconexão de cada um de uma pluralidade de relés, cada relé da pluralidade de relés correspondendo a uma da pluralidade de linhas de consumidor de eletricidade.

[0010] Em várias concretizações: (1) o aparelho ademais inclui um detector de fraude em comunicação com a cabeça de medidor; (2) o detector de fraude inclui uma luz e uma superfície reflexiva, e a cabeça de medidor é operável para instruir o módulo de controle de carga para desconectar todas as linhas de cliente se o detector de fraude prover notificação que a luz não é detectada refletindo da superfície reflexiva; (3) o aparelho ademais inclui uma caixa contendo a cabeça de medidor, o módulo de controle de carga e os relés, e em que o detector de fraude inclui um detector de luz ambiente entrando na caixa; (4) o aparelho ademais inclui uma caixa contendo a cabeça de medidor, o módulo de controle de carga e os relés, e em que a caixa é instalada em um poste de utilidade; (5) o aparelho ademais inclui meio para comparar energia de transformador à energia total usada pelas linhas de consumidor; (6) o aparelho ademais inclui meio para detectar fluxo de tensão inversa pelas linhas de consumidor; (7) o aparelho ademais inclui uma memória legível por computador em comunicação com a cabeça de medidor e um contador em comunicação com a cabeça de medidor, o contador correspondendo a uma linha de cliente e operável para contar abaixo uma quantidade de energia armazenada na memória, e a cabeça de medidor operável para enviar um sinal de desconexão ao módulo de controle de carga para desconectar a linha de cliente quando o contador alcança zero; (8) o aparelho ademais inclui uma memória legível por computador em comunicação com a cabeça de medidor, a memória operável para armazenar um limite de carga para uma linha de cliente, e a cabeça de medidor operável para enviar um sinal de desconexão ao módulo de controle de carga para desconectar a

linha de cliente quando o limite de carga é excedido; (9) o aparelho ademais inclui uma memória legível por computador em comunicação com a cabeça de medidor, a memória operável para armazenar um limite de uso para uma linha de cliente, e a cabeça de medidor operável para enviar um sinal de desconexão ao módulo de controle de carga para desconectar a linha de cliente quando o limite de uso é excedido; (10) o transponder é operável para se comunicar com o computador localizado remotamente através de linhas de potência de média tensão; (11) o aparelho ademais inclui uma unidade de exibição em comunicação com a cabeça de medidor e operável para exibir dados recebidos da cabeça de medidor; (12) a unidade de exibição é operável para exibir informação relativa a consumo de energia de um cliente; (13) a unidade de exibição é operável para exibir advertências relativas a uso de energia de um cliente ou roubo de energia suspeito; e (14) a unidade de exibição é operável para transmitir para dita cabeça de medidor informação entrada por um usuário.

Breve Descrição dos Desenhos

[0011] A Figura 1 é um diagrama de bloco/fiação mostrando a conexão de concretizações preferidas.

[0012] A Figura 2 é um diagrama de bloco mostrando a configuração física de concretizações preferidas.

[0013] As Figuras 3A-3B são diagramas esquemáticos de uma placa de CPU preferida de um Transponder de Varredura e MiniCloset.

[0014] A Figura 4 é um diagrama esquemático de uma fonte de energia de Transponder de Varredura preferida.

[0015] A Figura 5 é um diagrama esquemático de uma fonte de energia de MiniCloset preferida.

[0016] A Figura 6 é um diagrama esquemático de uma placa de circuito preferida para retornar informação de transformador de corrente a uma cabeça de medidor de MiniCloset.

[0017] As Figuras 7A-7C são diagramas esquemáticos de uma placa de circuito de Módulo de Controle de Carga preferida.

[0018] As Figuras 8A-8D são diagramas esquemáticos de uma placa de fonte de energia preferida que provê detecção de fraude óptica.

[0019] As Figuras 9A-9C são diagramas esquemáticos de uma placa de conexão de Guarda de Energia preferida.

[0020] A Figura 10 é um diagrama esquemático para uma placa de circuito de controle operável para prover controle de relé.

[0021] A Figura 11 é um diagrama de montagem de base de Guarda de Energia preferida.

[0022] As Figuras 12 e 13 são diagramas de barramentos de fase preferidos e construção dos mesmos.

[0023] A Figura 14 é um diagrama descrevendo construção e montagem de estrutura de barra de neutro preferida.

[0024] A Figura 15 descreve barras de transição preferidas;

[0025] A Figura 16 descreve colocação preferida de barras de transição.

[0026] As Figuras 17 e 18 descrevem construção de módulo de aceitante preferido.

[0027] A Figura 19 descreve um módulo sensor de corrente e relé integrado preferido.

[0028] A Figura 20 descreve uma vista explodida de um módulo sensor de corrente e relé integrado preferido.

[0029] A Figura 21 mostra vistas explodidas de módulos de medição preferidos.

[0030] A Figura 22 mostra os módulos de medição colocados em uma montagem de estrutura de EG e módulo de aceitante.

[0031] A Figura 23 mostra uma vista explodida de uma concretização preferida de Guarda de Energia.

[0032] A Figura 24 mostra uma vista explodida de uma montagem de EG preferida e montagem de base.

[0033] A Figura 25 mostra uma disposição de EG preferida.

[0034] As Figuras 26 e 27 são esquemas de módulo de medição preferidos.

[0035] A Figura 28 tem esquemas preferidos para uma placa de lugar traseiro.

[0036] A Figura 29 tem esquemas preferidos para uma placa de potência.

[0037] A Figura 30 tem esquemas preferidos para uma placa de extensão de I/O.

[0038] A Figura 31 tem esquemas preferidos para uma placa de CPU.

[0039] A Figura 32 tem esquemas preferidos para um módulo de controle.

[0040] A Figura 33 tem esquemas preferidos para circuitos de medição e fonte de energia para um módulo de exibição de cliente;

[0041] A Figura 34 tem esquemas preferidos para uma placa de exibição para o CDM.

[0042] A Figura 35 é um diagrama de bloco de uma extremidade dianteira analógica preferida para medição.

[0043] As Figuras 36 e 37 descrevem implementações de DSP preferidas.

[0044] A Figura 38 ilustra características de resposta a impulso e freqüência de filtro em fase preferidas.

[0045] A Figura 39 ilustra injetar sinais de PLC a harmônicos meio-ímpares de 60 Hz.

[0046] A Figura 40 descreve 12 possíveis modos nos quais um quadro de FFT recebido por um medidor pode estar fora de fase com um quadro de FFT transmitido por transponder de varredura.

[0047] A Figura 41 ilustra especificações de filtro de FIR preferidas.

[0048] A Figura 42 descreve tensão e corrente resultando de uma FFT preferida.

Descrição Detalhada de Concretizações Preferidas

[0049] Em uma concretização, um aparelho de medição de Guarda de Energia inclui um MiniCloset (quer dizer, um aparelho de medição operável para medir uma pluralidade de linhas de cliente); um Transponder de Varredura; um ou mais relés operáveis para desconectar serviço para clientes selecionados; um Módulo de Controle de Carga; e meio de detecção de fraude óptica.

[0050] O MiniCloset e Transponder de Varredura referidos aqui são em grande parte os mesmos como descritos na Patente US N° 6.947.854. Quer dizer, embora cada um tenha sido melhorado através dos anos, a funcionalidade e estrutura pertinentes a esta descrição podem ser tomadas para serem as mesmas como

descrito nessa patente.

[0051] Um aspecto da invenção inclui tomar a funcionalidade de medição de multicanal existente achada no MiniCloset e adicionar conexão e desconexão remotas por PLC. Prover tal funcionalidade adicional requerida adicionando novo hardware e software. O hardware adicionado inclui um Módulo de Controle de Carga (LCM) e relés de conexão/desconexão. Também adicionado eram circuitos de suporte para dirigir rastros de sinal para e do processador de medidor principal - a Cabeça de Medidor de MiniCloset. As adições de software incluem módulos de código que se comunicam com o hardware adicionado, como descrito nas tabelas abaixo. Figura 1 é um diagrama de bloco de conexões de uma concretização preferida. Linhas de potência de média tensão A, B, C e N (neutro) alimentam em Transformador de Distribuição 110. Linhas de baixa tensão conectam (por transformadores de corrente 120) o Transformador de Distribuição 110 à unidade de Guarda de Energia 140. A unidade de Guarda de energia 140 monitora os transformadores de corrente 120, e alimenta linhas de cliente monofásicas 1-24.

[0052] A Figura 2 é um diagrama de bloco de estrutura preferida de uma unidade de Guarda de Energia 140.

[0053] Transponder de Varredura 210 é o coletor de dados preferido para a unidade 140, pode estar localizado externo ou dentro do MiniCloset, e pode ser o coletor de dados principal para mais de um MiniCloset de cada vez. O Transponder de Varredura 210 preferivelmente: (a) verifica dados (cada comunicação preferivelmente começa com a verificação de relógio e identidade de medidor para assegurar integridade de dados); (b) coleta dados (coleta periodicamente um bloco de dados de cada unidade de medidor, com cada bloco contendo leituras de medidor previamente coletadas, leituras de intervalo, e registros de evento); (c) armazena dados (preferivelmente os dados são armazenados em memória não volátil por um período especificado (por exemplo, 40 dias)); e (d) relata dados (tanto por PLC, modem de telefone, conexão de RS-232, ou outro meio).

[0054] A placa de deslizamento 280 inclui uma cabeça de medidor de Minicloset e um módulo de controle de carga 240, que provê os sinais de controle para ativar

os relés. Toda a eletrônica preferivelmente é energizada por fonte de energia 250. A montagem de placa traseira 270 inclui múltiplos (por exemplo, 24) Transformadores de corrente e relés - agrupados, neste exemplo, como três conjuntos de 8 TCs e relés. Cabos de cliente são conectados por fios pelos TCs e se conectam ao circuito nas instalações de cliente 290. O Transponder de Varredura remotamente localizado 210 acessa a cabeça de medidor de Guarda de Energia e se comunica de modo bidirecional usando comunicação de portadora de linha de potência.

[0055] O fluxo de sinal mostrado nas Figuras 1 e 2 é preferivelmente realizado implementando módulos de código de software diferentes que trabalham simultaneamente para habilitar a habilidade de conexão/desconexão remota no Minicloset. Estes módulos de software, providos no Apêndice abaixo, são:

Módulo de Código	Localização	Função
lcm.def e pic.def	Módulo de Controle de Carga	Atuar conexão e desconexão de relés
pulse.c e pulse.h	Cabeça de Medidor	Estabelecer Comunicação com LCM
picend.def e picvars.def	Cabeça de Medidor	Prover sinais de controle para LCM
pulselink.def e pulseoutm.c	Cabeça de Medidor	Provê LCM com pulsos a serem usados para conectar e desconectar relés

[0056] Figuras 3-10 são esquemas de componentes preferidos, como descrito abaixo. Os relés de conexão/desconexão preferidos são relés de série K850 KG, mas aqueles qualificados na arte reconhecerão que outros relés podem ser usados sem partir da extensão da invenção.

Figura	esquema	Detalhe
3	PCB 107D	Placa de CPU do Transponder de Varredura e MiniCloset
4	PCB 135C	Fonte de energia para Transponder de Varredura

5	PCB 144C	Fonte de energia para MiniCloset
6	PCB 146C	Esta placa trás de volta informação de Transformador de Corrente para cabeça de medidor de MiniCloset
7	PCB 160A	Placa para Módulo de Controle de Carga
8	PCB 170	Placa de fonte de energia de EG que adiciona capacidade para detecção de fraude óptica
9	PCB 171	Placa de conexão de EG. Uma placa com trilhas para rotear o sinal
10	PCB 172	Placa de circuitos de controle para controle de relé

[0057] Em outra concretização, a implementação de Guarda de Energia tira vantagem da semelhança de arquitetura de painéis de disjuntor tradicionais, com o ambiente de medição de multicanal. Em um painel de disjuntor, eletricidade é alimentada ao painel e distribuída entre vários circuitos de cliente por disjuntores que provêem a habilidade para conectar ou desconectar os circuitos de cliente.

[0058] No MiniCloset/Guarda de Energia, múltiplos transformadores de corrente medem a corrente em circuitos de cliente e trazem estes dados de volta a uma unidade de processamento central, onde as quantidades de medição são calculadas. Porém, o MiniCloset/Guarda de Energia tem várias diferenças principais com um painel de disjuntor. Por exemplo, enquanto os disjuntores são achados perto de instalações de cliente, o Guarda de Energia tipicamente é instalado perto do transformador de distribuição de utilidade. As vantagens oferecidas por esta concretização alternada serão aparentes àqueles qualificados na arte. Por exemplo, esta concretização oferece dimensões melhoradas e tamanho global sobre as concretizações discutidas acima. Espaço é sempre um constrangimento quando adições de equipamento são feitas a instalações elétricas existentes. Esta versão do Guarda de Energia ("EG"), com dimensões preferidas de 71,12 cm X 55,88 cm X 27,94 cm provê uma vantagem significativa em situações onde constrangimentos volumétricos existem.

[0059] A descrição seguinte inclui detalhes de construção preferidos, esquemas detalhados, e descrições de software. Como com as concretizações discutidas

acima, esta concretização é operável para prover operações de desconexão/conexão remotas, prevenir roubo, detectar fraude, detectar tensão inversa, executar pré-pagamento e limitar carga, e executar validação de medidor.

Detalhes de Construção de EG Preferidos

[0060] Nesta concretização, componentes primários do EG são:

1. Montagem de Base de Guarda de Energia
2. Montagem de Guarda de Energia
   a. Barramentos de Fase e Barramentos de Neutro
   b. Barras de transição
   c. Módulo de Aceitante
3. Módulos de Medição de Guarda de Energia
   a. Módulos de Medição

Módulos de Sensor de Corrente e Relé Integrados

4. Eletrônica de Guarda de Energia
   a. PCB 203
   b. PCB 204
   c. PCB 234
   d. PCB 235
   e. PCB 202
   f. PCB 210
   g. PCB 230
   h. PCB 206

Base de Montagem de EG

[0061] A base de EG inclui um fundo de invólucro com parafusos e arruelas de retenção como um mecanismo de travamento para a cobertura de topo de EG, que está conectada em um lado por dobradiças. Veja Figura 11. O fundo de invólucro provê roteamento para os cabos de cliente.

Montagem de EG - Barramentos de Fase

[0062] Três barramentos de fase de alumínio são colocados para o centro da montagem de Guarda de Energia e alternados. Veja Figuras 12 e 13. Estes provêem

conexão aos módulos de medição de cliente pelo uso de barras de transição. Uma disposição de barramento alternada é descrita na Figura 13. Barramentos são mostrados em preto.

Barras de Neutro

[0063] O EG preferivelmente inclui 4 barras de neutro que formam um quadro para montagem de EG, por esse meio provendo um caminho para a corrente de neutro. Isto é mostrado na Figura 14. O borne na barra cruzada provê a alimentação de neutro do transformador de distribuição de utilidade. Também, há 2 barras de neutro de placa-mãe que levam a corrente de neutro ao módulo de controle.

Barras de Transição

[0064] As barras de transição completam a conexão mecânica e elétrica entre os módulos de medição de cliente e os barramentos de fase. Veja a Figura 15. Uma barra de transição para fase A e C é mostrada na Figura 15A; uma barra de transição para fase B é mostrada na Figura 15B. Figura 16 mostra as barras de transição em preto.

Módulo de Aceitante

[0065] Um módulo de aceitante preferivelmente é feito de plástico e aceita mecanicamente os módulos de medição que podem ser encaixados facilmente na montagem de EG. Cada EG tem 4 módulos de aceitante que são empilhados juntos e podem acomodar tanto 12 módulos de medição bifásicos ou 8 trifásicos. Veja Figura 17. O módulo de aceitante também provê uma rota mecânica para a barra neutra de placa-mãe que conecta ao módulo de controle. Veja Figura 18.

Módulos de Medição de Cliente

[0066] Módulos de medição de cliente preferidos provêem a metrologia requerida para medir o consumo para um cliente monofásico, bifásico ou trifásico. Um módulo individual funciona como um medidor independente completo que pode ser testado e avaliado como uma unidade de medição separada. Cada módulo preferivelmente inclui um módulo de sensor de corrente e relé integrado e eletrônica de metrologia, e provê uma conexão entre o circuito de cliente e os barramentos de fase. Figura 19 descreve um módulo de sensor de corrente e relé integrado preferido. Figura 20

descreve uma vista explodida de um módulo de sensor de corrente e relé integrado preferido.

[0067] Figura 21 mostra vistas explodidas de módulos de medição preferidos. Figura 22 mostra os módulos de medição (mostrados em preto) colocados na montagem de quadro de EG e módulo de aceitante.

[0068] Figura 23 mostra uma vista explodida de Guarda de Energia, e Figura 24 mostra uma vista explodida de uma Montagem de EG e Montagem de Base de EG preferidas.

Eletrônica

[0069] As caixas de Módulo de Controle preferivelmente incluem várias PCBs que funcionam simultaneamente para coletar dados de medição dos módulos de medição individuais e comunicar através de linhas de potência para transmitir estes dados a um dispositivo mestre, tal como um Transponder de Varredura ("ST").

[0070] Figura 25 mostra uma disposição de Guarda de Energia preferida para esta concretização. Cada linha de cliente tem um Módulo de Medição correspondente (PCB 203 e PCB 204, discutido abaixo) (esquemas mostrados nas Figuras 26 e 27).

[0071] Uma Placa de Colocação Atrás 2510 mostrada na Figura 25 (PCB 234; veja Figura 28 para diagrama de construção e esquemático) é o barramento comum que dirige sinais dentro do EG. Há dois tipos de opções de comunicação na Placa de Colocação Atrás 2510 para habilitar transferência de dados de Módulo de Controle 2520 para Módulos de Medição individuais PCB 203. Isto pode ser feito tanto usando a opção de I2C de 2 fios ou a opção serial de 1 fio.

[0072] O Módulo de Controle 2520 inclui uma Placa de Potência (PCB 210; veja Figura 29 para esquemático) é a placa de fonte de energia que também tem os circuitos de transmissão e recepção de PLC nela. A Placa de Potência provê energia à placa de CPU e às placas de eletrônica 203. O Módulo de Controle 2520 também inclui uma Placa de Extensão de I/O (PCB 230; veja Figura 30 para esquemático) é uma placa com várias opções de extensão de I/O que habilitam comunicação de Módulos de Medição à placa de CPU.

[0073] Módulo de Controle 2520 também inclui uma Placa de CPU (PCB 202; veja Figura 31 para esquemático), que tem um processador de Processamento de Sinal Digital (DSP) a bordo.

[0074] Finalmente, Módulo de Controle 2520 inclui uma placa de roteamento (PCB 235; veja Figura 32 para esquemático) com trilhas e um coletor sem componentes eletrônicos nele.

[0075] Cada Módulo de Exibição de Cliente (CDM) 2530 é instalado nas instalações do cliente e pode se comunicar de modo bidirecional com o EG instalado no transformador de distribuição servindo o cliente. PLC bidirecional habilita comunicação de utilidade-cliente através de linhas de potência de baixa tensão e permite a utilidade enviar informação regular, advertências, informação especial sobre interrupções, etc., para o cliente. Cada CDM 2530 inclui uma combinação selecionada de medição e fonte de energia junto com circuitos de PLC na mesma placa (PCB 240; veja Figura 33 para esquemático). Cada CDM preferivelmente também tem uma placa de exibição de 9 dígitos (PCB 220; veja Figura 34 para esquemático). Esta exibição se comunica com EG e mostra informação sobre consumo, precauções, advertências e outras mensagens de utilidade.

Implementação de Hardware

[0076] Em uma concretização, o Guarda de Energia implementa Transformada de Fourier Rápida (FFT) no sinal de comunicação de PLC ambos no ST e no medidor, e para propósitos de medição executa análise de harmônico detalhada. Esta seção discute um esquema de implementação dos Módulos de Medição, comunicação com Módulos de Controle e comunicação de PLC do Módulo de Controle com um Transponder de Varredura remotamente localizado.

[0077] O Módulo de Controle 2520 inclui fonte de energia e circuitos de PLC (PCB 210; veja Figuras 25 e 29); extensão de I/O (PCB 230; veja Figura 30) e placa de CPU nomeada Medidor D (PCB 202; veja Figura 31). A fonte de energia provê energia ao medidor D e extensão de I/O e contém os circuitos de transmissor e receptor de PLC. PCB 235 provê um roteamento de trilha e conexão de coletor entre várias placas.

[0078] O Módulo de Medição pode ter duas versões: bifásica ou trifásica. A versão bifásica pode ser programada por software para funcionar como um único medidor bifásico ou dois medidores monofásicos. A versão bifásica inclui um medidor B2 (PCB 203, esquemático mostrado na Figura 26), enquanto a versão trifásica inclui um medidor B3 (PCB 204 esquemático mostrado na Figura 27). Os medidores B atuam como escravos para o medidor D no Módulo de Controle 2520. Os medidores D e B podem se comunicar por um protocolo de ASCII serial.

[0079] Os vários medidores B são interconectados por BPB 2510 a 2520, que provê energia, uma referência de 1 Hz e comunicações seriais para o medidor D. A máquina de DSP preferida para o medidor B é o chip 56F8014VFAE Freescale. O microprocessador preferido usado para implementar a CPU no medidor D é o entre a família de ColdFire Integrated Microprocessadores, MCF5207. O uso de um processador específico é determinado por requisitos de RAM e Flash ditados pela versão de medidor. Uma fonte de energia separada e placa de LCD completam a porção eletrônica do medidor D como um produto. À parte de atuar como um mestre para medidores B, o medidor D também é um medidor trifásico e mede a saída de transformador total em que o EG está instalado. Como uma característica anti-roubo, este total é comparado com o consumo total relatado pelos vários medidores B.

$$\sum_{n=1}^{24} kWh_n$$ = Saída de Transformador Total

[0080] A constituição de fluxos de sinal é como segue:

B2: Duas tensões, Duas correntes e Nenhum Canal de Portadora de Linha de Potência (PLC).

B3: Três tensões, Três correntes e Nenhum Canal de PLC.

D: Três tensões, Três correntes e um Canal de PLC.

[0081] Cada fluxo tem um circuito associado para efetuar amplificação analógica e 'antialiasing'.

[0082] Específico para o medidor D é a implementação preferida de:

- Uma Malha Travada por Fase (PLL) para travar a amostragem dos fluxos de sinal a um múltiplo da linha de A/C entrante (amostragem síncrona para a

linha de potência).

- Um Oscilador Controlado por Tensão (VCO) a 90-100 MHz controlado por processador de DSP por dois módulos de PWM acionando diretamente o relógio de sistema, conseqüentemente fazendo o DSP coerente com o PLL.
- Um detector de fase síncrono que só responde ao fundamental da onda de freqüência de linha entrante e não a seus harmônicos.
- Opção para executar esquemas de modulação de FSK e PSK.

[0083] Cada canal de medição e comunicação preferivelmente inclui circuitos analógicos de extremidade dianteira seguidos pelo processamento de sinal. Único aos circuitos analógicos é um filtro 'anti-aliasing' com ganho fixo que provê rastreamento de temperatura de primeira ordem, conseqüentemente eliminando a necessidade para recalibrar medidores quando derivas de temperatura são encontradas. Isto é discutido a seguir, e então uma implementação de processamento de sinal preferida é discutida.

Cadeia de Sinal Analógico de Tensão e Corrente

[0084] A extremidade dianteira analógica para canais de tensão (corrente) inclui elementos sensores de tensão (corrente) e um atenuador programável, seguido por um filtro 'antialiasing'. O atenuador reduz o nível de sinal entrante de forma que nenhum corte ocorra depois do filtro 'anti-aliasing'. O filtro 'anti-aliasing' de ganho constante restabelece o sinal a valor total na entrada do Conversor Analógico para Digital (ADC). Para medição, o filtro 'anti-aliasing' corta fora freqüências acima de 5 kHz. As entradas são então alimentadas no ADC que faz parte do DSP. Veja Figura 35, que é um diagrama de bloco de uma extremidade dianteira analógica preferida para medição.

[0085] Enquanto uma implementação típica incluiria um Amplificador de Ganho Programável (PGA) seguido por um filtro 'anti-aliasing' de baixo ganho, a invenção, nesta concretização, implementa um atenuador programável seguido por um filtro de ganho fixo grande. Além disso, a implementação de ambos os filtros 'anti-aliasing' em um único chip é a mesma usando os mesmos Amplificadores Operacionais Quádruplos junto com resistores de 25 ppm e capacitores de NPO/COG. Esta

implementação única casando os filtros 'anti-aliasing' assegura que as derivas de fase encontradas em ambos os canais de tensão e corrente sejam exatamente idênticas e conseqüentemente precisão do cálculo de potência (dado pelo produto de V e I) não é comprometido. Isto provê um meio para ambos canais de V e I rastrearem derivas de temperatura até primeira ordem sem recalibrar o medidor.

[0086] Em contraste, usando um PGA junto com um filtro de ganho baixo não pode rastrear o deslocamento de fase nos sinais V e I introduzidos devido à temperatura. Isto é porque o deslocamento de fase introduzido por PGA é uma função do ganho.

Cadeia de Sinal Digital de Tensão, Corrente e PLC

[0087] Figura 36 é um diagrama de bloco da placa de PCB 202; as funções de cada bloco serão aparentes àqueles qualificados na arte. Figura 36 mostra uma implementação de DSP preferida.

[0088] Esta concretização preferivelmente usa uma PLL para travar a amostragem dos fluxos de sinal a um múltiplo da freqüência de linha de A/C entrante. Na concretização discutida acima, a amostragem é a uma taxa assíncrona à linha de potência. No medidor D, há um VCO a 90-100 MHz que é controlado pela máquina de DSP por dois módulos de PWM. O VCO excita diretamente o relógio de sistema do chip de DSP (desabilitando a PLL interna), assim o DSP se torna uma parte integrante da PLL. Travar o relógio de sistema do DSP à linha de potência facilita o alinhamento da amostragem para a forma de onda da linha de potência. O detector de fase deveria funcionar assim para só responder ao fundamental da onda de 60Hz entrante e não a seus harmônicos. Figura 37 é um diagrama de bloco desta implementação de DSP preferida.

[0089] Um DSP BIOS ou código de comutação de contexto voluntário provê três pilhas, cada uma para fundo, comunicações de PLC e comunicações seriais. O pequeno micro se comunica com o DSP usando um excitador I2C. O circuito integrado MSP430F2002 mede as fontes de energia, porta de fraude, temperatura e tensão de bateria. As tarefas do MSP430F2002 incluem:

manter um RTC;

ii. medir a tensão de bateria;

iii. medir a temperatura;

iv. medir a fonte de energia +U;

v. reajustar o DSP na interrupção parcial;

vi. prover um circuito de vigilância adicional; e

vii. prover uma referência de 1 segundo para entrar no DSP para uma referência de tempo para medir a referência de 1 segundo contra o relógio de sistema do VCO.

Cadeia de Sinal de Comunicação de PLC de Medidor D

[0090] Uma instalação típica consiste em múltiplos EGs e STs se comunicando através das linhas de potência. O medidor D se comunica de modo bidirecional com um Transponder de Varredura localizado remotamente pelo transformador de distribuição. Para habilitar isto, esta concretização usa uma banda de 10-25 kHz para comunicação de PLC. O sinal de PLC é amostrado a cerca de 240 kHz (212 * 60), síncrono com tensão de linha, seguindo o que um filtro de Resposta a Impulso Finita (FIR) é aplicado para dizimar os dados. Especificações de FIR preferidas são dadas abaixo: Banda de 10-25 kHz

Número de Derivações	65
Atenuação de Banda de Parada	71,23 dB
Freqüência Superior de Banda Passante	25 kHz
Freqüência Inferior de Banda de Parada	35 kHz
Entrada Amostrada	60 * 4096
Saída Mostrada	30* 2048

[0091] Veja Figura 38 para características preferidas de resposta de freqüência de filtro em fase e resposta a impulso.

[0092] Depois que a dizimação é feita a 60 kHz (2^{11} * 30), uma FFT de 2048 pontos é então executada nos dados dizimados. A taxa de dados é assim determinada ser 30 baud dependendo da escolha de filtros de FIR. Toda FFT produz dois bits aproximadamente a todo 66 ms ao usar FIR na banda de 10-25 kHz para se comunicar por transformadores de distribuição.

[0093] Para evitar o problema de se comunicar na presença de ruído de linha, esta concretização preferivelmente implementa uma técnica única para comunicação robusta e segura. Isto é feito injetando sinais de PLC a freqüências que são harmônicos meio-ímpares da freqüência de linha (60Hz). Isto é discutido abaixo, para uma concretização usando um espectro de ruído típico achado em linhas de CA na gama de 12-12,2 kHz. Figura 39 ilustra injetar sinais de PLC a harmônicos meio-ímpares de 60 Hz. Desde que FFT é feita a todo 30 Hz e os harmônicos são separados por 60 Hz, os bits de dados residem na caixa correspondendo ao 201,5º e 202,5º harmônico de 60 Hz na Figura 39. O algoritmo considera estas duas caixas de freqüências e compara a amplitude do sinal nos dois para determinar 1 ou 0. Este esquema de FSK usa duas freqüências e produz uma taxa de dados de 30 baud. Alternativamente, QFSK, que usa 4 freqüências, pode ser implementado para produzir 60 baud.

[0094] Ao atravessar por transformadores, preferivelmente ambos STs e medidores D executam FFT no PLC e sinais de dados a todo 30 Hz em uma gama de 10-25 kHz. Porque as Malhas Travadas em Fase (PLLs) implementadas ambos no ST e no medidor D estão travadas à linha, os quadros de dados estão sincronizados à freqüência de linha (60 Hz) igualmente. Porém, os quadros de dados podem se deslocar em fase devido a:

1. várias configurações de transformador que podem existir no caminho entre o ST e medidor (delta -ípsilon, etc.); e

2. um deslocamento em fase devido ao fato que STs estão travados em uma fase particular, enquanto medidores mono e polifásicos podem ser energizados por outras fases.

[0095] A relação de sinal para ruído (SNR) é maximizada quando o quadro de dados de medidor e quadros de dados de ST estão alinhados perto de perfeição. Do ponto de vista de um medidor, isto requer receber sinal de PLC de todos os possíveis STs que pode "ouvir", decodificar o sinal, verificar por SNR alinhando quadros de dados, e então responder ao ST que está produzindo SNR máxima. Figura 40 descreve os 12 possíveis modos nos quais o quadro de FFT recebido pelo

medidor pode estar fora de fase com quadro de FFT de ST. Linhas pontilhadas correspondem a uma rotação de 30 graus considerando um transformador delta no caminho de sinal entre ST e o medidor.

[0096] Além disso, porque os quadros de dados estão disponíveis a todo 30 Hz em uma linha de 60 Hz, há duas possibilidades correspondendo às 2 possíveis fases obtidas dividindo 60 Hz por 2. Conseqüentemente, há 24 modos que os quadros de dados de medidor podem estar desalinhados com quadros de dados de ST.

[0097] Em cada quadro do ST, há um número integral ímpar de ciclos da freqüência de portadora. Desde que o esquema de modulação preferido é Chaveamento por Deslocamento de Freqüência (FSK), se houver n ciclos para transmitir bit 1, bit 0 é transmitido usando n+2 ciclos da freqüência de portadora. Fica vital para o medidor reconhecer seus próprios 2 ciclos de 60 Hz a fim de ser capaz de decodificar seus bits de dados que estão disponíveis a todo 1/30 de um segundo.

[0098] Se o medidor D decodificar sinais com quadros de dados desalinhados, há energia que transborda nas freqüências adjacentes (meio-ímpares separadas). Se o nível de sinal que cai na caixa de freqüência "adjacente" for menos do que o fundo de ruído, o sinal pode ser decodificado corretamente. Porém, se o derramamento for mais do que o fundo de ruído, a habilidade para distinguir entre 1 e 0 diminui, e conseqüentemente a SNR global cai, resultando em um erro em decodificação. Em conclusão:

a. Se os quadros estiverem desalinhados, contaminação de bits de dados ocorre e a SNR degrada.

b. No evento que a freqüência muda e há quadros de dados desalinhados, há uma quantia significativa de energia que transborda nas caixas de FFT adjacentes, conseqüentemente interferindo com os outros STs no sistema que se comunicam usando freqüências nessas caixas específicas.

[0099] Uma vez que o deslocamento de relógio seja determinado correspondendo à SNR mais alta, o medidor então se trava até que uma mudança significante em relação de SNR seja encontrada pelo medidor, em qual caso o processo repete.

Implementação de Medição em Medidor D e B usando FFT

[00100] Enquanto versões do medidor B e do medidor D executam medição, o medidor D também é responsável por coletar a informação de medição dos vários medidores B por PCB 234. Cada fluxo de dados nos medidores tem um circuito associado para efetuar amplificação analógica e 'anti-aliasing'. Cada uma das seções de extremidade dianteira analógica tem um atenuador programável que é controlado pelo código de nível mais alto. O fluxo de dados é amostrado a 60 kHz (2^{10} * 60) e então um filtro de FIR é aplicado para dizimar o fluxo de dados a ~15 kHz (2^{8} * 60). Especificações de filtro preferidas são mostradas na tabela abaixo e Figura 41.

Número de Derivações	29
Atenuação de Banda de Parada	80,453dB
Freqüência Superior de Banda Passante	3kHz
Freqüência Inferior de Banda de Parada	12kHz

[00101] Desde que só os dados até 3 kHz são de interesse, preferivelmente uma faixa de 3-12 kHz no FIR de dizimação é usada com taxa de amostra de ~15KHZ. As freqüências de 0-3 kHz ou 12-15 kHz são mapeadas em 0-3 kHz. Uma FFT real é executada para produzir 2 fluxos de dados que podem ser decompostos ademais em 4 fluxos de dados: Tensão Real e Imaginária e Corrente Real e Imaginária. Isto é alcançado adicionando e subtraindo freqüências de espelho positivas e negativas para as partes real e imaginária, respectivamente. Desde que o sinal 'aliased' na gama de 12-15 kHz cai abaixo de 80 dB, a precisão é alcançada usando o filtro de FIR acima discutido. Alternativamente, uma FFT complexa de 256 pontos pode ser executada em toda fase do fluxo de dados dizimado. Isto produz 2 pares de fluxos de dados: uma parte real, que é a tensão, e uma parte imaginária, que é a corrente. Esta abordagem requer uma FFT complexa de 256 pontos a todo 16,667 milissegundos.

[00102] Os resultados de executar qualquer FFT são a tensão e corrente mostradas na Figura 42, onde a notação $V_{m,n}$ denota o m-ésimo harmônico do n-ésimo número de ciclo. Por exemplo, V_{11} e I_{11} correspondem ao fundamental do

primeiro ciclo, e V_{21} e I_{21} para o primeiro harmônico do primeiro ciclo, etc., como mostrado na Figura 42, que descreve quadros de FFT para tensão, indicando os harmônicos.

[00103] As partes real e imaginária do conteúdo harmônico de qualquer k-ésimo ciclo são dadas por:

$$V_{mk} = Re\,(V_{mk}) + iIm(V_{mk});\ m = 1, \ldots, M$$

$$I_{mk} = Re\,(I_{mk}) + iIm(I_{mk});\ k = 1, \ldots, n$$

[00104] A parte imaginária de tensão é a medida de falta de sincronização entre a PLL e a freqüência de linha. A fim de calcular quantidades de medição, os cálculos são feitos no domínio de tempo. No domínio de tempo, a funcionalidade de FFT oferece a flexibilidade para calcular quantidades de medição tanto usando só o fundamental ou incluindo os harmônicos. Usando a forma complexa da tensão e corrente obtidas da FFT, as quantidades de medição são calculadas como:

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

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

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

$$\text{Fator de Potência} = W/P$$

[00105] Porém, nas fórmulas anteriores, quando os harmônicos estão incluídos (V_{mk} e I_{mk}; m = 1, ... M; k = 1, ... n), todas as quantidades de medição incluem os efeitos de harmônicos. Por outro lado, quando só o fundamental é usado (V_{1k} e I_{1k}), todas as quantidades calculadas representam só a contribuição de 60 Hz. Como um exemplo, nós mostramos os cálculos só quando o fundamental é usado para executar cálculos. Só V_1e I_1 são usados de todos os quadros de dados de FFT. As quantidades seguintes são calculadas para um dado conjunto de N quadros e uma freqüência de linha de f_{line}:

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

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

$$kVAh = \sum_{i=1}^{N} |V_{1i}| * |I_{1i}| * \Delta t_i * 10^{-3} ; \Delta t = 1/ f_{line}$$

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

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

[00106] O fator de potência de deslocamento é dado por:

$$\text{Cos}(\theta) = \left| \frac{W}{VA} \right|;$$

onde W e VA incluem só os fundamentais e

$VA_1 = V_1RMS * I_1RMS$; onde

$$V_1RMS = \sqrt{\sum_{n=1}^{N} |V_{1,n}|^2} \;\&\; I_1RMS = \sqrt{\sum_{n=1}^{N} |I_{1,n}|^2} :$$

para N ciclos.

[00107] Esta flexibilidade para tanto incluir ou excluir os harmônicos ao calcular quantidades de medição se traduz a uma melhoria significante sobre as capacidades oferecidas pela concretização acima descrita. Ainda outra característica oferecida por esta concretização é o cálculo de Distorção Harmônica Total (THD). A THD é a medição da distorção harmônica presente, e é definida como a relação da soma das potências de todos os componentes harmônicos para a potência do fundamental. Para o n-ésimo ciclo, isto é avaliado como:

$$VTHD_n = \frac{\sqrt{\sum_{m=2}^{M} V_{mn}^2}}{V_{1n}} \;\&\; ITHD_n = \frac{\sqrt{\sum_{m=2}^{M} I_{mn}^2}}{I_{1n}}$$

V_{mn} (I_{mn}) é o m-ésimo harmônico do n-ésimo ciclo obtido da FFT, onde

$V^2_{m,n} = Re(V_{m,n})^2 + Im(V_{m,n})^2$ e $I^2_{m,n} = Re(I_{m,n})^2 + Im(I_{m,n})^2$.

Módulo de Exibição de Cliente

[00108] O módulo de exibição de cliente é instalado nas instalações de cliente, se comunica com o Guarda de Energia perto do transformador, e inclui: PCB 240, fonte de energia e circuitos de PLC (veja a Figura 33); e PCB 220, mostrador de LCD (veja a Figura 34). Em uma concretização, a unidade de exibição de cliente instalada na

residência do cliente é um unidade de PLC bidirecional que se comunica com EG. Por exemplo, não só pode a utilidade enviar mensagens, o cliente também pode pedir uma verificação de consumo com o EG instalado no poste.

[00109] Enquanto certas concretizações específicas da invenção foram descritas aqui para propósitos ilustrativos, a invenção não está limitada aos detalhes específicos, dispositivos representativos, e exemplos ilustrativos mostrados e descritos aqui. Várias modificações podem ser feitas sem partir do espírito ou extensão da invenção definida pelas reivindicações anexas e seus equivalentes.

APÊNDICE

LCM.DEF

```
; QLC LCM (Load Control Module) program for PIC 16C63A;
;
        include 'C:\pictools\16c63a.inc'
;       include 'C:\pictools\16c63.inc'          ;

; Configuration bits

        FUSES _BODEN_OFF ; Brown-out reset disabled
        FUSES _CP_OFF            ; No code protect
        FUSES _PWRTE_OFF ; Disable power-up timer
        FUSES _WDT_ON            ; Watchdog Timer Enabled
        FUSES _RC_OSC            ; Oscillator = RC

; Program Parameter Equates

Software_type  equ     7       ; Which PIC program is this?
                               ; 7 = LCM program
Software_version equ   2       ; Version #

LCM_base_addr equ      16      ; Address of first LCM

Ram_start      equ     020h    ; Beginning of available RAM

RESET_VECTOR       equ     0       ; Location of reset vector

ROM_start      equ     008h    ; first available program location

Num_start_bits equ     12      ; Number of start bits
;
;       Port definitions
;
comm_port      equ     PORTA

data_out       equ     3
data_in        equ     4
clock_in       equ     5

ctl_portequ    PORTA

driver_enable  equ     0               ; output bit to turn on the drivers

Driver_enable_mask     equ     1 << driver_enable

;       Initialization values for I/O ports

init_TRISA     equ     00110000b       ; Data direction for port A

init_TRISB     equ     000h            ; B port data direction - all ports output
init_TRISC     equ     000h            ; C port data direction - all out
```

```
read_TRISC    equ    01fh            ; C port data direction for jumper read - 0-4 in

Init_comm_port_value  equ    1 << data_out + Driver_enable_mask

option_reg_val equ    10001111b ; Option register init value

;              bit 7 = 1:       Port B pullups disabled
;              bit 6 = 0:       Int on falling edge
;              bit 5 = 0:       Timer 0 uses inst clock
;              bit 4 = 0:       Timer 0 counts on rising edge
;              bit 3 = 1:       Prescaler used by WDT
;              bit 2,1,0 = 111:Prescaler divide by 128

; NOTE - interrupts and timer 0 are not used by this program

ms_preset      equ    700     ; Time set for millisecond delay
;

; RAM locations

        ORG    Ram_start

Status_reg              ds     4       ; status register
            ORG    Status_reg
Status_MSB              ds     1

Clk_timeout_flag        equ    7       ; Timeout waiting for comm clock edge
bad_reset_flag          equ    6       ; Non-POR, non-WDT reset encountered
POR_flag                equ    5       ; Power-on reset detected
WDT_reset_flag                 equ    4       ; WDT reset, not from sleep

Status_LSB              ds     1

CPU_reset_flag          equ    7
Need_param_refresh_flag        equ    6
Parity_err_flag         equ    5

Global_cmd_timer        ds     1

Jumper_state            ds     1       ; current state of input jumpers

Parameter_reg           ds     4       ; Communication parameters
                        org    Parameter_reg
                        ds     1       ;
                        ds     2       ;
                        ds     1       ;
;
; Fast communication variables
;

Comm_state              ds     1       ; Stored state of fast comm uart
Wstate                  ds     1       ; Stored state of fast comm word handler
Comm_bit_count                 ds     1       ; Bit counter for fast comm
Comm_buffer             ds     1       ; fast comm word buffer
```

```
Work_addr           ds      1       ; Address read from jumpers
Temp                ds      1       ; temporary register
Tpar                ds      1       ; Transverse parity
Word_count          ds      1       ; Counter for sending and receiving 5-bit words
Command                     ds      1       ; Command received over comm link
Register_ID         ds      1       ; Register specified in comm command
PIC_addr            ds      1       ; Address of this PIC chip
Comm_buf            ds      5       ; Communication buffer

Input_hold          ds      1       ; Hold register for comm rcv value

Output_bit_reg      ds      4       ; Hold register for output bit mask

Flags               ds      1       ; Program control flags

Clk_wait_flag       equ     7       ; indicates which clock edge we are waiting for
Global_cmd_flag             equ     6       ; Processing global command

scratch             ds      1       ; Work register
Timeout_ctr         ds      2       ; Counter for pulse duration
Millisecs           ds      1       ; Counter for pulse duration milliseconds

Clk_timeout_ctr             ds      2       ; counter for clock timeout

NPR_timer           ds      2       ; counter for need parameter refresh

;

        ORG     0a0h

; Bank 1 RAM locations

        IF      $ - 1 and 0100h
        error - RAM overflow
        ENDIF

;

        ORG     RESET_VECTOR
        jmp     start                   ; Jump to first program location
;
;       Start of program space
;
        ORG     ROM_start

;
;       Initialization
;
Start
        setb    RP0                     ; point to upper register bank
        jnb     NOT_POR,got_POR                 ; Power on reset, set flags
        clrb    RP0                     ; point to RAM bank 0
        jnb     NOT_TO,WDT_reset        ; Watchdog reset
        setb    status_MSB.bad_reset_flag       ; Indicate unknown reset condition
        jmp     Do_initialize
```

```
WDT_reset
        setb    Status_MSB.WDT_reset_flag
        jmp     do_initialize
got_POR
        clrb    RP0                     ; point to RAM bank 0
        mov     Status_MSB,#1 << POR_flag ; Set POR flag
        clr     Status_LSB
Do_initialize
        mov     FSR,#Ram_start+2        ; Point past status flags
:clr_loop
        clr     INDF                    ; Clear RAM location
        inc     FSR                     ; Increment pointer
        jz      Init_state              ; If zero, done
        mov     W,FSR                   ; get pointer
        and     W,#7fH                  ; look at 7 LSB's
        jnz     :clr_loop               ; Not end of segment, continue
        add     FSR,#20h                ; Point to next segment
        jmp     :clr_loop               ; Not last segment, continue
Init_state

        setb    Status_LSB.CPU_reset_flag           ; Indicate CPU reset
        setb    Status_LSB.Need_param_refresh_flag  ; Indicate parameter refresh needed
        mov     comm_port,#Init_comm_port_value             ; initialize comm port state
        mov     Comm_state,#Edge_wait
        mov     Comm_bit_count,#Num_start_bits              ; Initialize fast comm state
machine
        clr     PORTB
        clr     PORTC
        jmp     Main_loop               ; start running main program

;
; Comm Receive routines
;
Edge_wait
                                                ; Initialize Watchdog timer
        mov     !OPTION,#option_reg_val                 ; Set option reg
        setb    comm_port.data_out              ; send a 1
        dec     Comm_bit_count                          ; decrement bit count
        jz      See_edge_one                    ; if zero, looking for a 1
        jb      Input_hold.data_in,Set_edge_wait        ; Looking for 0, reset bit count if 1
        jmp     Comm_ret                                ; done
See_edge_one
        jnb     Input_hold.data_in,Set_edge_wait ; looking for 1, reset if 0
        mov     Wstate,#Rcv_addr                ; First word is address
        mov     Tpar,#01fh                      ; Init parity
        jmp     Rcv_next_word

Rcv_stop
        mov     Comm_state,#Edge_wait
        mov     Comm_bit_count,#Num_start_bits      ; Preset state to wait for edge
        jnb     Input_hold.data_in,Parity_error ; stop bit must be 1
        jmp     Cmd_exec                        ; Done, process received buffer

Rcv_bits
        clc                                     ; copy ...
```

```
        snb     Input_hold.data_in          ;     ... input bit ...
        setc                                ;         ... to carry
        rl      Comm_buffer                 ; shift buffer
        jmp     Rcv_word_handler            ; Check for complete word
;
; Comm Transmit routines
;
Send_edge
        dec     Comm_bit_count                      ; decrement bit count
        jz      :One                        ; if zero, send a 1
        clrb    comm_port.data_out          ; send a 0
        jmp     Comm_ret
:One
        setb    comm_port.data_out          ; send a 1
        jmp     Do_xmit_word

Send_bits
        rl      Comm_buffer                 ; put xmit bit into carry
        jc      :one
        clrb    comm_port.data_out          ; send a '0'
        jmp     :check
:one
        setb    comm_port.data_out          ; send a '1'
:check
        jmp     Xmit_word_handler


;
;       Comm receive word handler routines
;

Rcv_word_handler
        djnz    Comm_bit_count,Comm_ret             ; If word not complete, continue
        and     Comm_buffer,#03fh                   ; only 6 bits
        clc
        mov     W,>>Comm_buffer                     ; get 5 MSBs of received word
        call    parity_lookup
        xor     W,Comm_buffer                       ; compare with received word
        jnz     Parity_error
        rr      Comm_buffer
        and     Comm_buffer,#01fh                   ; extract 5-bit value
        xor     Tpar,Comm_buffer                    ; calculate total parity
        mov     PCLATH,#Rcv_addr<
        mov     W,Wstate
        jmp     W                           ; Execute word handler routine

Rcv_addr
        clrb    Flags.Global_cmd_flag ; Clear global command flag
        mov     W,Comm_buffer                   ; Get received address
        mov     PIC_addr,W              ; save it
        xor     W,#1fh                  ; is it global command
        jnz     See_if_us               ; No, check for address match
        setb    Flags.Global_cmd_flag ; Set global command flag
        jmp     Its_us                  ; process rest of command
See_if_us
```

```
        mov     !PORTC,#read_TRISC ; Switch C port bit 0-4 to input
        clrb    ctl_port.driver_enable  ; Disable output drivers, enable jumper read
        mov     temp,#9
:loop
        djnz    temp,:loop              ; Wait at least 20 uSec
        mov     W,/PORTC                ; Get inverse of jumpers in W
        mov     Work_addr,W             ; Save jumper value
        setb    ctl_port.driver_enable  ; Enable output drivers
        mov     !PORTC,#init_TRISC ; Restore C port status
        mov     Jumper_state,Work_addr          ; Save jumper state
        and     Work_addr,#03h                  ; Ignore all but 2 LSBs
        add     Work_addr,#LCM_base_addr ; Add base offset
        cjne    Work_addr,PIC_addr,Set_edge_wait
Its_us
        mov     Wstate,#Rcv_cmd                 ; point to next routine
        jmp     Rcv_next_word

Rcv_cmd
        mov     Command,Comm_buffer                     ; save command
        jnb     Command.4,Comm_get_fast         ; Fast read command, skip Reg ID
        mov     Wstate,#Rcv_reg_ID              ; point to next routine
        jmp     Rcv_next_word
Comm_get_fast
        mov     Wstate,#Rcv_tpar                ; point to next routine
        jmp     Rcv_next_word

Rcv_reg_ID
        mov     register_ID,Comm_buffer                 ; save register number
        mov     Wstate,#Rcv_tpar                ; point to next routine
        jnb     Command.3,Rcv_next_word                 ; If no value, done
        mov     word_count,#7                   ; Init word count
        mov     Wstate,#Rcv_value               ; point to next routine
        jmp     Rcv_next_word

Rcv_value
        rl      Comm_buffer
        rl      Comm_buffer
        rl      Comm_buffer                     ; Shift rcvd bits to MSBs
        mov     Comm_bit_count,#5                       ; shift 5 bits
:shift_loop
        rl      Comm_buffer
        rl      Comm_buf+4
        rl      Comm_buf+3
        rl      Comm_buf+2
        rl      Comm_buf+1
        rl      Comm_buf                        ; Shift a bit from Comm_buf to comm_buf
        djnz    Comm_bit_count,:shift_loop
        djnz    word_count,Rcv_next_word        ; any more words?
        mov     Wstate,#Rcv_tpar                ; point to next routine
        jmp     Rcv_next_word

Rcv_tpar
        mov     W,Tpar                          ; Get result of parity calculations
        jnz     Parity_error                    ; If not 0, parity error
        jmp     Set_rcv_stop                    ; set up to receive stop bit
```

```
;
;       Comm transmit word handler routines
;

Xmit_word_handler
        djnz    Comm_bit_count,Comm_ret          ; If word not complete, continue
Do_xmit_word
        mov     PCLATH,#$<
        mov     W,Wstate
        jmp     W                                ; Execute word handler routine


Send_addr
        mov     Wstate,#Send_cmd>                ; point to next word routine
        mov     W,PIC_addr                       ; get address
        jmp     Send_next_word

Send_cmd
        mov     Wstate,#Send_value>              ; point to send value
        jnb     Command.4,Value_is_next                  ; Is it fast read command?
        mov     Wstate,#Send_reg_ID>             ; point to next word routine
Value_is_next
        mov     W,Command                        ; get command
        jmp     Send_next_word

Send_reg_ID
        mov     Wstate,#Send_value>              ; point to next word routine
        mov     W,Register_ID                    ; get Register ID
        jmp     Send_next_word

Send_value
        mov     Wstate,#Next_value>              ; point to next word routine
        mov     Word_count,#7                    ; value is 7 words long


Next_value
        mov     Comm_bit_count,#5
:loop
        rl      comm_buf+4
        rl      comm_buf+3
        rl      Comm_buf+2
        rl      Comm_buf+1
        rl      Comm_buf
        rl      Comm_buffer
        djnz    Comm_bit_count,:loop
        mov     W,Comm_buffer                            ; Get 5-bit word
        djnz    word_count,Send_next_word        ; if not last word, all set
        mov     Wstate,#Send_Tpar>               ; send parity next
        mov     W,Comm_buffer
        jmp     Send_next_word

Send_Tpar
        mov     Wstate,#Send_stop                ; point to next word routine
        mov     W,Tpar                           ; get transverse parity
```

```
        jmp     Send_next_word

Send_stop
        jmp     Set_edge_wait                   ; Go back to receive mode

;
;
;       Process received buffer
;
        org     100h

Cmd_exec
Process_cmd
        cjbe    Command,#7h,Fast_register_read      ; is it fast read?
        cje     Command,#10h,read_register  ; is it read register command?
        cje     Command,#18h,write_register ; is it write register?
        cje     Command,#19h,reset_status_bits      ; is it reset status?
        jmp     Cmd_process_done
Fast_register_read
        mov     register_ID,command ; Register number is same as command code
Read_register
        mov     W,#27
        mov     W,register_ID-W             ; Get index of register to read
        jnc     Cmd_process_done    ; if negative, not a valid register
        mov     Temp,W                      ; save index
        mov     PCLATH,#$<          ; set up High-order PC bits
        mov     W,Temp                      ; get index
        jmp     PC+W
        jmp     cpy_output_reg
        jmp     cpy_reg_28
        jmp     cpy_reg_29
        jmp     cpy_reg_30
Cpy_reg_31                          ; Reg 31 = Status register
        mov     Comm_buf,Status_reg
        mov     Comm_buf+1,Status_reg+1
        mov     Comm_buf+2,Status_reg+2
        mov     Comm_buf+3,Status_reg+3
        jmp     Setup_reply

Cpy_reg_30                          ; Reg 30 = Serial number
        call    serno
        mov     comm_buf+3,W
        call    serno+1
        mov     comm_buf+2,W
        call    serno+2
        mov     comm_buf+1,W
        call    serno+3
        mov     comm_buf,W
        jmp     Setup_reply

Cpy_reg_29                          ; Reg 29 = Software type and version
        mov     Comm_buf,#Software_type
        mov     Comm_buf+1,#Software_version
        mov     Comm_buf+2,#0
        mov     Comm_buf+3,#0               ; Send Software type and version
```

```
        jmp     Setup_reply

Cpy_reg_28                              ; Reg 28 = Parameter reg
        mov     Comm_buf,Parameter_reg
        mov     Comm_buf+1,Parameter_reg+1
        mov     Comm_buf+2,Parameter_reg+2
        mov     Comm_buf+3,Parameter_reg+3
        jmp     Setup_reply

Cpy_output_reg                          ; Reg 27 = Output state
        mov     Comm_buf,Output_bit_reg
        mov     Comm_buf+1,Output_bit_reg+1
        mov     Comm_buf+2,Output_bit_reg+2
        mov     Comm_buf+3,Output_bit_reg+3
        jmp     Setup_reply

Write_register
        mov     Comm_bit_count,#5                       ; need to shift buffer by 5
bits
:loop
        rl      comm_buf+4
        rl      comm_buf+3
        rl      comm_buf+2
        rl      comm_buf+1
        rl      comm_buf
        djnz    Comm_bit_count,:loop
        cjne    Register_ID,#28,See_write_output_reg ; Not reg 28, see if output reg
        mov     Parameter_reg,Comm_buf
        mov     Parameter_reg+1,Comm_buf+1
        mov     Parameter_reg+2,Comm_buf+2
        mov     Parameter_reg+3,Comm_buf+3      ; Copy received data to reg
        clrb    Status_LSB.Need_param_refresh_flag ; Reset need parameter refresh flag
        jmp     Cpy_reg_28                      ; send contents back in reply
See_write_output_reg
        cjne    Register_ID,#27,Cmd_process_done ; if not output reg, ignore
        mov     Output_bit_reg,Comm_buf
        mov     Output_bit_reg+1,Comm_buf+1
        mov     Output_bit_reg+2,Comm_buf+2
        mov     Output_bit_reg+3,Comm_buf+3     ; Copy received data to reg
        call    Update_outputs
        jmp     Cpy_output_reg                  ; send contents back in reply

Reset_status_bits
        mov     Comm_bit_count,#5                       ; need to shift buffer by 5
bits
:loop
        rl      comm_buf+4
        rl      comm_buf+3
        rl      comm_buf+2
        rl      comm_buf+1
        rl      comm_buf
        djnz    Comm_bit_count,:loop
        mov     W,/Comm_buf+1                   ; get byte
        and     Status_reg+1,W                  ; Clear bits
        mov     W,/Comm_buf                     ; get MSB
```

```
        and     Status_reg,W                            ; Clear bits
        mov     Register_ID,#31                                 ; indicate status reg in reply
        jmp     Cpy_reg_31

Setup_reply
        jb      Flags.Global_cmd_flag,Set_edge_wait     ; If it was global, don't reply
        mov     W,Status_LSB                            ; Get status LSB
        mov     Comm_buf+4,W                                    ; preset stored flags
        and     W,#01fh                                 ; Turn off 3 MSBs
        or      W,Status_MSB                            ; Combine 5 LSbs with Status_MSB
        sz
        setb    Comm_buf+4.7                            ; If any other flag bits set, indicate in
MSB
        mov     Tpar,#1fh                               ; Initialize parity word
        mov     Comm_state,#Send_edge                           ; Go to send_edge state
        mov     Wstate,#Send_addr                       ; Point to next routine
        mov     Comm_bit_count,#Num_start_bits                  ; Init number of bits
        jmp     Comm_ret

Cmd_process_done
        jmp     Set_edge_wait                           ; No reply needed, done

;
; Exit routines for comm
;

Parity_error
        setb    status_LSB.Parity_err_flag      ; Indicate parity error
Set_edge_wait
        mov     Comm_state,#Edge_wait                   ; Go to edge wait state
        mov     Comm_bit_count,#Num_start_bits          ; Init number of bits
        jmp     Comm_ret

Set_send_reply
        mov     Comm_state,#Send_edge                   ; Go to send edge state
        mov     Comm_bit_count,#Num_start_bits
        jmp     Comm_ret

Rcv_next_word
        mov     Comm_state,#Rcv_bits            ; set up to receive 6-bit word
        mov     Comm_bit_count,#6
        jmp     Comm_ret

Send_next_word
        and     W,#01fh                                 ; only 5 bits
        xor     Tpar,W                          ; update parity calculation
        call    parity_lookup
        mov     Comm_buffer,W                           ; Save word with parity (6 bits)
        rl      Comm_buffer
        rl      Comm_buffer                     ; move to MSbs
        mov     Comm_state,#Send_bits           ; set up to send 6-bit word
        mov     Comm_bit_count,#6
        jmp     Comm_ret

Set_rcv_stop
```

```
        mov     Comm_state,#Rcv_stop        ; point to next routine
        jmp     Comm_ret

        ORG     200h
Serno
;       retw    05ah,092h,05fh,000h
;       retw    0ffh,0ffh,0ffh,0ffh
        ds      4
        ret

;
;
;       Returns 6-bit value corresponding to passed 5-bit value with odd parity
;
parity_lookup
        mov     Temp,W              ; Save index
        mov     PCLATH,#:table<     ; Set up high-order bits
        mov     W,Temp              ; get index
        jmp     pc+w
:table
        retw
        1,2,4,7,8,11,13,14,16,19,21,22,25,26,28,31,32,35,37,38,41,42,44,47,49,50,52,55,56,5
9,61,62


;
;       Sets output bits according to received data
;       Comm_buf = Pulse duration in milliseconds
;                       0:      Continuous output (infinite duration)
;                       1-255:  Set output state for specified duration
;                               After duration is over, turn all outputs off
;
;       Comm_buf+1[3] = Bit map of output states - MSB first
;                       bits 23-18:     Unused
;                       bits 17-0:      Outputs 18-1
;

Update_outputs
        mov     W,Comm_buf+3                ; get lsbs
        and     W,#03h              ; only need 2 bits
        mov     Temp,W
        add     Temp,W                      ; Shift into bits 1 and 2 (X1 and X2)
        mov     W,PORTA                     ; Get present status of port A pins
        and     W,#0f9h                     ; clear output control bits
        or      W,Temp                      ; Set desired output state
        mov     PORTA,W                     ; output new value
        rr      comm_buf+1
        rr      comm_buf+2
        rr      comm_buf+3
        rr      comm_buf+1
        rr      comm_buf+2
        rr      comm_buf+3          ; shift buffer left 2 bits
        mov     PORTB,comm_buf+2
        mov     PORTC,comm_buf+3
        mov     !PORTA,#init_TRISA
```

```
        mov     !PORTB,#init_TRISB
        mov     !PORTC,#init_TRISC
        mov     Millisecs,Comm_buf      ; first byte is pulse duration
        jz      done_output_set
ms_loop
        mov     Timeout_ctr,#ms_preset<
        mov     Timeout_ctr+1,#ms_preset>
:loop
        djnz    Timeout_ctr+1,:loop
        djnz    Timeout_ctr,:loop
        djnz    Millisecs,ms_loop       ; delay for pulse duration
        and     PORTA,#0f9h             ; turn off X1 and X2
        clr     PORTB
        clr     PORTC                   ; turn off other outputs
done_output_set
        ret



;
;       Main Program start
;

Main_loop
        mov     !PORTA,#Init_TRISA  ; Initialize port A direction
        mov     !PORTB,#Init_TRISB  ; Initialize port B direction
        mov     !PORTC,#Init_TRISC  ; Init port C direction
        mov     Clk_timeout_ctr,#12 ; preset ...
        clr     Clk_timeout_ctr+1   ; ... clock timeout counter
        clrb    Flags.Clk_wait_flag ; Indicate waiting for clock low
Clkwait_loop
        djnz    scratch,Clk_cont
        clr     WDT                             ; Reset watchdog
        djnz    Clk_timeout_ctr+1,Clk_cont
        djnz    Clk_timeout_ctr,Clk_cont
        jb      Status_MSB.POR_flag,got_POR     ; Ignore clock timeout until POR
flag cleared
        setb    Status_MSB.Clk_timeout_flag ; Indicate timeout on comm clock
        jmp     do_initialize
Clk_cont
        jb      Comm_port.clock_in,:high        ; clock high
:low
        jb      Flags.Clk_wait_flag,Clkwait_loop
        mov     Input_hold,Comm_port            ; Save Data port value
        setb    Flags.Clk_wait_flag             ; Indicate waiting for clock high
        jmp     Clkwait_loop
:high
        jnb     Flags.Clk_wait_flag,Clkwait_loop
        djnz    NPR_timer+1,:cont               ; Dec LSB of need parm timer
        djnz    NPR_timer,:cont                         ; Dec MSB
        setb    Status_LSB.Need_param_refresh_flag  ; set flag every 64k clocks
:cont
        mov     PCLATH,#0                       ; Set up high-order bits for routines in page 0
        mov     W,Comm_state                    ; get vector
        jmp     W..                             ; Execute comm state machine
```

```
Comm_ret
        jmp     Main_loop
```

PIC.DEF

```
/*
include file for serial communication with PIC chips
*/

#pragma switch(ALL,FREQ)

 switch (PIC_serial_status)
   {
   case sending_bits:
   case sending_st5_bits:

#ifdef VATEST

     out_bit_value = (PICxmitBuf.bits.bufLong[0] and 0x80000000) != 0;   /* out bit value =
MSB */
     PICxmitBuf.bits.bufLong[0] = PICxmitBuf.bits.bufLong[0] << 1;
     if (PICxmitBuf.bits.bufLong[1] and 0x80000000)
       PICxmitBuf.bits.bufLong[0]++;
     PICxmitBuf.bits.bufLong[1] = PICxmitBuf.bits.bufLong[1] << 1;
     if (PICxmitBuf.bits.bufLong[2] and 0x80000000)
       PICxmitBuf.bits.bufLong[1]++;
     PICxmitBuf.bits.bufLong[2] = PICxmitBuf.bits.bufLong[2] << 1;     /* Shift 96-bit
buffer */

#else
     out_bit_value = 0;
     shift_LSB_ptr = andPICxmitBuf.bits.bufByte[12];  /* point past least significant word
of 96-bit register */
      asm(" MOVE.L  _shift_LSB_ptr,A0",          /* Get pointer to buffer, clear carry */
        " ROXL.W   -(A0)",                  /* rotate left and decrement pointer */
        " ROXL.W   -(A0)",                  /* rotate left and decrement pointer */
        " ROXL.W   -(A0)",                  /* rotate left and decrement pointer */
        " ROXL.W   -(A0)",                  /* rotate left and decrement pointer */
        " ROXL.W   -(A0)",                  /* rotate left and decrement pointer */
        " ROXL.W   -(A0)",                  /* 96 bits have been shifted, MSB is in carry
*/
        " ROXL.W   _out_bit_value"          /* Put MSB in out_bit_value */
        );

#endif
     break;
   case do_send_cmd:
     bit_count = NUM_START_BITS-1;
     PIC_serial_status = sending_start;
     out_bit_value = 0;
     break;
   case do_PIC_start :
     bit_count = 3 * BITS_PER_SEC;
     out_bit_value = 1;
     PIC_serial_status = sending_PIC_start;
     break;
   case sending_start:
     out_bit_value = 0;
     break;
```

```
    case sending_stop:
    case sending_start_1:
      out_bit_value = 1;
      break;
    default:
      out_bit_value = 1;
    }

 if (out_bit_value)
   fs1004.io.pulseOut2Off = 1;
 else
   fs1004.io.pulseOut2Off = 0;            /* Send bit out */
 in_bit_value = fs1004.io.pulseIn1State;    /* Get rcvd bit value */
 fs1004.io.pulseOut1Off = 1;              /* Send clock rising edge */

 switch (PIC_serial_status)
   {
   case waiting_for_start:
     if (in_bit_value == 0)
       zero_count++;
     else
       {
       if (zero_count == (NUM_START_BITS-1))
         {
         PIC_serial_status = rcving_bits;
         bit_count = PICrcvBuf.bitCount + 1;
         }
       else
         zero_count = 0;
       }
     break;
   case rcving_bits:

#ifdef VATEST

     PICrcvBuf.bits.bufLong[0] = PICrcvBuf.bits.bufLong[0] << 1;
     if (PICrcvBuf.bits.bufLong[1] and 0x80000000)
       PICrcvBuf.bits.bufLong[0]++;
     PICrcvBuf.bits.bufLong[1] = PICrcvBuf.bits.bufLong[1] << 1;
     if (PICrcvBuf.bits.bufLong[2] and 0x80000000)
       PICrcvBuf.bits.bufLong[1]++;
     PICrcvBuf.bits.bufLong[2] = (PICrcvBuf.bits.bufLong[2] << 1) +
       in_bit_value;
#else

     shift_LSB_ptr = andPICrcvBuf.bits.bufByte[12];   /* point past least significant word
of 96-bit register */
      asm(" MOVE.L  _shift_LSB_ptr,A0",          /* Get pointer to buffer, clear carry */
        " ROXL.W   -(A0)",                  /* rotate left and decrement pointer */
        " ROXL.W   -(A0)",                  /* rotate left and decrement pointer */
        " ROXL.W   -(A0)",                  /* rotate left and decrement pointer */
        " ROXL.W   -(A0)",                  /* rotate left and decrement pointer */
        " ROXL.W   -(A0)",                  /* rotate left and decrement pointer */
        " ROXL.W   -(A0)"                   /* 96 bits have been shifted, LSb is unknown
*/
```

```
        );
      PICrcvBuf.bits.bufByte[11] =
        (PICrcvBuf.bits.bufByte[11] and 0xfe) | in_bit_value;        /* add in received bit */

#endif

      break;
    default:
      {
      }
    }

  if (--bit_count <= 0)
    {
    switch (PIC_serial_status)
      {
      case rcving_bits:
        PIC_serial_status = rcving_stop;
        bit_count = 1;
        break;
      case sending_bits:
        PIC_serial_status = sending_stop;
        bit_count = 1;
        break;
      case sending_stop:
        if (PICxmitBuf.PIC_addr == GLOBAL_PIC_ADDR)
          PIC_serial_status = idle;
        else
          {
          PIC_serial_status = waiting_for_start;
          bit_count = PIC_wait_limit;
          zero_count = 0;
          }
        break;
      case sending_PIC_start:
        PIC_serial_status = idle;
        break;
      case sending_start:
        PIC_serial_status = sending_start_1;
        bit_count = 1;
        break;
      case sending_start_1:
        PIC_serial_status = sending_bits;
        bit_count = PICxmitBuf.bitCount;
        break;
      case waiting_for_start:
        PICrcvBuf.flags |= rcv_timeout_flag | rcv_data_ready_flag;
        PIC_serial_status = idle;
        break;
      case rcving_stop:
        if (in_bit_value == 0)
          PICrcvBuf.flags |= rcv_bad_length_flag;
        PIC_serial_status = idle;
        PICrcvBuf.flags |= rcv_data_ready_flag;
        bit_count = 1;
```

```
        break;
      default:
        PIC_serial_status = idle;
      }
    }

#pragma switch(ALL,NOFREQ)
```

PULSE.C

```
#include "mtrlink.def"
 /* #include "flash.def" */
#define PSTRU_DEFINED
#include "pulselink.def"

#include "pulse.h"
#include "ufloat.h"

#include "log.h"

#include "alarm.h"

#include "copymem.h"

#include "plcctrl.h"

#include <stdio.h>


#ifdef fakeMC5
   #undef NUMPH
   #define NUMPH 24
#endif

/* define WDT flags for the PIC communication routine */
#pragma region ("ram=WDTFlags")
short int  WDTpulseSec ;
#pragma region ("ram=ram")


#pragma region ("data=secondBack")
void (*pulseSecondp)(void)=pulseSecond;
#pragma region ("data=data")

#ifndef IS_MC5
#pragma region ("data=everySubsecond")
#ifdef IS_RSM
void (*pulseSubsecp)(void)=pulseSubsecond ;
#else
void (*pulseSubsecp)(void)=pulseService;
#endif
#pragma region ("data=data")
#endif

#pragma region ("data=powerUp")
void (*pulseStartupp)(void)=pulseStartup ;
#pragma region ("data=data")

#pragma region ("data=dayBack")
void (*pulseDayp)(void)=pulseDay ;
#pragma region ("data=data")
```

```
/* clear out PIC */
void pulseColdstart()
{
   int i ;
   for (i=0; i<num_PICs; i++)
      if (PIC_status[i].software_type == 3)
         {
         PIC_status[i].clear_pulse_regs = TRUE;
         PIC_status[i].get_status = TRUE;
         }
   clear_state = SEND_GLOBAL_CLR;
}

void pulseStartup(void)
{
   int i;

#ifndef IS_RSM
   int work_PIC_addr;
#endif
#ifdef IS_MC5
   int work_num_pulses;
#endif

#ifdef VATEST
/* test !!!!! */
   temp_command = SYSTEM_CONTROL;
   temp_reg_ID = 0;
   temp_command_data = READ_ENCODER_ID;
   temp_flags = rcv_data_ready_flag;
#endif

   for (i=0;(!dont_clear_PIC_stats) andand (i<(2*NUMPH));i++)
      {
      pulseData[i].error_cnt = 0;
      pulseData[i].ID_field[0]=0;
      pulseData[i].reading = 0;
      }

   for (i=0; i<MAX_PICS; i++)
      PIC_status[i].PIC_addr = 0;

   hold_num_pulse_ctrs = releaseCodep->option.numPulseCounters;
   if (hold_num_pulse_ctrs > NUMPH)
      num_pulse_ctrs = NUMPH;
   else
      num_pulse_ctrs = hold_num_pulse_ctrs;

   num_PICs = 1;

#ifdef IS_MC5
   PIC_status[0].PIC_addr = MC5_MUX_PIC_ADDR;  /* If MC5, add PDM PIC(s) to PIC
table */
```

```
  if(int_configltmessg.configLTM[PULSE1].accum)
    {
    work_num_pulses = num_pulse_ctrs;
    work_PIC_addr = 0;
    while (work_num_pulses > 0)
      {
      PIC_status[num_PICs++].PIC_addr = work_PIC_addr++;  /* Pulse counters for Q13
*/
      work_num_pulses -= 4;
      }
    }
  if(int_configltmessg.configLTM[PULSE2].accum)
    {
    work_num_pulses = num_pulse_ctrs;
    work_PIC_addr = 6;
    while (work_num_pulses > 0)
      {
      PIC_status[num_PICs++].PIC_addr = work_PIC_addr++;  /* Pulse counters for Q14
*/
      work_num_pulses -= 4;
      }
    }
  num_LCMs = releaseCodep->option.LCM_flags and 0x03;
  work_PIC_addr = LCM_BASE_ADDR;
  for (i=0;i<num_LCMs;i++)
    {
    PIC_status[num_PICs++].PIC_addr = work_PIC_addr++;  /* LCM's */
    }
#else
#ifdef IS_ST5
  PIC_status[0].PIC_addr = ST5_MUX_PIC_ADDR;  /* If ST5, add mux to PIC table */
  ATM_work = releaseCodep->option.couplers_mask;
  work_PIC_addr = ATM_BASE_ADDR;
  for (i=0;i<4;i++)
    {
    if ((ATM_work and 0x08) != 0)
      {                  /* Bit is set in mask, add ATM PIC to table */
      PIC_status[num_PICs++].PIC_addr = work_PIC_addr;
      }
    work_PIC_addr++;
    ATM_work = ATM_work << 1;
    }

#else
  if (num_pulse_ctrs == 0)
    num_PICs = 0;
#endif
#endif
  for (i=0; i<MAX_PICS; i++)
    {                                /* Build PIC table */
    PIC_status[i].get_serno = TRUE;
    PIC_status[i].get_version = TRUE;
    PIC_status[i].get_status = FALSE;
    PIC_status[i].reset_status = FALSE;
    PIC_status[i].update_parameter = FALSE;
```

```
    if (!dont_clear_PIC_stats)
      {
      PIC_status[i].PIC_reset_count = 0;
      PIC_status[i].comm_error_count = 0;
      PIC_status[i].PIC_data_err_count = 0;
      PIC_status[i].software_type = 0;
      PIC_status[i].software_version = 0;
      PIC_status[i].serial_number = 0;
      }

    PIC_status[i].curr_pulse_reg = 0;

#ifdef IS_MC5

    if (PIC_status[i].PIC_addr >= ST5_MUX_PIC_ADDR)
      PIC_status[i].reply_wait_limit = DIRECT_PIC_WAIT_LIMIT;
    else
      PIC_status[i].reply_wait_limit = BUFFERED_PIC_WAIT_LIMIT;
#else
    PIC_status[i].reply_wait_limit = DIRECT_PIC_WAIT_LIMIT;
#endif

    if (PIC_status[i].PIC_addr <= MAX_PULSE_PIC_ADDR)
      PIC_status[i].num_pulse_regs = PULSES_PER_PIC;
    else
      PIC_status[i].num_pulse_regs = 0;

    }

  current_PIC_index = 0;
  rcv_PIC_index = 0;

  if (dont_clear_PIC_stats)
    dont_clear_PIC_stats = FALSE;  /* Clear flag */
  else
    {
    PIC_bad_addr_count = 0;
    PICrcvBuf.flags = 0;
    PIC_serial_status = do_PIC_start;
    PIC_clk_state = clk_low;              /* Initialize serial comm variables */
    }

  clear_state = NO_GLOBAL_SEND;

  pulseOutMode = releaseCode.option.picMode ;   /* for compatibility with old code */

  if (startup.coldStart)
    {
    pulseColdstart() ;
    }
}
```

```
const unsigned char parity6[32] =
{1,2,4,7,8,11,13,14,16,19,21,22,25,26,28,31,32,35,37,38,41,42,44,47,49,50,52,55,56,59,61,62
};


void pulseDay()
{
   dont_clear_PIC_stats = TRUE;   /* Tell pulseStartup not to clear error counts */
   rebuild_timer = REBUILD_DELAY;  /* Set up for delayed table rebuild */
}

void pulseSecond()
{
#ifndef IS_ST5
  pulseService();
#endif
  if (Comm_background_flags.do_rebuild)
     {
     pulseStartup();
     Comm_background_flags.do_rebuild = FALSE;
     }
  if (Comm_background_flags.do_alarm)
     {
     putAlarm(andPIC_alarm,4);
     Comm_background_flags.do_alarm = FALSE;
     }
}

/* set_PLC_relays function                    */
/* Called from PLC routines to request update of   */
/* PLC port multiplexer PIC chip on ST5 power board */

#ifdef IS_ST5

void set_PLC_relays(int xmitRelay,int rcvRelay)
{
  while (MUX_control.active_flag || MUX_control.request_flag)
     {
     ccwait();
     }
   MUX_control.xmit_mask = xmitRelay;
   MUX_control.rcv_mask = rcvRelay;
   MUX_control.done_flag = FALSE;
   MUX_control.request_flag = TRUE;
   if ((MUX_control.xmit_mask != MUX_control.last_xmit_mask) ||
      (MUX_control.rcv_mask != MUX_control.last_rcv_mask))
      {
      while (!MUX_control.done_flag andand
           !(MUX_control.active_flag andand (PIC_serial_status == rcving_bits)) )
        {
        ccwait();
        }
      }
}
```

```
#endif

void pulseService(void)
{
  BOOLEAN   need_pulse_read,
            copy_rcvd_data,
            rcv_buf_err;

#ifdef IS_MC5
  BOOLEAN
            got_ver_3,
            got_ver_3_3;

  int       j;

  unsigned char *work_char_ptr;

  unsigned long work_bit_mask;

#endif

  int     work_pulse_index;
  phaccum accumXfer;
  int i ;

  if (num_PICs == 0)
     PIC_serial_status = idle;

  if (PIC_serial_status == idle)
     {
     if (!WDTpulseSec)
        WDTpulseSec=1; /*Tell WDT controller that we are OK*/
     }

  if (hold_num_pulse_ctrs != releaseCodep->option.numPulseCounters)
     {                                /* hdwr -n has changed */
     dont_clear_PIC_stats = FALSE;
     rebuild_timer = 1;       /* Force immediate table rebuild */
     }

  if ((PIC_serial_status == idle) andand
     (rebuild_timer != 0) andand
     (--rebuild_timer == 0))
     {
     Comm_background_flags.do_rebuild = TRUE;
     }
  else if ((PIC_serial_status == idle) andand
        (num_PICs != 0) andand
        (pulseOutMode != PIC_MC_SERIAL) andand
        (!Comm_background_flags.do_rebuild))       /* Serial processing enabled? */
     {
     if (PICrcvBuf.flags and rcv_data_ready_flag)
        {                                /* Process received message */
        PICrcvBuf.flags and= ~rcv_data_ready_flag;
```

```
        rcv_buf_err = !decode_rcv_buffer();

        if ((PIC_comm_mon.control_flags and RCV_DATA_RDY) == 0)
          {
          PIC_comm_mon.rcv_buf = PICrcvBuf;
          PIC_comm_mon.control_flags |= RCV_DATA_RDY;
          }

        if ((ext_comm.control_flags and CPY_RCV_DATA) != 0)
          {
          ext_comm.rcv_buf = PICrcvBuf;
          ext_comm.control_flags and= ~CPY_RCV_DATA;
          ext_comm.control_flags |= RCV_DATA_RDY;
          }

#ifdef IS_ST5
        if (ATM_control.active_flag)
          {
          ATM_control.result_flags = PICrcvBuf.flags;
          if (rcv_buf_err)
            {
            if (ATM_control.state != WAIT_ALIGN)
              {
              ATM_control.error_flag = TRUE;
              ATM_control.active_flag = FALSE;
              ATM_control.done_flag = TRUE;
              }
            }
          else
            {
            switch (ATM_control.state)
              {
              case DO_PRESET:
                {
                ATM_control.xmit_level = PICrcvBuf.command_data >> 24;
                ATM_control.coupler_level = (PICrcvBuf.command_data >> 16) and 0x0ff;
                ATM_control.active_flag = FALSE;
                }
                break;
              case SET_ALIGN:
                {
                ATM_control.state = WAIT_ALIGN;
                }
                break;
              case WAIT_ALIGN:
                {
                ATM_control.optimum_cap_code = PICrcvBuf.command_data and
 ATM_RC_CAP_MASK;
                ATM_control.optimum_reading = PICrcvBuf.command_data >> 16;
                ATM_control.result_flags |= (PICrcvBuf.command_data and 0xe000);
                                        /* Copy flags from reply */
                if ((ATM_control.result_flags and 0xc000) != 0)
                  {                     /* Analog levels out of range */
                  ATM_control.error_flag = TRUE;
                  }
```

```
            if (ATM_control.optimum_reading < ATM_MIN_READING)
              {
              ATM_control.result_flags |= 0x1000; /* Indicate reading too low */
              ATM_control.error_flag = TRUE;
              }
            ATM_control.active_flag = FALSE;
            }
            break;
          case DO_PLC_SET:
          case DO_ATM_DISCONNECT:
            {
            ATM_control.active_flag = FALSE;
            }
            break;
          default:
            {
            ATM_control.active_flag = FALSE;
            ATM_control.error_flag = TRUE;
            }
            break;
         }
        }
       if (!ATM_control.active_flag)
        ATM_control.done_flag = TRUE;
       }

      if (MUX_control.active_flag)
       {
       MUX_control.result_flags = PICrcvBuf.flags;
       if (rcv_buf_err)
        {
        MUX_control.error_flag = TRUE;
        MUX_control.last_xmit_mask = 0;
        MUX_control.last_rcv_mask = 0;
        }
       else
        {
        MUX_control.last_xmit_mask = PICrcvBuf.command_data and 0xff;
        MUX_control.last_rcv_mask = (PICrcvBuf.command_data >> 8) and 0xff;
        }
       MUX_control.active_flag = FALSE;
       MUX_control.done_flag = TRUE;
       }
#endif

      if (rcv_buf_err)
       {
#ifdef IS_ST5
       if (!ATM_control.active_flag || (ATM_control.state != WAIT_ALIGN))
#endif
        {
        last_comm_err_flag = PICrcvBuf.flags;
        copymem(sizeof (PICbitBuf), (char *) (andHoldrcvBuf), (char *)
(andPICrcvBuf));
        if ((PICrcvBuf.flags and rcv_bad_addr_flag) != 0)
```

```
            {       /* past end of valid PICs, address error */
            PIC_bad_addr_count++;
            set_PIC_alarm(PIC_adderr_0,PICrcvBuf.PIC_addr,
                    (PICxmitBuf.PIC_addr << 8)+PICxmitBuf.command);
            }
          else
            {       /* PIC address was valid, but comm failed */
            PIC_status[rcv_PIC_index].comm_error_count++;
            if (PIC_status[rcv_PIC_index].comm_error_count > 2)
              {
              set_PIC_alarm(PIC_comerr_0,PIC_status[rcv_PIC_index].PIC_addr,
                    PICrcvBuf.flags);
              }
                  /* Lost contact with PIC, read serial number and software version */
            PIC_status[rcv_PIC_index].get_serno = TRUE;
            PIC_status[rcv_PIC_index].get_version = TRUE;
            }
          }
        }
      else
        {                           /* Valid message received */
        if ((PICrcvBuf.command == READ_REGISTER) andand (PICrcvBuf.reg_ID ==
STATUS_REG))
            {
            process_status_bits();
            }
          else if (PICrcvBuf.command == SYSTEM_CONTROL)
            {
            if (PICrcvBuf.reg_ID == STATUS_REG)
              {                     /* Response frome reset status bits cmd */
              PIC_status[rcv_PIC_index].reset_status = FALSE;
              process_status_bits();
              }
#ifdef IS_MC5
            else if ( (PIC_status[rcv_PIC_index].software_type == 4) andand
                  ((PIC_status[rcv_PIC_index].last_cmd_data and 0xffff0000) ==
READ_ENCODER_ID) )
              {                     /* response from encoder read ID comd */
              work_pulse_index = (PICrcvBuf.PIC_addr * PULSES_PER_PIC) +
PICrcvBuf.reg_ID;
              i = (PIC_status[rcv_PIC_index].last_cmd_data and 0x0000ff00) >> 8;
              work_char_ptr = (unsigned char *) andPICrcvBuf.command_data;
              for (j=0; j<4 ; j++)
                {
                if ((*work_char_ptr == 0) || (i >= ID_FIELD_LENGTH))
                  {
                  j = 10;    /* indicate end of message found */
                  }
                else
                  {
                  pulseData[work_pulse_index].ID_buffer[i] = *work_char_ptr;
                  i++;
                  work_char_ptr++;
                  }
                }
```

```
            if (j>5)
              {                    /* found end of ID field from remote encoder */
              for (i=0; i<ID_FIELD_LENGTH; i++)
                {
                pulseData[work_pulse_index].ID_field[i] =
pulseData[work_pulse_index].ID_buffer[i];
                pulseData[work_pulse_index].ID_buffer[i] = 0;
                }
              }
            }
          }
        else if ((PICrcvBuf.command == READ_REGISTER) andand (PICrcvBuf.reg_ID
== ERROR_REG))
          {
          if (PIC_status[rcv_PIC_index].software_type == 4)
            {                  /* Remote encoder interface module */
            work_pulse_index = (PICrcvBuf.PIC_addr * PULSES_PER_PIC) +
(PICrcvBuf.command_data >> 24);
            if (work_pulse_index < (2 * NUMPH))
              {
              pulseData[work_pulse_index].error_cnt++;
              for (i=0; i<ID_FIELD_LENGTH;
pulseData[work_pulse_index].ID_buffer[i++] = 0);
              }
            }
#endif
          }
        else if ((PICrcvBuf.command == WRITE_REGISTER) andand (PICrcvBuf.reg_ID
== PARM_REG))
          {
          PIC_status[rcv_PIC_index].update_parameter = FALSE;
          }
        else if ((PICrcvBuf.command == READ_REGISTER) andand (PICrcvBuf.reg_ID
== VERS_REG))
          {
          PIC_status[rcv_PIC_index].get_version = FALSE;
          PIC_status[rcv_PIC_index].software_type =
          (PICrcvBuf.command_data and 0xff000000) >> 24;
          PIC_status[rcv_PIC_index].software_version =
          (PICrcvBuf.command_data and 0x00ff0000) >> 16;
          PIC_status[rcv_PIC_index].loop_rate =
          (PICrcvBuf.command_data and 0x0000ff00) >> 8;
          }
        else if ((PICrcvBuf.command == READ_REGISTER) andand (PICrcvBuf.reg_ID
== SERNO_REG))
          {
          PIC_status[rcv_PIC_index].get_serno = FALSE;
          PIC_status[rcv_PIC_index].serial_number = PICrcvBuf.command_data;
          }
        else if (((PICrcvBuf.command == READ_REGISTER) || (PICrcvBuf.command <=
MAX_FAST_READ))
            andand (PICrcvBuf.reg_ID <= MAX_PULSE_REG))
          {        /* Pulse reading */
          work_pulse_index = (PICrcvBuf.PIC_addr * PULSES_PER_PIC) +
PICrcvBuf.reg_ID;
```

```
            if (work_pulse_index < (NUMPH * 2))
              {
              pulseData[work_pulse_index].reading = PICrcvBuf.command_data;
              copy_rcvd_data = FALSE;
              if (PIC_status[rcv_PIC_index].get_version)
                {
                }
              else if (PIC_status[rcv_PIC_index].software_type == 4)
                {                   /* Remote encoder interface module */
                copy_rcvd_data = TRUE;
                accumXfer.low = pulseData[work_pulse_index].reading;
                }
              else if (PIC_status[rcv_PIC_index].software_type == 3)
                {                   /* Remote pulse counter module */
                copy_rcvd_data = TRUE;

                if (!PIC_status[rcv_PIC_index].get_serno)
                  {
                  sprintf((char *) pulseData[work_pulse_index].ID_field,"S%08ldP%d",
                        PIC_status[rcv_PIC_index].serial_number,
                        PICrcvBuf.reg_ID);
                  }

                if (releaseCode.option.countEveryEdge)
                   {
                   accumXfer.low = pulseData[work_pulse_index].reading;
                   }
                else
                   {
                   accumXfer.low = pulseData[work_pulse_index].reading >> 1;
                   }
                }
              accumXfer.high = 0L;
#ifndef VATEST

              if (copy_rcvd_data)
                {
#ifndef IS_MC5
            /* It is an RSM or ST5, so mapping for pulse 1 thru 4 is M1Q13, M1Q14, M2Q13,
M2Q14 */

ph[0][work_pulse_index/2][(work_pulse_indexand0x01)?PULSE2:PULSE1].accum =

ph[1][work_pulse_index/2][(work_pulse_indexand0x01)?PULSE2:PULSE1].accum =
                  accumXfer ;

#else
            /* It is an MC5, so mapping for pulse 1 thru 48 is M1Q13..M24Q13,
M1Q14..M24Q14   */


ph[0][work_pulse_index%NUMPH][(work_pulse_index/NUMPH)?PULSE2:PULSE1].accu
m =
```

```
ph[1][work_pulse_index%NUMPH][(work_pulse_index/NUMPH)?PULSE2:PULSE1].accu
m =
                accumXfer ;
#endif
                }

#endif
              }
            }

          if ((PICrcvBuf.flags and (PIC_error_flag | PIC_invalid_data_flag)) != 0)
            PIC_status[rcv_PIC_index].get_status = TRUE;
          if ((PICrcvBuf.flags and PIC_need_refresh_flag) != 0)
            PIC_status[rcv_PIC_index].update_parameter = TRUE;
          /* check status flags */
          }
        }

      /* Decide what command to send */

      PICxmitBuf.flags = 0;

#ifndef IS_ST5

      /* MC5 or RSM, check for cold-start processing to clear pulse registers */

      if (clear_state == SEND_GLOBAL_CLR)
        {                              /* Cold start, send global clr */
        PICxmitBuf.PIC_addr = GLOBAL_PIC_ADDR;
        PICxmitBuf.command = CLEAR_PULSE_REGISTERS;
        format_xmit_buffer();
        clear_state = SENDING_GLOBAL_CLR;
        PIC_serial_status = do_send_cmd;        /* Start UART */
        }
      else if (clear_state == SENDING_GLOBAL_CLR)
        {
        clear_state = GLOBAL_CLR_SENT;
        }
#else


    /* ST5, check for ATM (Automatic Tuning Module) operations */

      if (ATM_control.request_flag)
        {
        ATM_control.request_flag = FALSE;
        ATM_control.error_flag = FALSE;
        ATM_control.active_flag = TRUE;
        ATM_control.state = ATM_DONE;
        PICxmitBuf.PIC_addr = ATM_control.ATM_number + ATM_BASE_ADDR;

        switch (ATM_control.operation)
          {
          case SET_PRE_ALIGN:
```

```
          PICxmitBuf.command = WRITE_REGISTER;
          PICxmitBuf.reg_ID = OUTPUT_REG;
          PICxmitBuf.command_data = ATM_control.start_cap_code and
ATM_RC_CAP_MASK;
          PICxmitBuf.command_data |= ATM_PRESET_CODE;
          ATM_control.state = DO_PRESET;
          break;

        case DO_ALIGN:
          PICxmitBuf.command = WRITE_REGISTER;
          PICxmitBuf.reg_ID = CONTROL_REG;
          if (ATM_control.start_cap_code > ATM_control.end_cap_code)
            ATM_control.end_cap_code = ATM_control.start_cap_code;
          PICxmitBuf.command_data = ATM_control.start_cap_code and
ATM_RC_CAP_MASK;
          PICxmitBuf.command_data = (PICxmitBuf.command_data << 12 ) |
                          (ATM_control.end_cap_code and ATM_RC_CAP_MASK) |
                           (ATM_MEAS_TIME << 24);
          ATM_control.state = SET_ALIGN;
          ATM_control.align_timer = ATM_control.end_cap_code -
ATM_control.start_cap_code;
          ATM_control.align_timer *= ((ATM_DISC_TIME + ATM_MEAS_TIME + 4) *
(1.25 / 62.5));
          ATM_control.align_timer += 13; /* Time in 1/64 sec to wait before checking
result */
                                /* Number of steps * time per step + 25% + .2 second */
          break;

        case SET_NORMAL_PLC:
          PICxmitBuf.command = WRITE_REGISTER;
          PICxmitBuf.reg_ID = OUTPUT_REG;
          PICxmitBuf.command_data = ATM_control.optimum_cap_code and
ATM_RC_CAP_MASK;
          PICxmitBuf.command_data |= ATM_PLC_CODE;
          ATM_control.state = DO_PLC_SET;
          break;

        case DISCONNECT_COUPLER:
          PICxmitBuf.command = WRITE_REGISTER;
          PICxmitBuf.reg_ID = OUTPUT_REG;
          PICxmitBuf.command_data = ATM_control.optimum_cap_code and
ATM_RC_CAP_MASK;
          PICxmitBuf.command_data |= ATM_DISC_CODE;
          ATM_control.state = DO_ATM_DISCONNECT;
          break;


        default:
        case READ_XMIT_LEVEL:
          ATM_control.error_flag = TRUE;
          ATM_control.done_flag = TRUE;
          ATM_control.active_flag = FALSE;
          break;

        }
```

```
    if (ATM_control.active_flag)
      {
      format_xmit_buffer();
      PIC_wait_limit = DIRECT_PIC_WAIT_LIMIT;
      PIC_serial_status = do_send_cmd;        /* Start UART */
      }

    }
  else if (ATM_control.active_flag)
    {
    if ((ATM_control.state != WAIT_ALIGN) ||
      (--ATM_control.align_timer <= 0))
      {
      ATM_control.error_flag = TRUE;
      ATM_control.done_flag = TRUE;
      ATM_control.active_flag = FALSE;
      }
    else
      {                              /* poll ATM for auto-tune end */
      PICxmitBuf.command = 0;
      format_xmit_buffer();
      PIC_wait_limit = DIRECT_PIC_WAIT_LIMIT;
      PIC_serial_status = do_send_cmd;        /* Start UART */
      }
    }

  else if (MUX_control.request_flag)
    {
    MUX_control.request_flag = FALSE;
    MUX_control.error_flag = FALSE;
    if ((pulseOutMode == PIC_ST_SERIAL))
      {
      PICxmitBuf.bits.bufByte[0] = ((MUX_control.rcv_mask and 0x03) << 5) |
                    ((MUX_control.xmit_mask and 0x7f) >> 2);

      PICxmitBuf.bits.bufByte[1] = ((MUX_control.xmit_mask and 0x03) << 6);

      bit_count = 11;
      PIC_serial_status = sending_st5_bits;        /* Start UART */
      MUX_control.last_xmit_mask = MUX_control.xmit_mask;
      MUX_control.last_rcv_mask = MUX_control.rcv_mask;
      MUX_control.done_flag = TRUE;
      }
    else
      {
      MUX_control.active_flag = TRUE;
      PICxmitBuf.PIC_addr = ST5_MUX_PIC_ADDR;
      PICxmitBuf.command = WRITE_REGISTER;
      PICxmitBuf.reg_ID = OUTPUT_REG;
      PICxmitBuf.command_data = (MUX_control.rcv_mask << 8) +
MUX_control.xmit_mask;
      format_xmit_buffer();
      PIC_wait_limit = DIRECT_PIC_WAIT_LIMIT;
      PIC_serial_status = do_send_cmd;        /* Start UART */
```

```
        }
      }

#endif

    else if ((ext_comm.control_flags and (XMIT_DATA_RDY | XMIT_REQUEST)) ==
        (XMIT_DATA_RDY | XMIT_REQUEST))
      {                              /* Send command for external routine */
      PICxmitBuf = ext_comm.xmit_buf;
      ext_comm.control_flags and= ~XMIT_REQUEST;
      ext_comm.control_flags |= CPY_RCV_DATA;

      /*   Look up correct PIC_status index for reuested PIC */
      for (i=0; ((i<MAX_PICS) andand (PIC_status[i].PIC_addr != PICxmitBuf.PIC_addr));
i++);

      if (i < num_PICs)
        {
        PIC_status[i].last_cmd_data = PICxmitBuf.command_data;
        PIC_status[i].last_command = PICxmitBuf.command;
        }

#ifdef IS_MC5
      if (ext_comm.xmit_buf.PIC_addr < ST5_MUX_PIC_ADDR)
        PIC_wait_limit = BUFFERED_PIC_WAIT_LIMIT;
      else
#endif
        PIC_wait_limit = DIRECT_PIC_WAIT_LIMIT;

      format_xmit_buffer();
      PIC_serial_status = do_send_cmd;        /* Start UART */
      }

    else if ( ((ext_comm.control_flags and SUPPRESS_NORMAL_COMM) == 0)
#ifdef IS_ST5
          andand (--ST5_PIC_delay_ctr <= 0))
      {
      ST5_PIC_delay_ctr = ST5_PIC_DELAY;    /* Slow down routine communications in
the ST5 */
#else
                                          )
      {
#endif
      PICxmitBuf.PIC_addr = PIC_status[current_PIC_index].PIC_addr;
      need_pulse_read = TRUE;       /* Set defaults */

#ifdef IS_ST5

      if ((pulseOutMode == PIC_ST_SERIAL) andand
        (PICxmitBuf.PIC_addr == ST5_MUX_PIC_ADDR))
        {
        }
      else
```

```
#endif

        if (PIC_status[current_PIC_index].force_pulse_read andand
           (!PIC_status[current_PIC_index].get_version) andand
           (PIC_status[current_PIC_index].PIC_addr < ST5_MUX_PIC_ADDR) )
           {
           PIC_status[current_PIC_index].force_pulse_read = FALSE;
           }
        else
           {
           PIC_status[current_PIC_index].force_pulse_read = TRUE;

#ifndef IS_ST5
           if (PIC_status[current_PIC_index].clear_pulse_regs andand
              (clear_state == GLOBAL_CLR_SENT) )
              {                          /* Global clear sent, get status */
              need_pulse_read = FALSE;
              PICxmitBuf.command = READ_REGISTER;
              PICxmitBuf.reg_ID = STATUS_REG;
              format_xmit_buffer();
              }
           else

#endif
              if (PIC_status[current_PIC_index].get_version)
              {
              need_pulse_read = FALSE;
              PICxmitBuf.command = READ_REGISTER;
              PICxmitBuf.reg_ID = VERS_REG;
              format_xmit_buffer();
              }
           else if (PIC_status[current_PIC_index].get_serno)
              {
              need_pulse_read = FALSE;
              PICxmitBuf.command = READ_REGISTER;
              PICxmitBuf.reg_ID = SERNO_REG;
              format_xmit_buffer();
              }
           else if (PIC_status[current_PIC_index].update_parameter)
              {
              PICxmitBuf.command = WRITE_REGISTER;
              if (PIC_status[current_PIC_index].software_type == 3)
                 {                                  /* Pulse ctr */

                 /* Pulse ctr parameters:          */
                 /* Power-On sample rate:  42/sec  */
                 /* Power-off sample rate:  1/sec  */
                 /* Days to count if rcving pulses:  */
                 /*      RSM: 5 days               */
                 /*      PDM: 35 days              */
                 /* Days to count if no pulses rcvd:  */
                 /*      RSM: 1 day                */
                 /*      PDM: 3 days               */

                 /* on RSM, the 2 lsbs of          */
```

```
            /* load_shed_mask[0] are sent to    */
            /* the PIC output bits              */
#ifdef IS_RSM
            PICxmitBuf.command_data = 0x01150507 and
                        (load_shed_state[0] | 0xfffffffc);
#else
            PICxmitBuf.command_data = 0x0115230d;
#endif
          }
#ifdef IS_MC5
        else if (PIC_status[current_PIC_index].software_type == 4)
          {                                   /* Remote encoder reader */
          PICxmitBuf.command_data = READ_ENCODER_PARM;
          if ((releaseCodep->option.couplers_mask and 0x01) == 0)
           { /* If mask bit 0 is 0, no touch-pad compatibility */
            PICxmitBuf.command_data and= 0xff00ffff;
           }
          }
        else if (PIC_status[current_PIC_index].software_type == 5)
          {                                   /* MC5 mux */
          got_ver_3 = got_ver_3_3 = FALSE;

          for (i=0; i<num_PICs; i++)
            {
            if (PIC_status[i].get_version)
              {
              got_ver_3 = TRUE;
              got_ver_3_3 = TRUE;
              }
            else if (PIC_status[i].software_type == 3)
              {
              got_ver_3 = TRUE;
              if (PIC_status[i].software_version == 3)
                got_ver_3_3 = TRUE;
              }
            }
            if (got_ver_3)
              {
              if (got_ver_3_3 andand
                ((releaseCodep->option.couplers_mask and 0x02) != 0))
                {
                PICxmitBuf.command_data = 0x0c007000;  /* 32 bps, periodic wake-up
*/
                }
              else
                {
                PICxmitBuf.command_data = 0x0c007080;  /* 32 bps, no wake-up */
                }
              }
            else
              PICxmitBuf.command_data = 0x02030080;  /* 205 bps, no wake-up */

          }
#endif
#ifdef IS_ST5
```

```
            else if (PIC_status[current_PIC_index].software_type == 8)
              {                                   /* Auto tune module */
              PICxmitBuf.command_data = 0x02030000 | (ATM_DISC_TIME << 8) |
                              (ATM_RC_XMIT_ON >> 8) | ATM_NUM_CAP_BITS;
              /* xmit offset = 2, cplr offset = 3,                    */
              /* relay delay = ATM_DISC_TIME, Xmit mask = ATM_RC_XMIT_ON,   */
              /* number of capacitor relays = ATM_NUM_CAP_BITS          */
              }
            else if (PIC_status[current_PIC_index].software_type == 6)
              {                                   /* ST5 mux */
              PICxmitBuf.command_data = 0x00000000;
              }
#endif
            else if (PIC_status[current_PIC_index].software_type == 7)
              {                                   /* Load Control Module */
              PICxmitBuf.command_data = 0x00000000;
              }
            else
              {
              PIC_status[current_PIC_index].update_parameter = FALSE;
                          /* unknown software type, cancel command */
              }

            if (PIC_status[current_PIC_index].update_parameter)
              {
              need_pulse_read = FALSE;
              PICxmitBuf.reg_ID = PARM_REG;
              format_xmit_buffer();
              }
            }
          else if (PIC_status[current_PIC_index].reset_status)
            {
            need_pulse_read = FALSE;
            PICxmitBuf.command = SYSTEM_CONTROL;
            PICxmitBuf.command_data =
            (PIC_status[current_PIC_index].status_reg_value and 0xffbf);
            PICxmitBuf.command_data = PICxmitBuf.command_data << 16;  /* Move
status bits to MSb's */
            PICxmitBuf.reg_ID = STATUS_REG;
            format_xmit_buffer();
            }
          else if (PIC_status[current_PIC_index].get_status)
            {
            need_pulse_read = FALSE;
            PICxmitBuf.command = READ_REGISTER;
            PICxmitBuf.reg_ID = STATUS_REG;
            format_xmit_buffer();
            }

#ifdef IS_MC5
          if (need_pulse_read andand (PIC_status[current_PIC_index].software_type == 4))
            {
            need_pulse_read = FALSE;
            PICxmitBuf.command = SYSTEM_CONTROL;
```

```
            if (++PIC_status[current_PIC_index].curr_ID_reg >= PULSES_PER_PIC)
              PIC_status[current_PIC_index].curr_ID_reg = 0;

            if ( (((PIC_status[current_PIC_index].PIC_addr * PULSES_PER_PIC) +
                 PIC_status[current_PIC_index].curr_ID_reg) % 24) >= num_pulse_ctrs)
                          PIC_status[current_PIC_index].curr_ID_reg = 0;

            PICxmitBuf.reg_ID = PIC_status[current_PIC_index].curr_ID_reg;
            work_pulse_index = (PIC_status[current_PIC_index].PIC_addr *
PULSES_PER_PIC) + PICxmitBuf.reg_ID;
            for (i=0; i<ID_FIELD_LENGTH; i++)
              {
              if (pulseData[work_pulse_index].ID_buffer[i] == 0)
                {
                j = i - 1;
                i = ID_FIELD_LENGTH;
                }
              }

            if (j<0)
              {
              j = 0;
              }
            else if (j > (ID_FIELD_LENGTH-4))
              {
              j = ID_FIELD_LENGTH-4;
              }

            PICxmitBuf.command_data = READ_ENCODER_ID | (j << 8);;
            format_xmit_buffer();
            }
#endif
          }


        if (need_pulse_read)
          {
#ifdef IS_MC5
          if (PIC_status[current_PIC_index].software_type == 7)
            {
            i = PIC_status[current_PIC_index].PIC_addr - LCM_BASE_ADDR;  /* get
LCM # */
            work_bit_mask = (load_shed_state[0] >> (9 * i)) and 0x000001ff;
            PICxmitBuf.command_data = 0;
            for (i=0;i<9;i++)
              {
              if ((work_bit_mask and 0x100) != 0)
                PICxmitBuf.command_data = (PICxmitBuf.command_data << 2) + 1; /* lsb
turns relay on */
              else
                PICxmitBuf.command_data = (PICxmitBuf.command_data << 2) + 2; /* msb
turns relay on */
              work_bit_mask = work_bit_mask << 1;
              }
            PICxmitBuf.command_data |= 0x32000000;     /* Set pulse time = 50 mS */
```

```
            PICxmitBuf.command = WRITE_REGISTER;
            PICxmitBuf.reg_ID = OUTPUT_REG;
            format_xmit_buffer();
            need_pulse_read = FALSE;
            }
          else
#endif
            if (PIC_status[current_PIC_index].num_pulse_regs == 0)
            {
#ifndef IS_ST5
            PICxmitBuf.command = READ_REGISTER;
            PICxmitBuf.reg_ID = VERS_REG;
            format_xmit_buffer();
            need_pulse_read = FALSE;
#endif
            }
          else
            {
#ifdef IS_MC5
            if ( (((PIC_status[current_PIC_index].PIC_addr * PULSES_PER_PIC) +
                    PIC_status[current_PIC_index].curr_pulse_reg) % 24) >=
num_pulse_ctrs)
                PIC_status[current_PIC_index].curr_pulse_reg = 0;
#endif
            if (PIC_status[current_PIC_index].curr_pulse_reg <= MAX_FAST_READ)
              PICxmitBuf.command = PIC_status[current_PIC_index].curr_pulse_reg;
            else
              PICxmitBuf.command = READ_REGISTER;

            PICxmitBuf.reg_ID = PIC_status[current_PIC_index].curr_pulse_reg;
            format_xmit_buffer();
            need_pulse_read = FALSE;
            if (++PIC_status[current_PIC_index].curr_pulse_reg >=
PIC_status[current_PIC_index].num_pulse_regs)
              {
              PIC_status[current_PIC_index].curr_pulse_reg = 0;
#ifdef IS_RSM
              PIC_status[current_PIC_index].update_parameter = TRUE;
#endif
              }
            }
          }

        PIC_status[current_PIC_index].last_cmd_data = PICxmitBuf.command_data;
        PIC_status[current_PIC_index].last_command = PICxmitBuf.command;

        PIC_wait_limit = PIC_status[current_PIC_index].reply_wait_limit;

        rcv_PIC_index = current_PIC_index;

        if (++current_PIC_index == num_PICs)
          current_PIC_index = 0;

        if (!need_pulse_read)
          PIC_serial_status = do_send_cmd;       /* Start UART */
```

```
        }

      }
}

#ifdef IS_RSM   /* In RSM, generate serial stream from subseconds */

void pulseSubsecond(void)
{
  if (num_PICs != 0)
     {
     if (PIC_clk_state == clk_high)
       {
       /* Falling edge, just set clock low */
       fs1004.io.pulseOut1Off = 0;
       PIC_clk_state = clk_low;
       }
     else
       {
       PIC_clk_state = clk_high;

#include "pic.def"   /* Do PIC serial comm on clock rising edge */


       }
     }
}

#endif


void process_status_bits(void)
{
  unsigned int     work_pulse_index, work_status_bits;

   PIC_status[rcv_PIC_index].get_status = FALSE;
   PIC_status[rcv_PIC_index].status_reg_value = PICrcvBuf.command_data >> 16;
   PIC_status[rcv_PIC_index].time_since_global = (PICrcvBuf.command_data >> 8) and
0xff;

   if (PIC_status[rcv_PIC_index].status_reg_value != 0)
      {
      PIC_status[rcv_PIC_index].reset_status = TRUE;
      if ((PIC_status[rcv_PIC_index].status_reg_value and 0x8f10) != 0)
        {
        PIC_status[rcv_PIC_index].PIC_data_err_count++;   /* PIC detected bad data */
        set_PIC_alarm(PIC_flags_0,PIC_status[rcv_PIC_index].PIC_addr,
                PIC_status[rcv_PIC_index].status_reg_value);
        work_status_bits = (PIC_status[rcv_PIC_index].status_reg_value >> 8) and 0x0f;
        work_pulse_index = PIC_status[rcv_PIC_index].PIC_addr * PULSES_PER_PIC;
        while (work_status_bits > 0)
          {
          if (work_status_bits and 0x01)
            pulseData[work_pulse_index].error_cnt++;
```

```
            work_status_bits = work_status_bits >> 1;
            work_pulse_index++;
            }

        }
      if (((PIC_status[rcv_PIC_index].status_reg_value and 0x20) != 0) andand
        !PIC_status[rcv_PIC_index].clear_pulse_regs)
        {
        PIC_status[rcv_PIC_index].comm_error_count++;     /* PIC detected comm parity err
*/
        if (PIC_status[rcv_PIC_index].comm_error_count > 2)
          {
          set_PIC_alarm(PIC_comerr_0,PIC_status[rcv_PIC_index].PIC_addr,
                  PICrcvBuf.flags | 0x0800);
          }
        }
      if ((PIC_status[rcv_PIC_index].status_reg_value and 0x3080) != 0)
        {
        PIC_status[rcv_PIC_index].PIC_reset_count++;      /* PIC was reset */
        if (PIC_status[rcv_PIC_index].PIC_reset_count > 1)
          {
          set_PIC_alarm(PIC_reset_0,PIC_status[rcv_PIC_index].PIC_addr,
                  PIC_status[rcv_PIC_index].status_reg_value);
          }
        }
      }

    if (PIC_status[rcv_PIC_index].clear_pulse_regs andand
      (clear_state == GLOBAL_CLR_SENT) )
      {
      PIC_status[rcv_PIC_index].clear_pulse_regs = FALSE;
      if ( ((PIC_status[rcv_PIC_index].status_reg_value and 0x0f) != 0x0f) andand
         (PIC_status[rcv_PIC_index].software_type == 3) )
        {
        PIC_status[rcv_PIC_index].PIC_data_err_count++;  /* Clear cmd failed */
        set_PIC_alarm(PIC_flags_0,PIC_status[rcv_PIC_index].PIC_addr,
                PIC_status[rcv_PIC_index].status_reg_value);
        }
      }
    else if ( ((PIC_status[rcv_PIC_index].status_reg_value and 0x0f) != 0) andand
          ((PIC_status[rcv_PIC_index].status_reg_value and 0xf10) == 0) )
      {
      PIC_status[rcv_PIC_index].PIC_data_err_count += 1000;     /* Spurious clr cmd */
      set_PIC_alarm(PIC_flags_0,PIC_status[rcv_PIC_index].PIC_addr,
              PIC_status[rcv_PIC_index].status_reg_value);
      }
}

void set_PIC_alarm(alarmcodes pass_alarm_codes,
            int pass_PIC_addr,
            unsigned int pass_alarm_data)
{
  if (!Comm_background_flags.do_alarm)
    {
    PIC_alarm.alarm_codes = pass_alarm_codes + pass_PIC_addr;
```

```
    PIC_alarm.ack = pass_alarm_data;
    Comm_background_flags.do_alarm = TRUE;
    }
}

unsigned char workTpar;

BOOLEAN decode_rcv_buffer(void)
{
  int          i;
  unsigned int  workInt;

  workTpar = 0x1f;

  if (PICrcvBuf.flags == 0)
    {
    work_PIC_addr = Get5();
    if (PICrcvBuf.flags == 0)
      PICrcvBuf.PIC_addr = work_PIC_addr;   /* Valid address received */
    PICrcvBuf.command = Get5();
    if (PICrcvBuf.command > 0x0f)
      PICrcvBuf.reg_ID = Get5();
    else
      PICrcvBuf.reg_ID = PICrcvBuf.command;
    PICrcvBuf.command_data = 0;
    for (i=0; i<6; i++)
      PICrcvBuf.command_data = (PICrcvBuf.command_data << 5) + Get5();
    workInt = Get5();
    PICrcvBuf.command_data = (PICrcvBuf.command_data << 2) + (workInt >> 3);
    PICrcvBuf.flags |= workInt and 0x07;
    workInt = Get5();
    if (workTpar != 0)
      PICrcvBuf.flags |= rcv_tpar_error_flag;
    }
  /*  Look up correct PIC_status index for data in rcv buffer */
  for (i=0; ((i<MAX_PICS) andand (PIC_status[i].PIC_addr != PICrcvBuf.PIC_addr)); i++);

  if (i >= num_PICs)
    PICrcvBuf.flags |= rcv_bad_addr_flag; /* past end of valid PICs, address error */

  if (i == MAX_PICS)
    rcv_PIC_index = 0;
  else
    rcv_PIC_index = i;

#ifdef VATEST

#if VATEST==1
  PICrcvBuf.command = temp_command;
  PICrcvBuf.reg_ID = temp_reg_ID;
  PICrcvBuf.command_data = temp_command_data;
  PICrcvBuf.flags = temp_flags;

#endif
#endif
```

```
    return((PICrcvBuf.flags and
        ~(PIC_error_flag | PIC_need_refresh_flag | PIC_invalid_data_flag)) == 0);
}

void format_xmit_buffer(void)
{
    int         i,j,ShiftCount;
    unsigned int  WorkInt;
    unsigned long  WorkData;

    PICxmitBuf.bitCount = 0;
    PICxmitBuf.bits.bufLong[0] = 0;
    PICxmitBuf.bits.bufLong[1] = 0;
    PICxmitBuf.bits.bufLong[2] = 0;    /* Clear xmit bit buffer */

    workTpar = 0x1f;                   /* Initializa Tpar */

    Stuff6(PICxmitBuf.PIC_addr);
    Stuff6(PICxmitBuf.command);
    if (PICxmitBuf.command > 0x0f)
       {
       Stuff6(PICxmitBuf.reg_ID);
       if (PICxmitBuf.command > 0x17)
          {
          WorkData = PICxmitBuf.command_data;
          ShiftCount = 32;
          for (i = 0,i<7;i++)
             {
             WorkInt = 0;
             for (j=0;j<5;j++)
                {
                WorkInt = WorkInt << 1;
                if (WorkData and 0x80000000)
                   WorkInt += 1;
                WorkData = WorkData << 1;
                if (--ShiftCount == 0)
                   {
                   WorkInt = (WorkInt << 3) + (PICxmitBuf.flags and 0x07);
                   j = 5;
                   }
                }
             Stuff6(WorkInt);
             }
          }
       }
    Stuff6(workTpar);
                                       /* Buffer set, set up UART control */
    PICrcvBuf.bitCount = 66;

    if ((PIC_comm_mon.control_flags and XMIT_DATA_RDY) == 0)
       {
       PIC_comm_mon.xmit_buf = PICxmitBuf;
       PIC_comm_mon.control_flags |= XMIT_DATA_RDY;
       }
```

```
  PICrcvBuf.flags = 0;
  PICrcvBuf.PIC_addr = PICxmitBuf.PIC_addr;  /* preset address of remote PIC */
}


unsigned int Get5(void)
{
  int          byte_index, bit_index;
  unsigned int  work_result;

  work_result = 0;
  if ((PICrcvBuf.flags and
     ~(PIC_error_flag | PIC_need_refresh_flag | PIC_invalid_data_flag)) == 0)
     {
     byte_index = 96 - PICrcvBuf.bitCount;
     bit_index = byte_index and 0x07;
     byte_index = byte_index >> 3;
     work_result = (PICrcvBuf.bits.bufByte[byte_index] << 8) +
PICrcvBuf.bits.bufByte[byte_index+1];
     work_result = work_result >> (10 - bit_index);
     work_result and= 0x3f;
     if (parity6[work_result >> 1] != work_result)
        {
        work_result = 0;
        PICrcvBuf.flags |= rcv_parity_err_flag;
        }
     else
        {
        work_result = work_result >> 1;
        workTpar ^= work_result;
        PICrcvBuf.bitCount -= 6;
        if (PICrcvBuf.bitCount < 0)
           PICrcvBuf.flags |= rcv_bad_length_flag;
        }
     }
  return(work_result);
}

void Stuff6(int pass_data)
{
  int  byte_index, bit_index;
  workTpar ^= pass_data;
  byte_index = PICxmitBuf.bitCount >> 3;
  bit_index = PICxmitBuf.bitCount and 0x07;
  pass_data = parity6[pass_data] << (10 - bit_index);
  PICxmitBuf.bits.bufByte[byte_index] |= pass_data >> 8;
  PICxmitBuf.bits.bufByte[byte_index+1] |= pass_data and 0xff;
  PICxmitBuf.bitCount += 6;
}
```

PULSE.H

```
/*
*/

#ifndef BOOLEAN_DEFINED
#define BOOLEAN_DEFINED
typedef int BOOLEAN ;
#endif


/* subsecond processing for pulse counters */
void pulseSubsecond(void);

/* Daily PIC table rebuild */
void pulseDay(void);

/* Main PIC communication routine */
void pulseService(void);

/* PIC comm routine - once-per-second code */
void pulseSecond(void);

/* Subroutines for PIC communication */
void Stuff6(int pass_data);
void format_xmit_buffer(void);
BOOLEAN decode_rcv_buffer(void);
void process_status_bits(void);
unsigned int Get5(void);

void set_PIC_alarm(alarmcodes pass_alarm_type,
            int pass_PIC_addr,
            unsigned int pass_alarm_data);


/* Startup code for PIC communication */
void pulseStartup (void);

/* Routine to clear pulse registers after cold start */
void pulseColdstart(void);

/* defined in pt.c */
#define RELAY_A 0
#define RELAY_B 1
void pulseOut(int picNbr,int relayNbr,int ONoff);
void ptEveryMinute(void) ;
void ptEverySecond(void) ;
void set_PLC_relays(int xmitRelay,int rcvRelay) ;
```

PICEND.DEF

```
/*
include file at end of mtrsamp to control MC and ST pic chips from mtrsamp interrupt
*/

#ifdef IS_MC5

if (pulseOutMode==PIC_MC_SERIAL)
        {
        asm (" MOVE.L #0x80400E,A0");
        asm (" AND.W #0xFCFF,(A0)");   /* Clear clock and data ports */
        }

#endif

#ifdef IS_ST5

 asm (" MOVE.L #0x80400E,A0");
 asm (" AND.W #0xFEFF,(A0)");   /*  Insert clock falling edge */

if (PIC_serial_status == sending_st5_bits)
        {
 asm (" NOP");
 asm (" NOP");
 asm (" NOP");
 asm (" NOP");
 asm (" NOP");
 asm (" NOP");
 asm (" NOP");
 asm (" NOP");
 asm (" NOP");
 asm (" NOP");
 asm (" NOP");
 asm (" NOP");
 asm (" NOP");
 asm (" MOVE.L #0x80400E,A0");
 asm (" OR.W #0x0100,(A0)");  /* If sending bits, activate clock*/
        }

#endif
```

PICVARS.DEF

```
/*
local variables to control MC and ST pic chips from mtrsamp interrupt
meter/c/lib/picvars.def
*/

unsigned long bitOneFlag;
```

PULSELINK.DEF

```
/* determine if variables are external or defined here */
#undef ref
#ifdef PSTRU_DEFINED
#define ref
#else
#define ref extern
#endif


#if (NUMPH > 6)
#define IS_MC5
#else
#ifdef MAX_SCAN_METERS
#define IS_ST5
#else
#define IS_RSM
#endif
#endif

/* Values for hdwr -p (pulseOutMode)  */

#define STANDARD_PIC_COMM 0

#if 0      /* remove these equates from mtrlink.def */
#define PIC_PULSE_SERIAL 1
#define PIC_MC_SERIAL 2
#define PIC_ST_SERIAL 3
#endif     /* then uncomment them here */

/* #define VATEST      1 */


#ifdef VATEST


#undef IS_MC5
#undef IS_RSM
#undef IS_ST5

#define IS_RSM
#undef NUMPH
#define NUMPH 3
#undef MAX_SCAN_METERS

#endif

/*  Uncomment for water meters (Oakville Hydro) */
#define READ_ENCODER_ID 0x49ff0000
#define READ_ENCODER_PARM 0x03 12500c


/*  Uncomment for gas meters (Sonix)
#define READ_ENCODER_ID 0x49fe0389
```

```
#define READ_ENCODER_PARM 0x0300500c
                        */

#define ID_FIELD_LENGTH 13


#define PULSES_PER_PIC      4

#ifdef IS_MC5
#define BITS_PER_SEC        822
#define MAX_PICS    16
#else
#ifdef IS_ST5
#define BITS_PER_SEC        942
#define MAX_PICS    5
#else
#define BITS_PER_SEC   32
#define MAX_PICS    1
#endif
#endif

#define DIRECT_PIC_WAIT_LIMIT   24
#define BUFFERED_PIC_WAIT_LIMIT 10 * BITS_PER_SEC

#define NUM_START_BITS    12

#ifdef IS_ST5
#define  REBUILD_DELAY 30000
#else
#define  REBUILD_DELAY 1000
#endif

typedef
        struct {
                unsigned long   serial_number;
    unsigned long  last_cmd_data;
    unsigned char       PIC_addr;
    unsigned char       software_type;
    unsigned char       software_version;
                unsigned char   loop_rate;
                unsigned int    status_reg_value;
                unsigned char   time_since_global;
                unsigned char   num_pulse_regs;
                unsigned char   curr_pulse_reg;
    unsigned char  curr_ID_reg;
    unsigned char  last_command;
                unsigned int    reply_wait_limit;
                unsigned int    PIC_data_err_count;
                unsigned int    PIC_reset_count;
                unsigned int    comm_error_count;
                unsigned        force_pulse_read: 1;
                unsigned        get_serno       : 1;
                unsigned        get_version     : 1;
                unsigned        get_status      : 1;
                unsigned        reset_status    : 1;
```

```
                unsigned        update_parameter: 1;
                unsigned        clear_pulse_regs: 1;
                }
                        PIC_status_t;

typedef
        enum    {
    sending_bits,
    sending_st5_bits,
                rcving_bits,
                do_PIC_start,
                sending_PIC_start,
                do_send_cmd,
                sending_start,
                sending_start_1,
                sending_stop,
                waiting_for_start,
                rcving_stop,
                idle
                }
                        PIC_s_stat_t;

typedef
        enum    {
    SET_PRE_ALIGN,
    DO_ALIGN,
    SET_NORMAL_PLC,
    DISCONNECT_COUPLER,
    READ_XMIT_LEVEL
                }
                        ATM_op_t;

typedef
        enum    {
    DO_PRESET,
    DO_PLC_SET,
    SET_ALIGN,
    WAIT_ALIGN,
    DO_ATM_DISCONNECT,
    ATM_DONE
                }
                        ATM_state_t;


typedef
     struct{
          unsigned char   PIC_addr;
          unsigned char   command;
          unsigned char   reg_ID;
          unsigned long   command_data;
                        unsigned int        flags;
          int               bitCount;
          union {
               unsigned long bufLong[3];
               unsigned char bufByte[12];
```

```
        } bits;
    }
        PICbitBuf;

#ifndef BOOLEAN_DEFINED
#define BOOLEAN_DEFINED
typedef int BOOLEAN ;
#endif

/* Definition of bits in Flags field of PIC comm buffers */

/* NOTE- Top 4 bits (0xf000) reserved for use by ATM code  */

#define rcv_bad_addr_flag    0x0100
#define rcv_tpar_error_flag  0x0080
#define rcv_data_ready_flag  0x0040
#define rcv_bad_length_flag  0x0020
#define rcv_timeout_flag     0x0010
#define rcv_parity_err_flag  0x0008
#define PIC_error_flag       0x0004
#define PIC_need_refresh_flag 0x0002
#define PIC_invalid_data_flag 0x0001

/* Command codes for PIC communication */

#define MAX_FAST_READ            0x07
#define CLEAR_PULSE_REGISTERS 0x08
#define READ_REGISTER            0x10
#define WRITE_REGISTER           0x18
#define SYSTEM_CONTROL  0x19

/* Register ID codes for PIC communication */

#define MAX_PULSE_REG            15
#define CONTROL_REG      16
#define ERROR_REG        26
#define OUTPUT_REG       27
#define PARM_REG                 28
#define VERS_REG                 29
#define SERNO_REG              30
#define STATUS_REG             31

/* Special-purpose PIC addresses */

#define GLOBAL_PIC_ADDR   31
#define MC5_MUX_PIC_ADDR  30
#define ST5_MUX_PIC_ADDR  29

#define LCM_BASE_ADDR     16 /* First LCM is at address 16 */
#define ATM_BASE_ADDR     20 /* First ATM is at address 20 */
#define MAX_PULSE_PIC_ADDR 15 /* pulse couters are from 0 to 15 */


/*  Bit usage in the ATM relay control word */
```

```
#define ATM_RC_XMIT_ON     0x00008000
#define ATM_RC_MEAS_XMIT   0x00004000
#define ATM_RC_MEAS_CPLR   0x00002000
#define ATM_RC_RES_SHORTED 0x00001000

#define ATM_RC_CAP_MASK    0x000003ff
#define ATM_NUM_CAP_BITS   10

#define ATM_PRESET_CODE    ATM_RC_XMIT_ON | ATM_RC_MEAS_XMIT |
ATM_RC_MEAS_CPLR
#define ATM_PLC_CODE       ATM_RC_XMIT_ON | ATM_RC_RES_SHORTED
#define ATM_DISC_CODE      0

/* Other ATM control equates */

#define ATM_MEAS_TIME      20L     /* mSec to wait between cap change and ADC
reading */
#define ATM_DISC_TIME      20L     /* mSec to wait before switching cap relays */
#define ATM_MIN_READING    0x0038  /* If reading at end of autotune less than this, fail
*/


/* Equates for ST5 comm */

#define ST5_PIC_DELAY      66

#pragma region ("ram=ram")

ref unsigned short         current_PIC_index, rcv_PIC_index;
ref PIC_status_t    PIC_status[MAX_PICS];

struct          {
    unsigned long    reading;
    int              error_cnt;
    unsigned char    ID_field[ID_FIELD_LENGTH];
    unsigned char    ID_buffer[ID_FIELD_LENGTH];
                } ref      pulseData[2*NUMPH];

struct         {
    PICbitBuf      xmit_buf;
    PICbitBuf      rcv_buf;
    int          control_flags;
                } ref        ext_comm, PIC_comm_mon;

struct           {
    ATM_op_t       operation;
    unsigned int   ATM_number;
    ATM_state_t    state;
    unsigned       request_flag : 1;
    unsigned       done_flag : 1;
    unsigned       error_flag : 1;
    unsigned       active_flag : 1;
    unsigned int   start_cap_code;
    unsigned int   end_cap_code;
    unsigned int   optimum_cap_code;
```

```
    unsigned int  optimum_reading;
    unsigned int  xmit_level;
    unsigned int  coupler_level;
    unsigned int  result_flags;
    long int      align_timer;
              } ref       ATM_control;

struct          {
    unsigned      request_flag : 1;
    unsigned      done_flag : 1;
    unsigned      error_flag : 1;
    unsigned      active_flag : 1;
    unsigned int  xmit_mask;
    unsigned int  rcv_mask;
    unsigned int  result_flags;
    unsigned int  last_xmit_mask;
    unsigned int  last_rcv_mask;
              } ref       MUX_control;

struct          {
    unsigned      do_rebuild : 1;
    unsigned      do_alarm : 1;
              } ref       Comm_background_flags;

/* Bit definitions in ext_buf.control_flags */
#define  SUPPRESS_NORMAL_COMM   0x0001
#define  XMIT_REQUEST           0x0002  /* External routine requests send of xmit_buf */
#define  CPY_RCV_DATA           0x0004  /* External cmd was sent, copy rcvd buffer to
rcv_buf */
#define  RCV_DATA_RDY           0x0008  /* Got reply from external cmd, reply is in
rcv_buf */
#define  XMIT_DATA_RDY          0x0010  /* xmit_buf has copy of last cmd sent */

ref PICbitBuf               PICxmitBuf, PICrcvBuf, HoldrcvBuf;
ref PIC_s_stat_t    PIC_serial_status;
ref unsigned int    num_PICs;
ref unsigned int    num_LCMs;
ref unsigned int    rebuild_timer;
ref unsigned int    ATM_work;
ref unsigned int    work_PIC_addr;
ref unsigned int    num_pulse_ctrs;
ref unsigned int    hold_num_pulse_ctrs;
ref unsigned int    bit_count;
ref unsigned int          out_bit_value;
ref unsigned char   *shift_LSB_ptr;
ref unsigned char   in_bit_value;
ref unsigned int    PIC_wait_limit;
ref unsigned int    zero_count;
ref unsigned int    last_comm_err_flag;
ref BOOLEAN         dont_clear_PIC_stats;
ref int             ST5_PIC_delay_ctr;
ref unsigned int    PIC_bad_addr_count;
ref alarmstru       PIC_alarm;

#ifdef VATEST
```

```
/* test !!!! */
ref int                 temp_command;
ref int                 temp_reg_ID;
ref long        temp_command_data;
ref int                 temp_flags;
/* test !!!! */
#endif

ref enum {
        clk_high,
        clk_low
        }               PIC_clk_state;

ref enum {
        NO_GLOBAL_SEND,
        SEND_GLOBAL_CLR,
        SENDING_GLOBAL_CLR,
        GLOBAL_CLR_SENT
        }               clear_state;

/*  Old equates maintained for compatibility */
#if 0
#define RELAY_A_BIT 1
#define RELAY_B_BIT 2
#define IDLE 0
#define TX 1
#define RX 2
#define DATA_READY 3
#define WAKE 4
#define PIC_RESET 5

#define WAKE_COUNT_RELOAD (64*4)        /* Four Seconds */
#define RESET_COUNT_RELOAD 10       /* 1/4 Seconds */
#define RELAY_A_OFF_CMD 5                   /*active low trigger*/
#define RELAY_A_ON_CMD  4
#define RELAY_B_OFF_CMD 7
#define RELAY_B_ON_CMD  6
#define CLEAR_ACC_CMD 8
#define SUBACC_CMD 12
#define ECHO_CMD 14
#define SLEEP_CMD 15
#define PULSE_SERIAL_COM_MAX 15
#define WAKE_CMD 16
#define RESET_CMD 17
#define PULSE_COM_NUM 18
#define NO_PULSE_BIT_DATA 0xFFFFFFFF
#define RELAY_CMD_MIN 4
#define RELAY_CMD_MAX 7
#endif
```

PULSEOUTM.DEF

```
#include "clklink.def"
#include "mtrlink.def"

#include "plc.def"
#include "serlink.def"
#include "pulselink.def"

#ifndef NO_PULSES

#include "scan.h"
#include "pulse.h"
#include "log.h"

void pulseout_second_back(void);
void pulseout_ram_init1(void);
ADDRFN pulseoutaddrfn ;

#pragma region ("data=ramInit1")
void (*pulseout_ram_init1p)(void)=pulseout_ram_init1;
#pragma region ("data=data")

#pragma region ("data=secondBack")
void (*pulseout_second_backp)(void)=pulseout_second_back ;
#pragma region ("data=data")

ADDRFN_RET pulseoutaddrfn(reg8stru reg8)
{
  /* write the load shed event to flash */
  putEvent(LOAD_SHED_EVENT,1,andreg8.ulong);

  return(0);
}

void pulseout_ram_init1()
{
  addrfn_tbl[GET_FROM_SLAVE][PULSEOUT_STATE] = pulseoutaddrfn ;
}
void pulseout_second_back()
{
}
#endif
```

REIVINDICAÇÕES

1. Aparelho para medição de multi-canal de eletricidade, caracterizado pelo fato de que compreende:

uma cabeça de medidor localizada em um lado secundário de um transformador e operável para separadamente medir uso de eletricidade para cada uma de uma pluralidade de linhas de consumidor de eletricidade;

a dita cabeça de medidor em comunicação direta com um transponder localizado em um lado primário do dito transformador e operável para transmitir dados para e receber dados do dito transponder por comunicação de linha de potência direta, o dito transponder operável para transmitir dados e para receber dados de um computador localizado remotamente;

um módulo de controle de carga em comunicação com dita cabeça de medidor e operável para atuar a conexão e desconexão de cada um de uma pluralidade de relés, cada relé de dita pluralidade de relés correspondendo a uma da dita pluralidade de linhas de consumidor de eletricidade; e

uma caixa contendo a dita cabeça de medidor, o dito módulo de controle de carga e os ditos relés.

2. Aparelho de acordo com reivindicação 1, caracterizado pelo fato de que compreende adicionalmente um detector de fraude em comunicação com dita cabeça de medidor.

3. Aparelho de acordo com reivindicação 2, caracterizado pelo fato de que dito detector de fraude compreende uma luz e uma superfície reflexiva, e em que dita cabeça de medidor é operável para instruir dito módulo de controle de carga para desconectar todas de ditas linhas de cliente se dito detector de fraude provê notificação de que dita luz não é detectada refletindo de dita superfície reflexiva.

4. Aparelho de acordo com reivindicação 2, caracterizado pelo fato de que dito detector de fraude compreende um detector de luz ambiente entrando na dita caixa.

5. Aparelho de acordo com reivindicação 1, caracterizado pelo fato de que a dita caixa é instalada em um poste de utilidade.

6. Aparelho de acordo com reivindicação 1, caracterizado pelo fato de que compreende ainda um dispositivo para comparar energia de transformador à energia total usada por ditas linhas de consumidor.

7. Aparelho de acordo com reivindicação 1, caracterizado pelo fato de que compreende adicionalmente um dispositivo para detectar fluxo de tensão inversa por ditas linhas de consumidor.

8. Aparelho de acordo com reivindicação 1, caracterizado pelo fato de que compreende adicionalmente uma memória legível por computador em comunicação com dita cabeça de medidor e um contador em comunicação com dita cabeça de medidor, dito contador correspondendo a uma linha de cliente e operável para contar regressivamente uma quantidade de energia armazenada em dita memória, e dita cabeça de medidor operável para enviar um sinal de desconexão para o dito módulo de controle de carga para desconectar dita linha de cliente quando dito contador alcança zero.

9. Aparelho de acordo com reivindicação 1, caracterizado pelo fato de que compreende adicionalmente uma memória legível por computador em comunicação com dita cabeça de medidor, dita memória operável para armazenar um limite de carga para uma linha de cliente, e dita cabeça de medidor operável para enviar um sinal de desconexão para o dito módulo de controle de carga para desconectar dita linha de cliente quando dito limite de carga é excedido.

10. Aparelho de acordo com reivindicação 1, caracterizado pelo fato de que compreende adicionalmente uma memória legível por computador em comunicação com dita cabeça de medidor, dita memória operável para armazenar um limite de uso para uma linha de cliente, e dita cabeça de medidor operável para enviar um sinal de desconexão para um módulo de controle de carga para desconectar dita linha de cliente quando dito limite de uso é excedido.

11. Aparelho de acordo com reivindicação 1, caracterizado pelo fato de que dito transponder é operável para se comunicar com dito computador localizado remotamente através de linhas de potência de média tensão.

12. Aparelho de acordo com reivindicação 1, caracterizado pelo fato de

que compreende adicionalmente uma unidade de exibição em comunicação com dita cabeça de medidor e operável para exibir dados recebidos de dita cabeça de medidor.

13. Aparelho de acordo com reivindicação 12, caracterizado pelo fato de que dita unidade de exibição é operável para exibir informação relativa a consumo de energia de um cliente.

14. Aparelho de acordo com reivindicação 12, caracterizado pelo fato de que dita unidade de exibição é operável para exibir advertências relativas a uso de energia de um cliente ou a suspeito roubo de energia.

15. Aparelho de acordo com reivindicação 12, caracterizado pelo fato de que dita unidade de exibição é operável para transmitir para dita cabeça de medidor informação entrada por um cliente.

16. Aparelho para medição remota de multi-canal de eletricidade usando comunicação de linha de potência, caracterizado pelo fato de que compreende:

uma pluralidade de pontos medidores localizados em um lado secundário de um transformador e operáveis para medir separadadamente uso de eletricidade para cada uma de uma pluralidade de linhas de consumidor de eletricidade;

o aparelho em comunicação direta com um transponder localizado em um lado primário do transformador e operável para transmitir dados para e receber dados do transponder por comunicação de linha de potência direta, o transponder operável para transmitir dados para e para receber dados de um computador localizado remotamente;

o aparelho operável para controlar um ou mais módulos de controle de carga operáveis para atuar conexão e desconexão de cada um de uma pluralidade de relés, cada relé da pluralidade de relés correspondendo a uma da pluralidade de linhas de consumidor de eletricidade; e

uma caixa contendo a pluralidade de pontos medidores, o módulo de controle de carga, e os relés, em que o transformador de distribuição converte voltagens de distribuição de média tensão em voltagens de baixa tensão apropriadas para fornecer energia aos consumidores, e em que o aparelho é

operável para injetar sinais em e receber sinais de linhas de potência de baixa tensão que abastecem consumidores com energia elétrica; ditos sinais provendo comunicação de via dupla entre a cabeça de medidor e o transponder, e atravessando o transformador de distribuição

17. Aparelho de acordo com reivindicação 16, caracterizado pelo fato de que compreende ainda um detector de fraude em comunicação com dita cabeça de medidor.

18. Aparelho de acordo com reivindicação 17, caracterizado pelo fato de que dito detector de fraude inclui uma luz e uma superfície reflexiva, e em que dita cabeça de medidor é operável para instruir dito módulo de controle de carga para desconectar todas de ditas linhas de cliente se dito detector de fraude prover notificação que dita luz não é detectada refletindo de dita superfície reflexiva.

19. Aparelho de acordo com reivindicação 17, caracterizado pelo fato do dito detector de fraude incluir um detector de luz ambiente entrando na caixa.

20. Aparelho de acordo com reivindicação 16, caracterizado pelo fato de que compreende ainda um dispositivo para comparar energia de transformador à energia total usada por ditas linhas de consumidor.

21. Aparelho de acordo com reivindicação 16, caracterizado pelo fato de que compreende ainda um dispositivo para detectar fluxo de tensão inversa por ditas linhas de consumidor.

22. Aparelho de acordo com reivindicação 16, caracterizado pelo fato de que compreende ainda uma memória legível por computador em comunicação com dita cabeça de medidor e um contador em comunicação com dita cabeça de medidor, dito contador correspondendo a uma linha de cliente e operável para contar regressivamente uma quantidade de energia armazenada em dita memória, e dita cabeça de medidor operável para enviar um sinal de desconexão ao módulo de controle de carga para desconectar dita linha de cliente quando dito contador alcança zero.

23. Aparelho de acordo com reivindicação 16, caracterizado pelo fato de que compreende ainda uma memória legível por computador em comunicação com

dita cabeça de medidor, dita memória operável para armazenar um limite de carga para uma linha de cliente, e dita cabeça de medidor operável para enviar um sinal de desconexão para o módulo de controle de carga para desconectar dita linha de cliente quando dito limite de carga é excedido.

24. Aparelho de acordo com reivindicação 16, caracterizado pelo fato de que compreende ainda uma memória legível por computador em comunicação com dita cabeça de medidor, dita memória operável para armazenar um limite de uso para uma linha de cliente, e dita cabeça de medidor operável para enviar um sinal de desconexão para o módulo de controle de carga para desconectar dita linha de cliente quando dito limite de uso é excedido.

25. Aparelho de acordo com reivindicação 16, caracterizado pelo fato de que dito transponder é operável para se comunicar com dito computador localizado remotamente através de linhas de potência de média tensão.

26. Aparelho de acordo com reivindicação 16, caracterizado pelo fato de que compreende ainda uma unidade de exibição em comunicação com dita cabeça de medidor e operável para exibir dados recebidos de dita cabeça de medidor.

27. Aparelho de acordo com reivindicação 26, caracterizado pelo fato de que dita unidade de exibição é operável para exibir informação relativa a consumo de energia de um cliente.

28. Aparelho de acordo com reivindicação 26, caracterizado pelo fato de que dita unidade de exibição é operável para exibir advertências relativas a uso de energia de um cliente ou a suspeito roubo de energia.

29. Aparelho de acordo com reivindicação 26, caracterizado pelo fato de que dita unidade de exibição é operável para transmitir para dita cabeça de medidor informação entrada por um cliente.

30. Aparelho de acordo com reivindicação 16, caracterizado pelo fato do transponder ser operável para se comunicar com o computador localizado remotamente via pelo menos um de: comunicações de rádio, fibra óptica, e linhas telefônicas.

31. Aparelho de acordo com reivindicação 16, caracterizado pelo fato de

ser operável para transmitir dados diretamente ao transponder a freqüências dentro da faixa de 10 a 25 kHz.

32. Aparelho de acordo com reivindicação 16, caracterizado pelo fato de ser operável para transmitir dados diretamente ao transponder a freqüências correspondendo a harmônicos meio ímpares de 60 Hz.

33. Aparelho para medição de multi-canal de eletricidade, caracterizado pelo fato de que compreende:

um módulo de controle em comunicação com um circuito secundário de um transformador de distribuição, em que o transformador de distribuição converte voltagens de distribuição de média tensão em voltagens de baixa tensão apropriadas para fornecer energia aos consumidores;

o módulo de controle em comunicação direta com um transponder em comunicação com um circuito primário do transformador e operável para transmitir dados para e receber dados do transponder por comunicação de linha de potência direta, através do transformador de distribuição, o transponder operável para transmitir dados para e para receber dados de um computador localizado remotamente;

uma pluralidade de módulos de medição em comunicação com o módulo de controle, cada módulo de medição operável para medir uso de eletricidade em uma de uma pluralidade de linhas de consumidor de eletricidade alimentadas pelo circuito secundário do transformador de distribuição;

um ou mais relés em comunicação com o módulo de controle e operáveis para atuar conexão e desconexão de eletricidade às linhas de consumidor de eletricidade; e

uma caixa contendo o módulo de controle, os módulos de medição e os relés, em que o aparelho é operável para injetar sinais em e receber sinais de linhas de potência de baixa tensão que abastecem consumidores com energia elétrica;

ditos sinais provendo comunicação de via dupla entre a cabeça de medidor e o transponder, e atravessando o transformador de distribuição para se comunicar, pelo menos parcialmente, sobre linhas de potência de média tensão,

com o computador localizado remotamente.

34. Aparelho de acordo com reivindicação 33, caracterizado pelo fato do um ou mais relés serem operáveis para controlar cada fase das linhas de consumidor de eletricidade.

35. Aparelho de acordo com reivindicação 16, caracterizado pelo fato do transponder ser operável para transmitir dados para e receber dados de um computador localizado remotamente via um ou mais transponders intermediários.

36. Aparelho para medição remota de multi-canal de eletricidade usando comunicação de linha de potência, caracterizado pelo fato de que compreende:

uma pluralidade de pontos medidores localizados em um lado secundário de um transformador e operáveis para medir separadadamente uso de eletricidade para cada uma de uma pluralidade de linhas de consumidor de eletricidade;

o aparelho em comunicação com um transponder localizado em um lado primário do transfornmador e operável para transmitir dados para e receber dados do transponder por comunicação de linha de potência, o transponder operável para transmitir dados para e para receber dados de um computador localizado remotamente;

um ou mais módulos de controle de carga em comunicação com a cabeça de medidor e operáveis para atuar cada um de uma pluralidade de relés, cada relé da pluralidade de relés operável para serviço de conectar e desconectar em uma da pluralidade de linhas de consumidor de eletricidade; e

uma caixa contendo a pluralidade de pontos medidores, o módulo de controle de carga, e os relés, em que o transformador de distribuição converte voltagens de distribuição de média tensão em voltagens de baixa tensão apropriadas para fornecer energia aos consumidores, e em que o aparelho é operável para injetar sinais em e receber sinais de linhas de potência de baixa tensão que abastecem consumidores com energia elétrica; ditos sinais provendo comunicação de via dupla entre a cabeça de medidor e o transponder, e atravessando o transformador de distribuição.

37. Aparelho para medição de multi-canal de eletricidade, caracterizado

pelo fato de que compreende:

um módulo de controle em comunicação com um circuito secundário de um transformador de distribuição;

o módulo de controle em comunicação direta com um transponder em comunicação com um circuito primário do transformador e operável para transmitir dados para e receber dados do transponder por comunicação de linha de potência direta, através do transformador de distribuição, o transponder operável para transmitir dados para e para receber dados de um computador localizado remotamente;

uma pluralidade de módulos de medição em comunicação com o módulo de controle, cada módulo de medição operável para medir uso de eletricidade em uma de uma pluralidade de linhas de consumidor de eletricidade alimentadas pelo circuito secundário do transformador de distribuição;

um ou mais relés em comunicação com o módulo de controle e operáveis para atuar conexão e desconexão de eletricidade para as linhas de consumidor de eletricidade; e

uma caixa contendo o módulo de controle, os módulos de medição, e os relés.

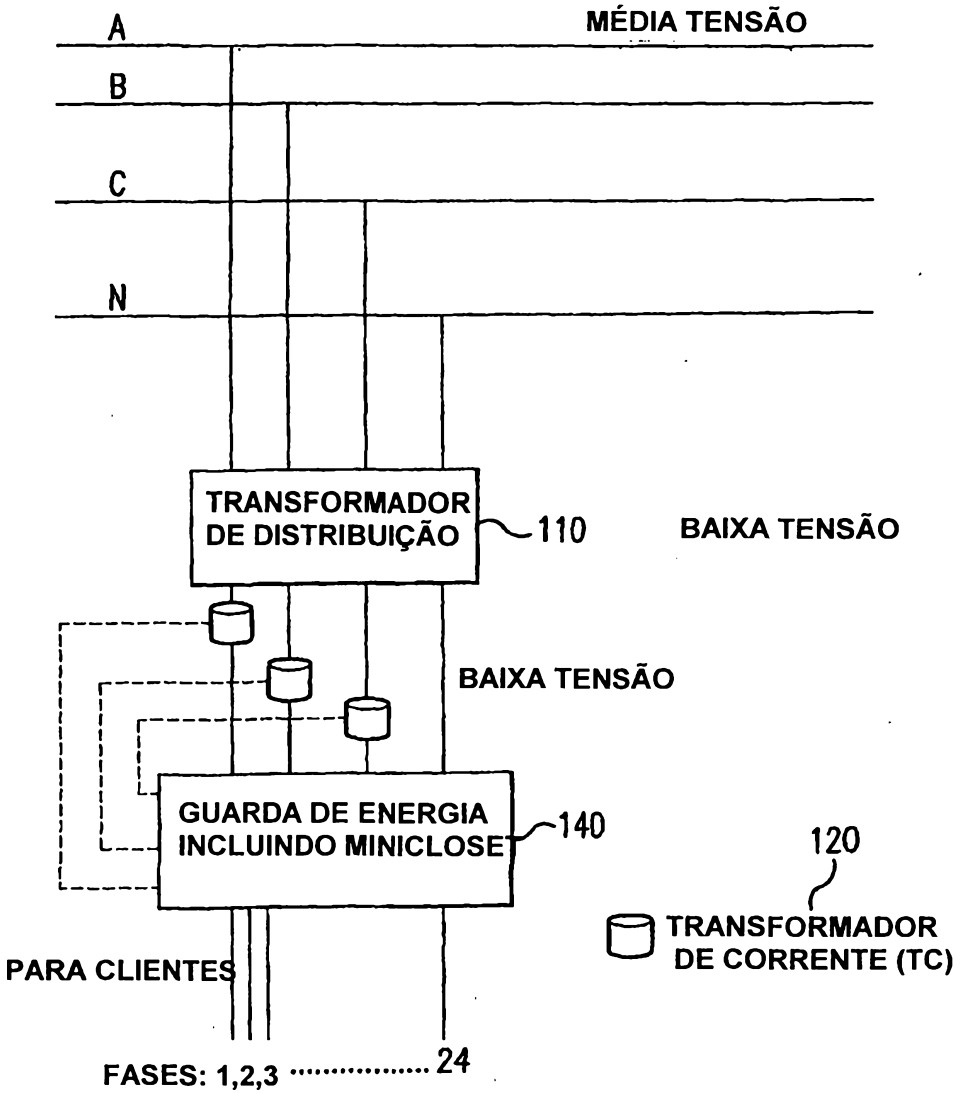

A
MÉDIA TENSÃO
B
C
N
TRANSFORMADOR
DE DISTRIBUIÇÃO
110
BAIXA TENSÃO
BAIXA TENSÃO
GUARDA DE ENERGIA
INCLUINDO MINICLOSE
140
120
TRANSFORMADOR
DE CORRENTE (TC)
PARA CLIENTES
FASES: 1,2,3 24

FIG.1

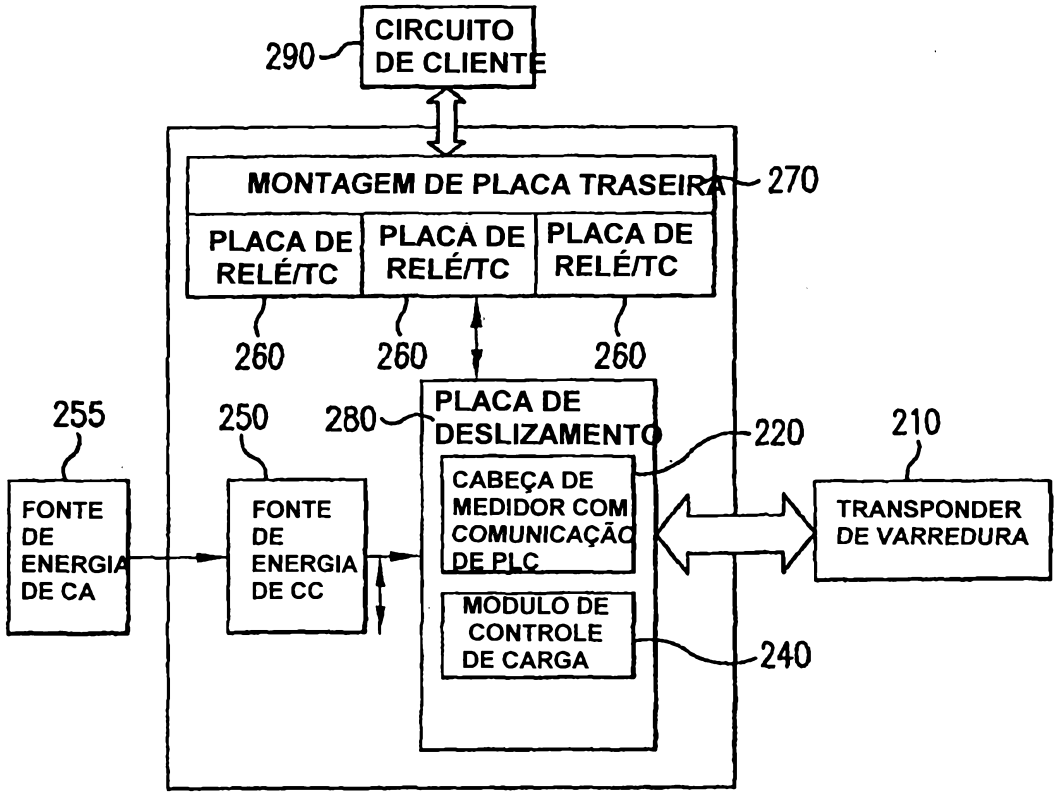

290
CIRCUITO DE CLIENTE
MONTAGEM DE PLACA TRASEIRA
270
PLACA DE RELÉ/TC
PLACA DE RELÉ/TC
PLACA DE RELÉ/TC
260
260
260
255
250
280
PLACA DE DESLIZAMENTO
220
210
FONTE DE ENERGIA DE CA
FONTE DE ENERGIA DE CC
CABEÇA DE MEDIDOR COM COMUNICAÇÃO DE PLC
TRANSPONDER DE VARREDURA
MODULO DE CONTROLE DE CARGA
240

FIG.2

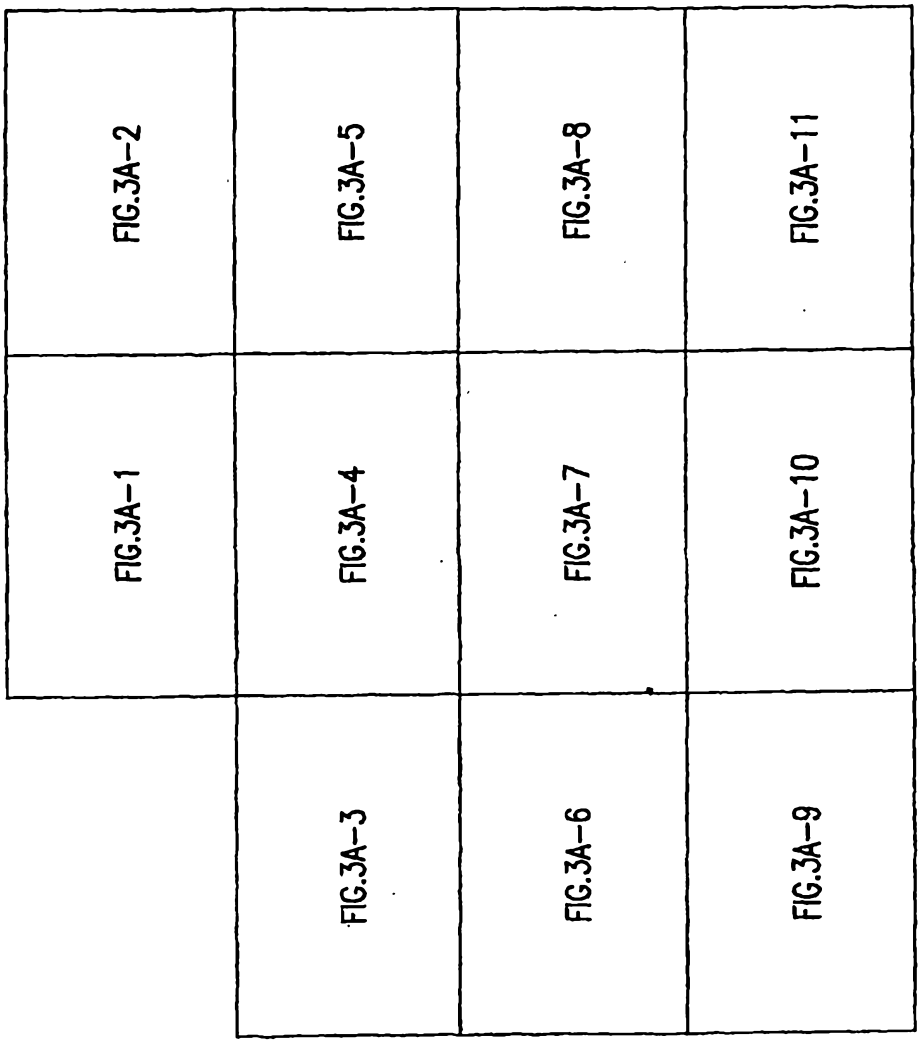


FIG.3A-1
FIG.3A-2
FIG.3A-3
FIG.3A-4
FIG.3A-5
FIG.3A-6
FIG.3A-7
FIG.3A-8
FIG.3A-9
FIG.3A-10
FIG.3A-11

FIG.3A

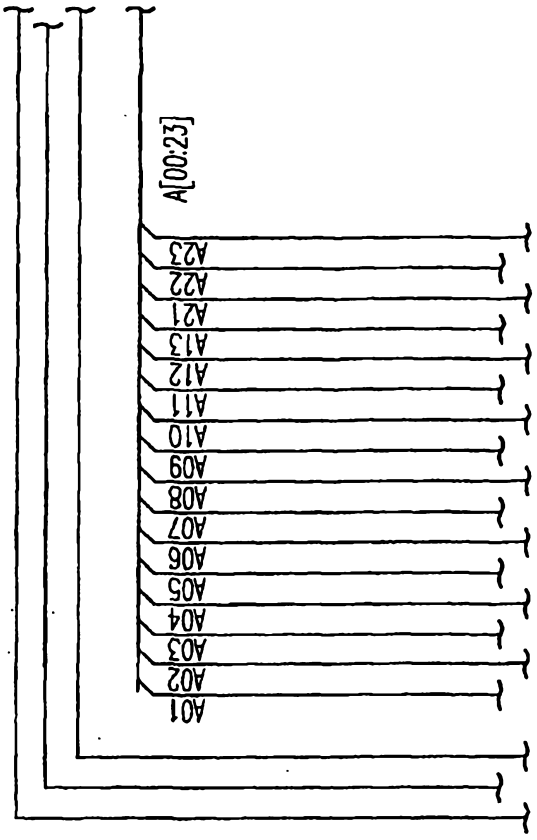

A[00:23]
A23
A22
A21
A13
A12
A11
A10
A09
A08
A07
A06
A05
A04
A03
A02
A01

FIG.3A-1

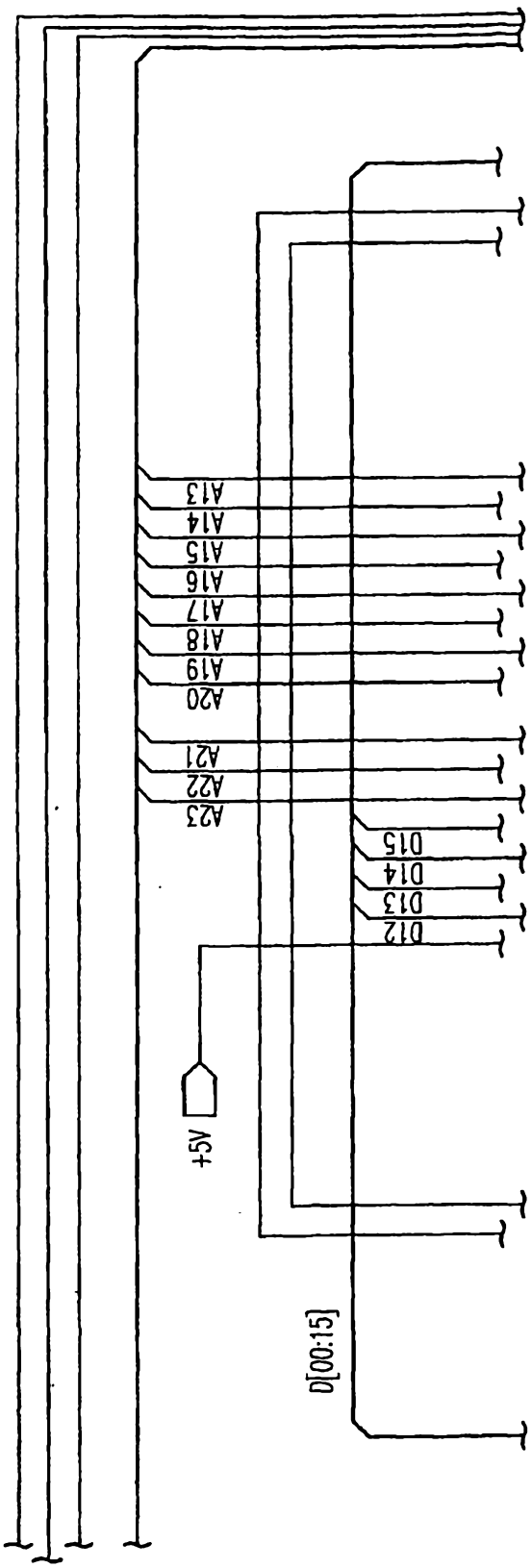

A13
A14
A15
A16
A17
A18
A19
A20
A21
A22
A23
D15
D14
D13
D12
+5V
D[00:15]

FIG.3A-2

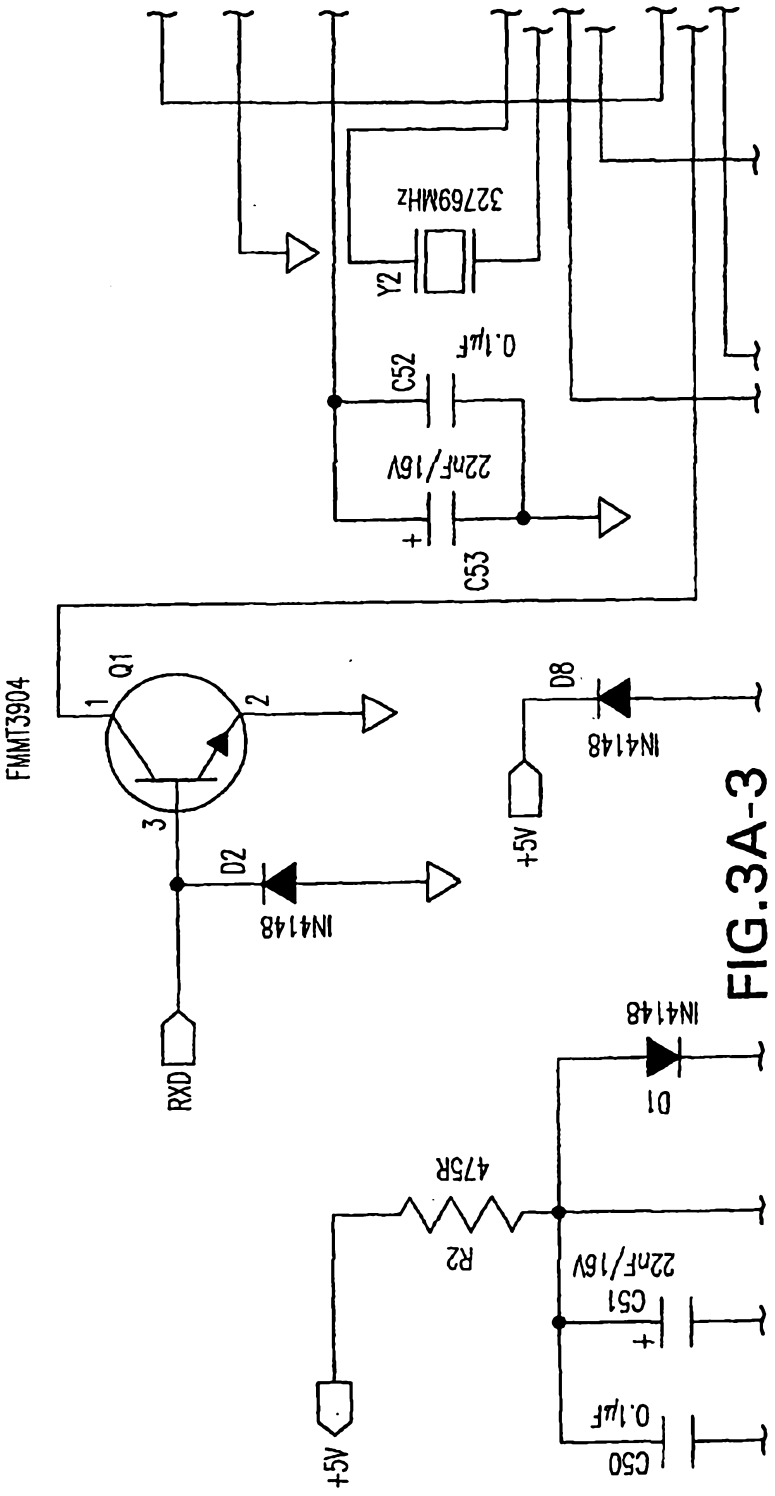

32769MHZ
Y2
0.1μF
C52
22nF/16V
+
C53
FMMT3904
Q1
1
2
3
D8
IN4148
+5V
D2
IN4148
RXD
IN4148
D1
475R
R2
22nF/16V
C51
+
0.1μF
C50
+5V

FIG.3A-3

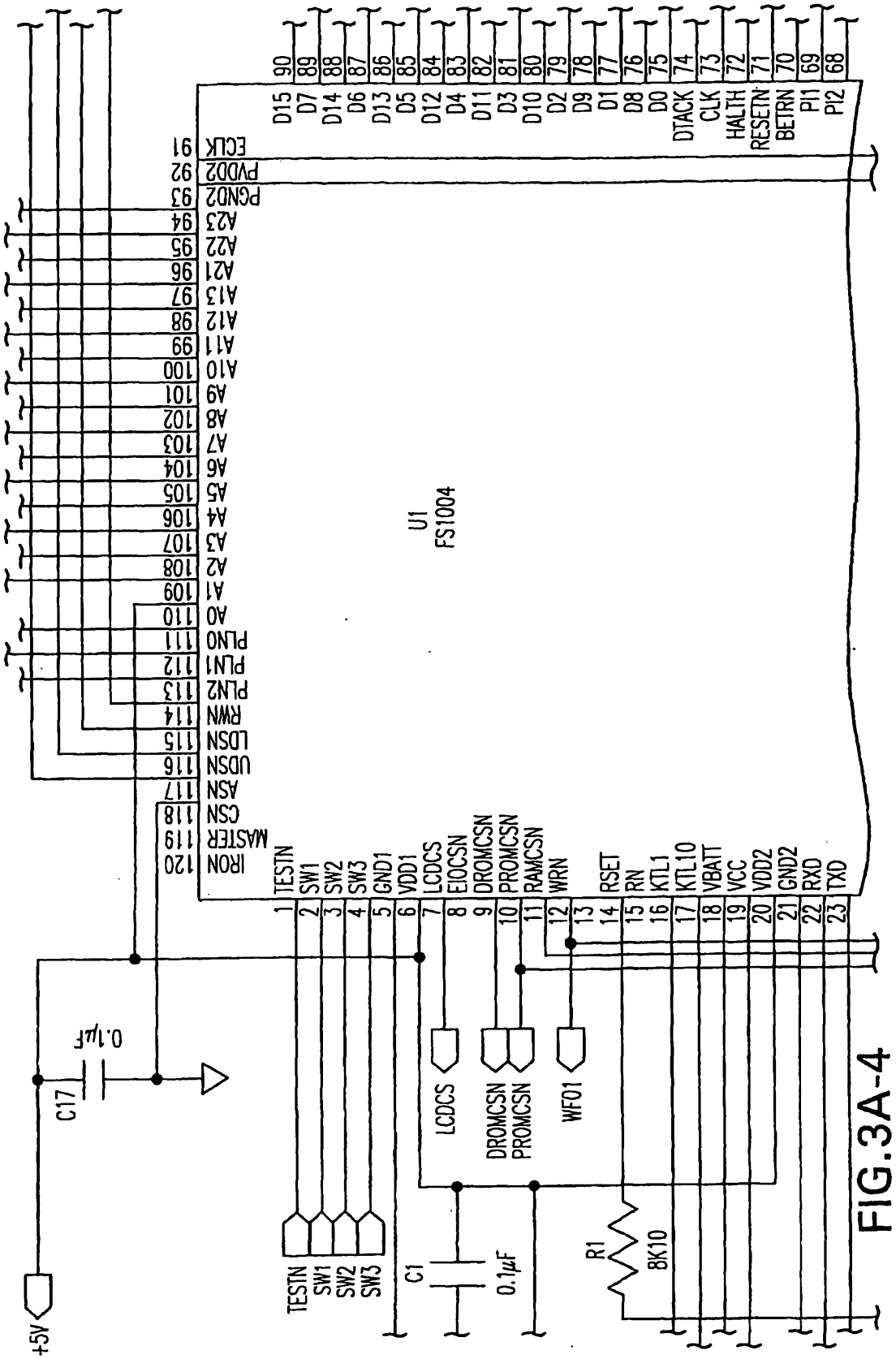

U1
FS1004
TESTIN
SW1
SW2
SW3
GND1
VDD1
LCDCS
EIOCSN
DROMCSN
PROMCSN
RAMCSN
WRN
RSET
RN
KTL1
KTL10
VBATT
VCC
VDD2
GND2
RXD
TXD
IRON
MASTER
CSN
ASN
UDSN
LDSN
RWN
PLN2
PLN1
PLN0
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A22
A23
PGND2
PVDD2
ECLK
D15
D7
D14
D6
D13
D5
D12
D4
D11
D3
D10
D2
D9
D1
D8
D0
DTACK
CLK
HALTH
RESETN
BETRN
PI1
PI2
+5V
C17
0.1μF
C1
0.1μF
R1
8K10
TESTIN
SW1
SW2
SW3
LCDCS
DROMCSN
PROMCSN
WF01

FIG.3A-4

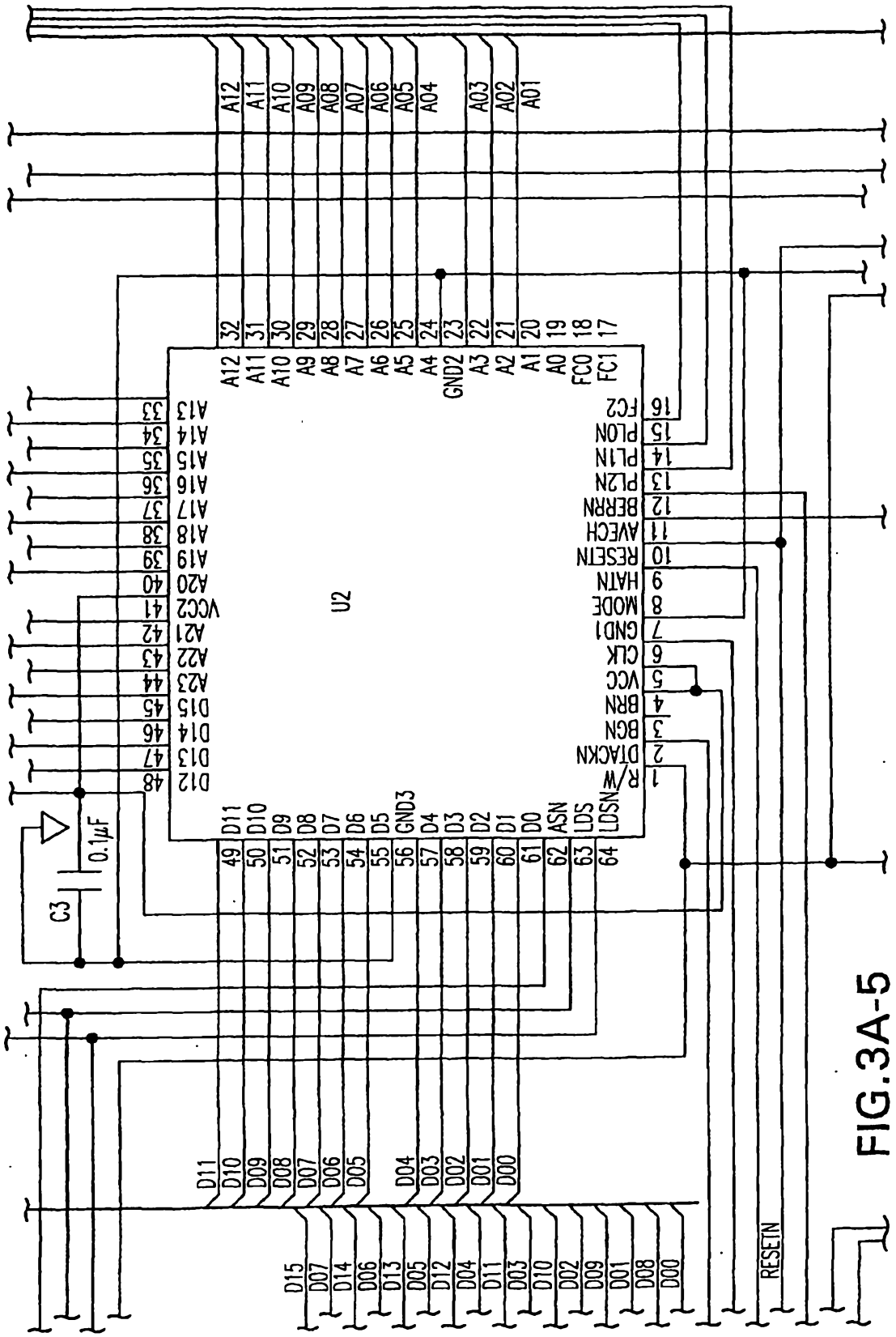

U2
A12 A11 A10 A09 A08 A07 A06 A05 A04 GND2 A03 A02 A01 FC0 FC1
32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17
A13 A14 A15 A16 A17 A18 A19 A20 VCC2 A21 A22 A23 D15 D14 D13 D12
33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
FC2 PLON PL1N PL2N BERRN AVECH RESETN HATN MODE GND1 CLK VCC BRN BGN DTACKN R/W
16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
D11 D10 D09 D08 D07 D06 D05 GND3 D04 D03 D02 D01 D00 ASN LDS LDSN
49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64
C3
0.1µF
D11 D10 D09 D08 D07 D06 D05 D04 D03 D02 D01 D00
D15 D07 D14 D06 D13 D05 D12 D04 D11 D03 D10 D02 D09 D01 D08 D00
RESETN

FIG.3A-5

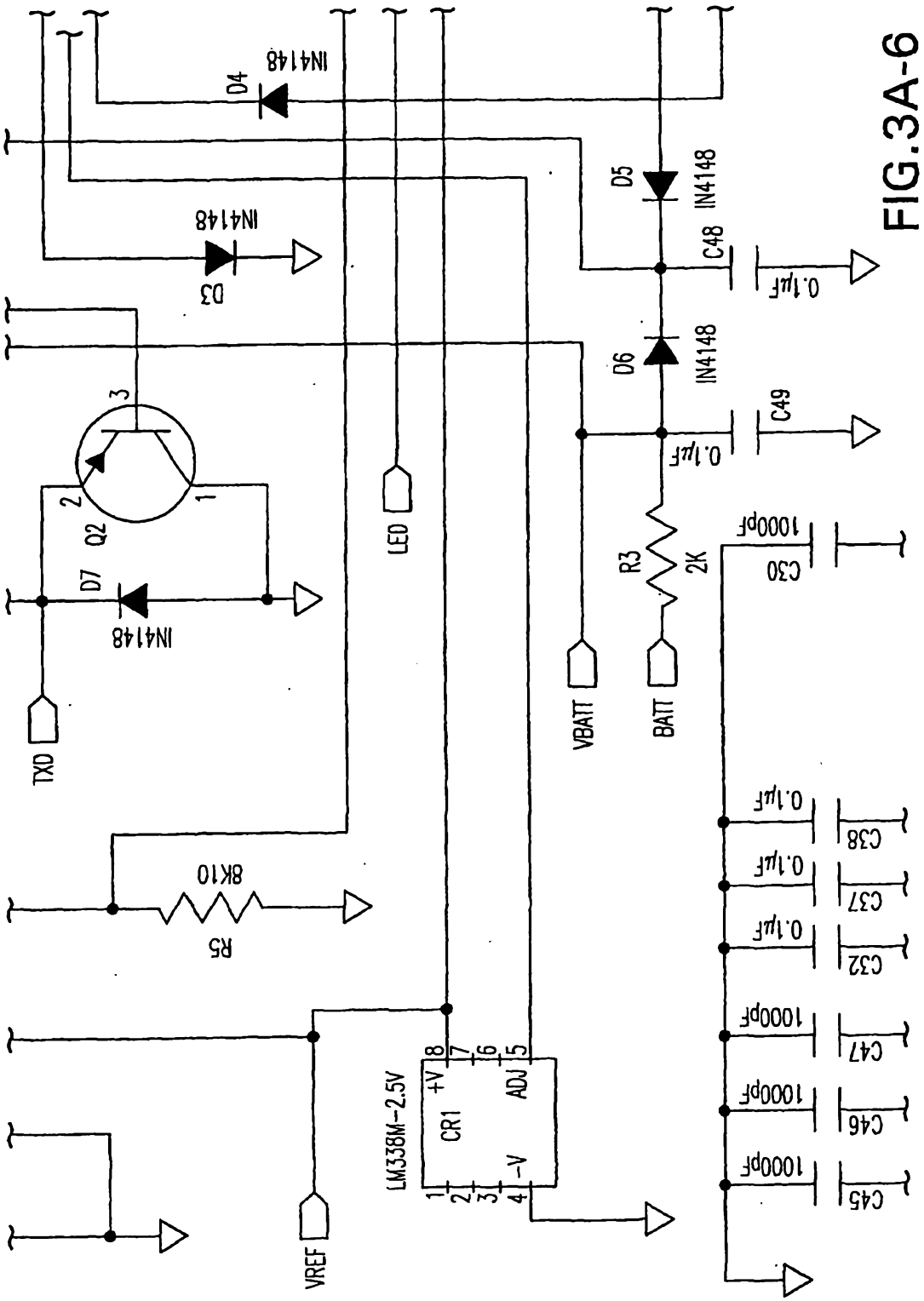

D4
IN4148
D5
IN4148
C48
0.1μF
D3
IN4148
D6
IN4148
C49
0.1μF
Q2
LED
R3
2K
1000pF
C30
D7
IN4148
TXD
VBATT
BATT
0.1μF
C38
0.1μF
C37
0.1μF
C32
8K10
R5
LM338M-2.5V
CR1
+V
ADJ
-V
1000pF
C47
1000pF
C46
1000pF
C45
VREF

FIG.3A-6

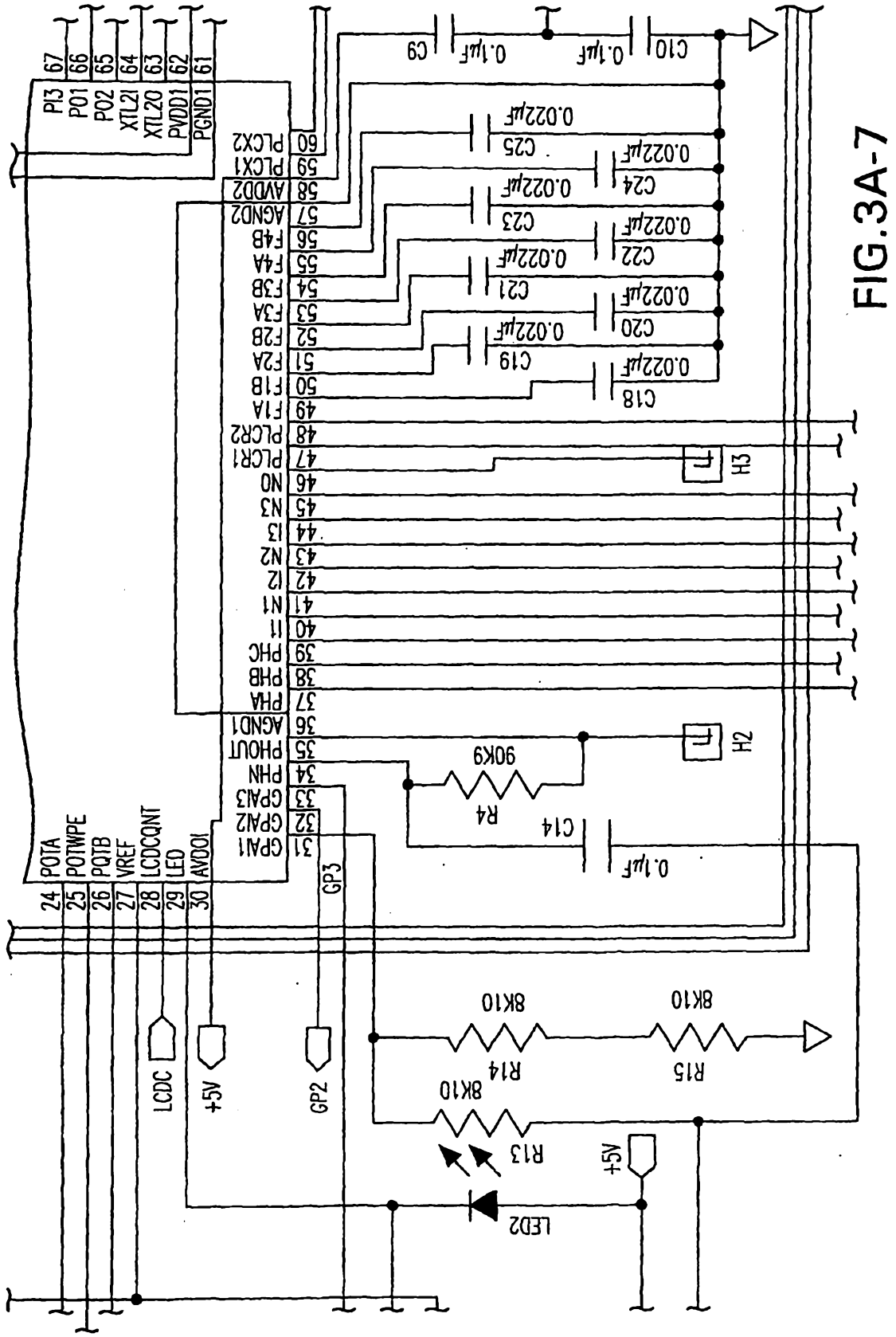

PI3 67
PO1 66
PO2 65
XTL2I 64
XTL2O 63
PVDD1 62
PGND1 61
PLCX2 60
PLCX1 59
AVDD2 58
AGND2 57
F4B 56
F4A 55
F3B 54
F3A 53
F2B 52
F2A 51
F1B 50
F1A 49
PLCR2 48
PLCR1 47
N0 46
N3 45
I3 44
N2 43
I2 42
N1 41
I1 40
PHC 39
PHB 38
PHA 37
AGND1 36
PHOUT 35
PHN 34
GPAI3 33
GPAI2 32
GPAI1 31
POTA 24
POTWPE 25
POTB 26
VREF 27
LCDCONT 28
LED 29
AVDD1 30
C9 0.1μF
C10 0.1μF
C25 0.022μF
C24 0.022μF
C23 0.022μF
C22 0.022μF
C21 0.022μF
C20 0.022μF
C19 0.022μF
C18 0.022μF
H3
H2
90K9
R4
C14 0.1μF
GP3
8K10
R14
8K10
R15
8K10
R13
LED2
+5V
GP2
+5V
LCDC

FIG.3A-7

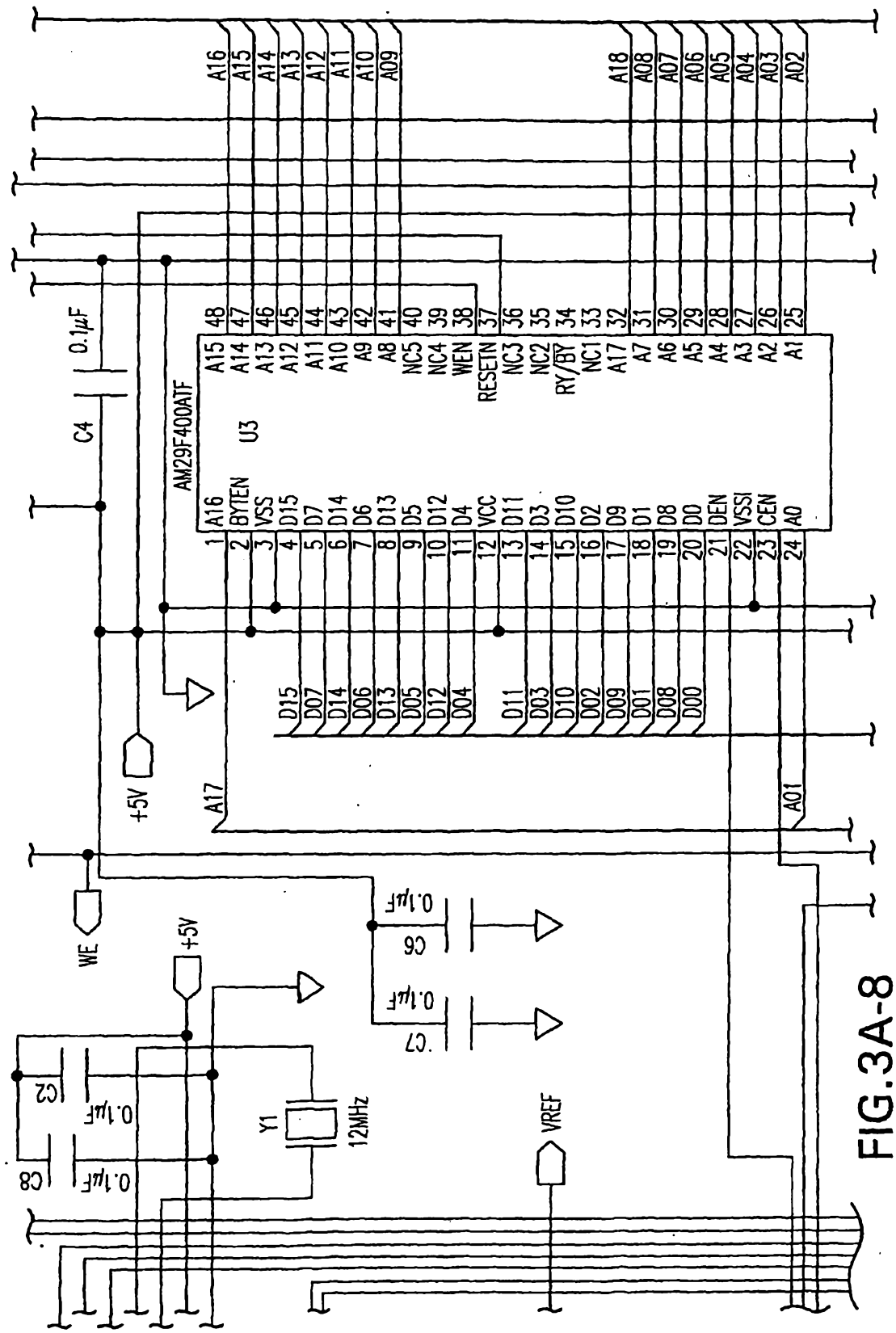

AM29F400ATF
U3
A16
BYTEN
VSS
D15
D7
D14
D6
D13
D5
D12
D4
VCC
D11
D3
D10
D2
D9
D1
D8
D0
DEN
VSSI
CEN
A0
A15
A14
A13
A12
A11
A10
A9
A8
NC5
NC4
WEN
RESETN
NC3
NC2
RY/BY
NC1
A17
A7
A6
A5
A4
A3
A2
A1
A18
A08
A07
A06
A05
A04
A03
A02
A01
C4
0.1μF
+5V
WE
C6
C7
C2
C8
Y1
12MHz
VREF

FIG.3A-8

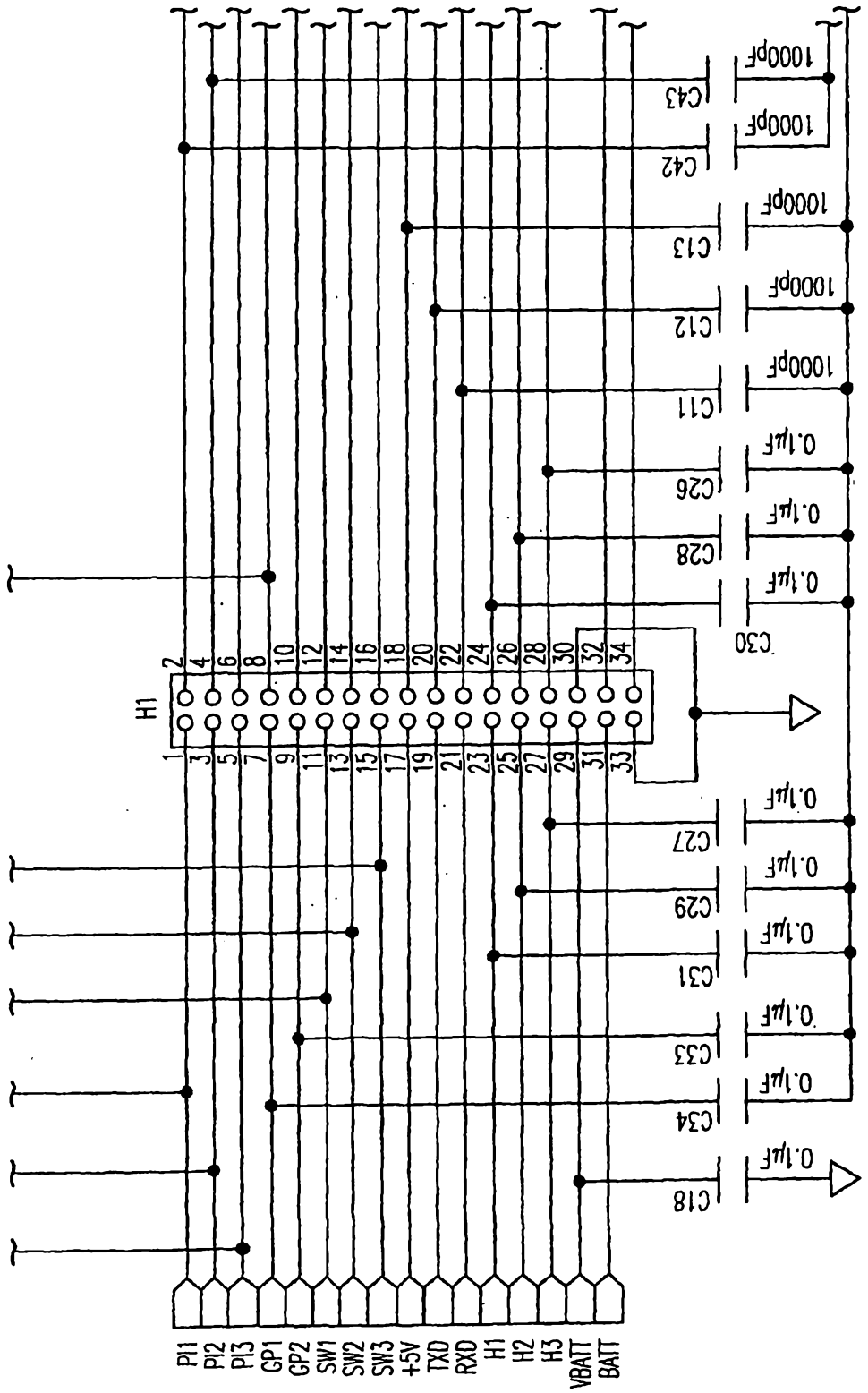

PI1
PI2
PI3
GP1
GP2
SW1
SW2
SW3
+5V
TXD
RXD
H1
H2
H3
VBATT
BATT
H1
C18
0.1μF
C34
0.1μF
C33
0.1μF
C31
0.1μF
C29
0.1μF
C27
0.1μF
C30
0.1μF
C28
0.1μF
C26
0.1μF
C11
1000pF
C12
1000pF
C13
1000pF
C42
1000pF
C43
1000pF

FIG.3A-9

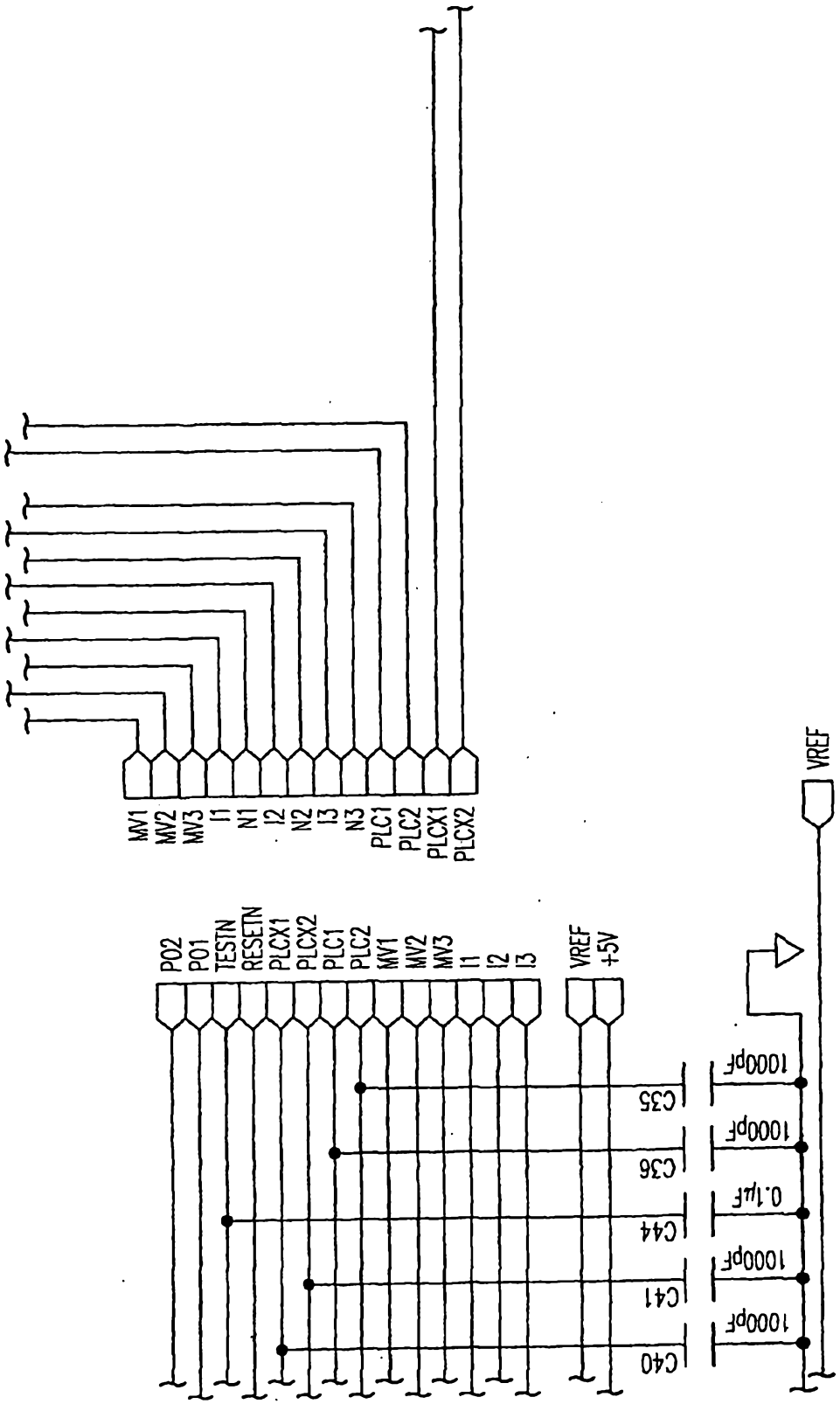

MV1
MV2
MV3
I1
N1
I2
N2
I3
N3
PLC1
PLC2
PLCX1
PLCX2
PO2
PO1
TESTN
RESETN
PLCX1
PLCX2
PLC1
PLC2
MV1
MV2
MV3
I1
I2
I3
VREF
+5V
VREF
C35
C36
C44
C41
C40
1000pF
1000pF
0.1μF
1000pF
1000pF

FIG.3A-10

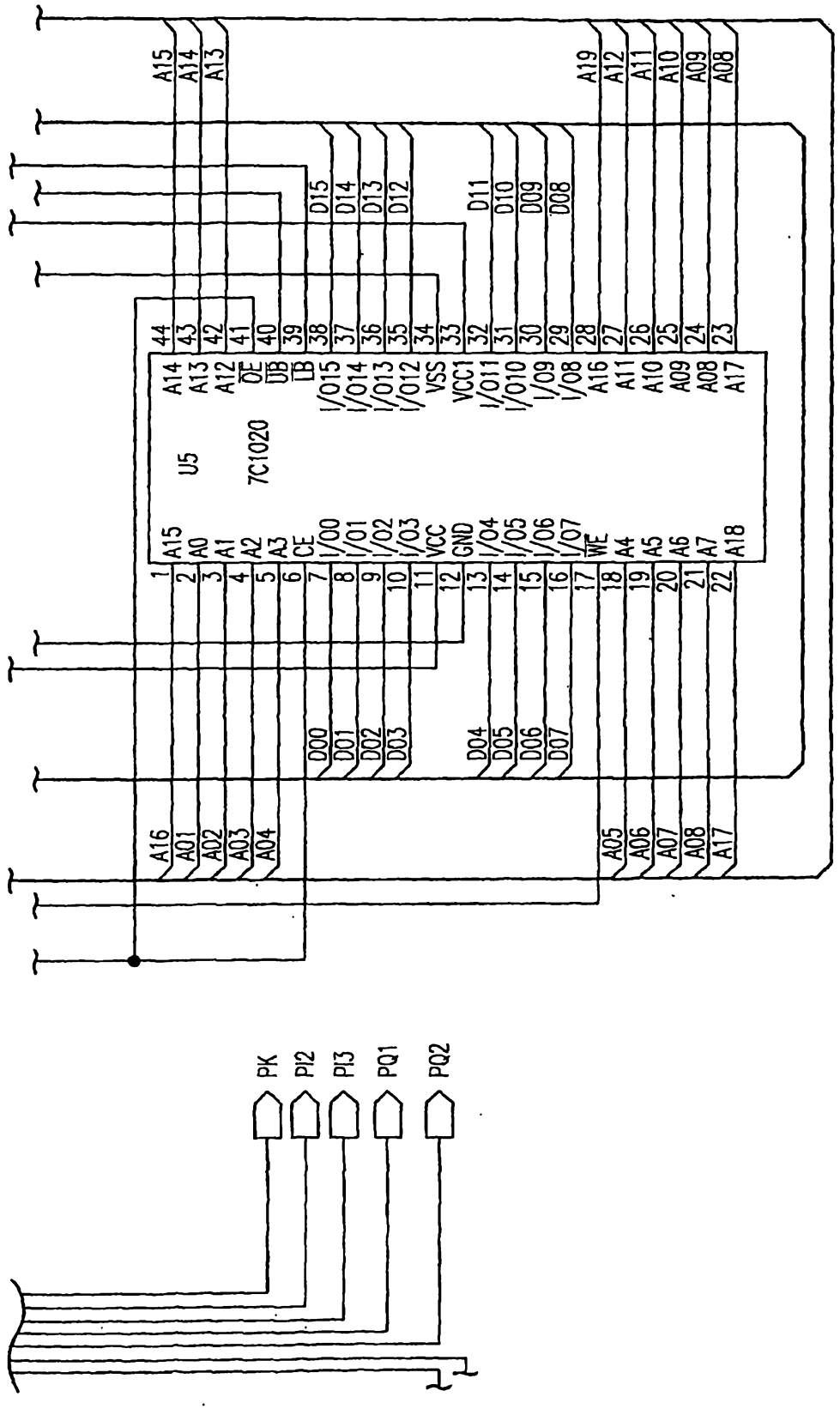

U5
7C1020
A15
A14
A13
A12
OE
UB
LB
I/O15
I/O14
I/O13
I/O12
VSS
VCC1
I/O11
I/O10
I/O9
I/O8
A16
A11
A10
A09
A08
A17
A19
A12
A11
A10
A09
A08
D15
D14
D13
D12
D11
D10
D09
D08
A15
A0
A1
A2
A3
CE
I/O0
I/O1
I/O2
I/O3
VCC
GND
I/O4
I/O5
I/O6
I/O7
WE
A4
A5
A6
A7
A18
D00
D01
D02
D03
D04
D05
D06
D07
A16
A01
A02
A03
A04
A05
A06
A07
A08
A17
PK
PI2
PI3
PQ1
PQ2

FIG.3A-11

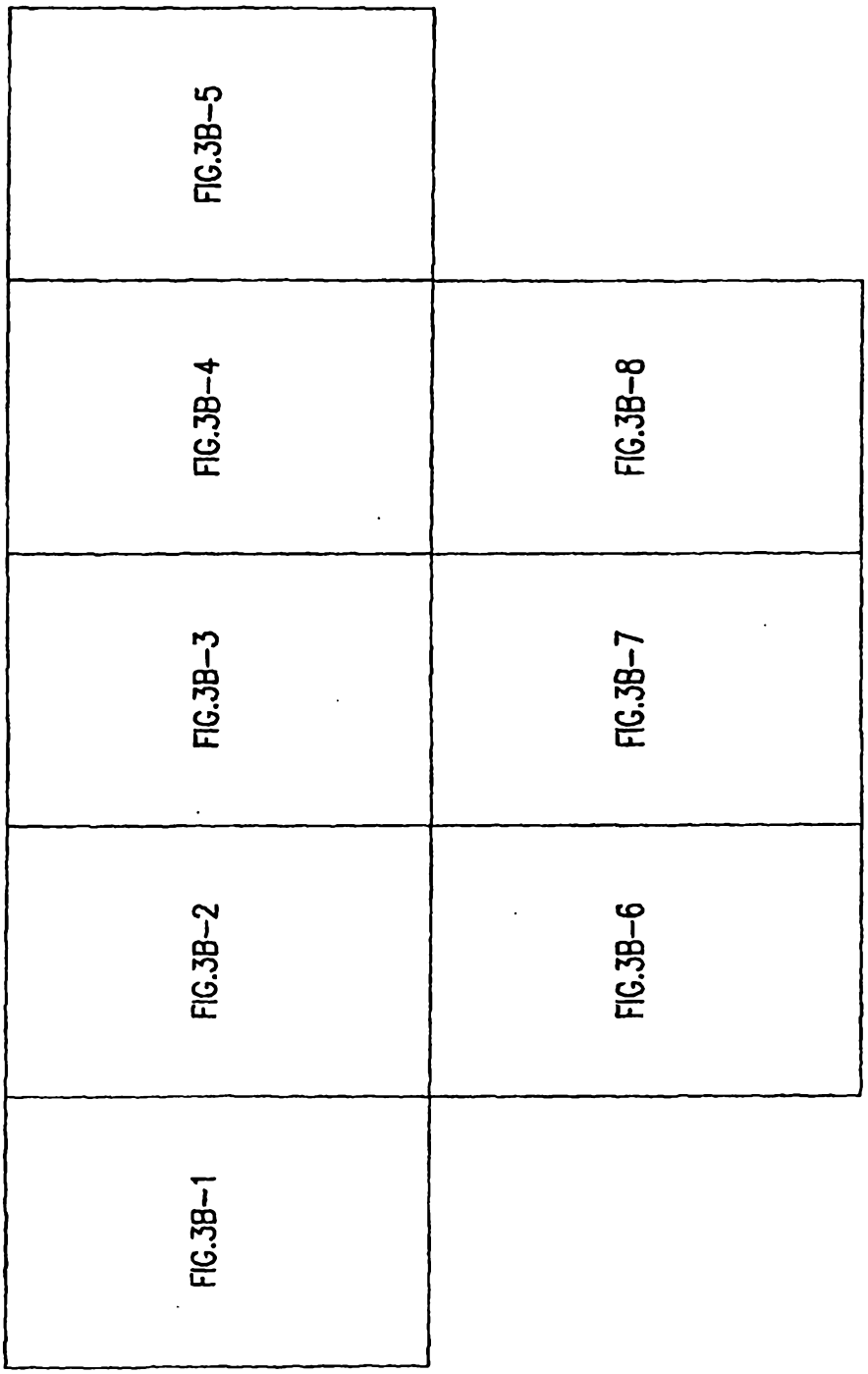


FIG.3B

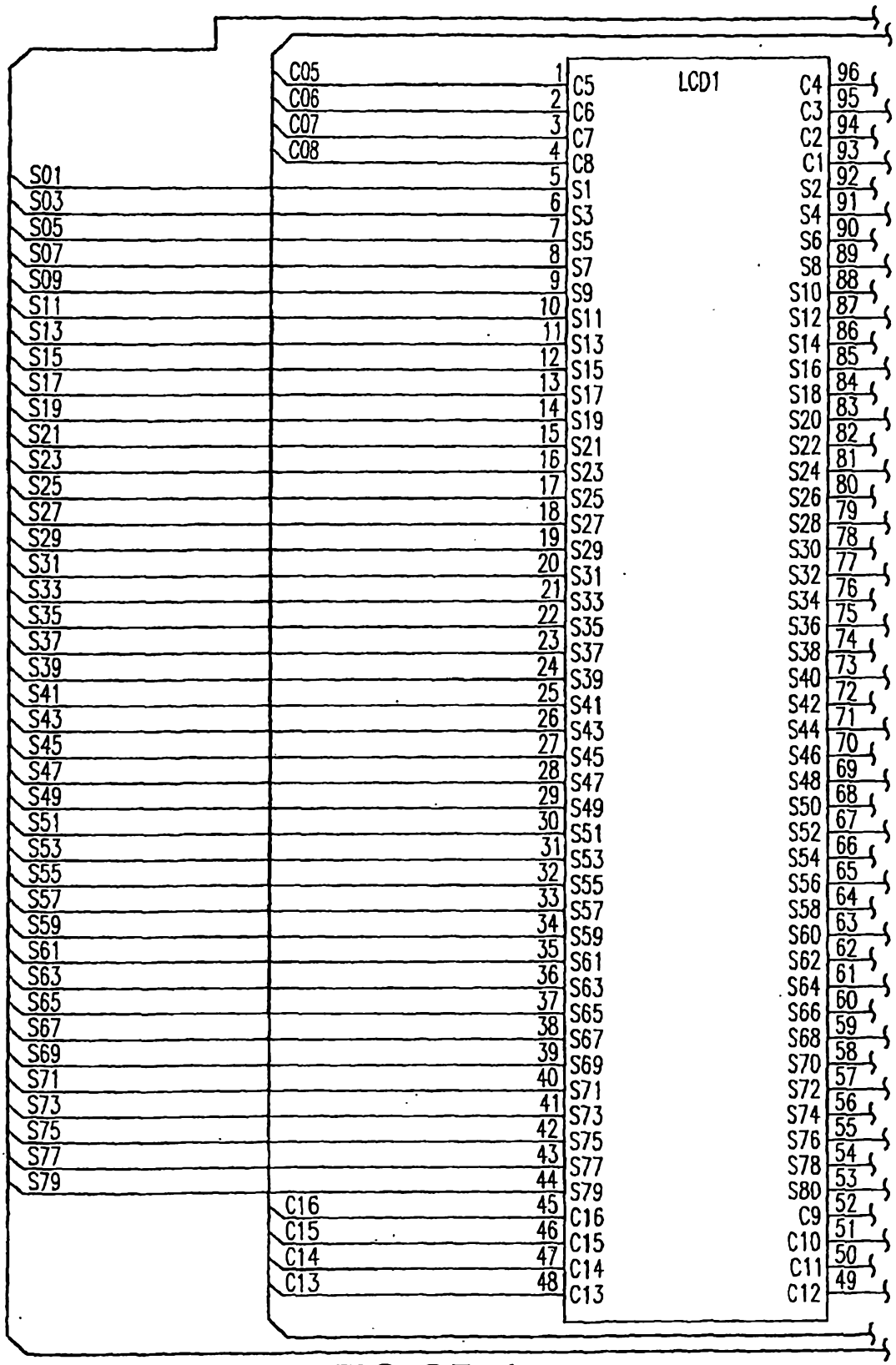

C05 1
C06 2
C07 3
C08 4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
C16
C15
C14
C13
S01
S03
S05
S07
S09
S11
S13
S15
S17
S19
S21
S23
S25
S27
S29
S31
S33
S35
S37
S39
S41
S43
S45
S47
S49
S51
S53
S55
S57
S59
S61
S63
S65
S67
S69
S71
S73
S75
S77
S79
C5
C6
C7
C8
S1
S3
S5
S7
S9
S11
S13
S15
S17
S19
S21
S23
S25
S27
S29
S31
S33
S35
S37
S39
S41
S43
S45
S47
S49
S51
S53
S55
S57
S59
S61
S63
S65
S67
S69
S71
S73
S75
S77
S79
C16
C15
C14
C13
LCD1
C4 96
C3 95
C2 94
C1 93
S2 92
S4 91
S6 90
S8 89
S10 88
S12 87
S14 86
S16 85
S18 84
S20 83
S22 82
S24 81
S26 80
S28 79
S30 78
S32 77
S34 76
S36 75
S38 74
S40 73
S42 72
S44 71
S46 70
S48 69
S50 68
S52 67
S54 66
S56 65
S58 64
S60 63
S62 62
S64 61
S66 60
S68 59
S70 58
S72 57
S74 56
S76 55
S78 54
S80 53
C9 52
C10 51
C11 50
C12 49

FIG.3B-1

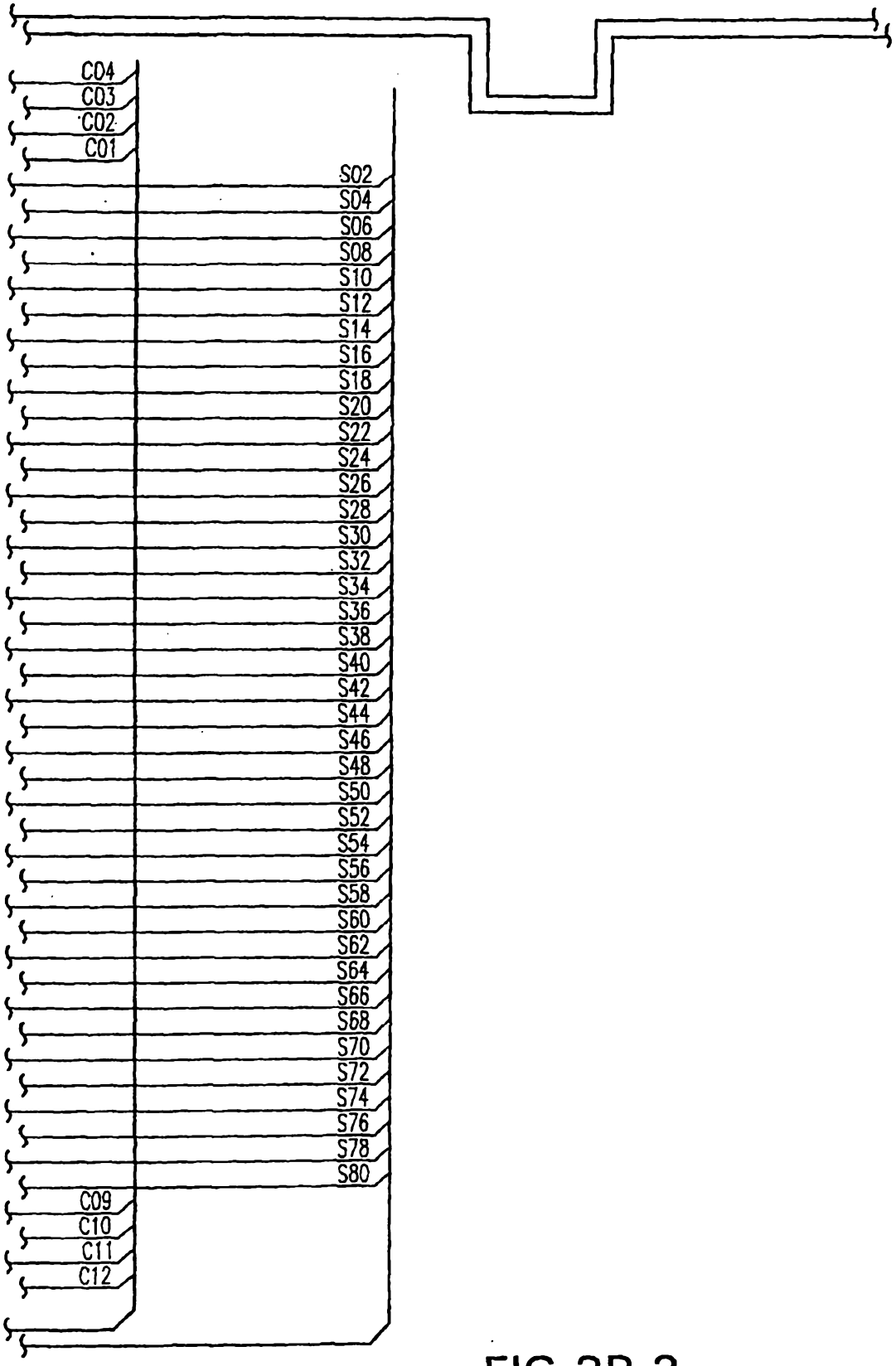

C04
C03
C02
C01
S02
S04
S06
S08
S10
S12
S14
S16
S18
S20
S22
S24
S26
S28
S30
S32
S34
S36
S38
S40
S42
S44
S46
S48
S50
S52
S54
S56
S58
S60
S62
S64
S66
S68
S70
S72
S74
S76
S78
S80
C09
C10
C11
C12

FIG.3B-2

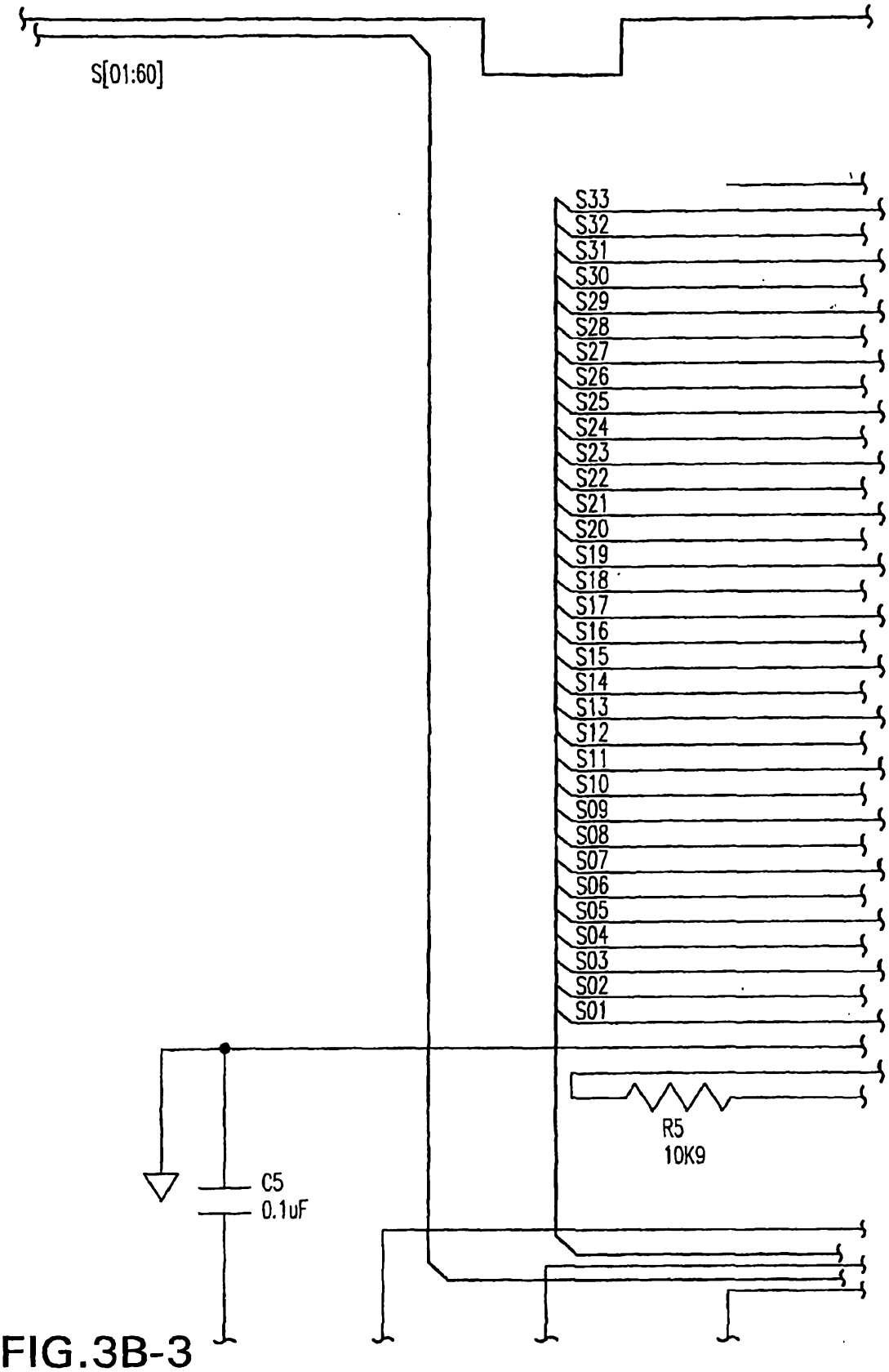

S[01:60]
S33
S32
S31
S30
S29
S28
S27
S26
S25
S24
S23
S22
S21
S20
S19
S18
S17
S16
S15
S14
S13
S12
S11
S10
S09
S08
S07
S06
S05
S04
S03
S02
S01
R5
10K9
C5
0.1uF

FIG.3B-3

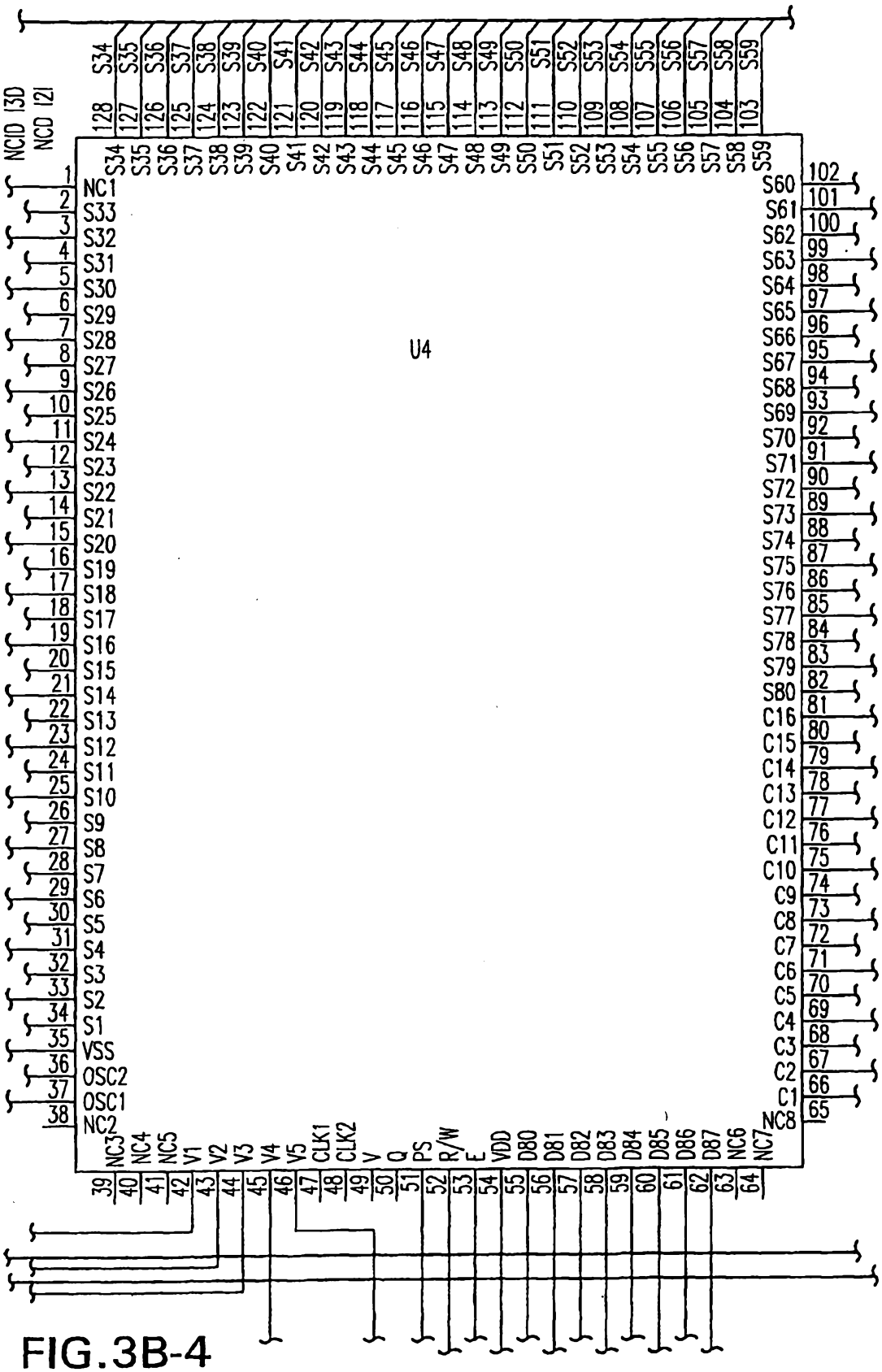

U4
NC1
S33
S32
S31
S30
S29
S28
S27
S26
S25
S24
S23
S22
S21
S20
S19
S18
S17
S16
S15
S14
S13
S12
S11
S10
S9
S8
S7
S6
S5
S4
S3
S2
S1
VSS
OSC2
OSC1
NC2
S60
S61
S62
S63
S64
S65
S66
S67
S68
S69
S70
S71
S72
S73
S74
S75
S76
S77
S78
S79
S80
C16
C15
C14
C13
C12
C11
C10
C9
C8
C7
C6
C5
C4
C3
C2
C1
NC8
NC3
NC4
NC5
V1
V2
V3
V4
V5
CLK1
CLK2
V
Q
PS
R/W
E
VDD
D80
D81
D82
D83
D84
D85
D86
D87
NC6
NC7

FIG.3B-4

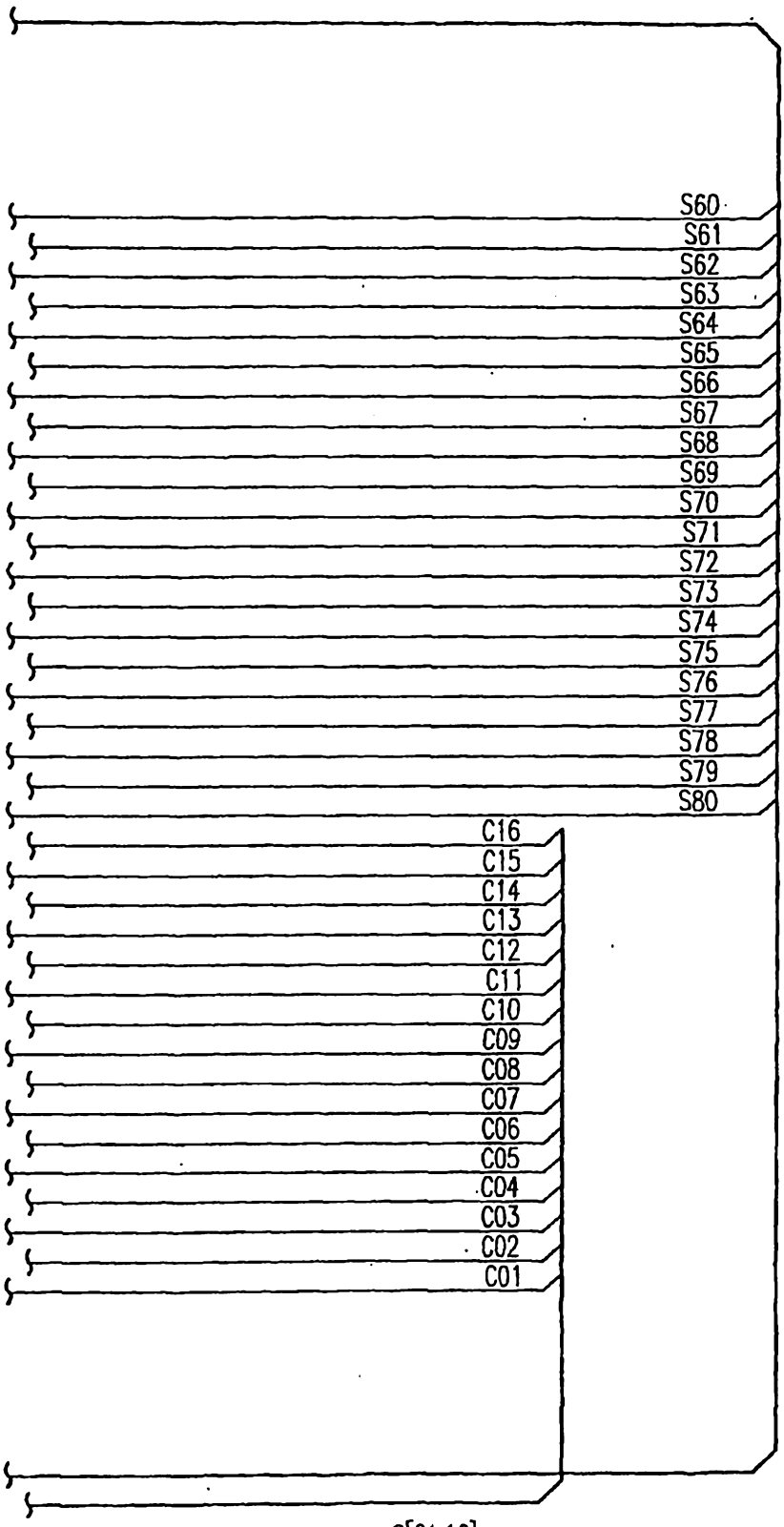

S60
S61
S62
S63
S64
S65
S66
S67
S68
S69
S70
S71
S72
S73
S74
S75
S76
S77
S78
S79
S80
C16
C15
C14
C13
C12
C11
C10
C09
C08
C07
C06
C05
C04
C03
C02
C01
C[01:16]

FIG.3B-5

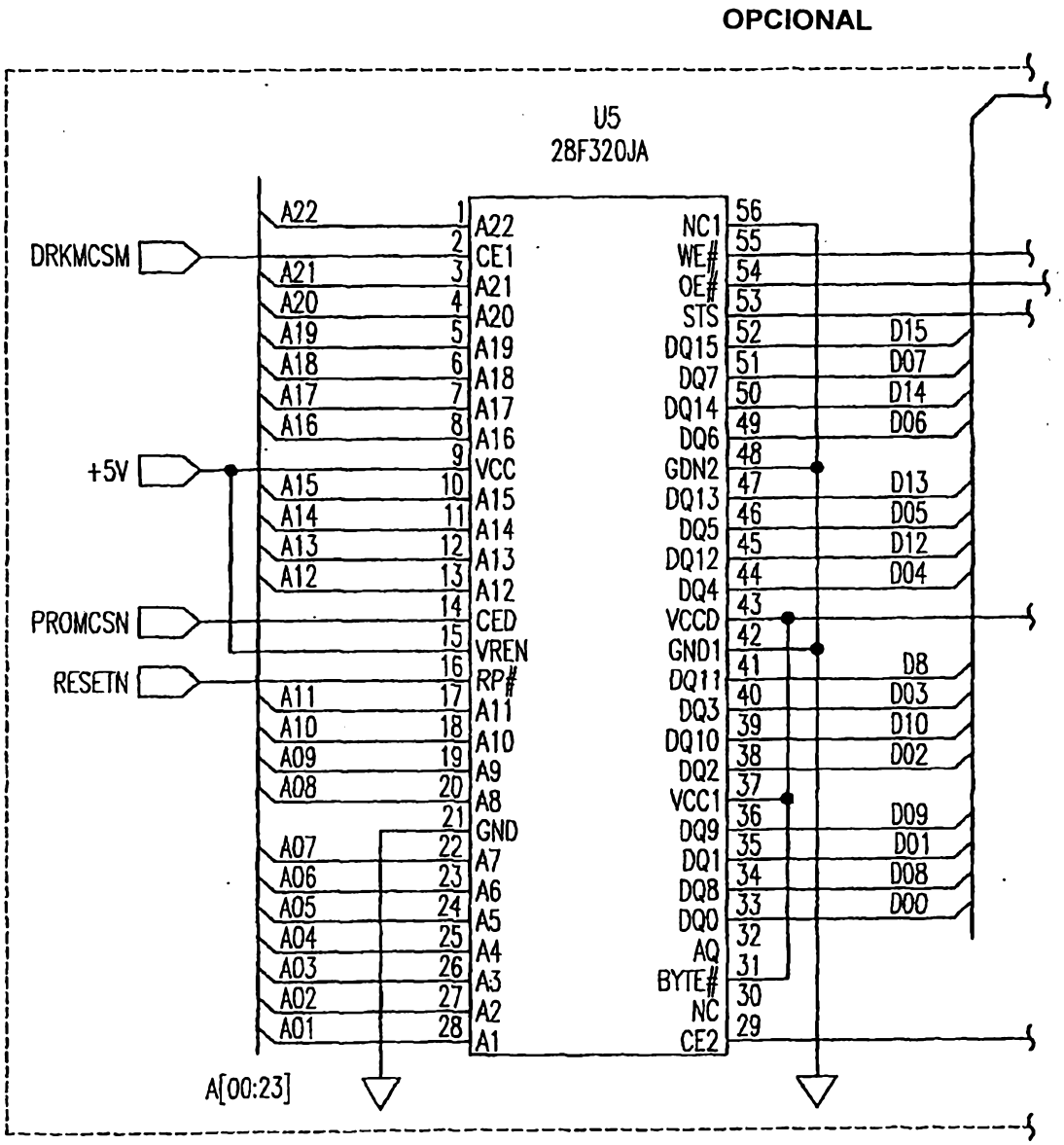

OPCIONAL
U5
28F320JA
DRKMCSM
+5V
PROMCSN
RESETN
A[00:23]
A22 1
A21 3
A20 4
A19 5
A18 6
A17 7
A16 8
9
A15 10
A14 11
A13 12
A12 13
14
15
16
A11 17
A10 18
A09 19
A08 20
21
A07 22
A06 23
A05 24
A04 25
A03 26
A02 27
A01 28
A22 1
CE1 2
A21 3
A20 4
A19 5
A18 6
A17 7
A16 8
VCC 9
A15 10
A14 11
A13 12
A12 13
CED 14
VREN 15
RP# 16
A11 17
A10 18
A9 19
A8 20
GND 21
A7 22
A6 23
A5 24
A4 25
A3 26
A2 27
A1 28
NC1 56
WE# 55
OE# 54
STS 53
DQ15 52
DQ7 51
DQ14 50
DQ6 49
GDN2 48
DQ13 47
DQ5 46
DQ12 45
DQ4 44
VCCD 43
GND1 42
DQ11 41
DQ3 40
DQ10 39
DQ2 38
VCC1 37
DQ9 36
DQ1 35
DQ8 34
DQ0 33
AQ 32
BYTE# 31
NC 30
CE2 29
D15
D07
D14
D06
D13
D05
D12
D04
D8
D03
D10
D02
D09
D01
D08
D00

FIG.3B-6

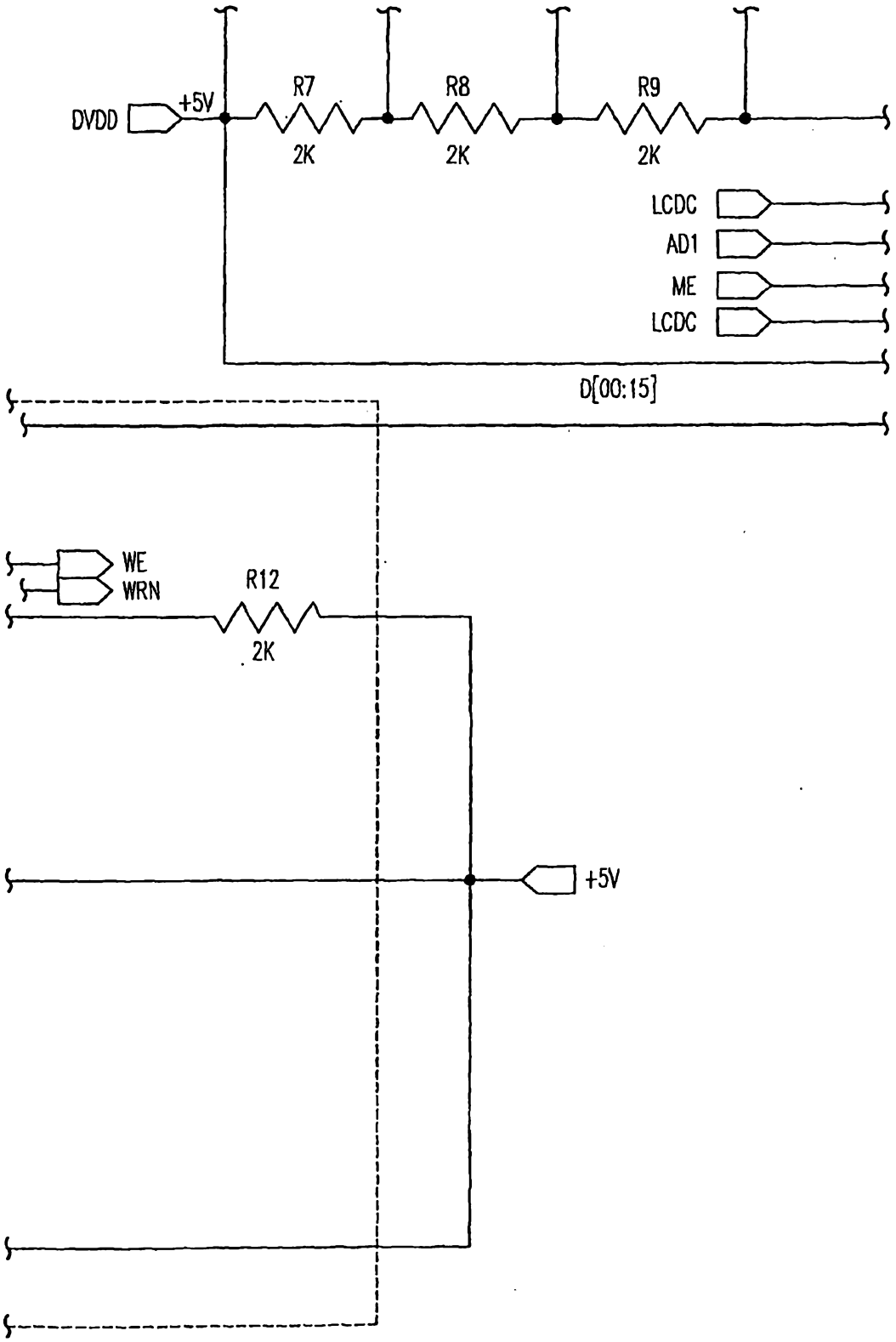


FIG.3B-7

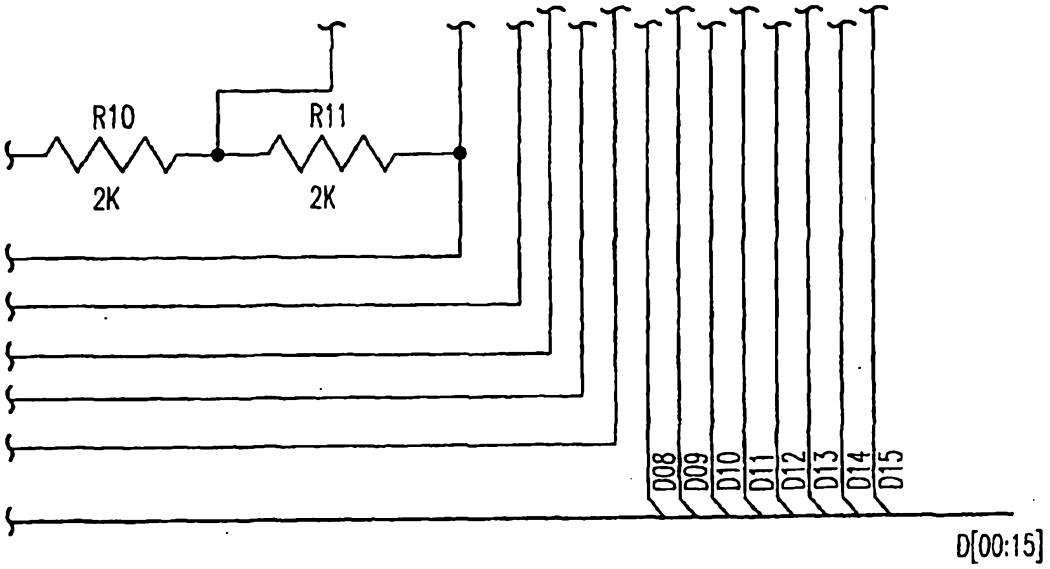


FIG.3B-8

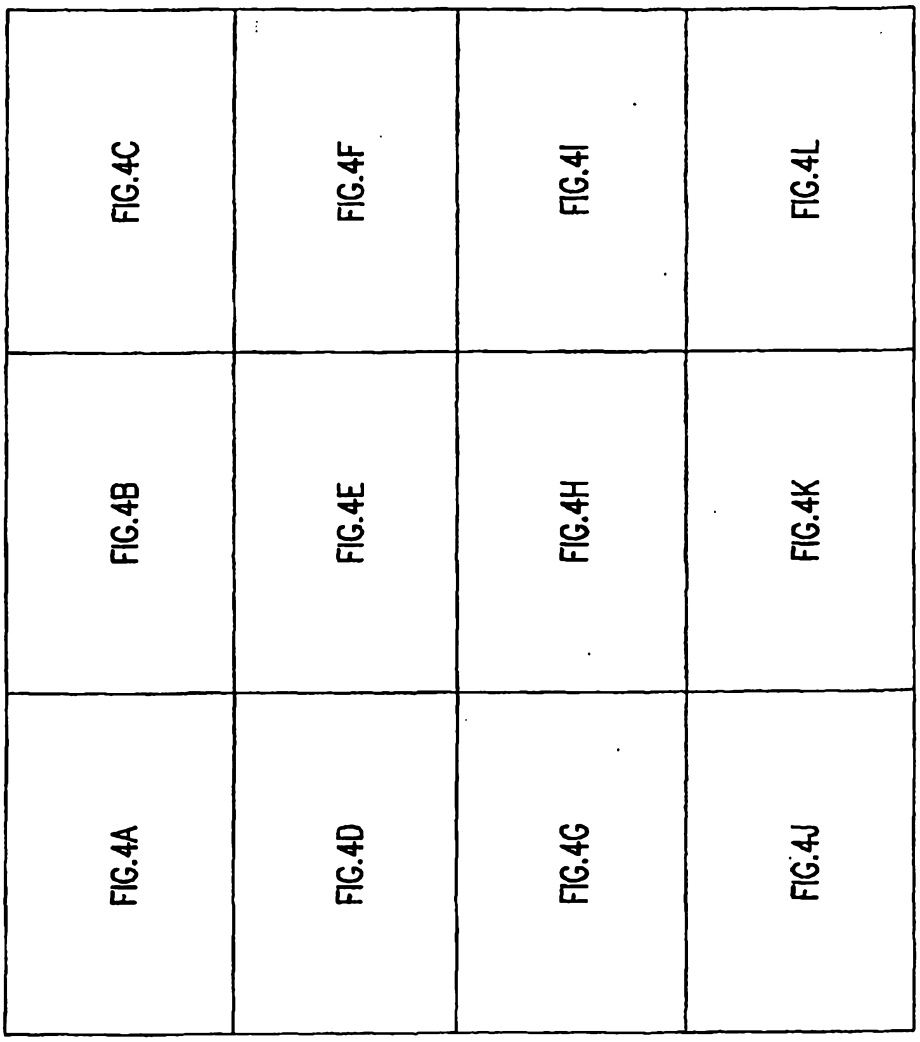


FIG.4C
FIG.4F
FIG.4I
FIG.4L
FIG.4B
FIG.4E
FIG.4H
FIG.4K
FIG.4A
FIG.4D
FIG.4G
FIG.4J

FIG.4

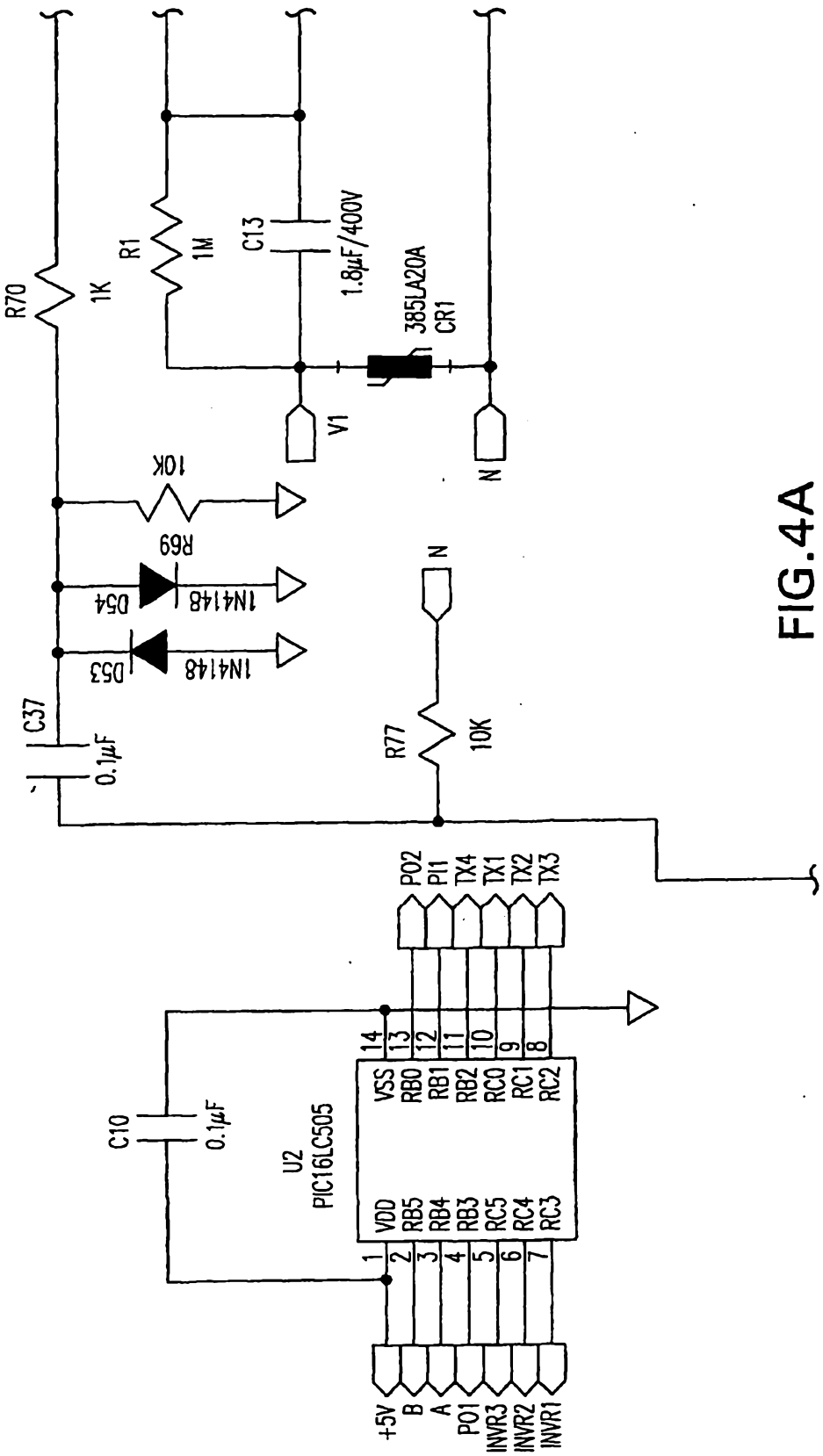

R70
1K
R1
1M
C13
1.8µF/400V
385LA20A
CR1
V1
N
10K
R69
D54
1N4148
D53
1N4148
C37
0.1µF
R77
10K
N
P02
PI1
TX4
TX1
TX2
TX3
14
13
12
11
10
9
8
VSS
RB0
RB1
RB2
RC0
RC1
RC2
C10
0.1µF
U2
PIC16LC505
VDD
RB5
RB4
RB3
RC5
RC4
RC3
1
2
3
4
5
6
7
+5V
B
A
PO1
INVR3
INVR2
INVR1

FIG.4A

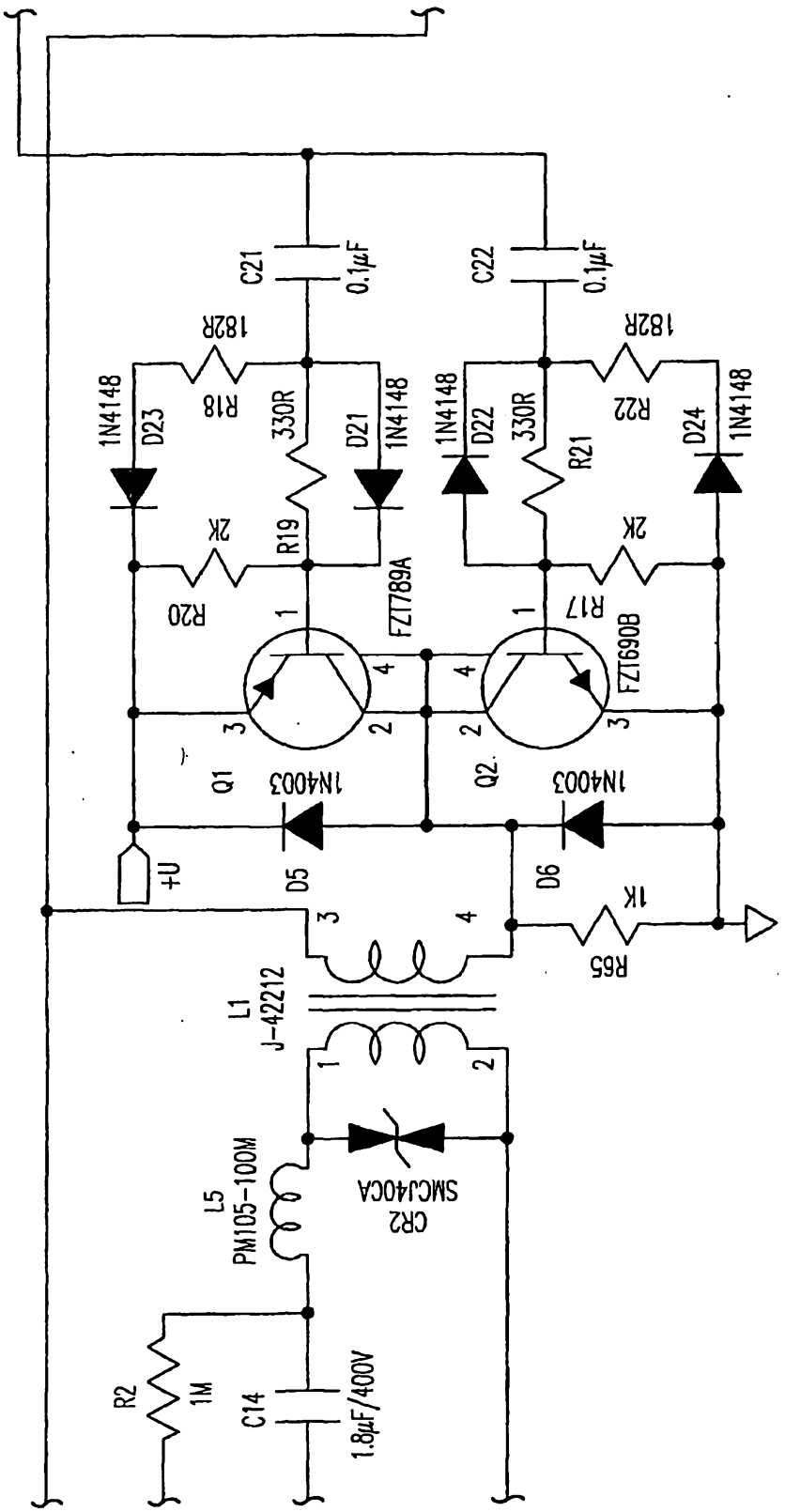

C21
0.1µF
C22
0.1µF
182R
182R
1N4148
D23
R18
330R
D21
1N4148
1N4148
D22
330R
R21
R22
D24
1N4148
2K
R19
2K
R20
R17
FZT789A
FZT690B
Q1
Q2
1N4003
1N4003
D5
D6
+U
1K
R65
L1
J-42212
L5
PM105-100M
CR2
SMCJ40CA
R2
1M
C14
1.8µF/400V

FIG.4B

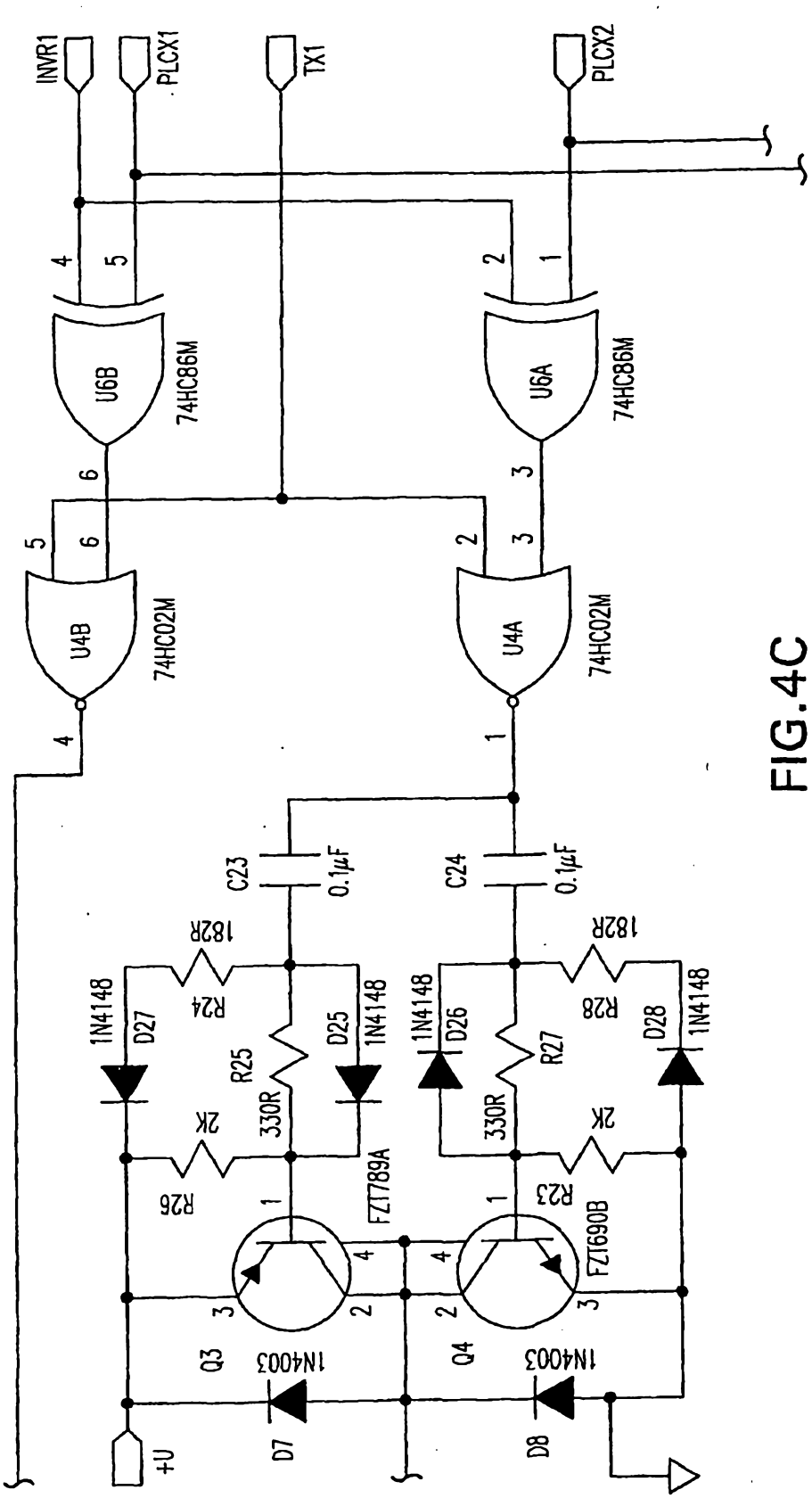

INVR1
PLCX1
TX1
PLCX2
U6B
74HC86M
U6A
74HC86M
U4B
74HC02M
U4A
74HC02M
C23
0.1µF
C24
0.1µF
182R
R24
R28
1N4148
D27
D25
D26
D28
R25
330R
R27
2K
R26
R23
FZT789A
FZT690B
Q3
Q4
1N4003
D7
D8
+U

FIG.4C

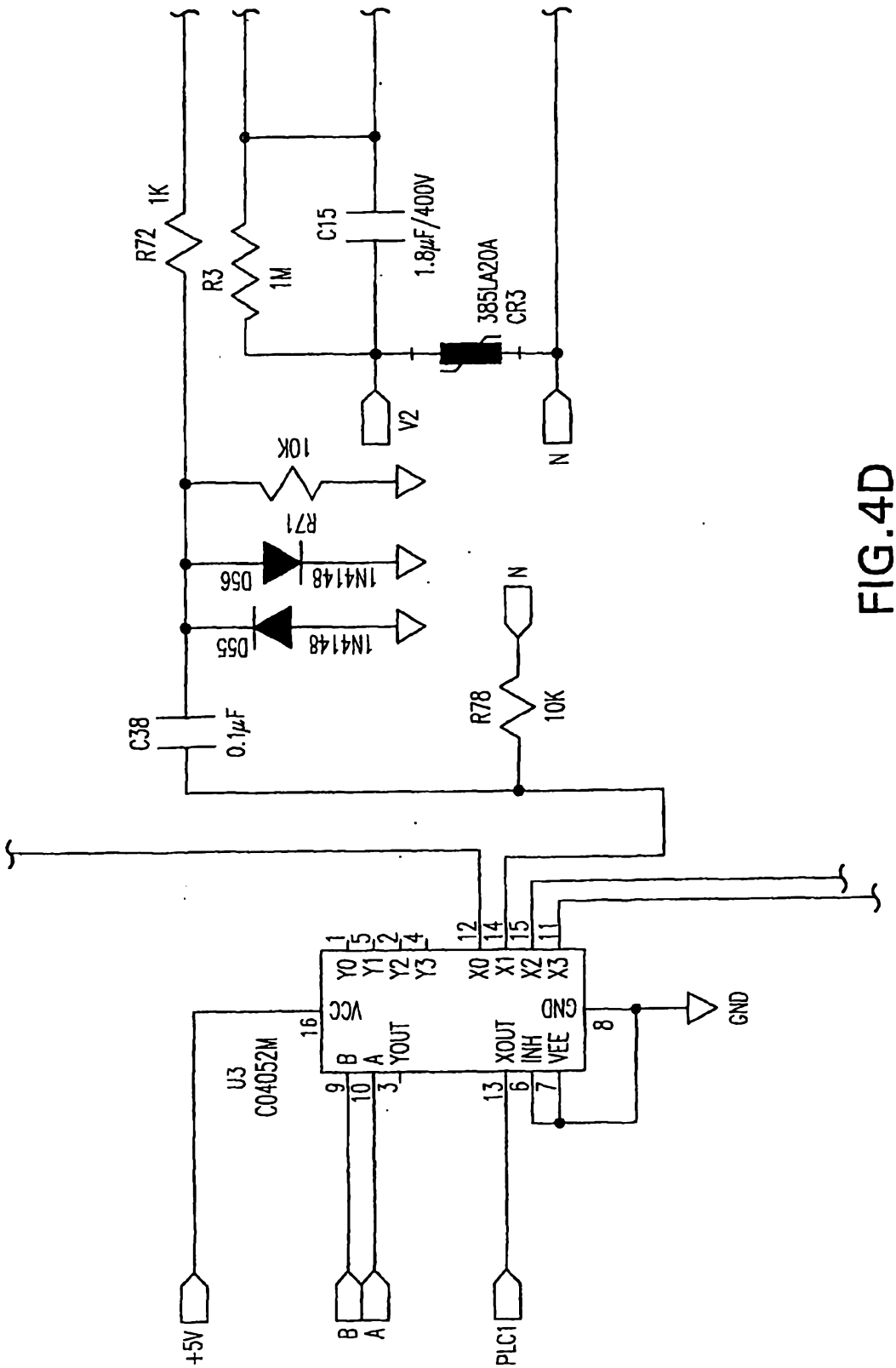

R72 1K
R3 1M
C15 1.8µF/400V
385LA20A CR3
V2
N
10K
R71
D56 1N4148
D55 1N4148
C38 0.1µF
R78 10K
U3 CD4052M
Y0 Y1 Y2 Y3 X0 X1 X2 X3
VCC GND
B A YOUT XOUT INH VEE
+5V
B A
PLC1
GND

FIG.4D

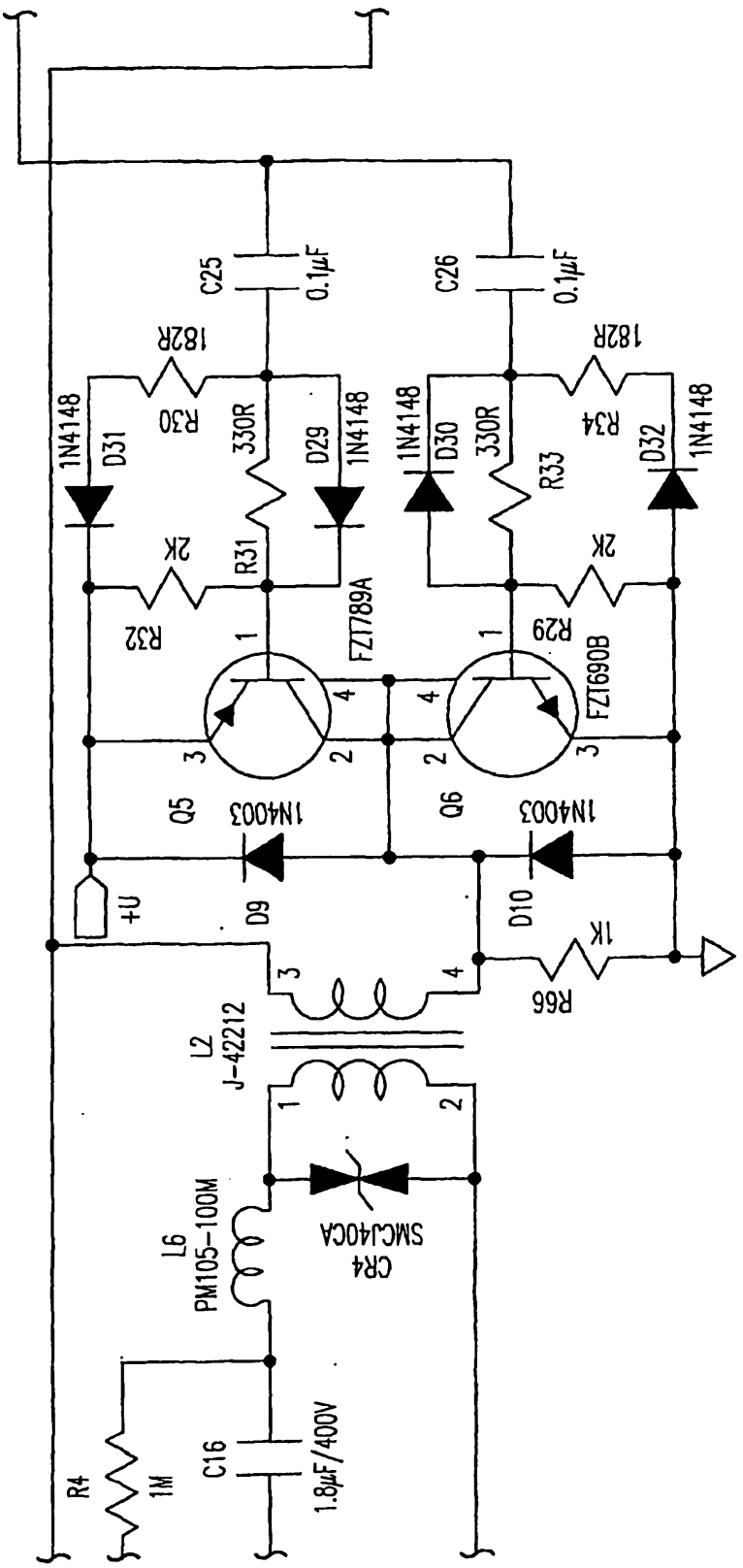

C25
0.1µF
C26
0.1µF
182R
R30
330R
R31
2K
R32
1N4148
D31
D29
1N4148
182R
R34
330R
R33
2K
R29
1N4148
D30
D32
1N4148
FZT789A
FZT690B
Q5
Q6
1N4003
1N4003
D9
D10
+U
1K
R66
L2
J-42212
L6
PM105-100M
SMCJ40CA
CR4
R4
1M
C16
1.8µF/400V

FIG.4E

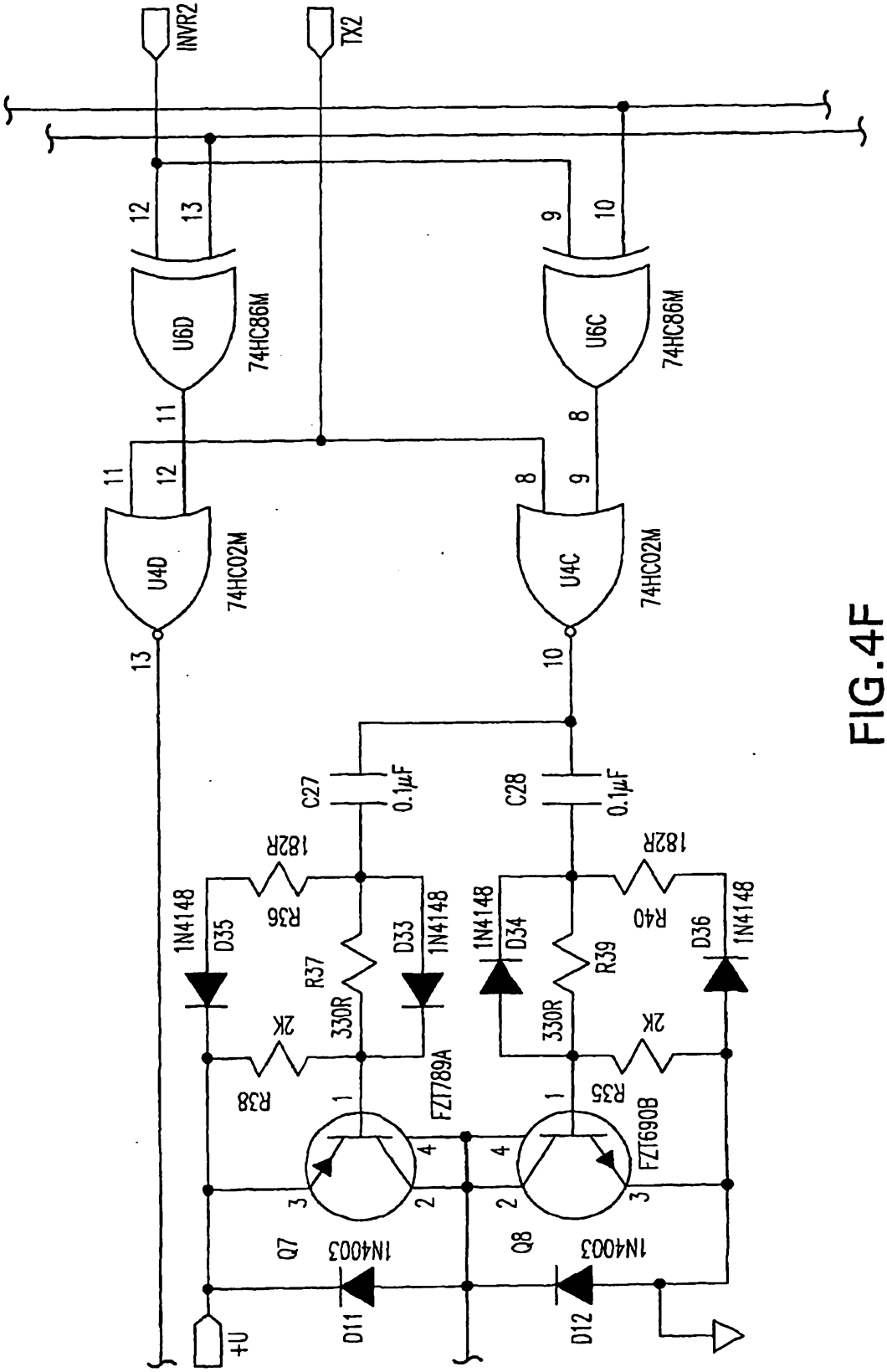

INVR2
TX2
U6D
74HC86M
U6C
74HC86M
U4D
74HC02M
U4C
74HC02M
C27
0.1µF
C28
0.1µF
182R
R36
R40
1N4148
D35
D33
D34
D36
R37
R39
330R
2K
R38
R35
FZT789A
FZT690B
Q7
Q8
1N4003
D11
D12
+U

FIG.4F

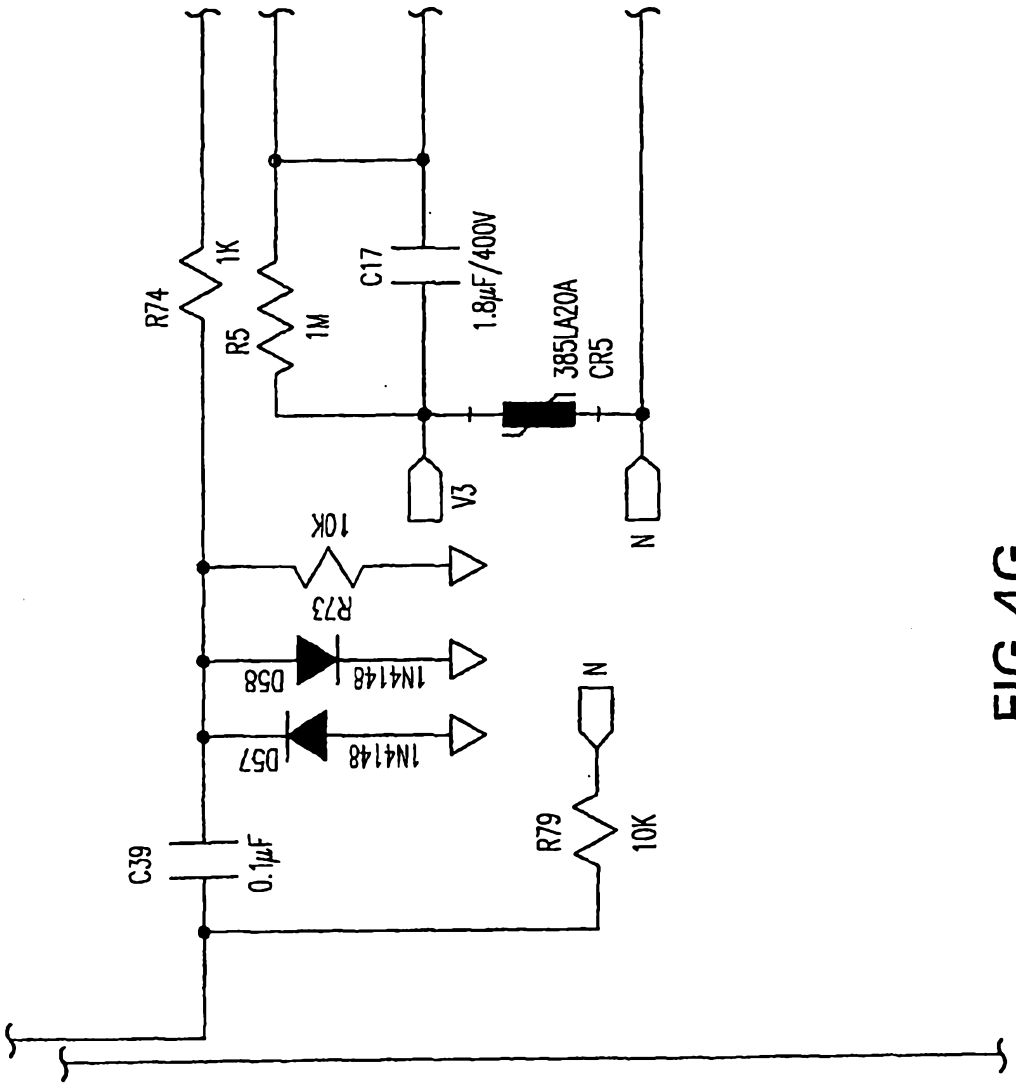

R74
1K
R5
1M
C17
1.8μF/400V
385LA20A
CR5
V3
N
10K
R73
D58
1N4148
D57
1N4148
N
R79
10K
C39
0.1μF

FIG.4G

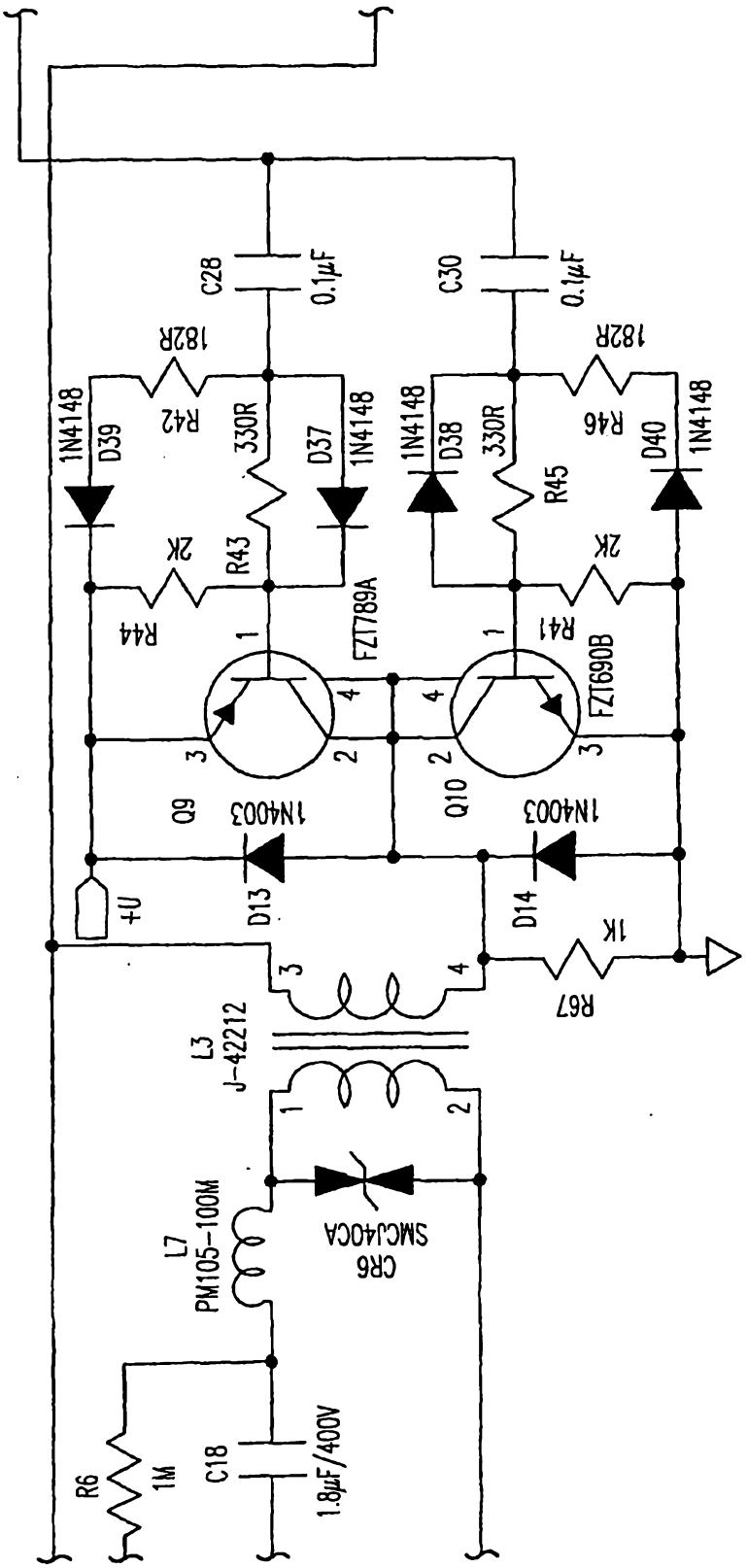

C28
0.1µF
C30
0.1µF
182R
182R
1N4148
D39
R42
330R
D37
1N4148
1N4148
D38
330R
R45
R46
D40
1N4148
2K
R43
2K
R44
R41
FZT789A
FZT690B
Q9
Q10
1N4003
1N4003
+U
D13
D14
1K
R67
L3
J-42212
L7
PM105-100M
CR6
SMCJ40CA
R6
1M
C18
1.8µF/400V

FIG.4H

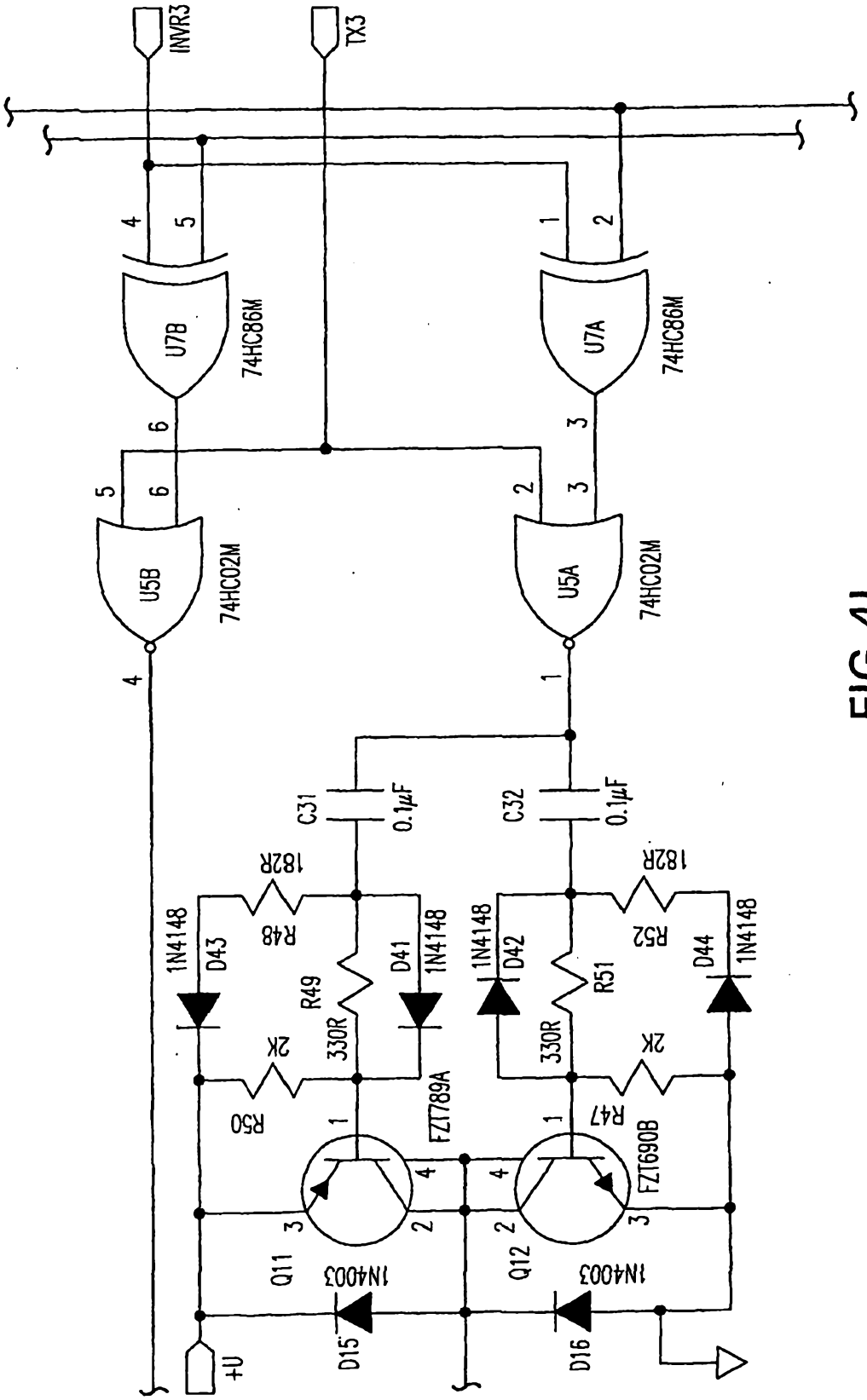

INVR3
TX3
U7B
74HC86M
U7A
74HC86M
U5B
74HC02M
U5A
74HC02M
C31
0.1µF
C32
0.1µF
182R
182R
R48
R52
1N4148
D43
1N4148
D41
1N4148
D42
1N4148
D44
R49
330R
R51
330R
2K
2K
R50
R47
FZT789A
FZT690B
Q11
Q12
1N4003
1N4003
D15
D16
+U

FIG.4I

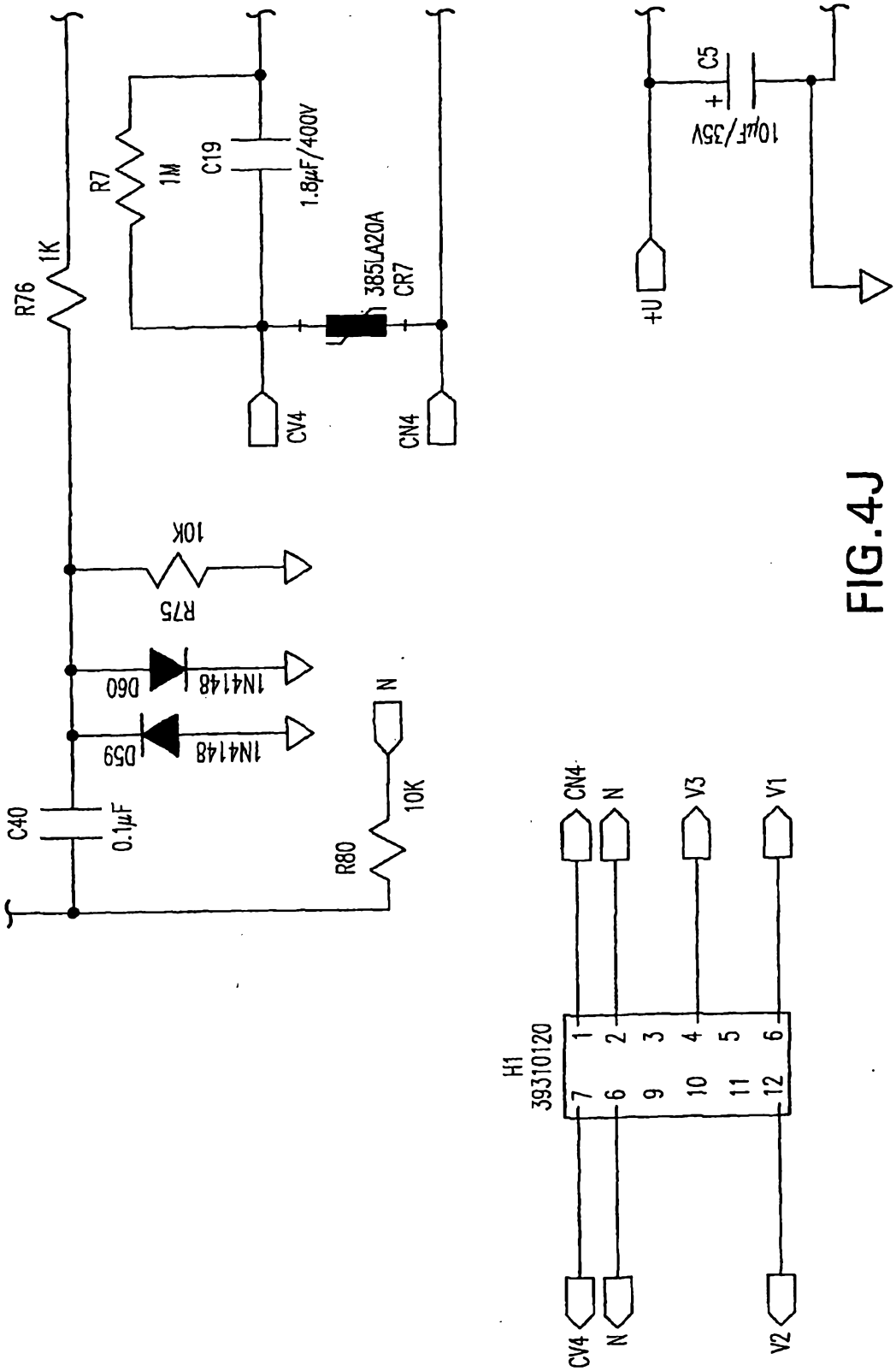

R76
1K
R7
1M
C19
1.8µF/400V
385LA20A
CR7
CV4
CN4
+U
C5
10µF/35V
R75
10K
D60
1N4148
D59
1N4148
C40
0.1µF
R80
10K
N
H1
39310120
CN4
N
V3
V1
CV4
N
V2

FIG.4J

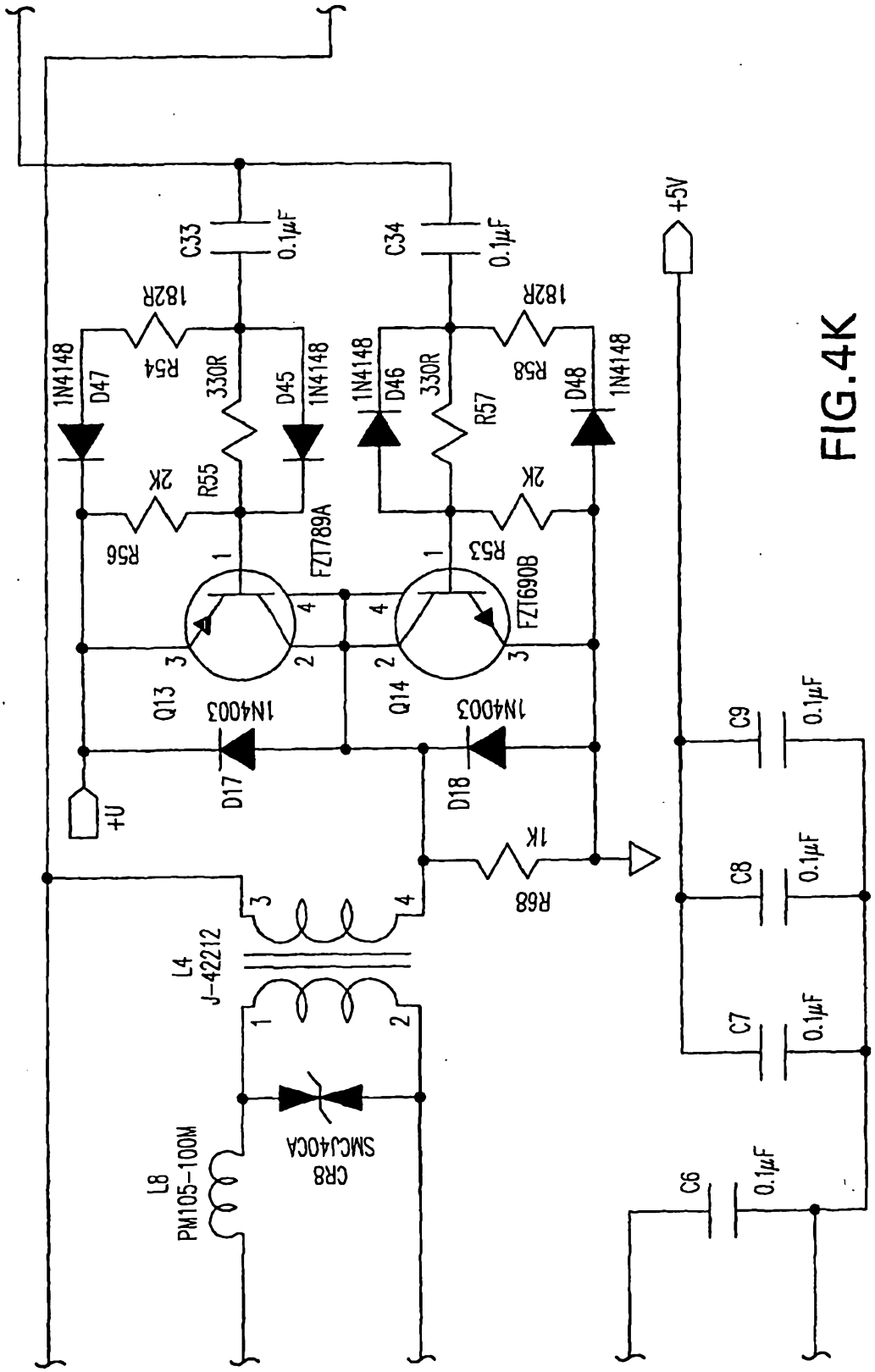

C33
0.1µF
C34
0.1µF
+5V
182R
182R
1N4148
D47
R54
330R
D45
1N4148
1N4148
D46
330R
R57
R58
D48
1N4148
2K
R55
2K
R56
R53
FZT789A
FZT690B
Q13
Q14
1N4003
1N4003
D17
D18
+U
1K
R68
C9
0.1µF
C8
0.1µF
C7
0.1µF
C6
0.1µF
L4
J-42212
L8
PM105-100M
CR8
SMCJ40CA

FIG.4K

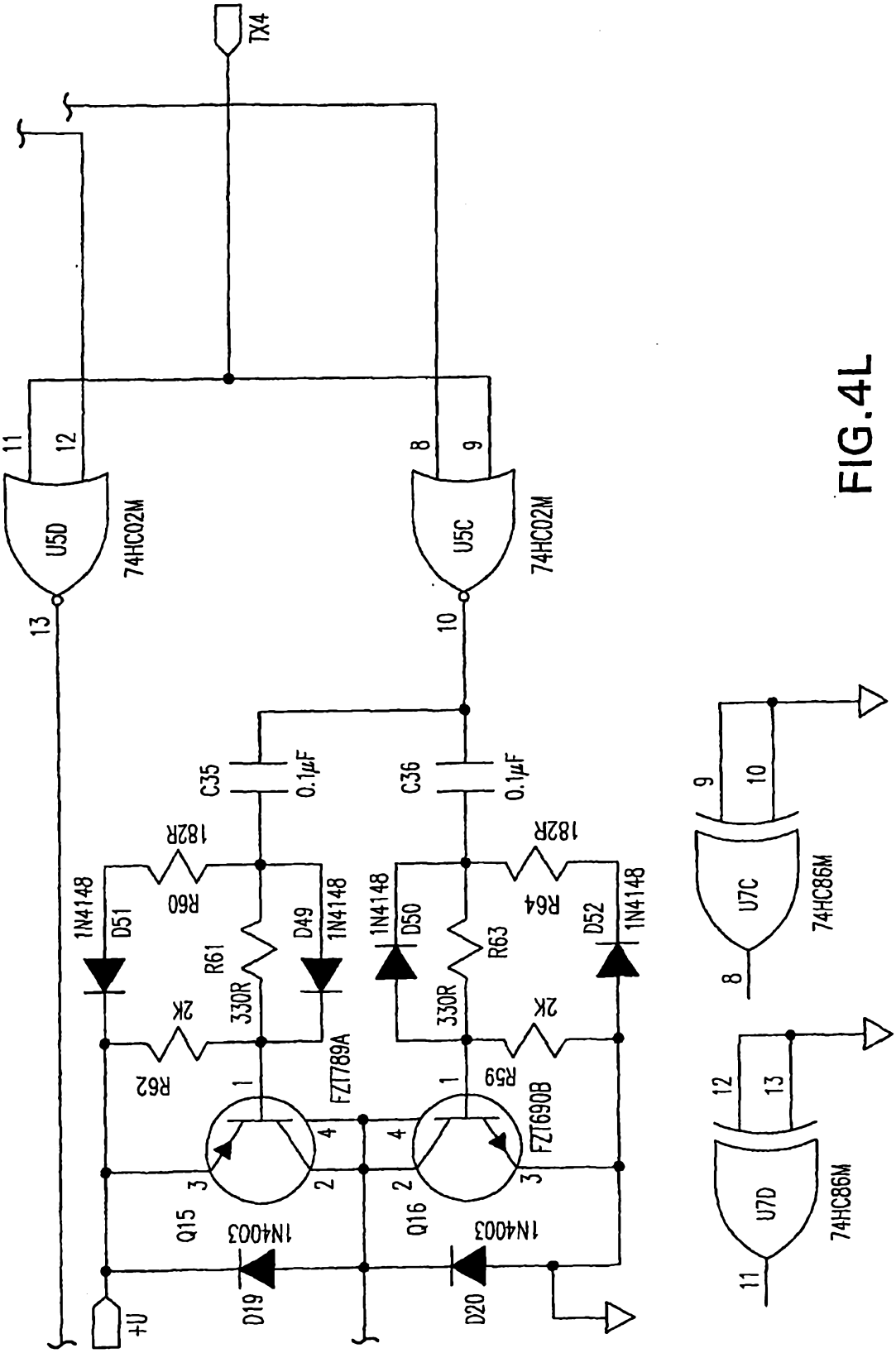

TX4
U5D
74HC02M
U5C
74HC02M
C35
0.1µF
C36
0.1µF
182R
R60
182R
R64
1N4148
D51
1N4148
D49
1N4148
D50
1N4148
D52
R61
330R
R63
330R
2K
R62
2K
R59
FZT789A
FZT690B
Q15
Q16
1N4003
D19
1N4003
D20
+U
U7C
74HC86M
U7D
74HC86M

FIG.4L

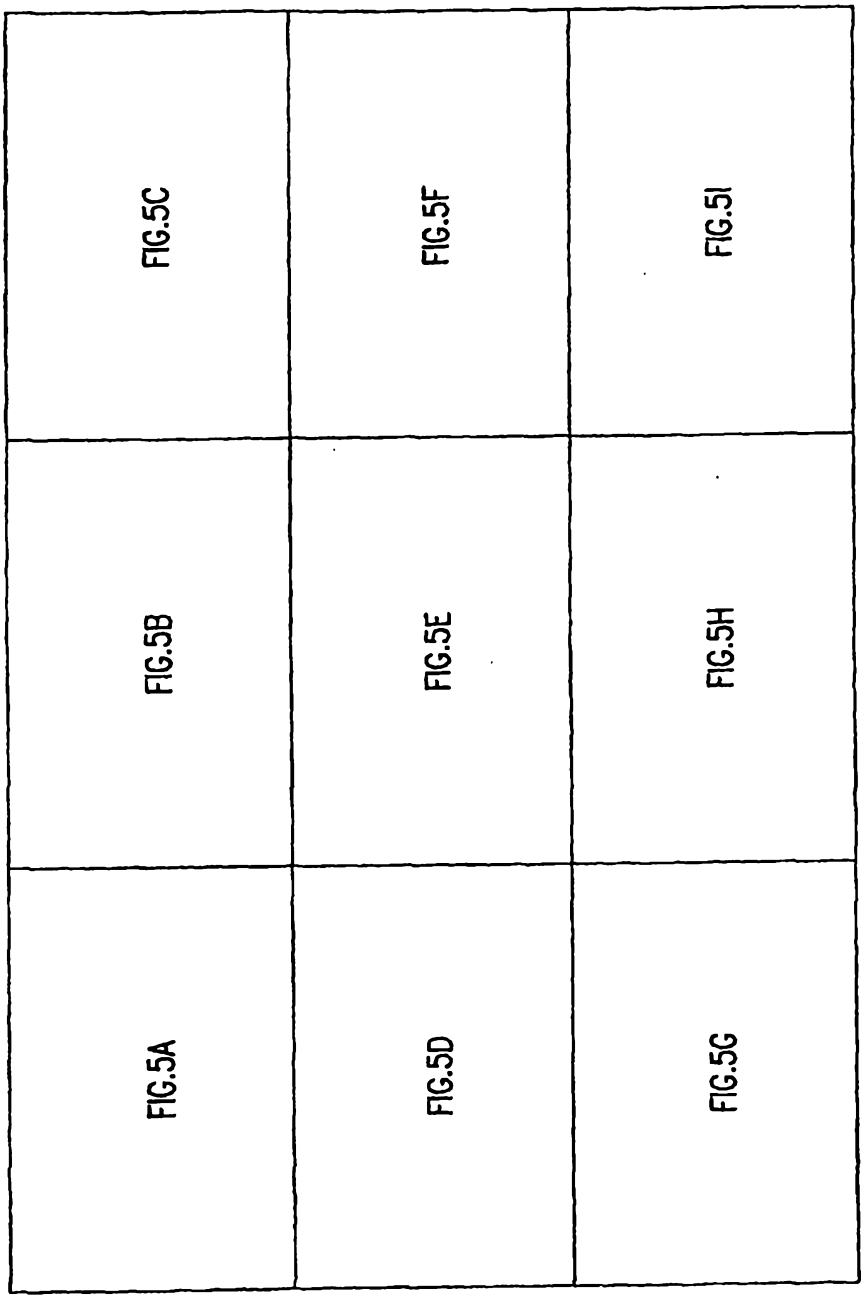


FIG.5

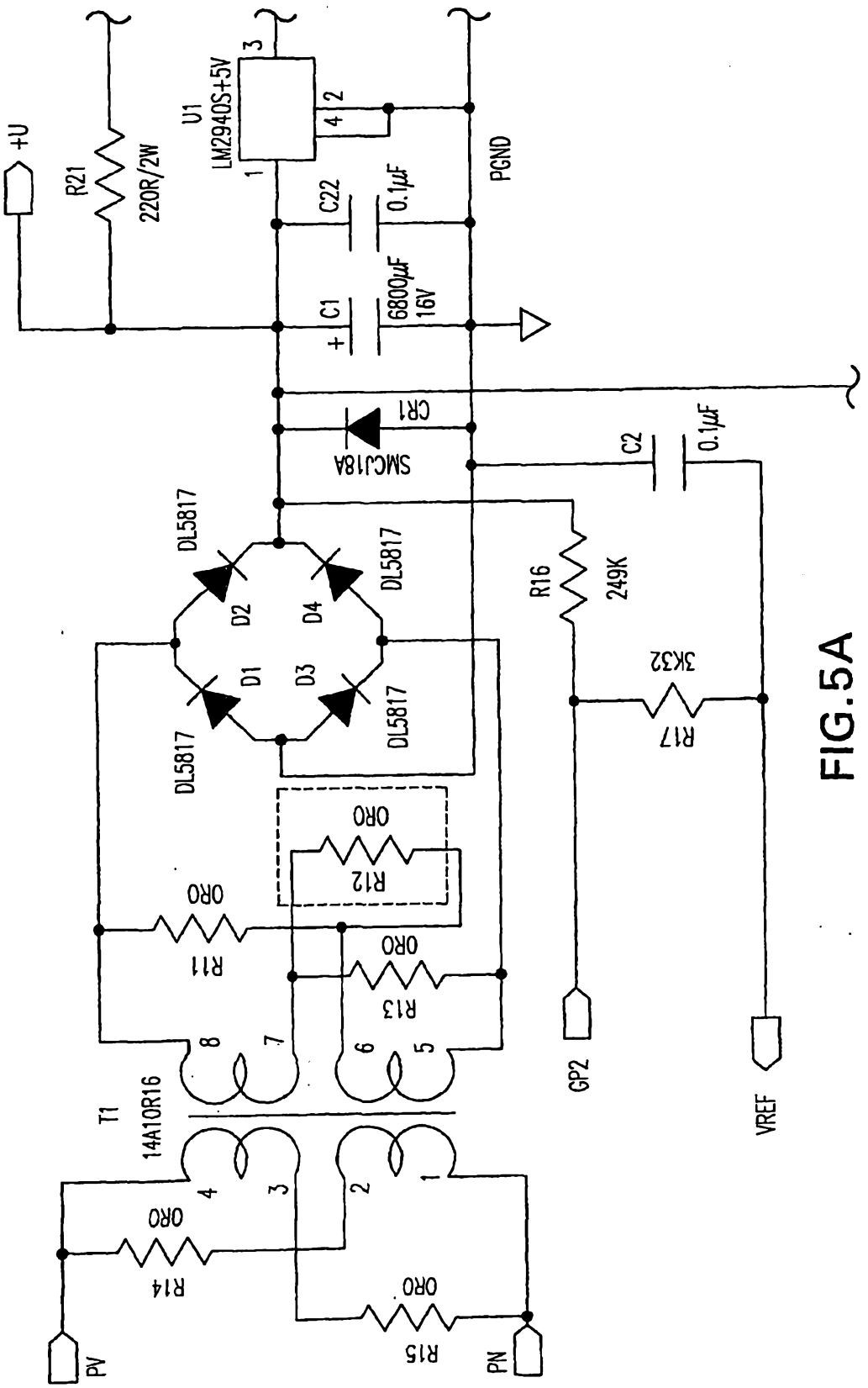

+U
R21
220R/2W
U1
LM2940S+5V
PGND
C22
0.1µF
C1
6800µF
16V
CR1
SMCJ18A
C2
0.1µF
DL5817
D2
D4
D1
D3
R16
249K
3K32
R17
R12
ORO
R11
R13
T1
14A10R16
R14
R15
PV
PN
GP2
VREF

FIG.5A

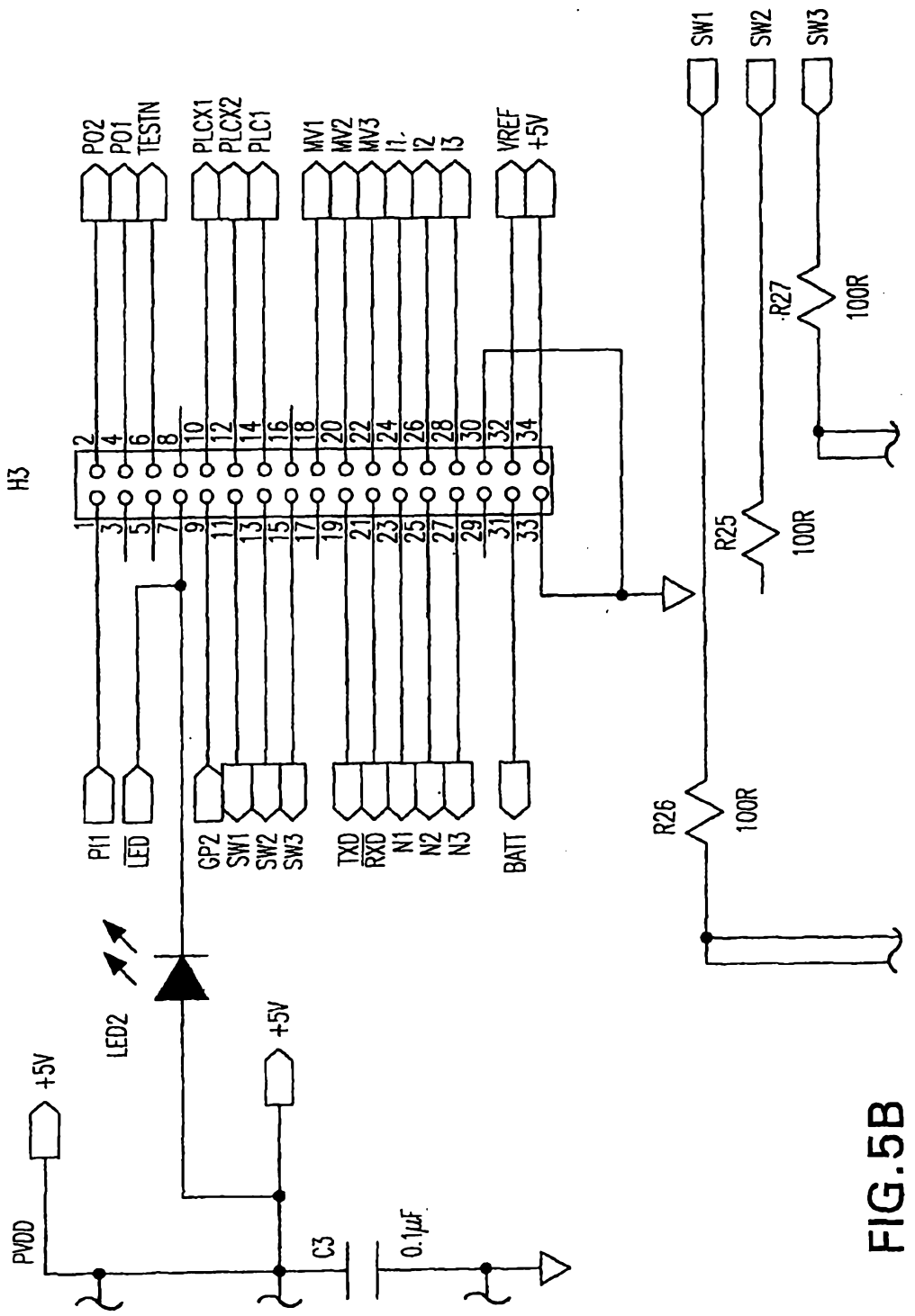

H3
PO2
PO1
TESTN
PLCX1
PLCX2
PLC1
MV1
MV2
MV3
I1
I2
I3
VREF
+5V
PI1
LED
GP2
SW1
SW2
SW3
TXD
RXD
N1
N2
N3
BATT
SW1
SW2
SW3
R25
100R
R26
100R
R27
100R
LED2
+5V
+5V
PVDD
C3
0.1µF

FIG.5B

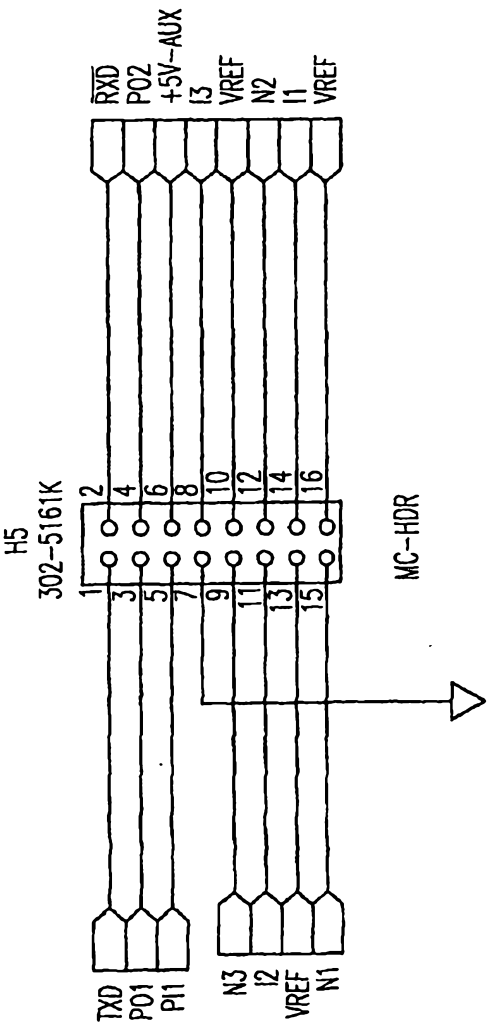

H5
302-5161K
MC-HDR
RXD
PO2
+5V-AUX
I3
VREF
N2
I1
VREF
TXD
PO1
PI1
N3
I2
VREF
N1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

FIG.5C

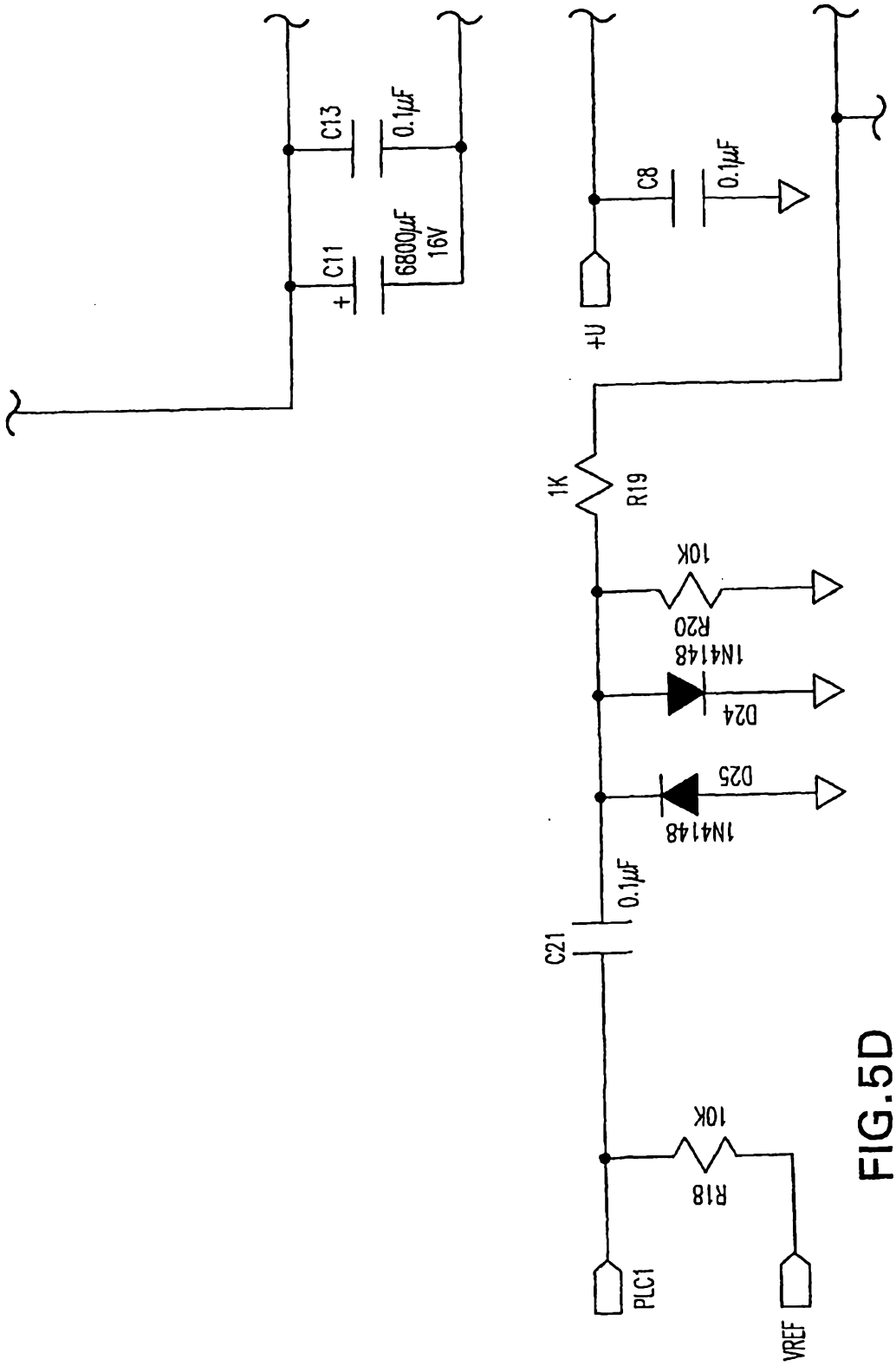

C13
0.1µF
C11
+
6800µF
16V
C8
0.1µF
+U
1K
R19
10K
R20
1N4148
D24
D25
1N4148
0.1µF
C21
10K
R18
PLC1
VREF

FIG.5D

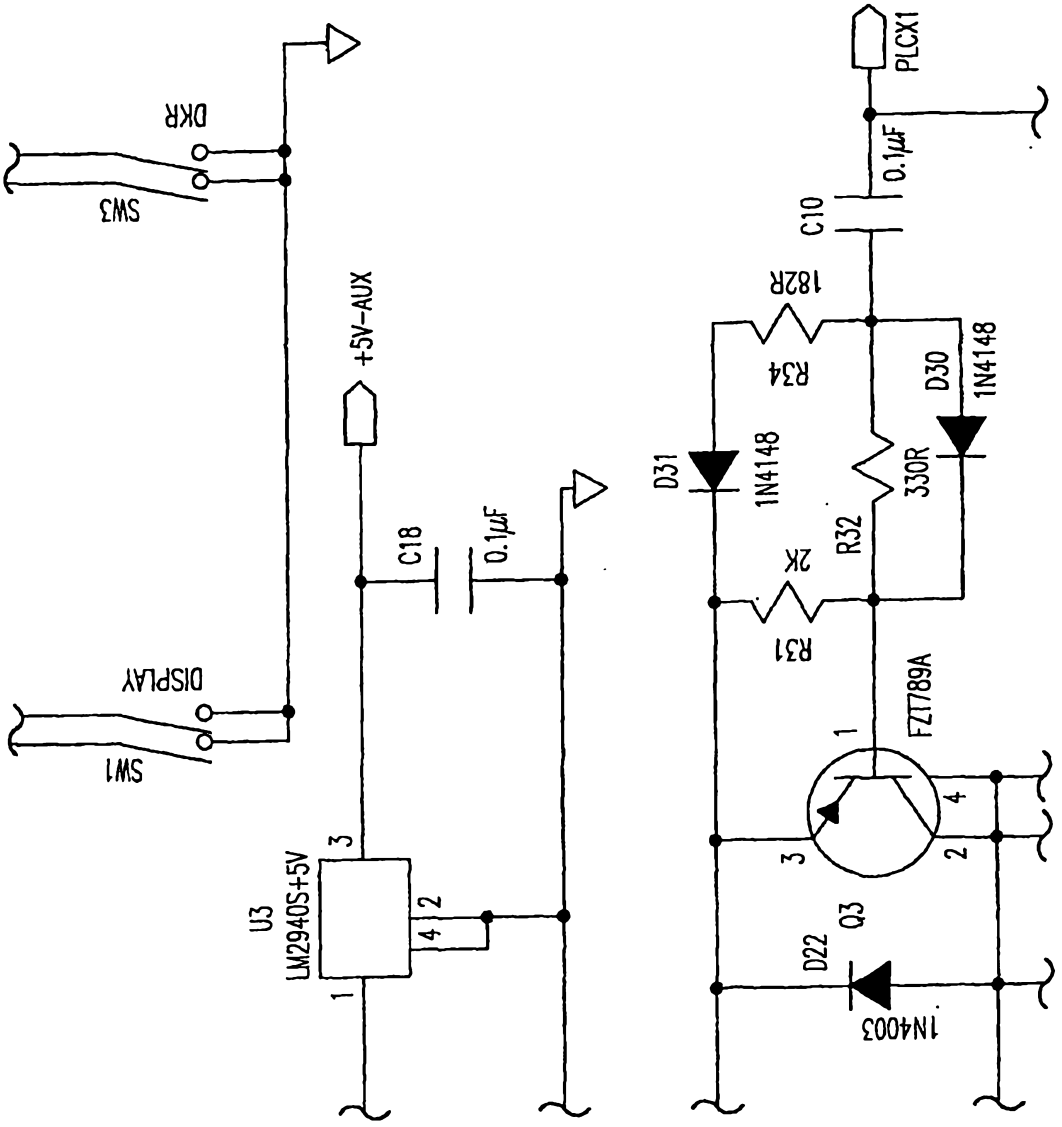

DKR
SW3
DISPLAY
SW1
+5V-AUX
C18
0.1µF
U3
LM2940S+5V
PLCX1
0.1µF
C10
182R
R34
D30
1N4148
D31
1N4148
330R
R32
2K
R31
FZT789A
Q3
D22
1N4003

FIG.5E

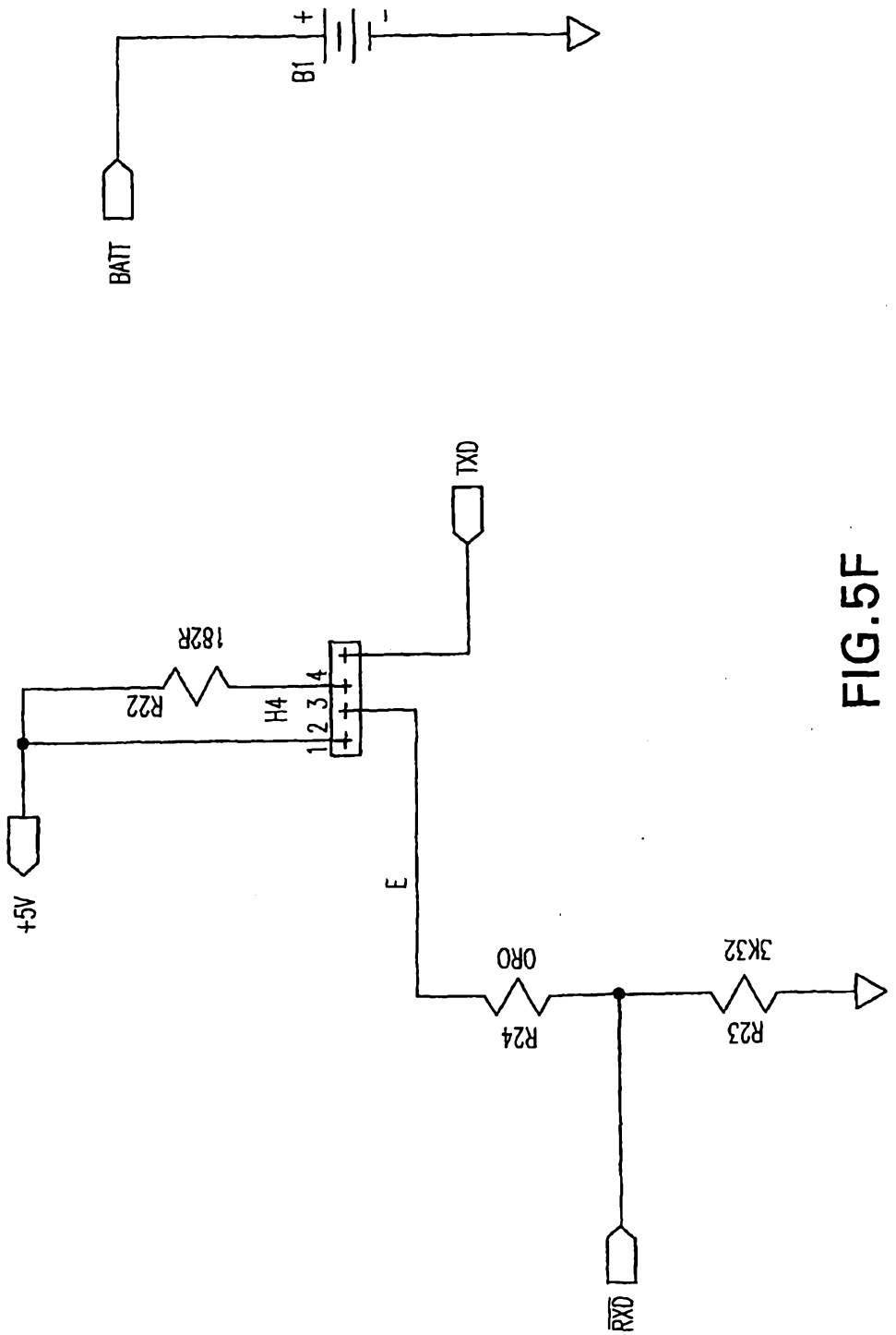


FIG.5F

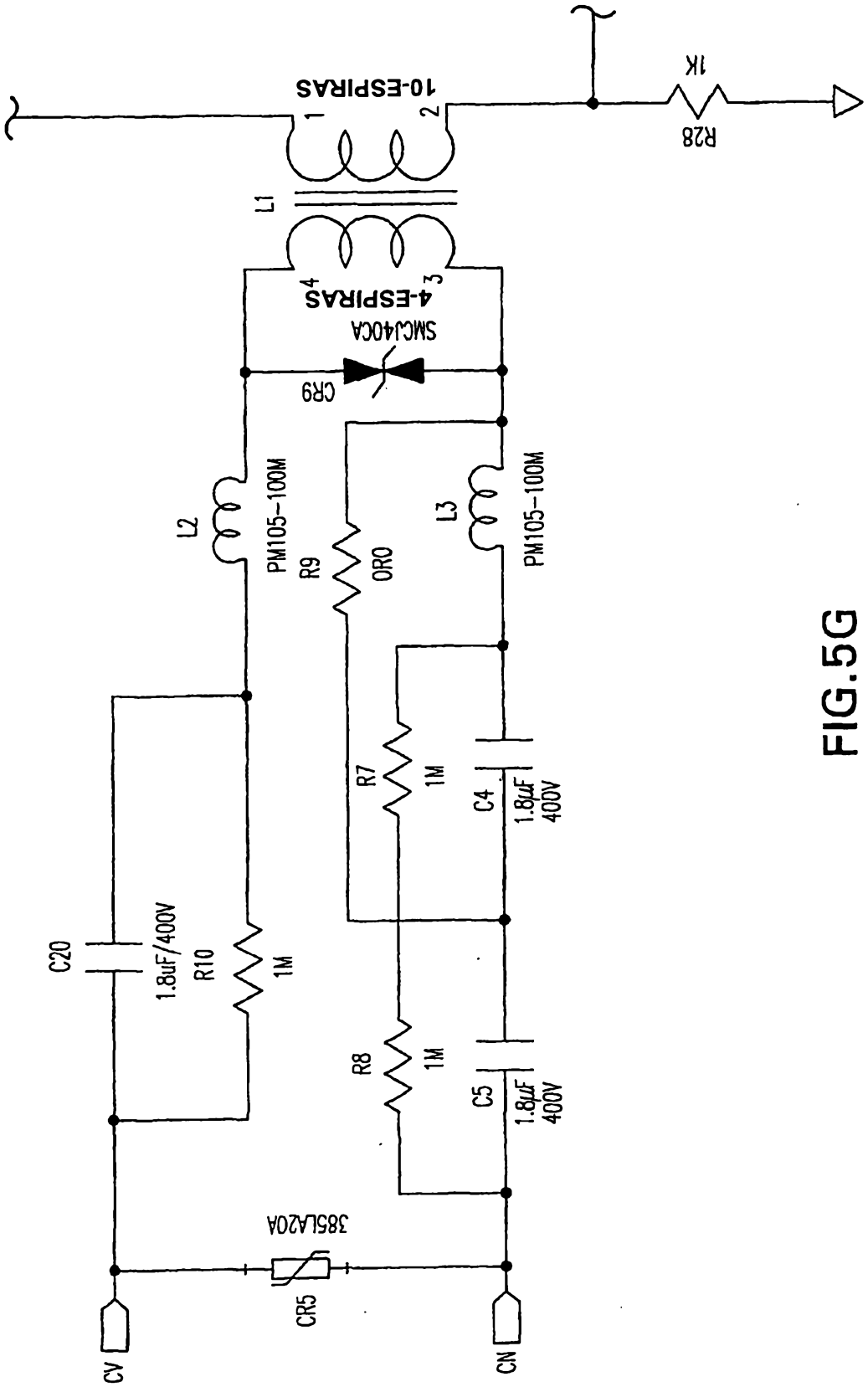
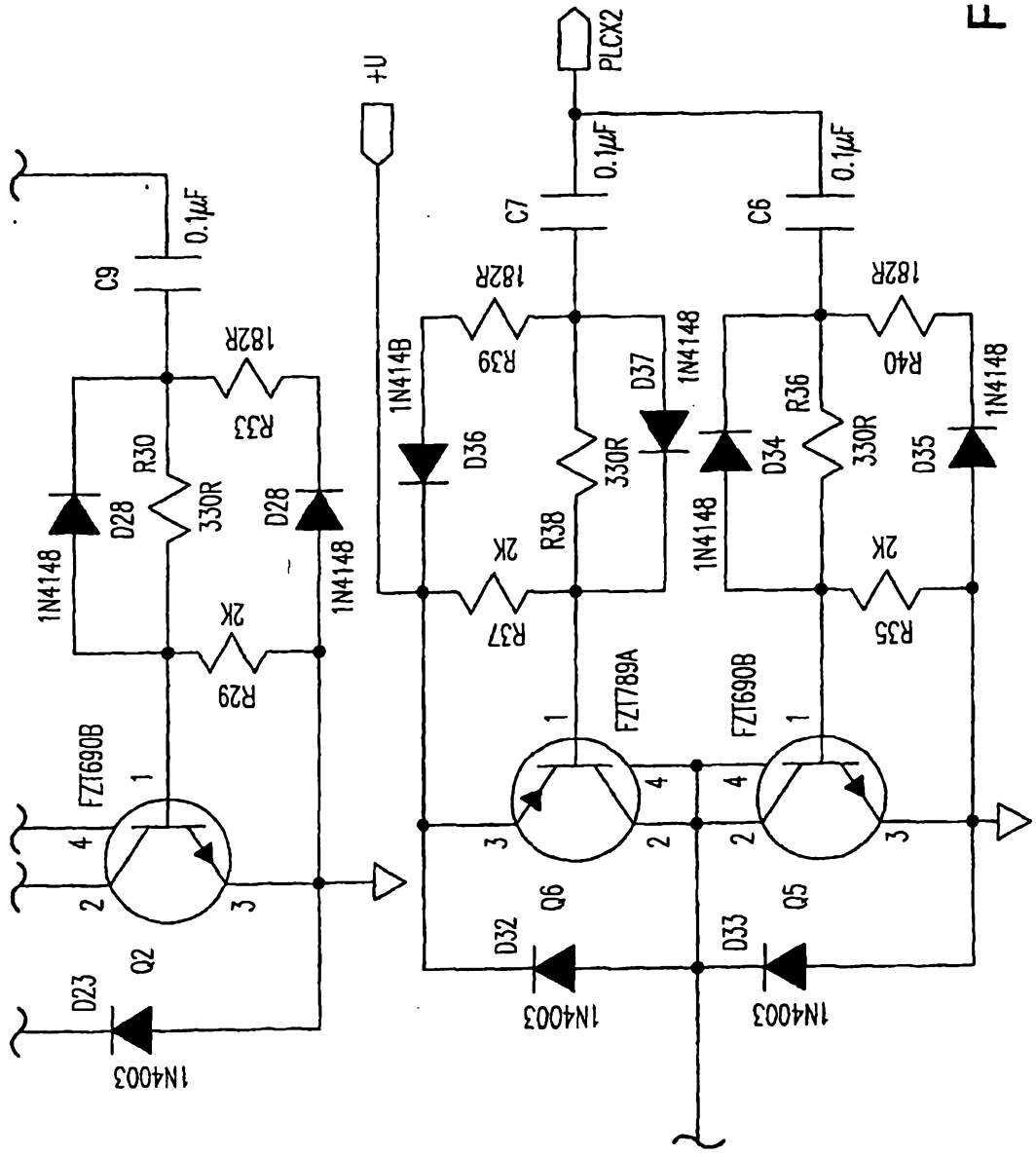

10-ESPIRAS
1
2
1K
R28
L1
4-ESPIRAS
4
3
SMCJ40CA
CR9
L2
PM105-100M
L3
PM105-100M
R9
0R0
R7
1M
C4
1.8µF
400V
C20
1.8uF/400V
R10
1M
R8
1M
C5
1.8µF
400V
385LA20A
CR5
CV
CN

FIG.5G



+U
PLCX2
C9
0.1μF
C7
0.1μF
C6
0.1μF
182R
R33
R30
330R
D28
D28
1N4148
1N4148
2K
R29
FZT690B
Q2
D23
1N4003
182R
R39
1N414B
D36
330R
R38
2K
R37
D37
1N4148
182R
R40
R36
330R
D34
D35
1N4148
1N4148
2K
R35
FZT789A
FZT690B
Q6
Q5
D32
D33
1N4003
1N4003

FIG.5H

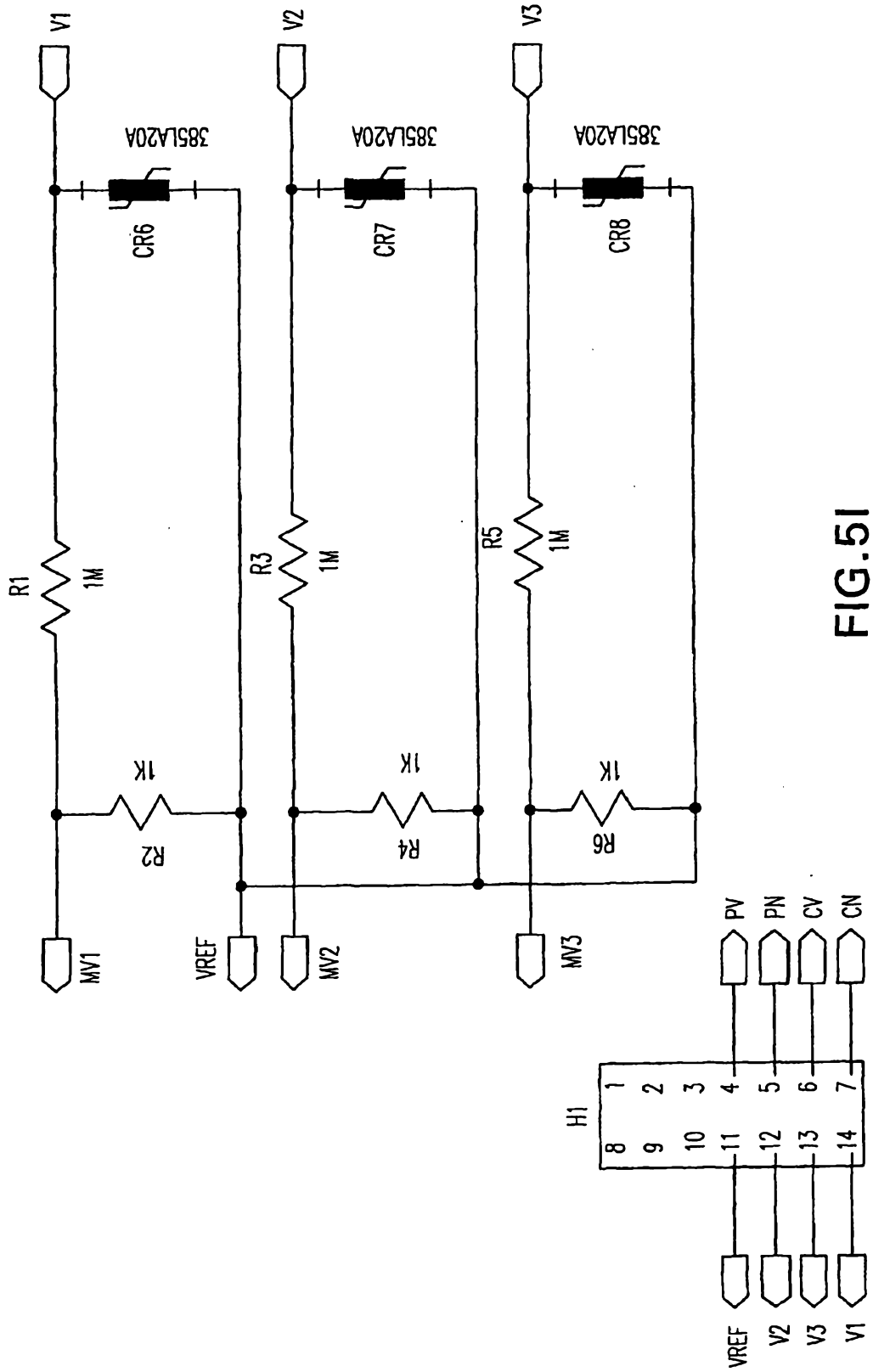

V1
V2
V3
3SLA20A
3SLA20A
3SLA20A
CR6
CR7
CR8
R1
1M
R3
1M
R5
1M
1K
1K
1K
R2
R4
R6
MV1
VREF
MV2
MV3
PV
PN
CV
CN
H1
1
2
3
4
5
6
7
8
9
10
11
12
13
14
VREF
V2
V3
V1

FIG.5I

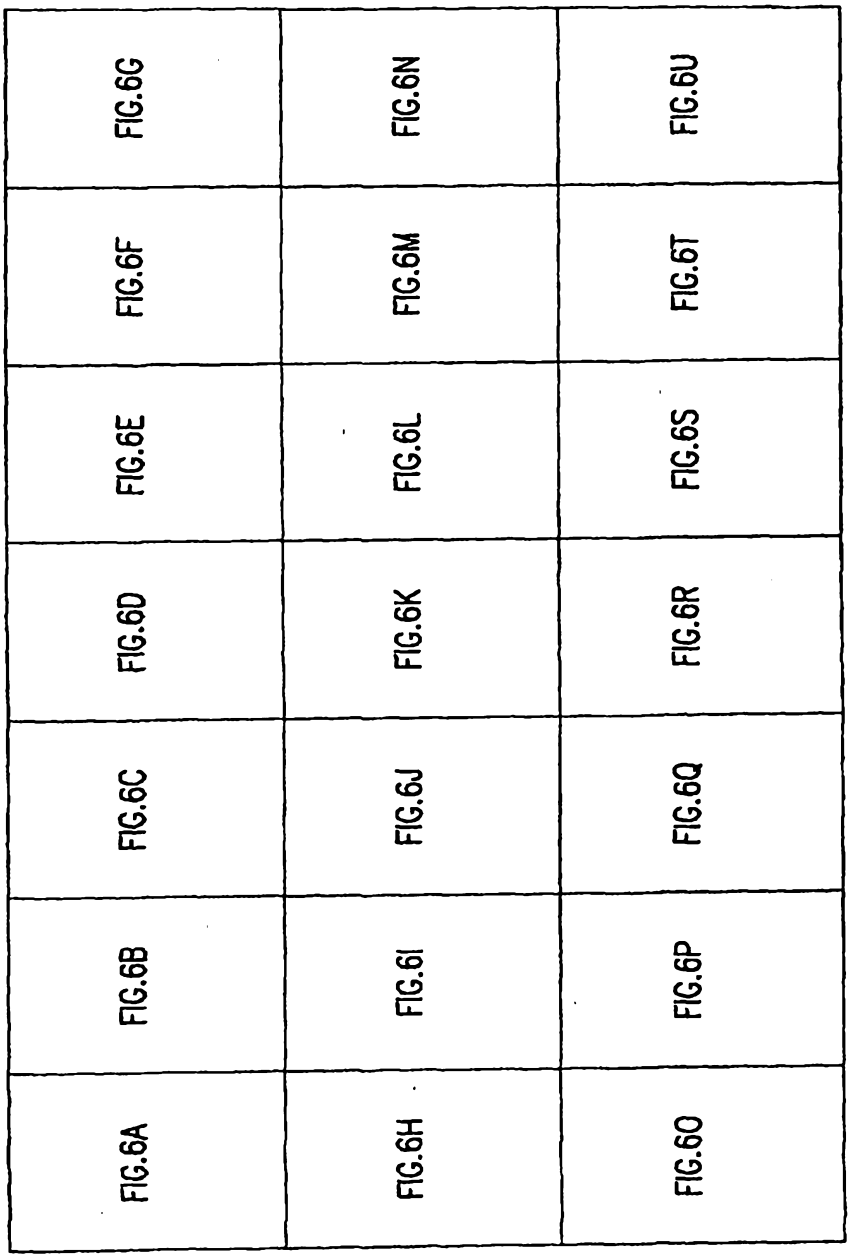


FIG.6A
FIG.6B
FIG.6C
FIG.6D
FIG.6E
FIG.6F
FIG.6G
FIG.6H
FIG.6I
FIG.6J
FIG.6K
FIG.6L
FIG.6M
FIG.6N
FIG.6O
FIG.6P
FIG.6Q
FIG.6R
FIG.6S
FIG.6T
FIG.6U

FIG.6

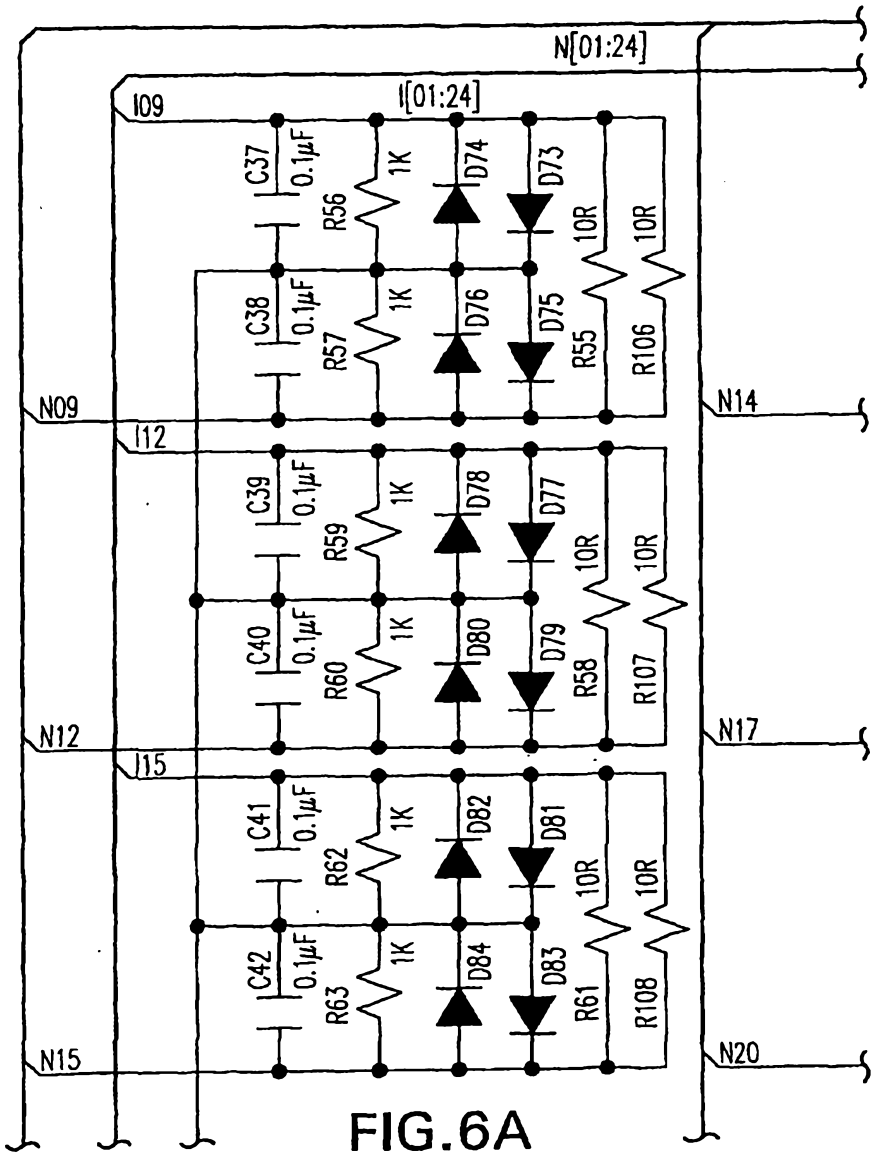

N[01:24]
I[01:24]
I09
I12
I15
N09
N12
N15
N14
N17
N20
C37 0.1µF
C38 0.1µF
C39 0.1µF
C40 0.1µF
C41 0.1µF
C42 0.1µF
R56 1K
R57 1K
R59 1K
R60 1K
R62 1K
R63 1K
D74
D73
D76
D75
D78
D77
D80
D79
D82
D81
D84
D83
R55 10R
R106
R58 10R
R107
R61 10R
R108

FIG.6A

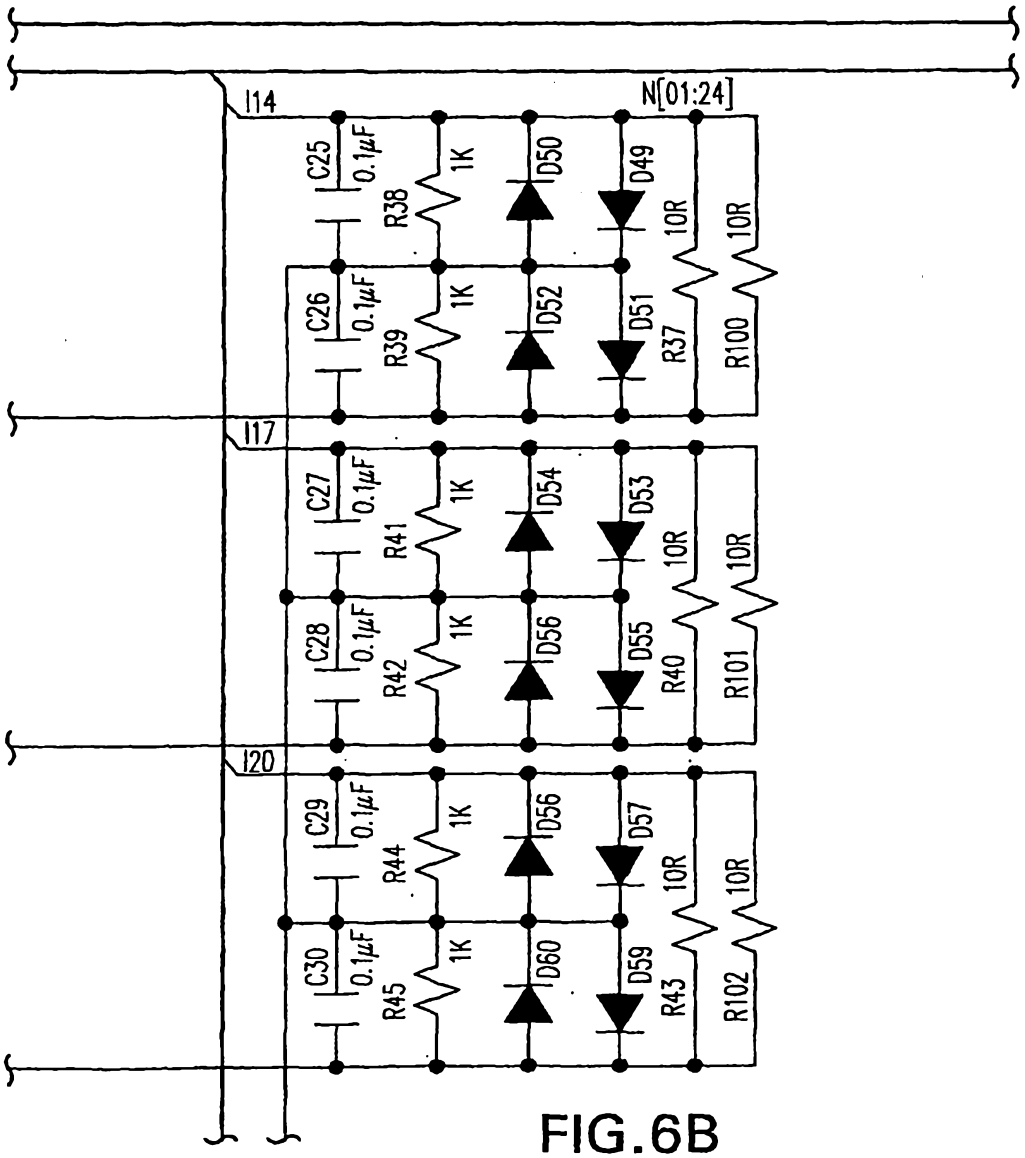

I14
N[01:24]
C25
0.1μF
R38
1K
D50
D49
10R
10R
C26
0.1μF
R39
1K
D52
D51
R37
R100
I17
C27
0.1μF
R41
1K
D54
D53
10R
10R
C28
0.1μF
R42
1K
D56
D55
R40
R101
I20
C29
0.1μF
R44
1K
D56
D57
10R
10R
C30
0.1μF
R45
1K
D60
D59
R43
R102

FIG.6B

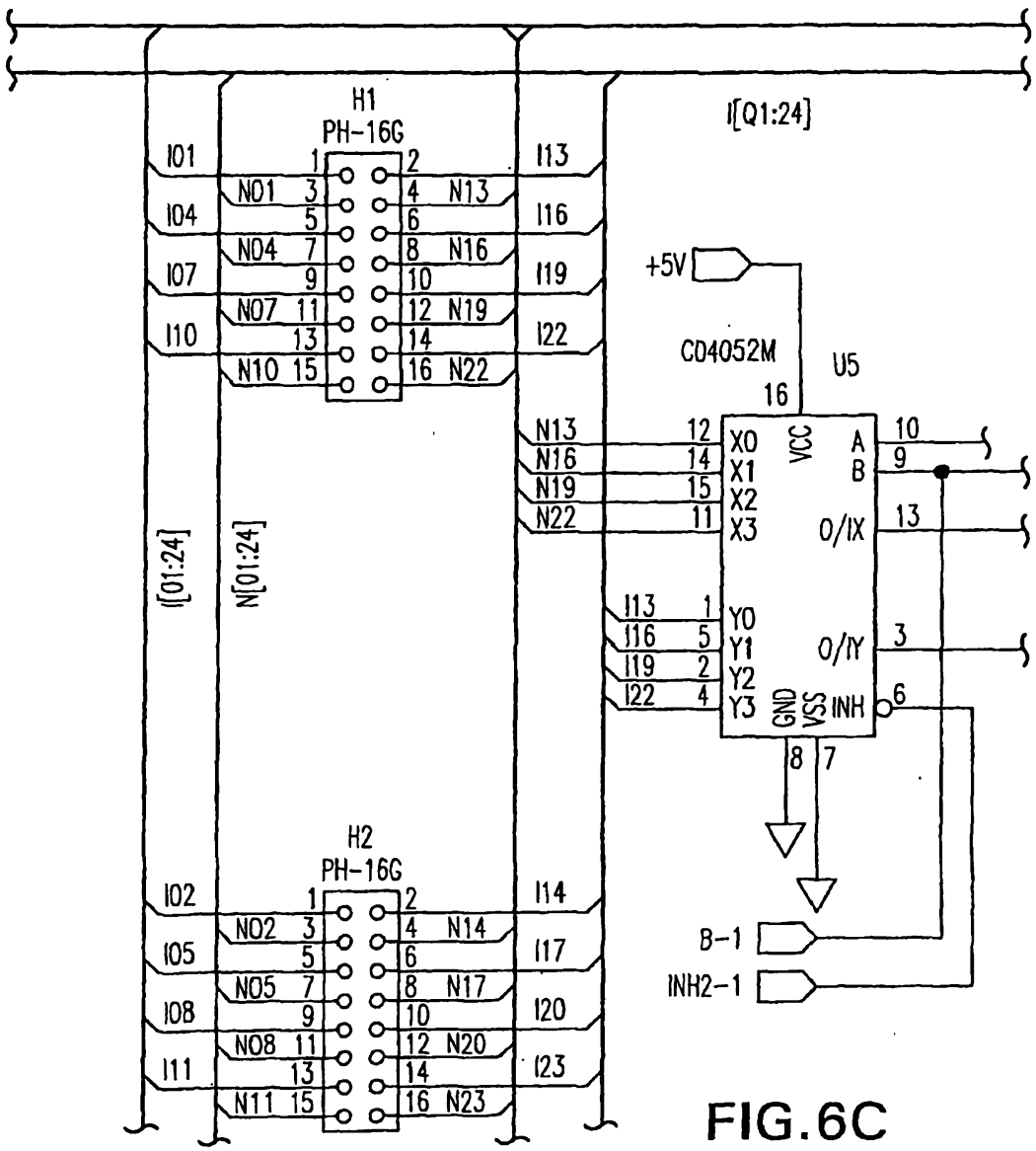

H1
PH-16G
I01
I04
I07
I10
NO1
NO4
NO7
N10
N13
N16
N19
N22
I13
I16
I19
I22
I[Q1:24]
+5V
CD4052M
U5
VCC
GND
VSS
X0
X1
X2
X3
Y0
Y1
Y2
Y3
A
B
O/IX
O/IY
INH
I[01:24]
N[01:24]
H2
PH-16G
I02
I05
I08
I11
NO2
NO5
NO8
N11
N14
N17
N20
N23
I14
I17
I20
I23
B-1
INH2-1

FIG.6C

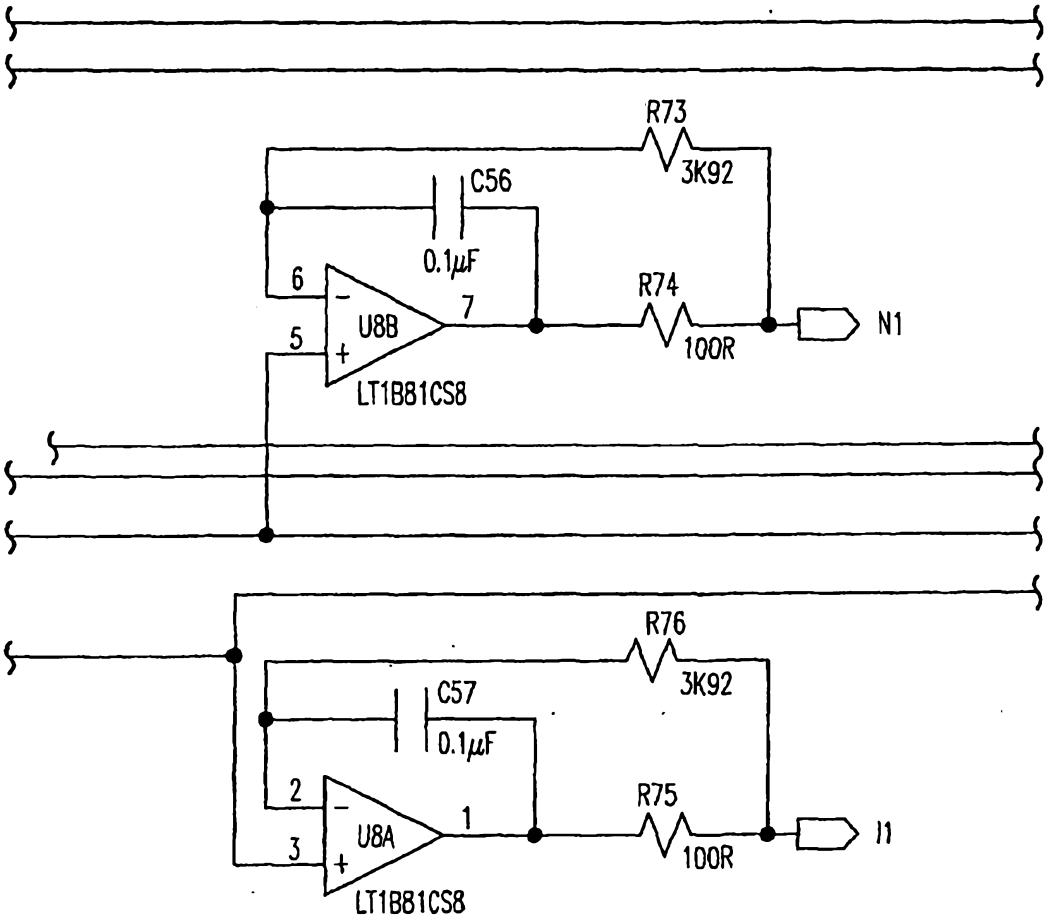

R73
3K92
C56
0.1μF
6
5
U8B
7
LT1B81CS8
R74
100R
N1
R76
3K92
C57
0.1μF
2
3
U8A
1
LT1B81CS8
R75
100R
I1

FIG.6D

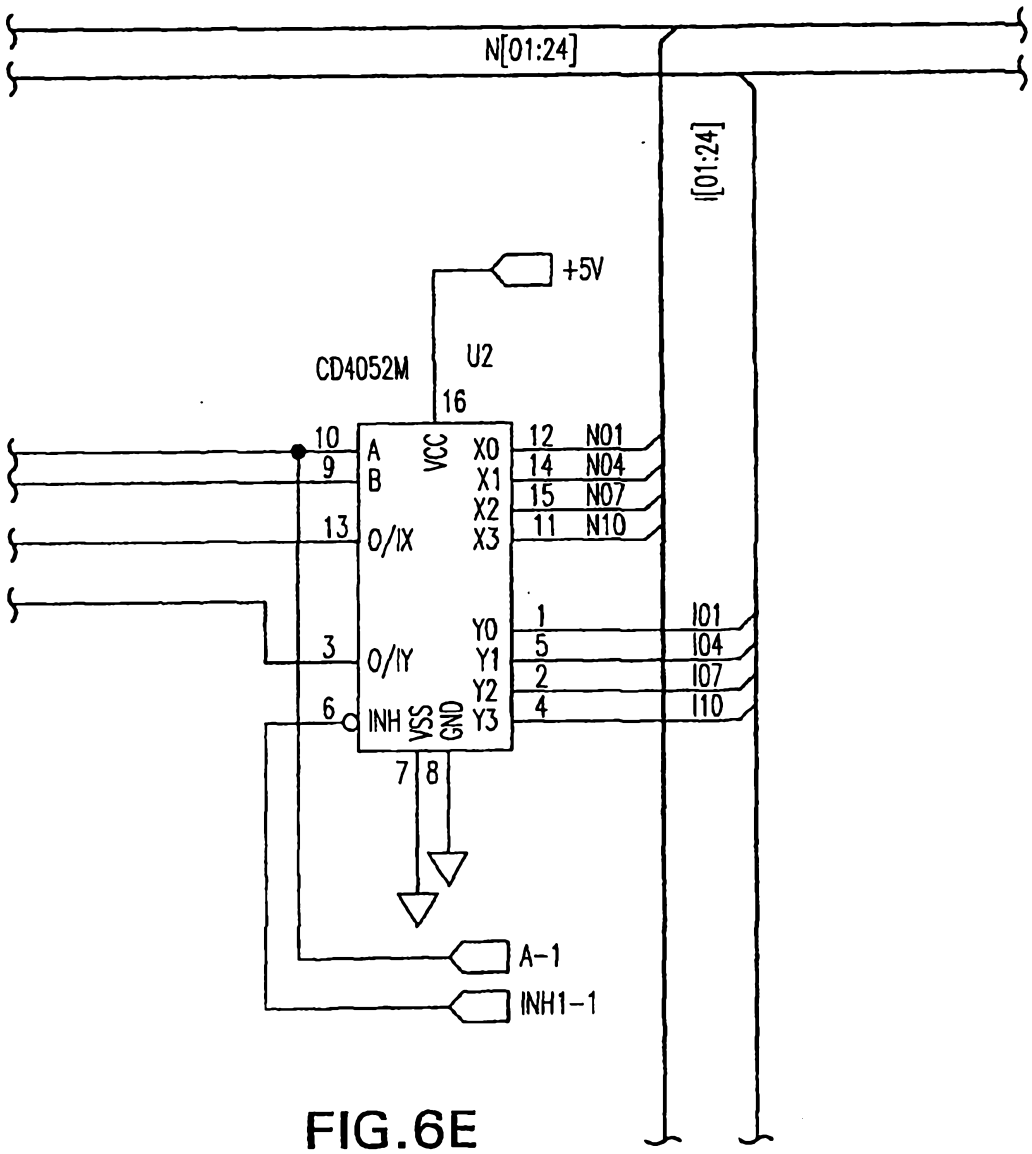

N[01:24]
I[01:24]
+5V
CD4052M
U2
16
10
9
13
3
6
7
8
A
B
O/IX
O/IY
INH
VCC
VSS
GND
X0
X1
X2
X3
Y0
Y1
Y2
Y3
12
14
15
11
1
5
2
4
N01
N04
N07
N10
I01
I04
I07
I10
A-1
INH1-1

FIG.6E

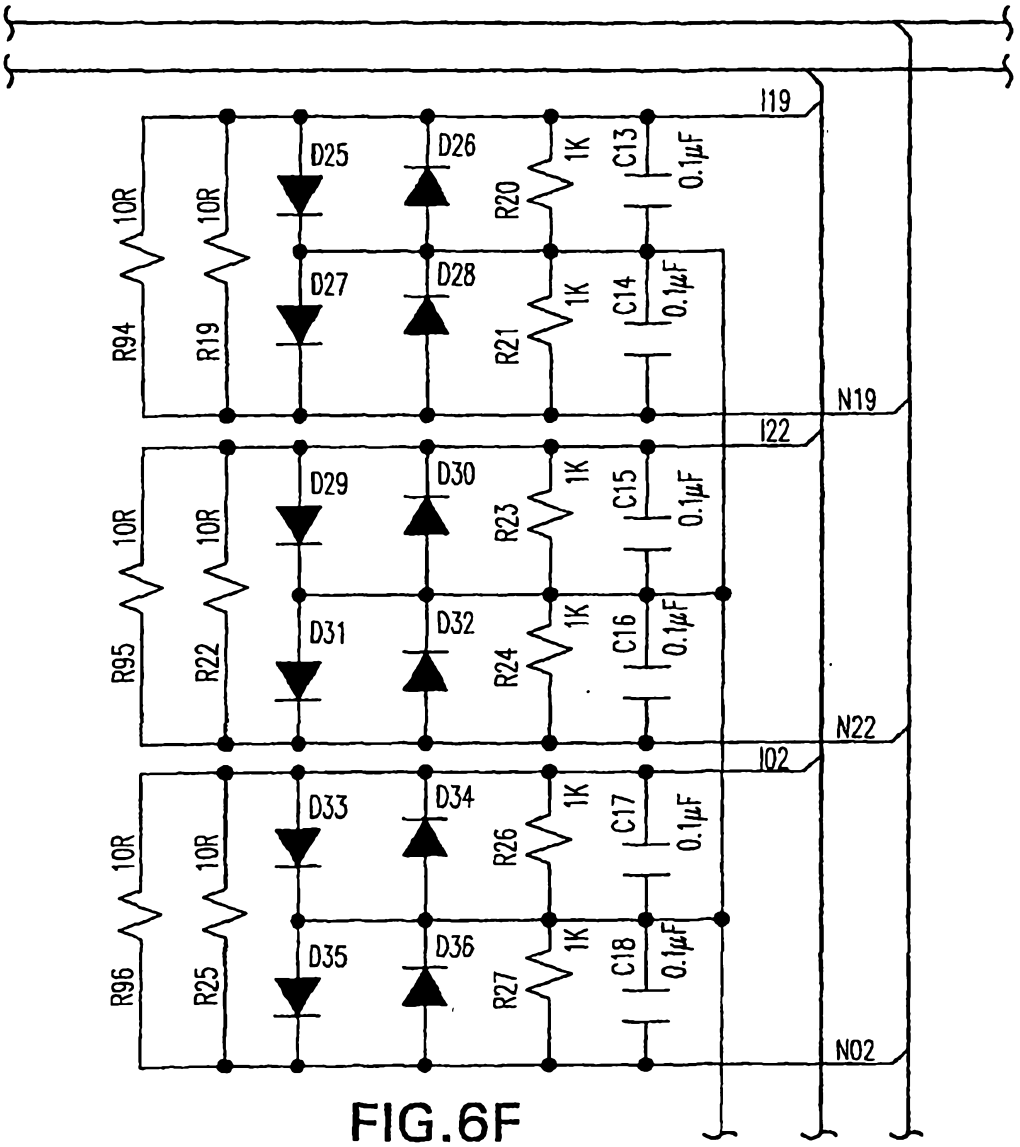

I19
D25
D26
R20
1K
C13
0.1μF
10R
10R
R94
R19
D27
D28
R21
1K
C14
0.1μF
N19
I22
D29
D30
R23
1K
C15
0.1μF
10R
10R
R95
R22
D31
D32
R24
1K
C16
0.1μF
N22
I02
D33
D34
R26
1K
C17
0.1μF
10R
10R
R96
R25
D35
D36
R27
1K
C18
0.1μF
N02

FIG.6F

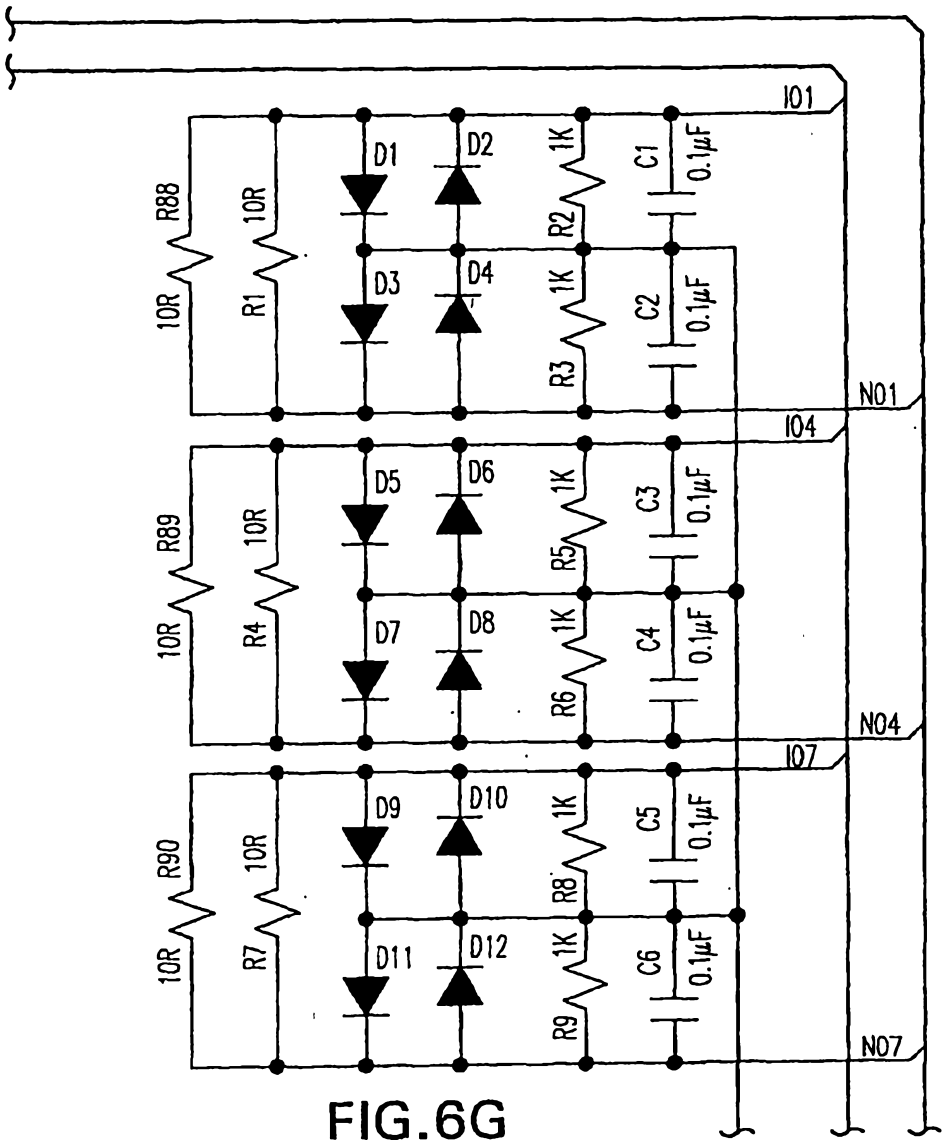

I01
D1
D2
1K
C1
0.1μF
R2
R88
10R
10R
R1
D3
D4
1K
C2
0.1μF
R3
N01
I04
D5
D6
1K
C3
0.1μF
R5
R89
10R
10R
R4
D7
D8
1K
C4
0.1μF
R6
N04
I07
D9
D10
1K
C5
0.1μF
R8
R90
10R
10R
R7
D11
D12
1K
C6
0.1μF
R9
N07

FIG.6G

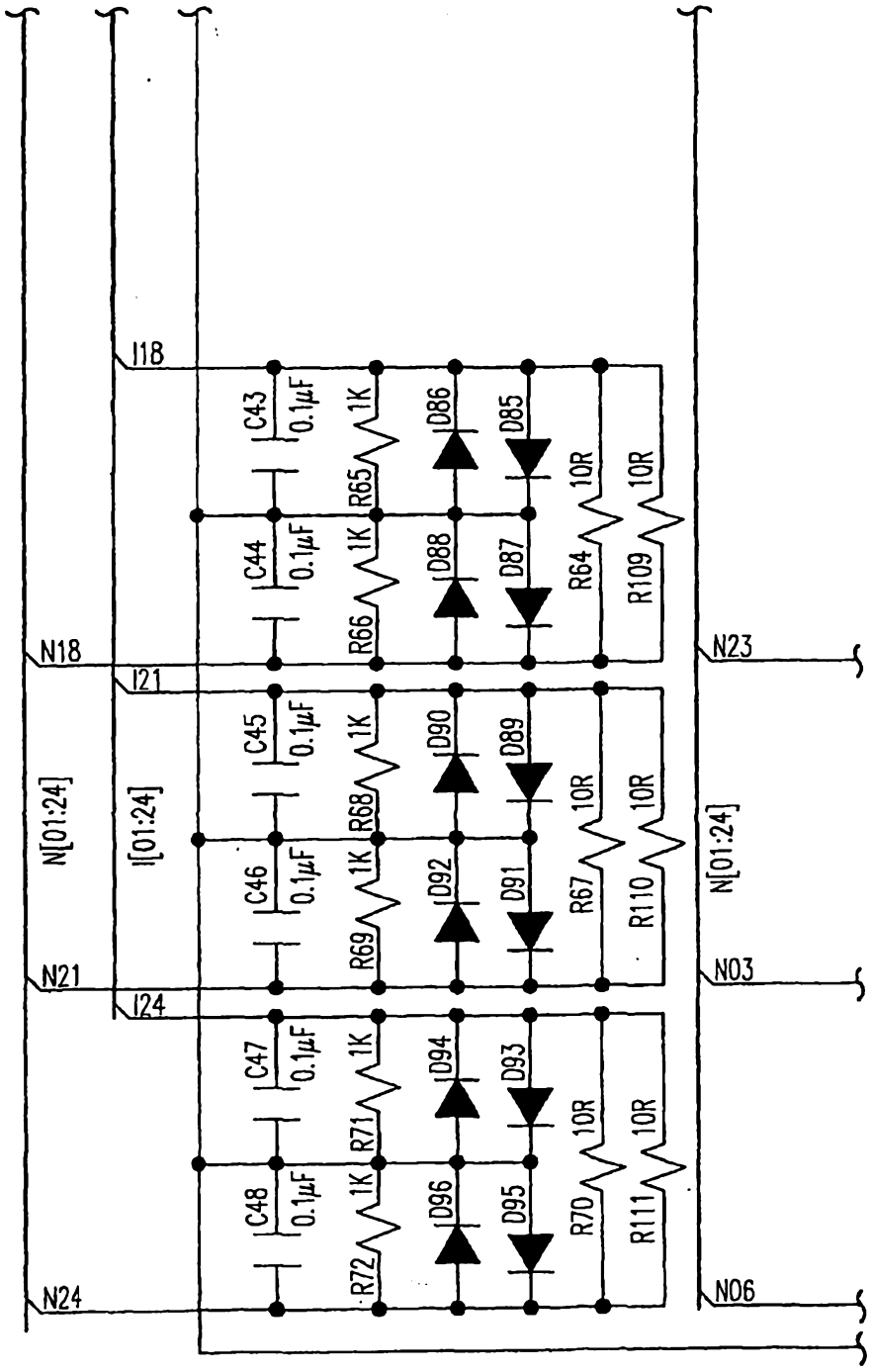

I18
I21
I24
N18
N21
N24
N23
N03
N06
N[01:24]
I[01:24]
N[01:24]
C43
C44
C45
C46
C47
C48
0.1µF
R65
R66
R68
R69
R71
R72
1K
D86
D85
D88
D87
D90
D89
D92
D91
D94
D93
D96
D95
R64
R109
R67
R110
R70
R111
10R

FIG.6H

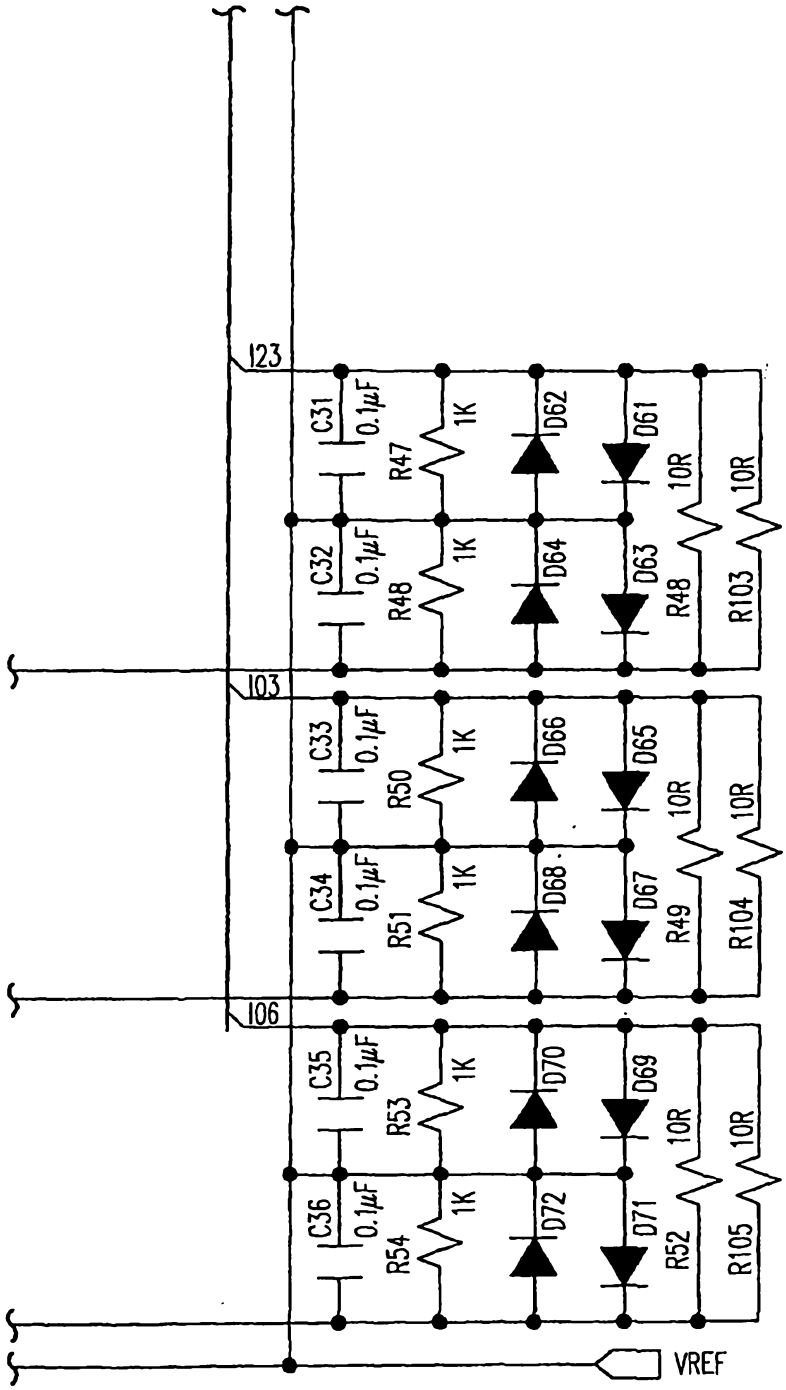

123
103
106
C31
0.1μF
C32
0.1μF
C33
0.1μF
C34
0.1μF
C35
0.1μF
C36
0.1μF
R47
1K
R48
1K
R50
1K
R51
1K
R53
1K
R54
1K
D62
D64
D66
D68
D70
D72
D61
D63
D65
D67
D69
D71
10R
R48
10R
R49
10R
R52
R103
R104
R105
VREF

FIG.6I

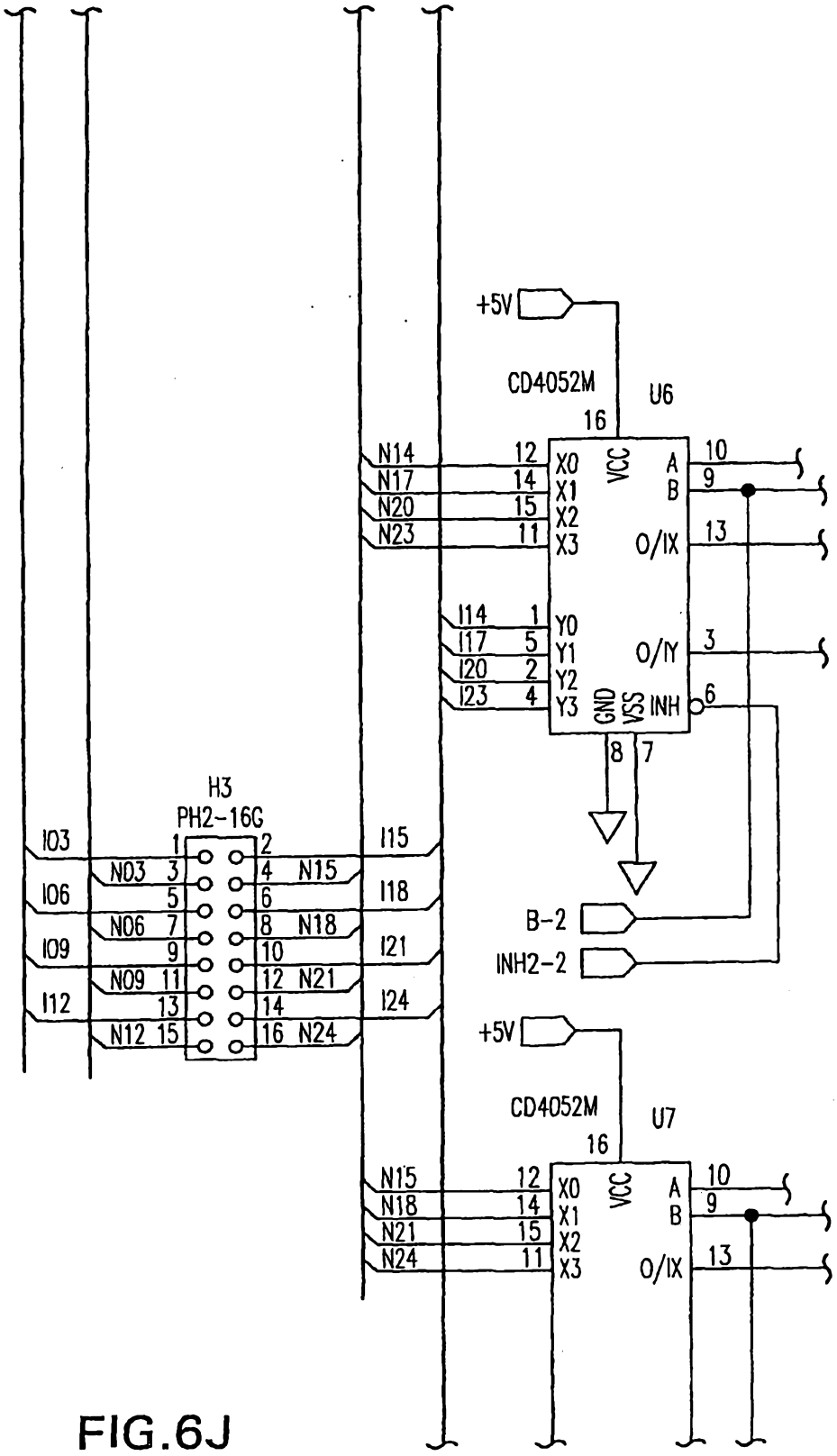

+5V
CD4052M
U6
16
N14
N17
N20
N23
12
14
15
11
X0
X1
X2
X3
VCC
A
B
O/IX
10
9
13
I14
I17
I20
I23
1
5
2
4
Y0
Y1
Y2
Y3
O/IY
3
GND
VSS
INH
6
8
7
H3
PH2-16G
I03
I06
I09
I12
I15
I18
I21
I24
N03
N06
N09
N12
N15
N18
N21
N24
B-2
INH2-2
+5V
CD4052M
U7
16
N15
N18
N21
N24
12
14
15
11
X0
X1
X2
X3
VCC
A
B
O/IX
10
9
13

FIG.6J

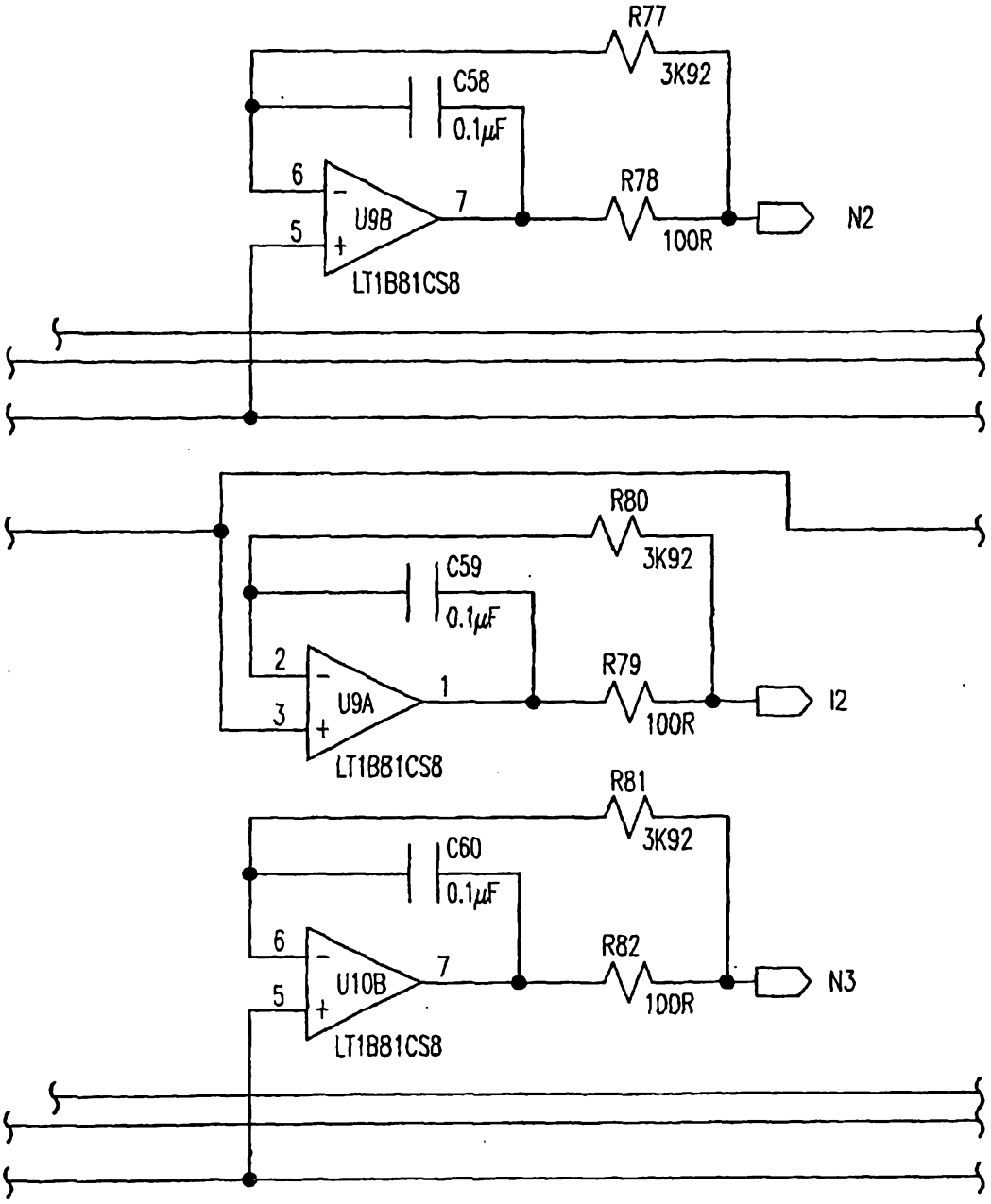

R77
3K92
C58
0.1µF
6
5
U9B
7
R78
100R
N2
LT1B81CS8
R80
3K92
C59
0.1µF
2
3
U9A
1
R79
100R
I2
LT1B81CS8
R81
3K92
C60
0.1µF
6
5
U10B
7
R82
100R
N3
LT1B81CS8

FIG.6K

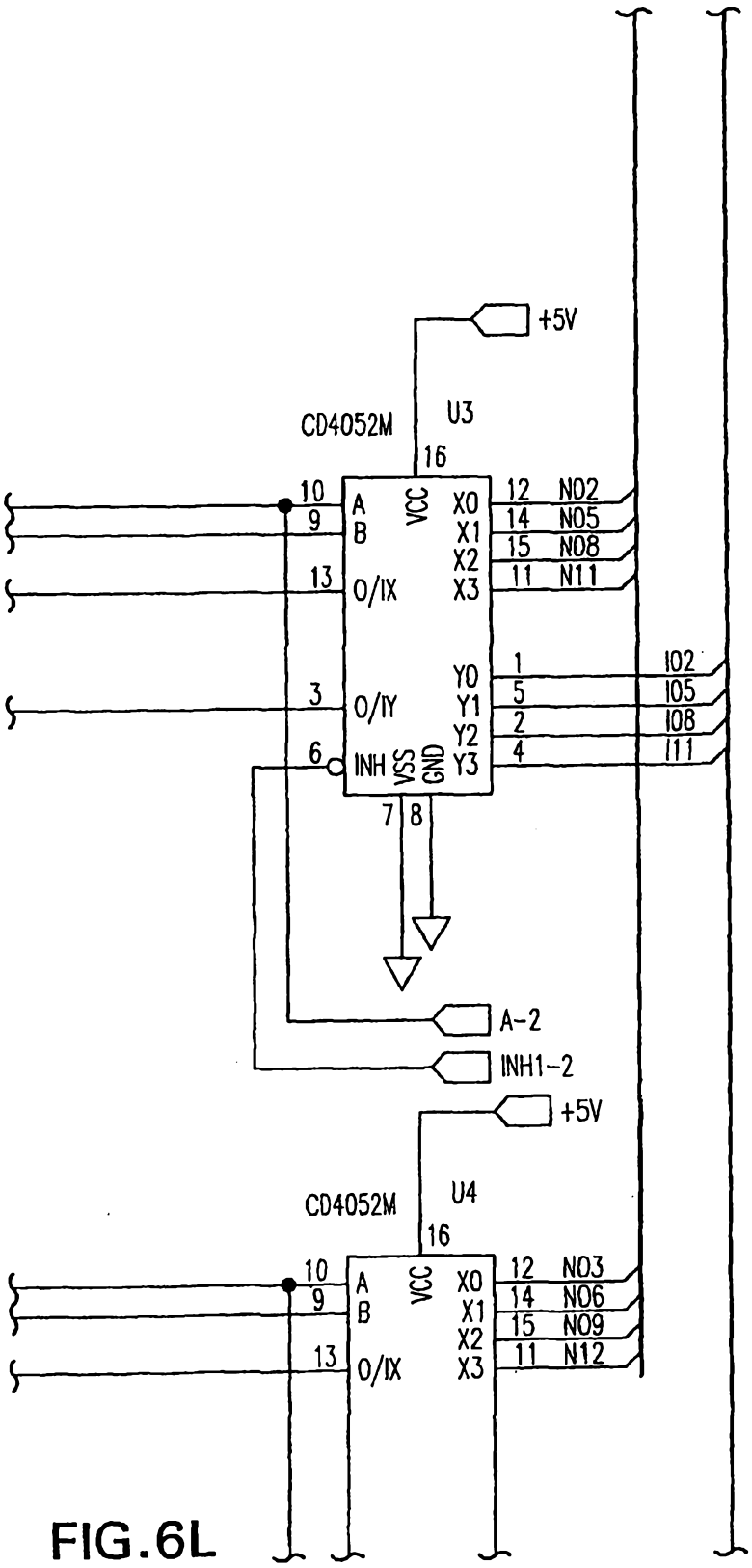

+5V
CD4052M
U3
16
10
9
13
3
6
7
8
A
B
O/IX
O/IY
INH
VCC
VSS
GND
X0
X1
X2
X3
Y0
Y1
Y2
Y3
12
14
15
11
1
5
2
4
NO2
NO5
NO8
N11
IO2
IO5
IO8
I11
A-2
INH1-2
+5V
CD4052M
U4
16
10
9
13
A
B
O/IX
VCC
X0
X1
X2
X3
12
14
15
11
NO3
NO6
NO9
N12

FIG.6L

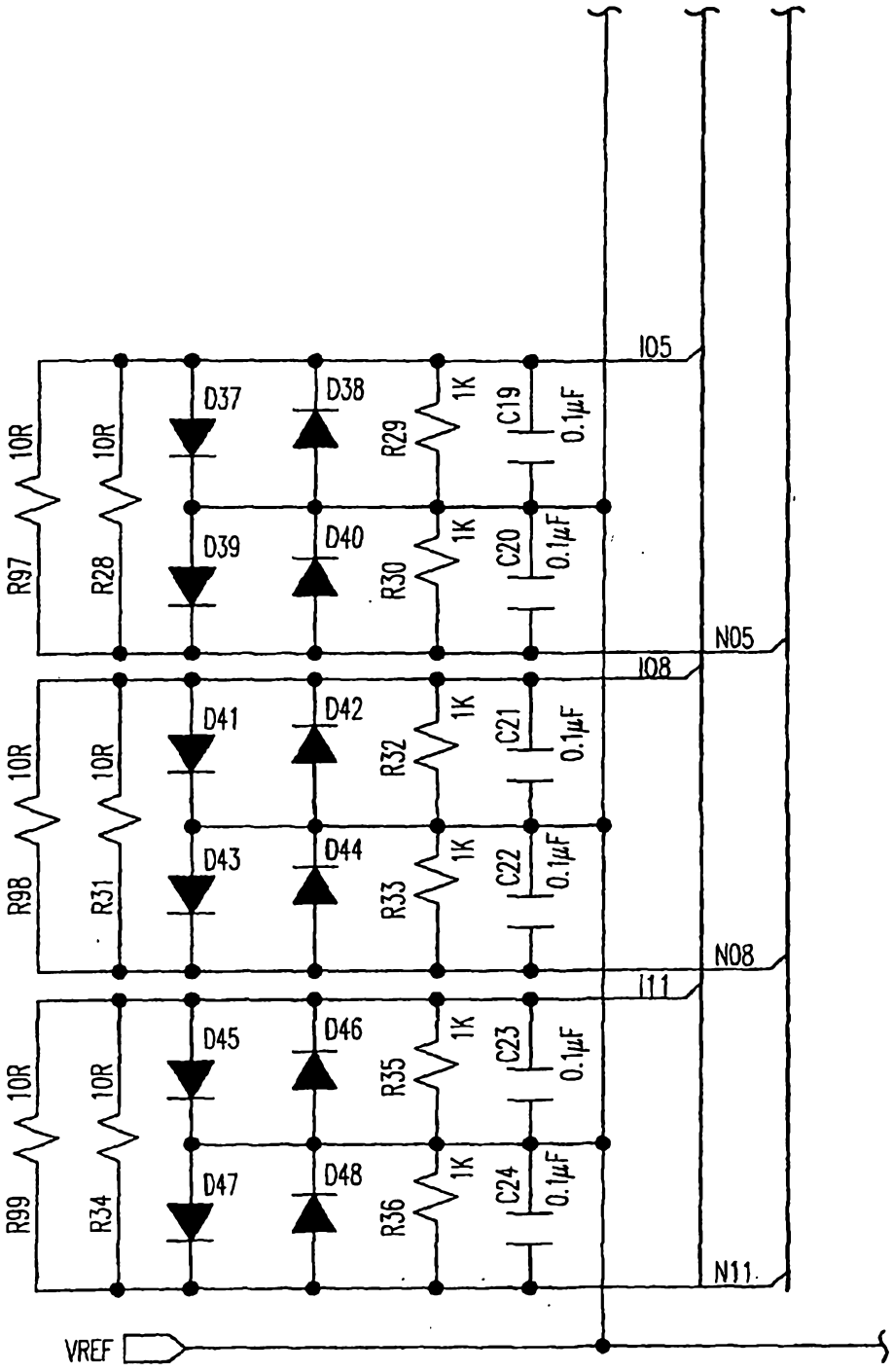

I05
D37
D38
R29
1K
C19
0.1μF
10R
10R
R97
R28
D39
D40
R30
1K
C20
0.1μF
N05
I08
D41
D42
R32
1K
C21
0.1μF
10R
10R
R98
R31
D43
D44
R33
1K
C22
0.1μF
N08
I11
D45
D46
R35
1K
C23
0.1μF
10R
10R
R99
R34
D47
D48
R36
1K
C24
0.1μF
N11
VREF

FIG.6M

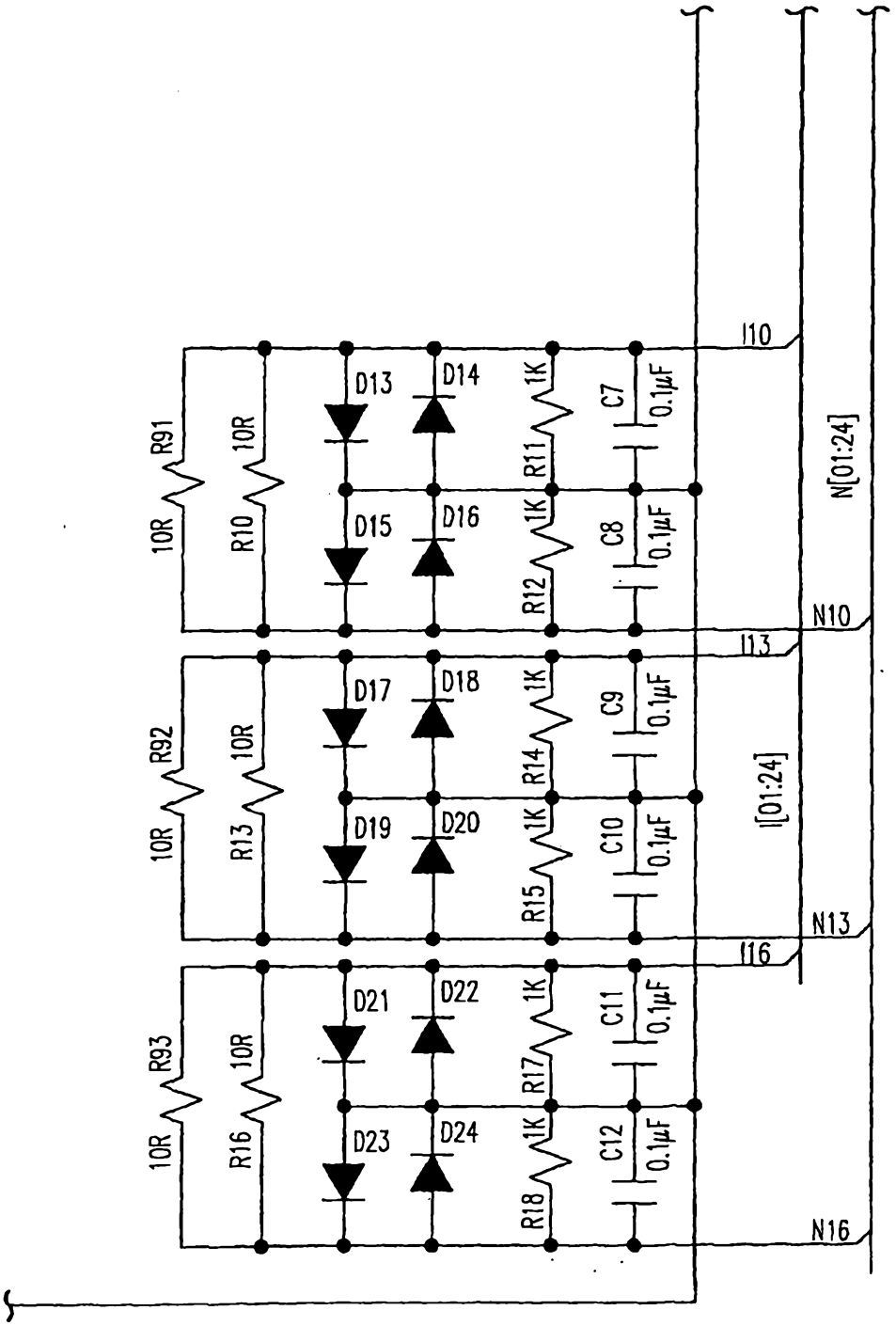

I10
N[01:24]
D13
D14
1K
C7
0.1µF
R91
10R
10R
R10
R11
D15
D16
1K
C8
0.1µF
R12
N10
I13
I[01:24]
D17
D18
1K
C9
0.1µF
R92
10R
10R
R13
R14
D19
D20
1K
C10
0.1µF
R15
N13
I16
D21
D22
1K
C11
0.1µF
R93
10R
10R
R16
R17
D23
D24
1K
C12
0.1µF
R18
N16

FIG.6N

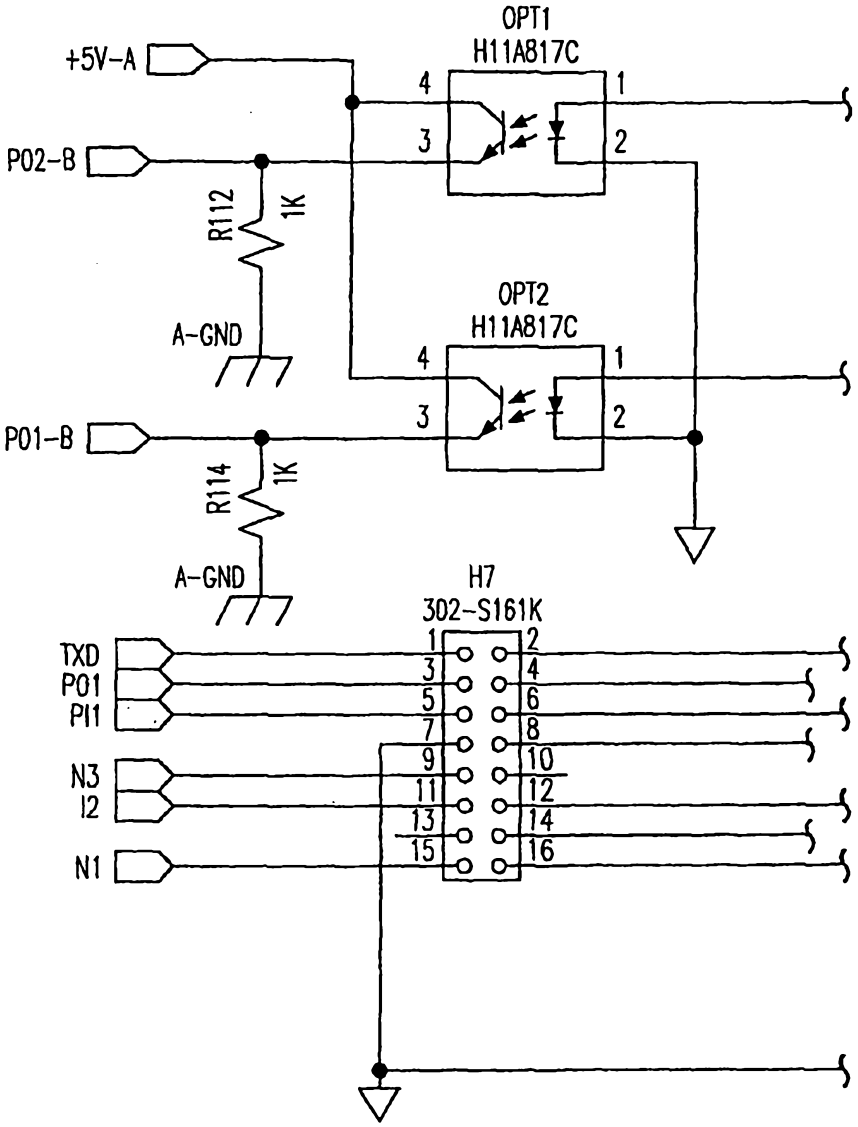

OPT1
H11A817C
+5V-A
P02-B
R112
1K
A-GND
OPT2
H11A817C
P01-B
R114
1K
A-GND
H7
302-S161K
TXD
P01
PI1
N3
I2
N1

FIG.60

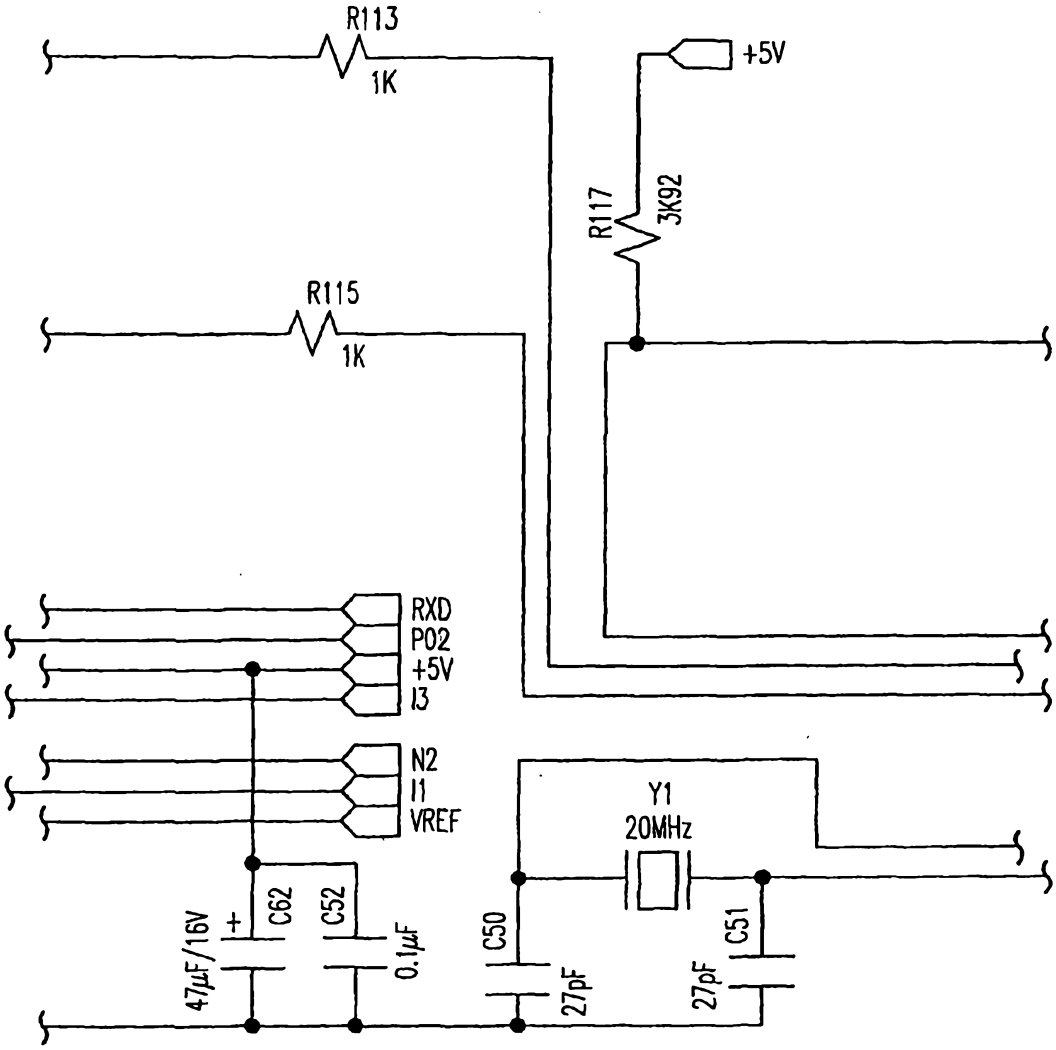

R113
1K
+5V
R117
3K92
R115
1K
RXD
P02
+5V
I3
N2
I1
VREF
Y1
20MHz
C62
47μF/16V
C52
0.1μF
C50
27pF
C51
27pF

FIG.6P

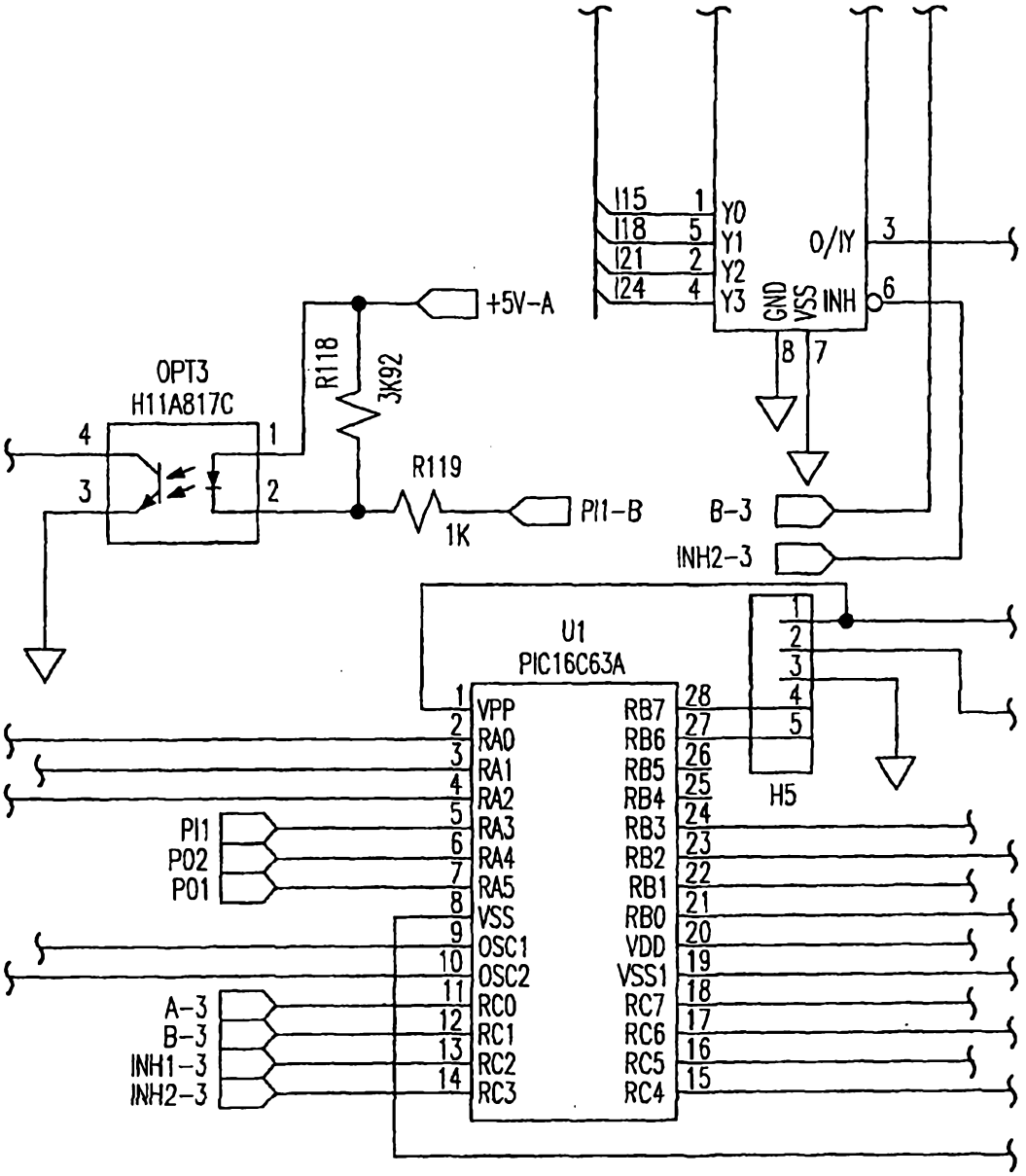

I15
I18
I21
I24
Y0
Y1
Y2
Y3
O/IY
GND
VSS
INH
+5V-A
R118
3K92
OPT3
H11A817C
R119
1K
PI1-B
B-3
INH2-3
U1
PIC16C63A
VPP
RA0
RA1
RA2
RA3
RA4
RA5
VSS
OSC1
OSC2
RC0
RC1
RC2
RC3
RB7
RB6
RB5
RB4
RB3
RB2
RB1
RB0
VDD
VSS1
RC7
RC6
RC5
RC4
H5
PI1
PO2
PO1
A-3
B-3
INH1-3
INH2-3

FIG.6Q

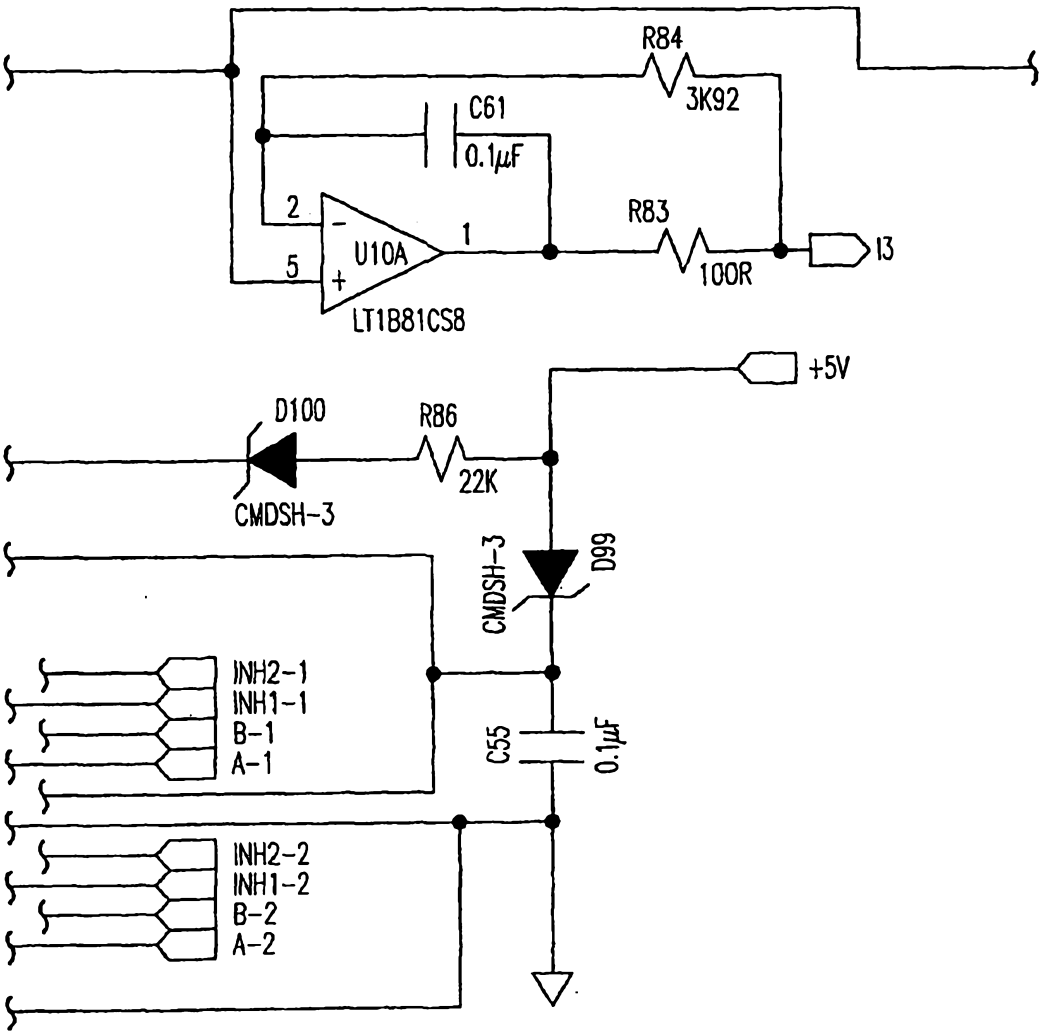

R84
3K92
C61
0.1μF
2
5
U10A
LT1B81CS8
1
R83
100R
I3
+5V
D100
CMDSH-3
R86
22K
CMDSH-3
D99
INH2-1
INH1-1
B-1
A-1
C55
0.1μF
INH2-2
INH1-2
B-2
A-2

FIG.6R

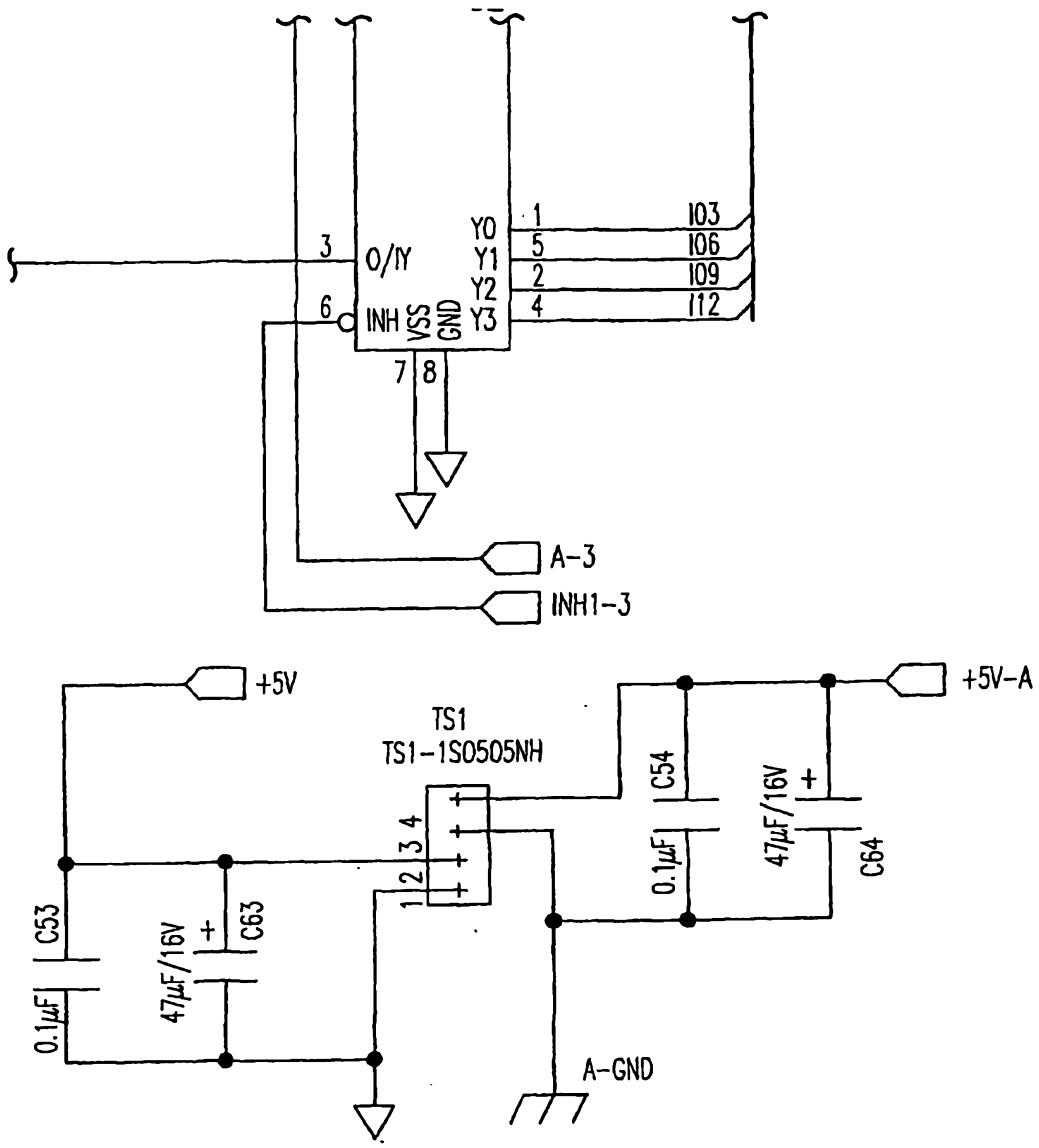

3
O/IY
6
INH
VSS
GND
7
8
Y0
Y1
Y2
Y3
1
5
2
4
103
106
109
112
A-3
INH1-3
+5V
TS1
TS1-1S0505NH
1
2
3
4
C53
0.1μF
47μF/16V
C63
C54
0.1μF
47μF/16V
C64
+5V-A
A-GND

FIG.6S

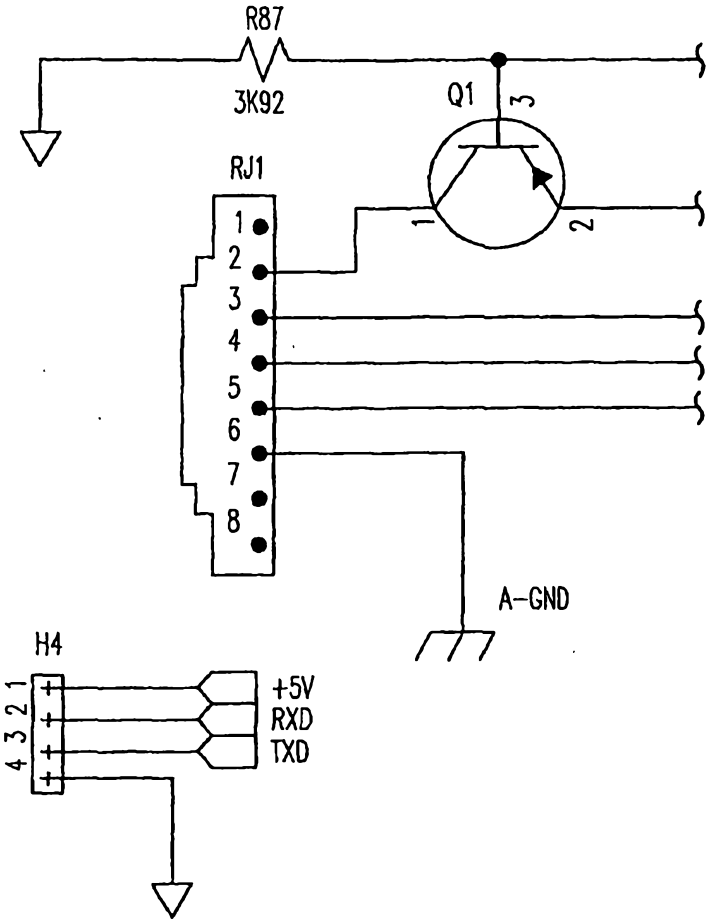


FIG.6T

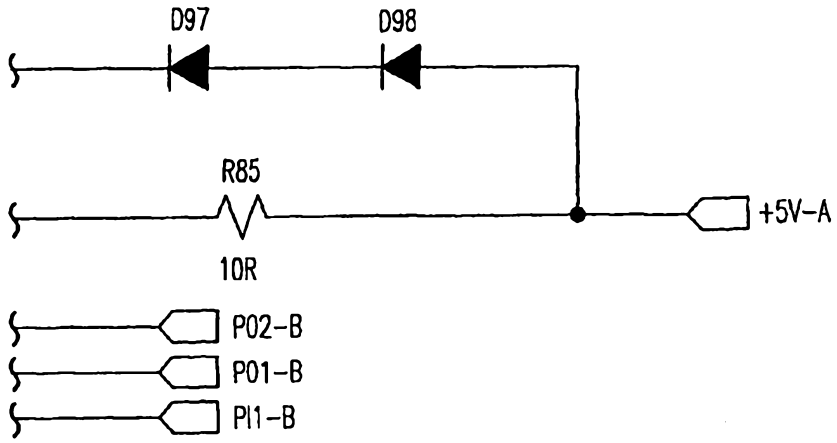


FIG.6U

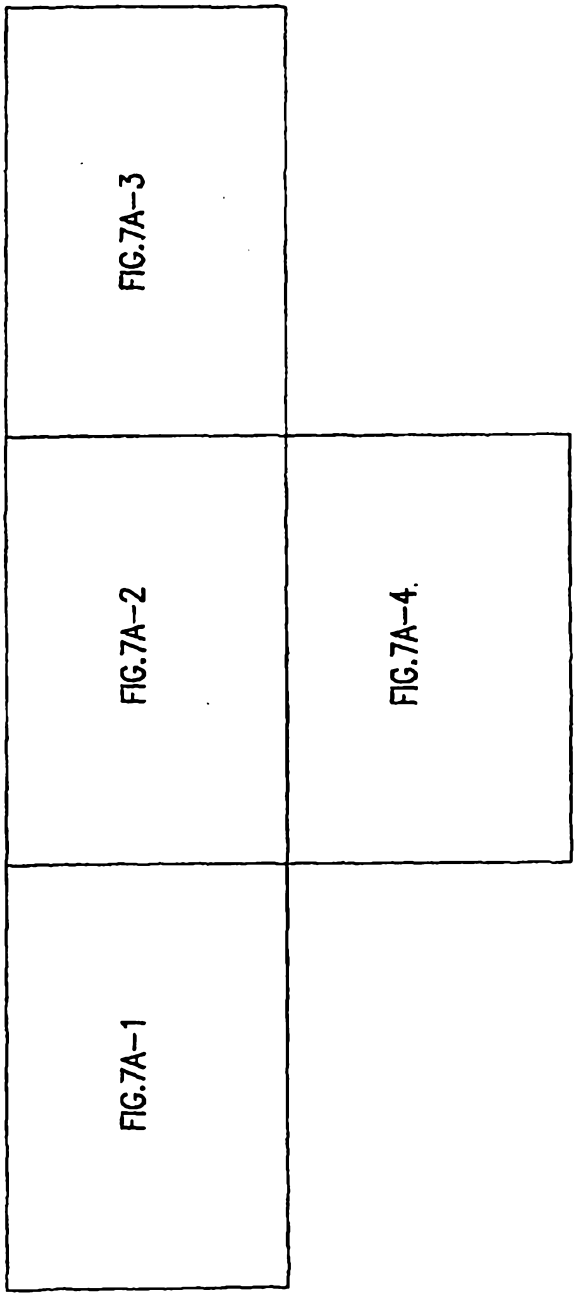


FIG.7A

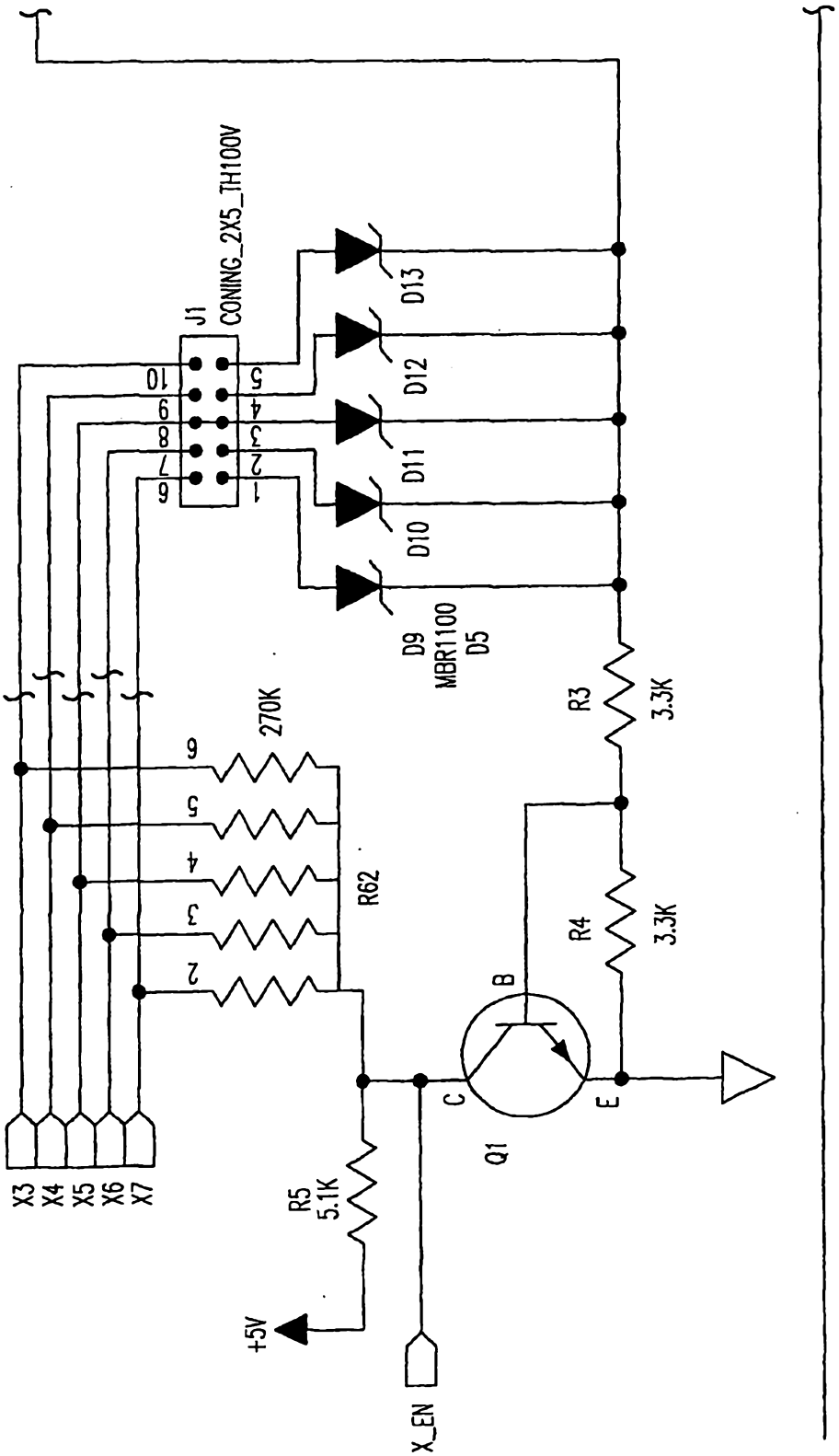

J1
CONNING_2X5_TH100V
10
9
8
7
6
5
4
3
2
1
D13
D12
D11
D10
D9
MBR1100
D5
R3
3.3K
270K
6
5
4
3
2
R62
R4
3.3K
B
C
E
Q1
X3
X4
X5
X6
X7
R5
5.1K
+5V
X_EN

FIG.7A-1

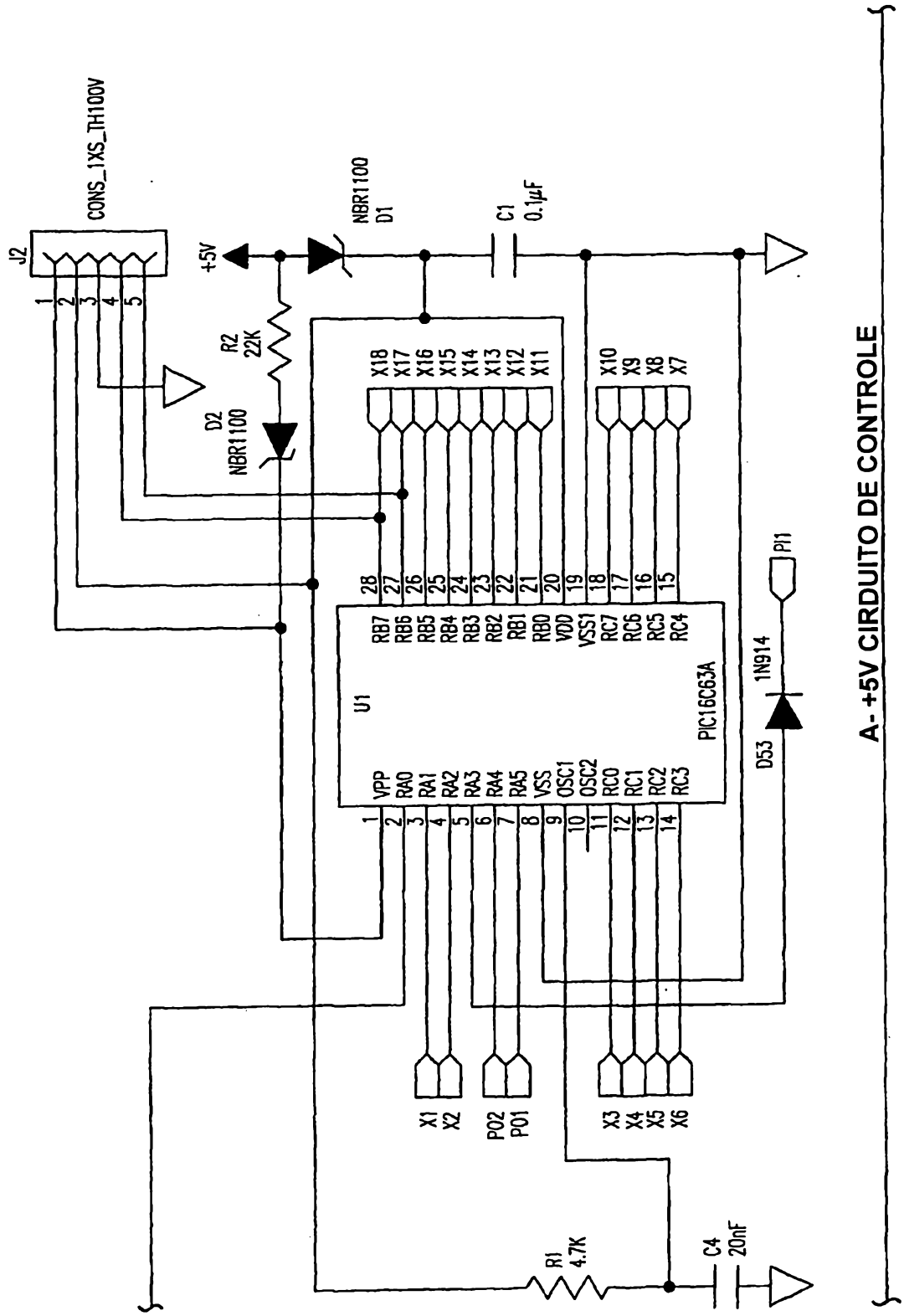

J2
CONS_1XS_TH100V
+5V
NBR1100
D1
C1
0.1µF
R2
22K
D2
NBR1100
X18
X17
X16
X15
X14
X13
X12
X11
X10
X9
X8
X7
U1
PIC16C63A
RB7
RB6
RB5
RB4
RB3
RB2
RB1
RB0
VDD
VSS1
RC7
RC6
RC5
RC4
VPP
RA0
RA1
RA2
RA3
RA4
RA5
VSS
OSC1
OSC2
RC0
RC1
RC2
RC3
PI1
1N914
D53
X1
X2
PO2
PO1
X3
X4
X5
X6
R1
4.7K
C4
20nF
A- +5V CIRDUITO DE CONTROLE
+12V CIRCUITO DE EXCITAÇÃO

FIG.7A-2

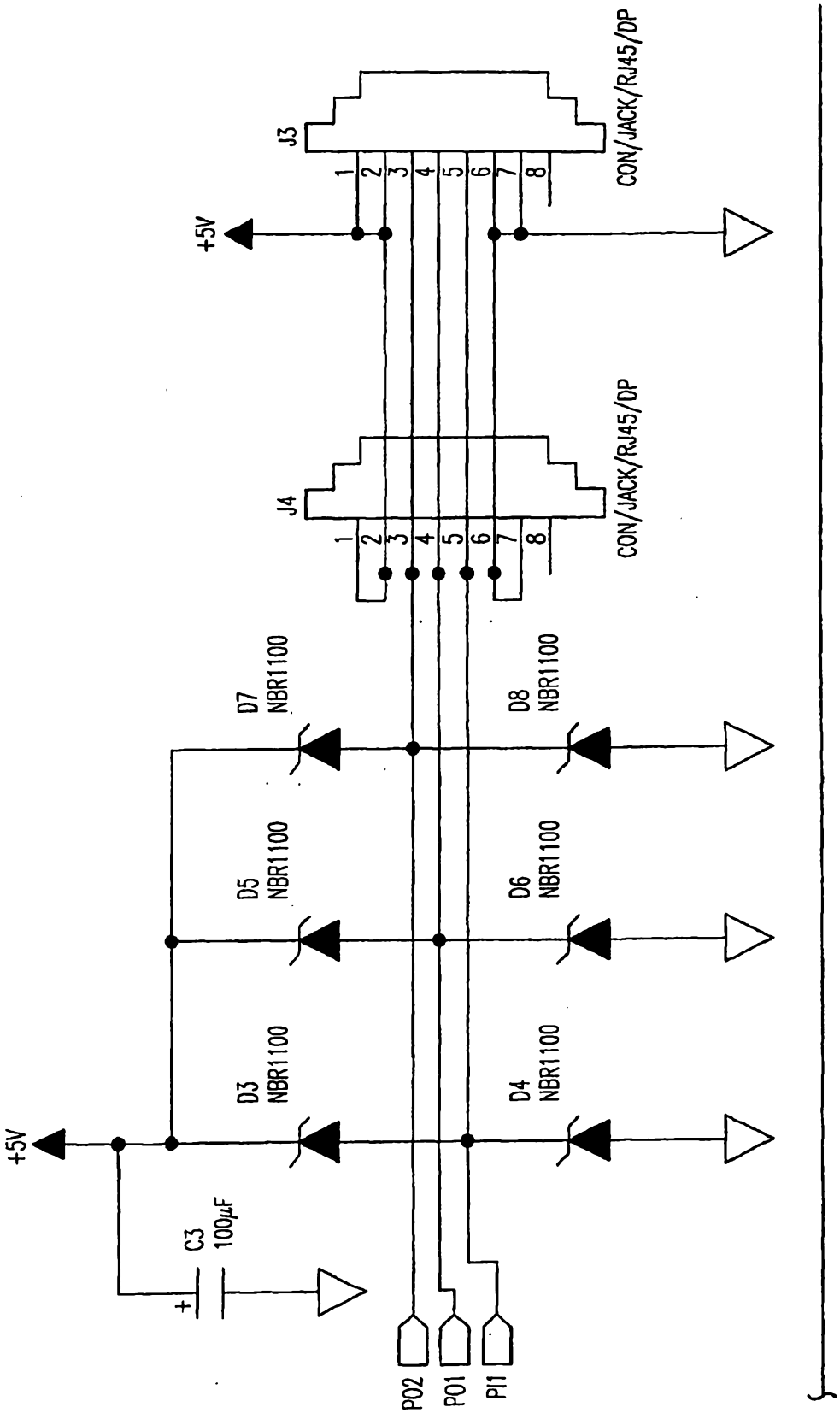

J3
CON/JACK/RJ45/DP
J4
CON/JACK/RJ45/DP
+5V
+5V
D7
NBR1100
D8
NBR1100
D5
NBR1100
D6
NBR1100
D3
NBR1100
D4
NBR1100
C3
100μF
PO2
PO1
PI1

FIG.7A-3

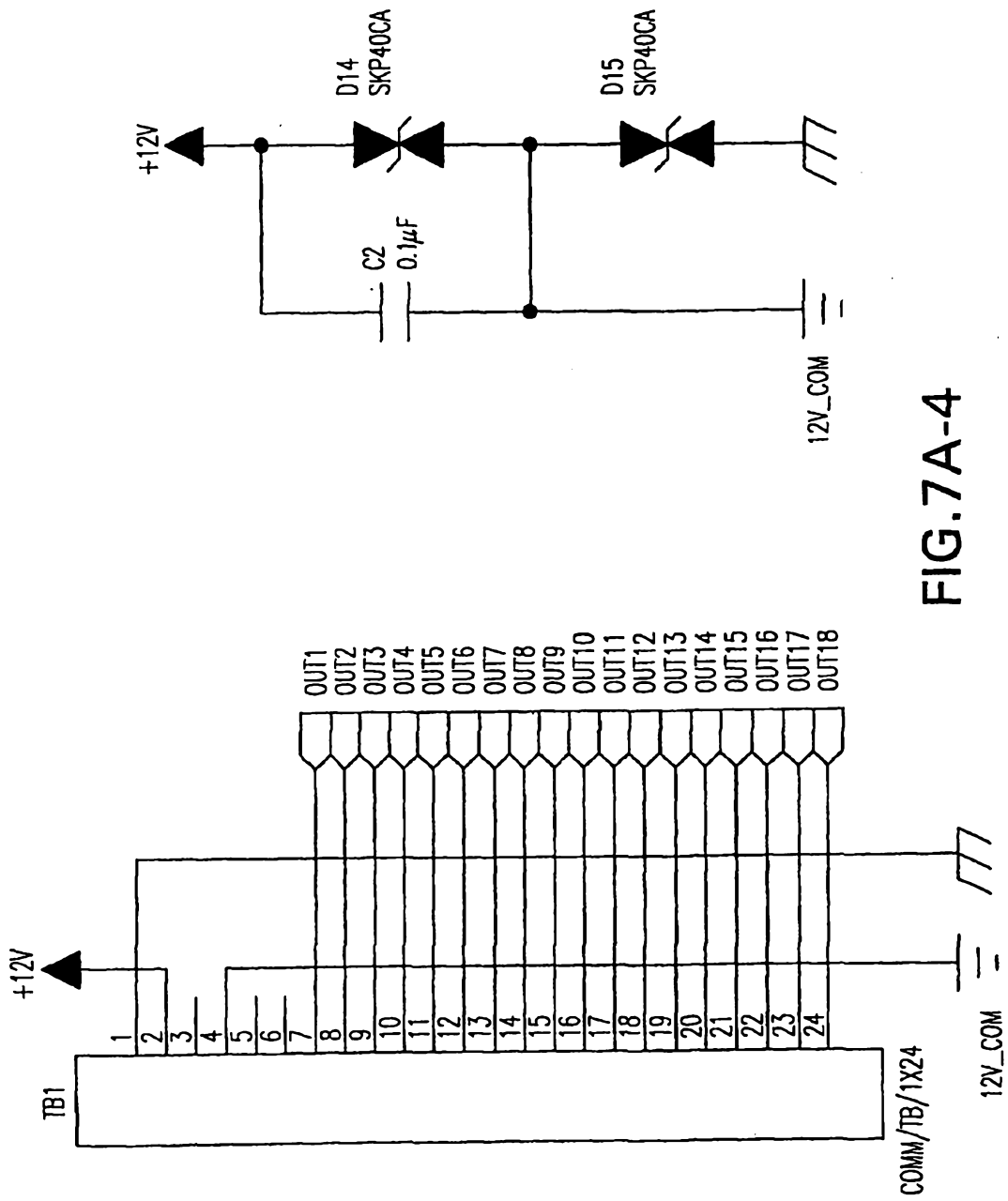

+12V
D14
SKP40CA
D15
SKP40CA
C2
0.1μF
12V_COM
TB1
COMM/TB/1X24
OUT1
OUT2
OUT3
OUT4
OUT5
OUT6
OUT7
OUT8
OUT9
OUT10
OUT11
OUT12
OUT13
OUT14
OUT15
OUT16
OUT17
OUT18
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
+12V
12V_COM

FIG.7A-4

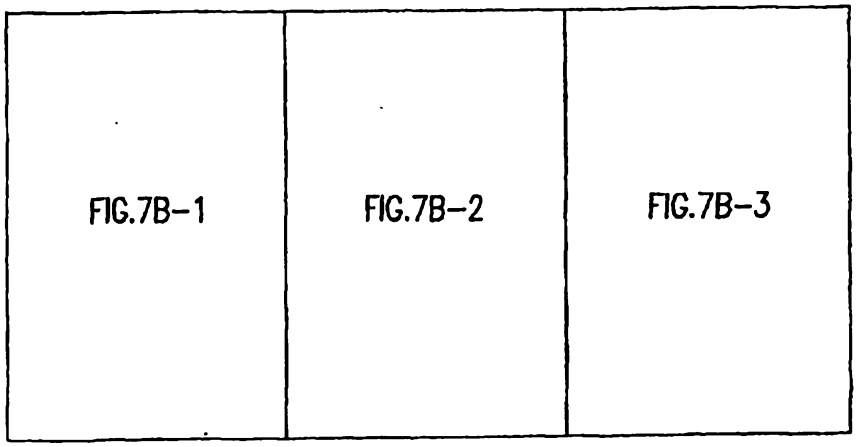


FIG.7B

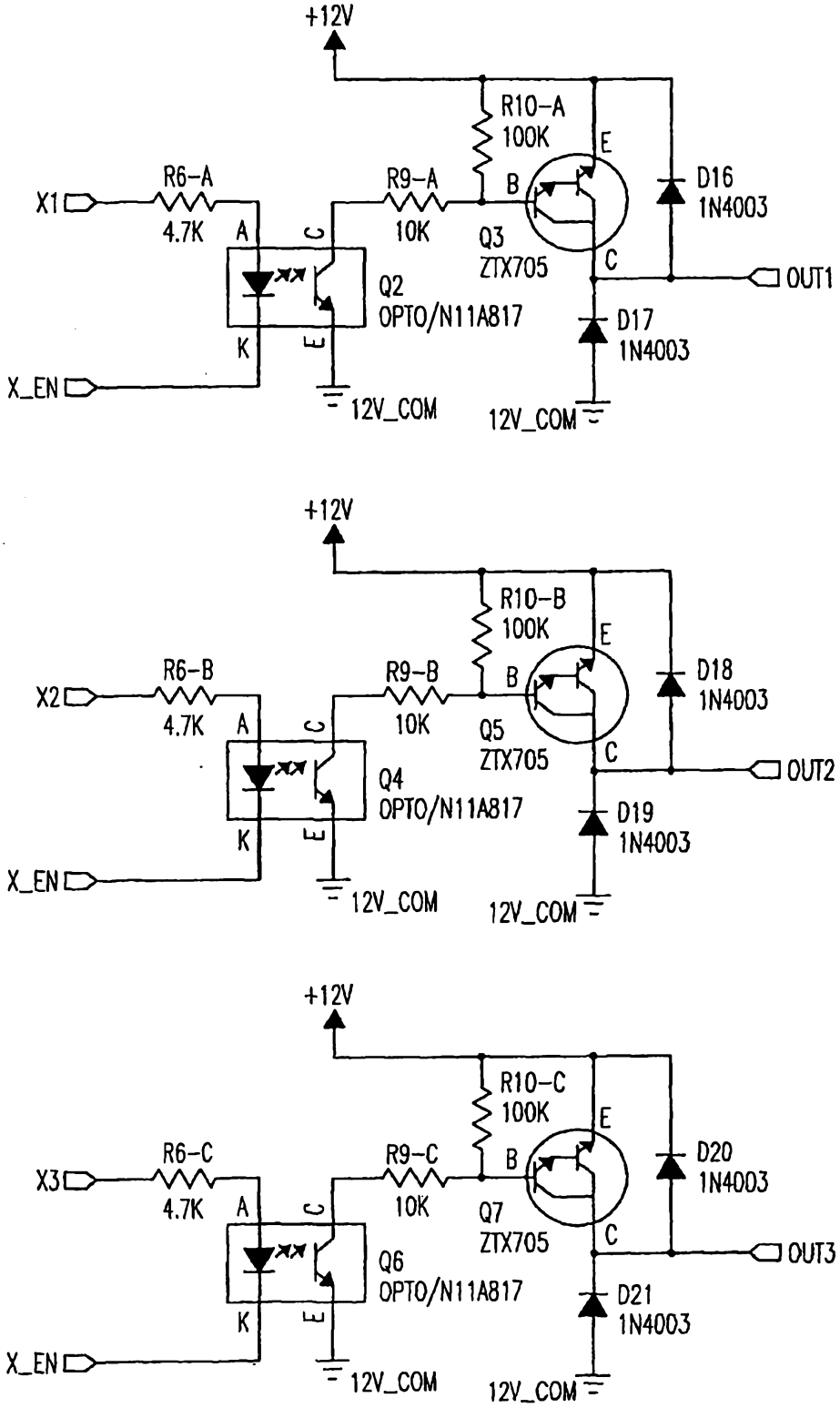

+12V
R10-A
100K
E
R6-A
R9-A
B
D16
1N4003
X1
4.7K
A
C
10K
Q3
ZTX705
C
OUT1
Q2
OPTO/N11A817
D17
1N4003
K
E
X_EN
12V_COM
12V_COM
+12V
R10-B
100K
E
R6-B
R9-B
B
D18
1N4003
X2
4.7K
A
C
10K
Q5
ZTX705
C
OUT2
Q4
OPTO/N11A817
D19
1N4003
K
E
X_EN
12V_COM
12V_COM
+12V
R10-C
100K
E
R6-C
R9-C
B
D20
1N4003
X3
4.7K
A
C
10K
Q7
ZTX705
C
OUT3
Q6
OPTO/N11A817
D21
1N4003
K
E
X_EN
12V_COM
12V_COM

FIG.7B-1

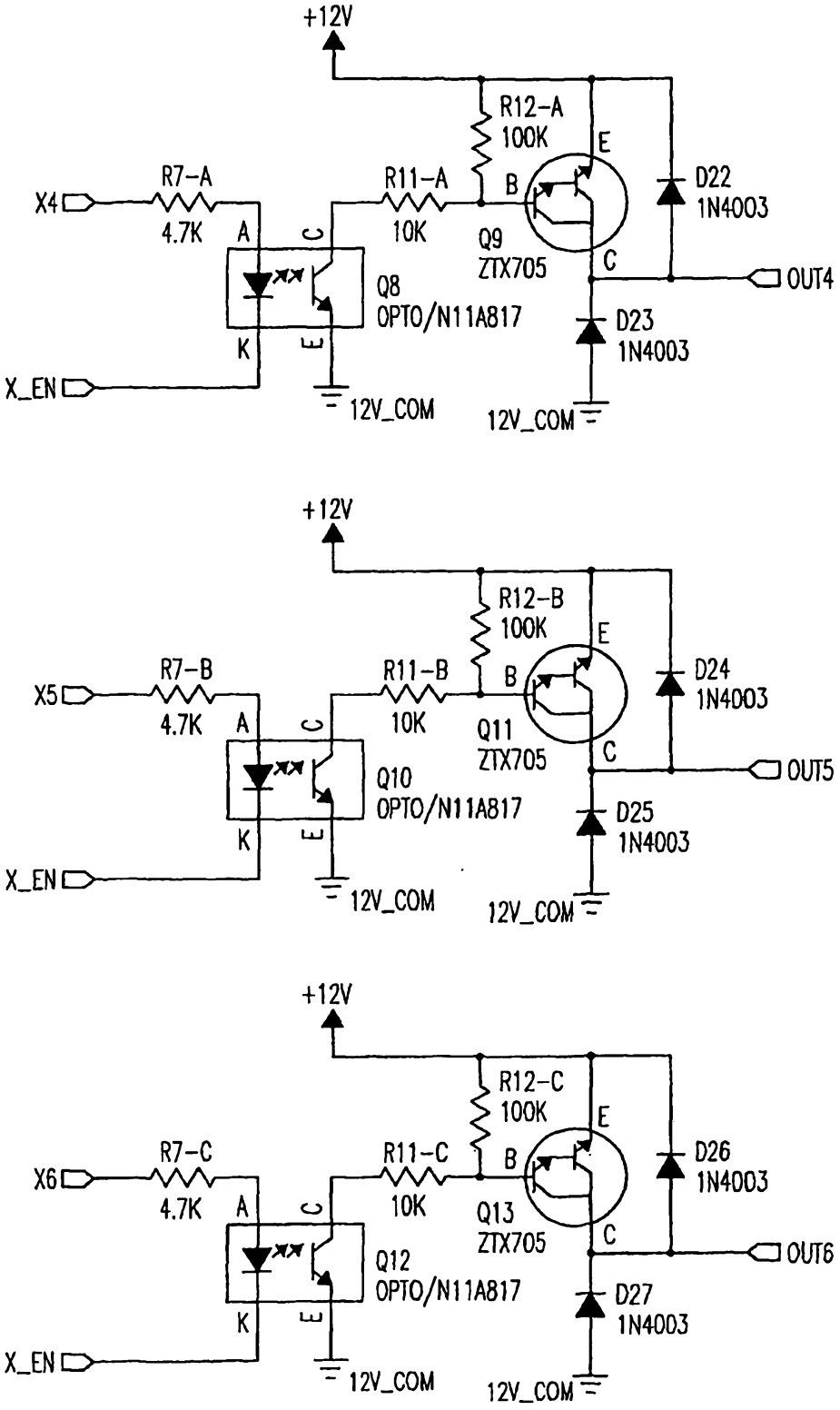

+12V
R12-A
100K
E
R7-A
R11-A
B
D22
1N4003
X4
4.7K
A
C
10K
Q9
ZTX705
C
OUT4
Q8
OPTO/N11A817
D23
1N4003
K
E
X_EN
12V_COM
12V_COM
+12V
R12-B
100K
E
R7-B
R11-B
B
D24
1N4003
X5
4.7K
A
C
10K
Q11
ZTX705
C
OUT5
Q10
OPTO/N11A817
D25
1N4003
K
E
X_EN
12V_COM
12V_COM
+12V
R12-C
100K
E
R7-C
R11-C
B
D26
1N4003
X6
4.7K
A
C
10K
Q13
ZTX705
C
OUT6
Q12
OPTO/N11A817
D27
1N4003
K
E
X_EN
12V_COM
12V_COM

FIG.7B-2

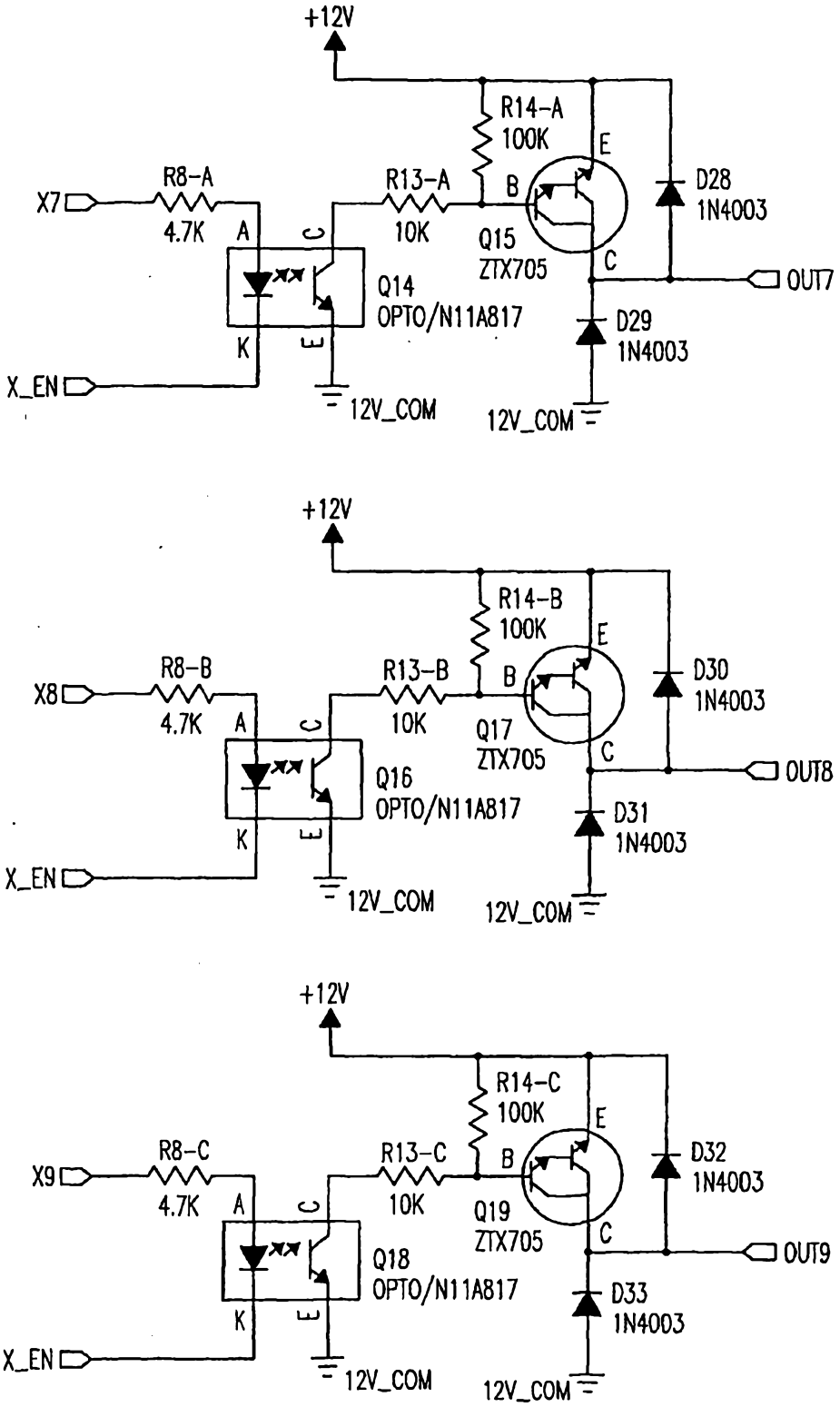

+12V
R14-A
100K
E
R8-A
R13-A
B
D28
1N4003
X7
4.7K
A
C
10K
Q15
ZTX705
C
OUT7
Q14
OPTO/N11A817
D29
1N4003
K
E
X_EN
12V_COM
12V_COM
+12V
R14-B
100K
E
R8-B
R13-B
B
D30
1N4003
X8
4.7K
A
C
10K
Q17
ZTX705
C
OUT8
Q16
OPTO/N11A817
D31
1N4003
K
E
X_EN
12V_COM
12V_COM
+12V
R14-C
100K
E
R8-C
R13-C
B
D32
1N4003
X9
4.7K
A
C
10K
Q19
ZTX705
C
OUT9
Q18
OPTO/N11A817
D33
1N4003
K
E
X_EN
12V_COM
12V_COM

FIG.7B-3

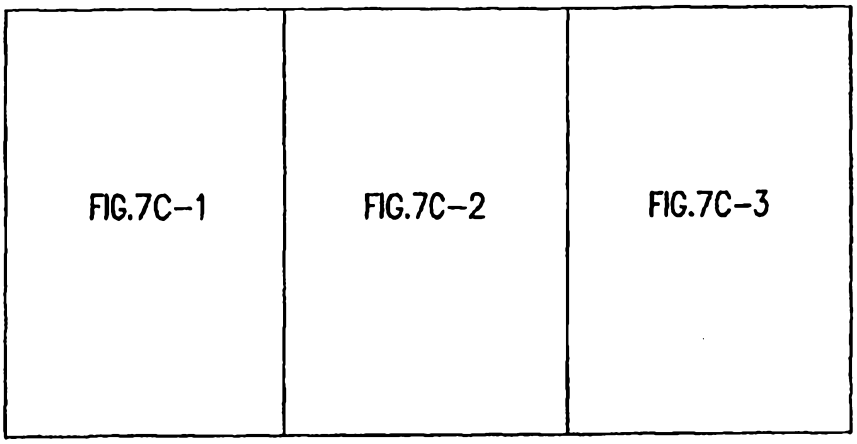


FIG.7C

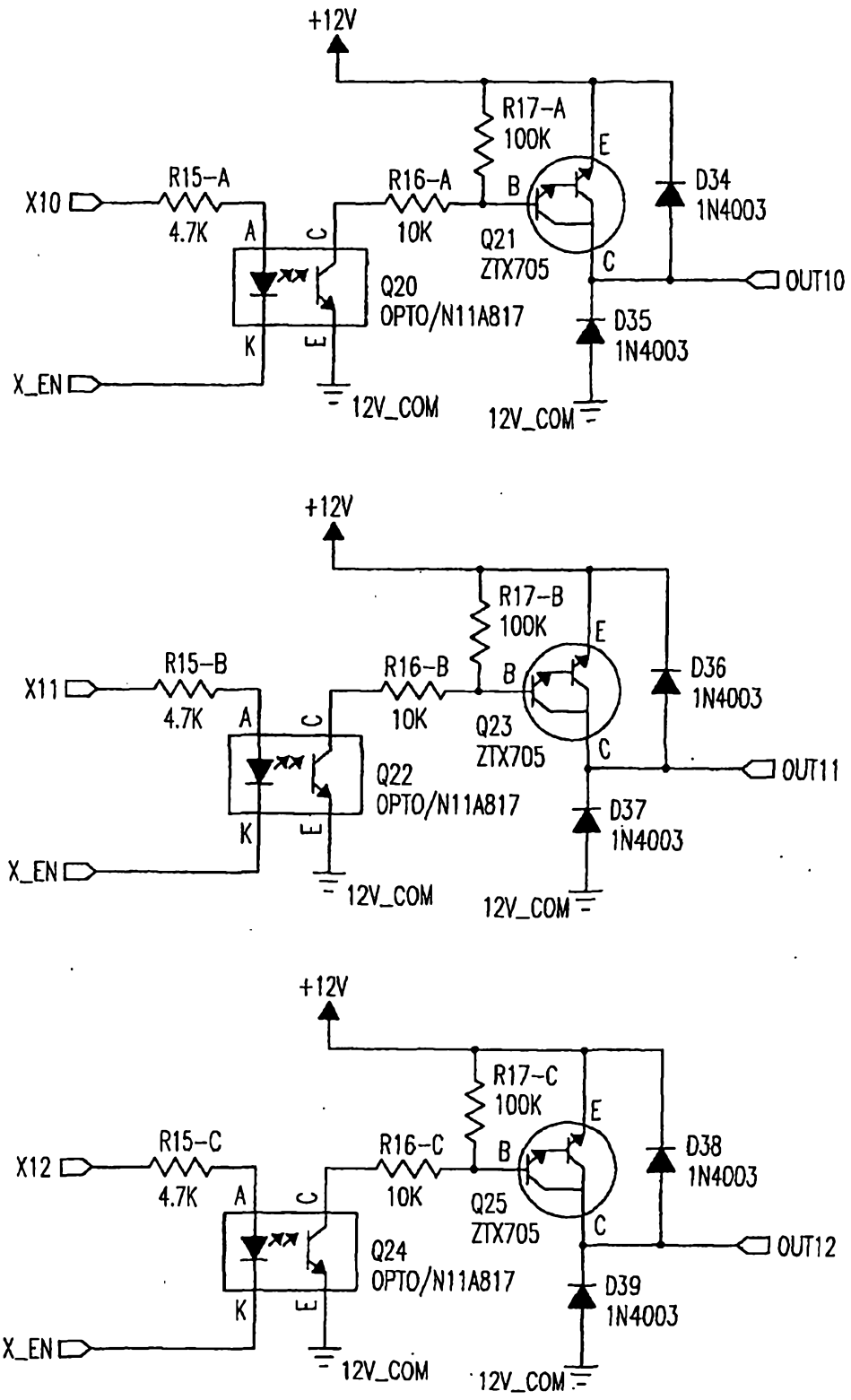

+12V
R17-A
100K
E
X10
R15-A
4.7K
A
C
R16-A
10K
B
Q21
ZTX705
D34
1N4003
C
OUT10
Q20
OPTO/N11A817
D35
1N4003
K
E
X_EN
12V_COM
12V_COM
+12V
R17-B
100K
E
X11
R15-B
4.7K
A
C
R16-B
10K
B
Q23
ZTX705
D36
1N4003
C
OUT11
Q22
OPTO/N11A817
D37
1N4003
K
E
X_EN
12V_COM
12V_COM
+12V
R17-C
100K
E
X12
R15-C
4.7K
A
C
R16-C
10K
B
Q25
ZTX705
D38
1N4003
C
OUT12
Q24
OPTO/N11A817
D39
1N4003
K
E
X_EN
12V_COM
12V_COM

FIG.7C-1

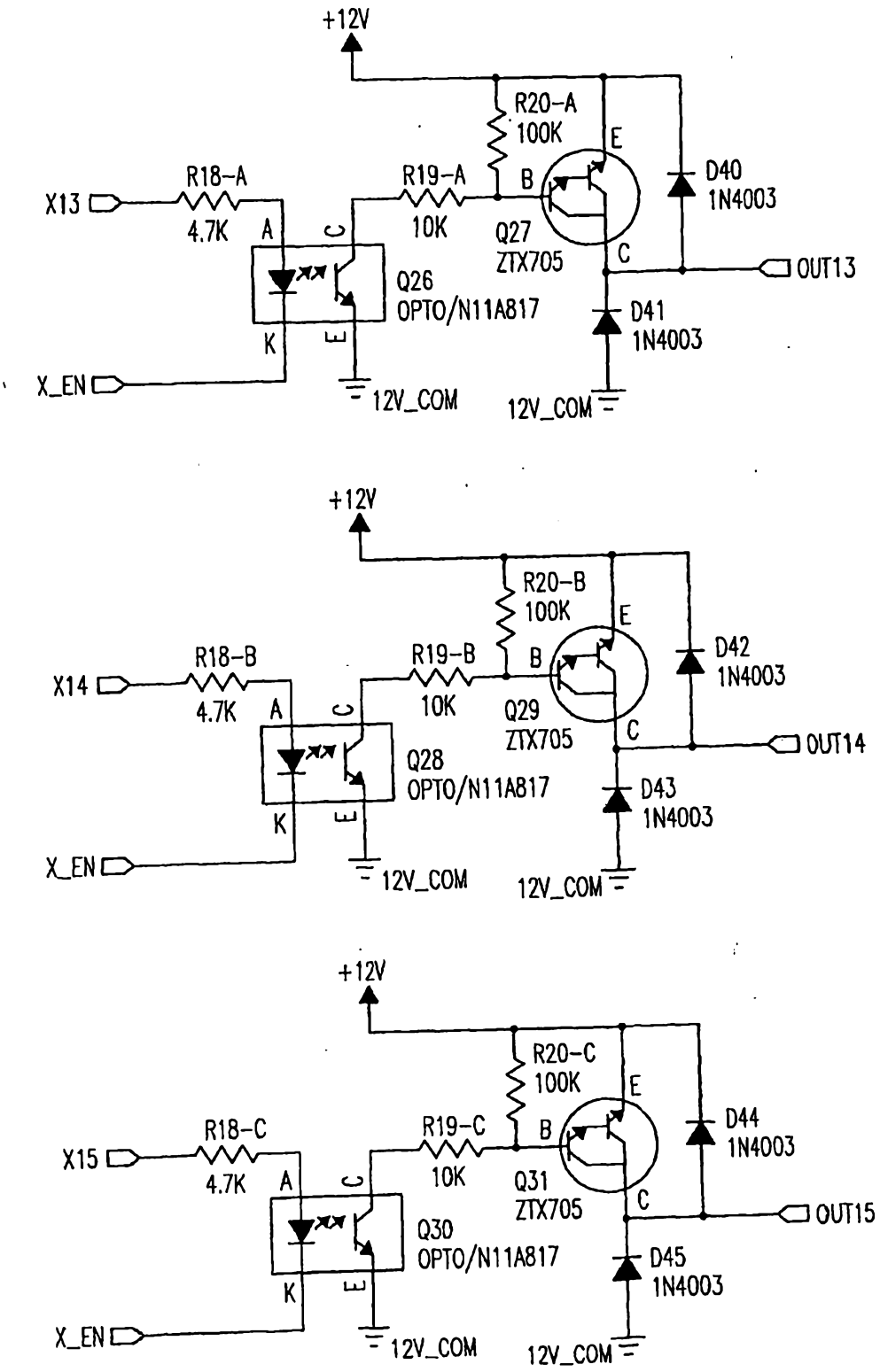

+12V
R20-A
100K
E
D40
1N4003
X13
R18-A
4.7K
A
C
R19-A
10K
B
Q27
ZTX705
C
OUT13
Q26
OPTO/N11A817
D41
1N4003
K
E
X_EN
12V_COM
12V_COM
+12V
R20-B
100K
E
D42
1N4003
X14
R18-B
4.7K
A
C
R19-B
10K
B
Q29
ZTX705
C
OUT14
Q28
OPTO/N11A817
D43
1N4003
K
E
X_EN
12V_COM
12V_COM
+12V
R20-C
100K
E
D44
1N4003
X15
R18-C
4.7K
A
C
R19-C
10K
B
Q31
ZTX705
C
OUT15
Q30
OPTO/N11A817
D45
1N4003
K
E
X_EN
12V_COM
12V_COM

FIG.7C-2

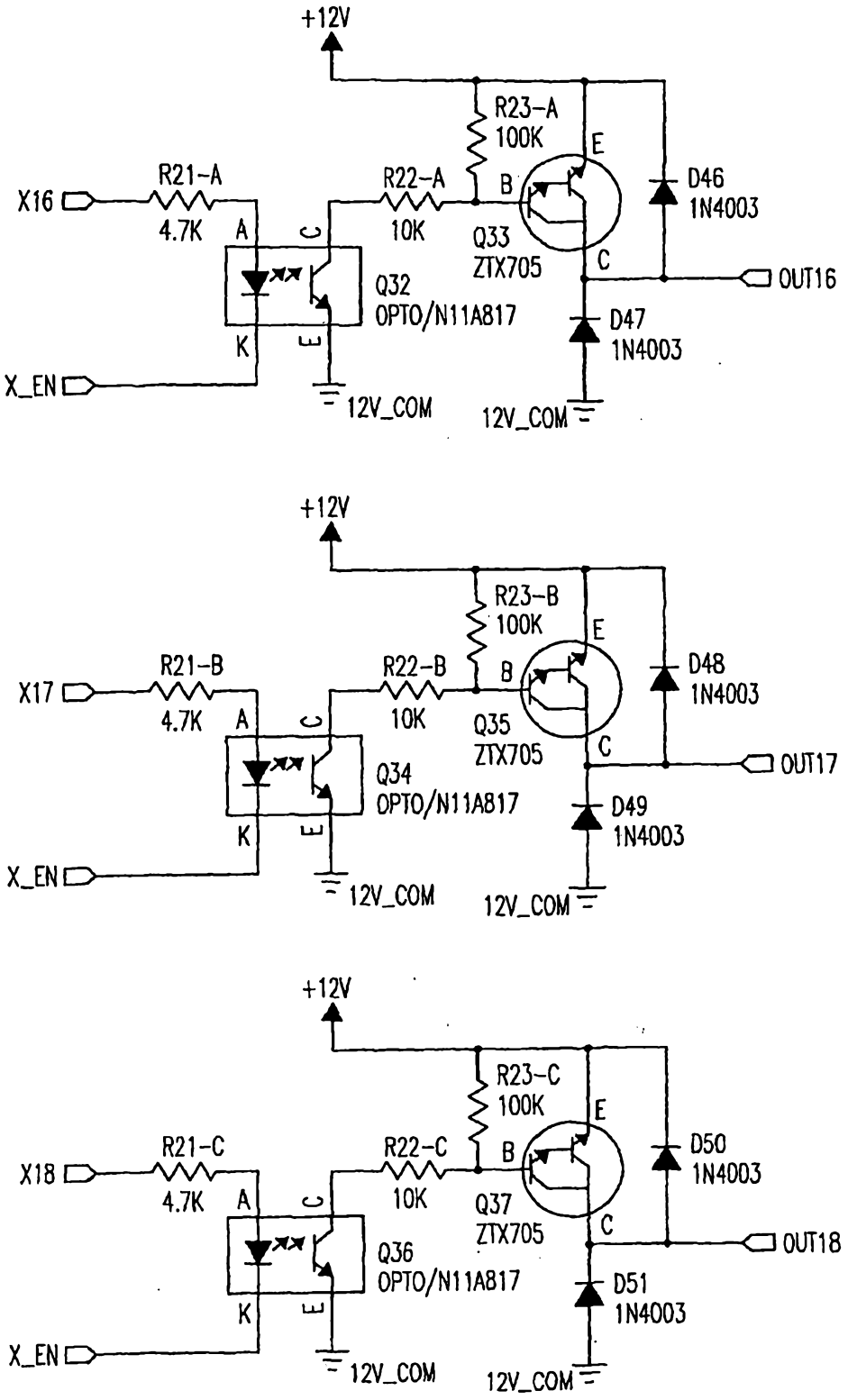

+12V
R23-A
100K
E
R21-A
R22-A
B
D46
1N4003
X16
4.7K
A
C
10K
Q33
ZTX705
C
OUT16
Q32
OPTO/N11A817
D47
1N4003
K
E
X_EN
12V_COM
12V_COM
+12V
R23-B
100K
E
R21-B
R22-B
B
D48
1N4003
X17
4.7K
A
C
10K
Q35
ZTX705
C
OUT17
Q34
OPTO/N11A817
D49
1N4003
K
E
X_EN
12V_COM
12V_COM
+12V
R23-C
100K
E
R21-C
R22-C
B
D50
1N4003
X18
4.7K
A
C
10K
Q37
ZTX705
C
OUT18
Q36
OPTO/N11A817
D51
1N4003
K
E
X_EN
12V_COM
12V_COM

FIG.7C-3

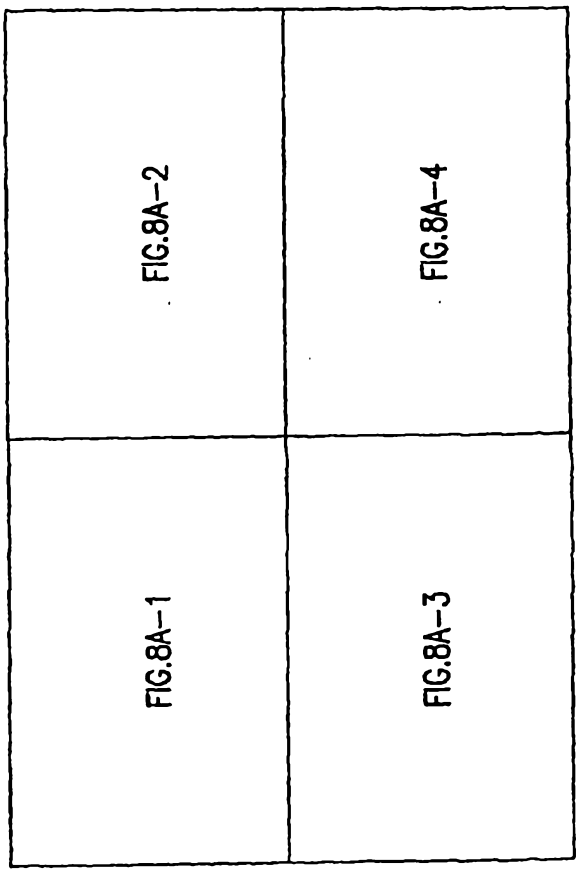


FIG.8A

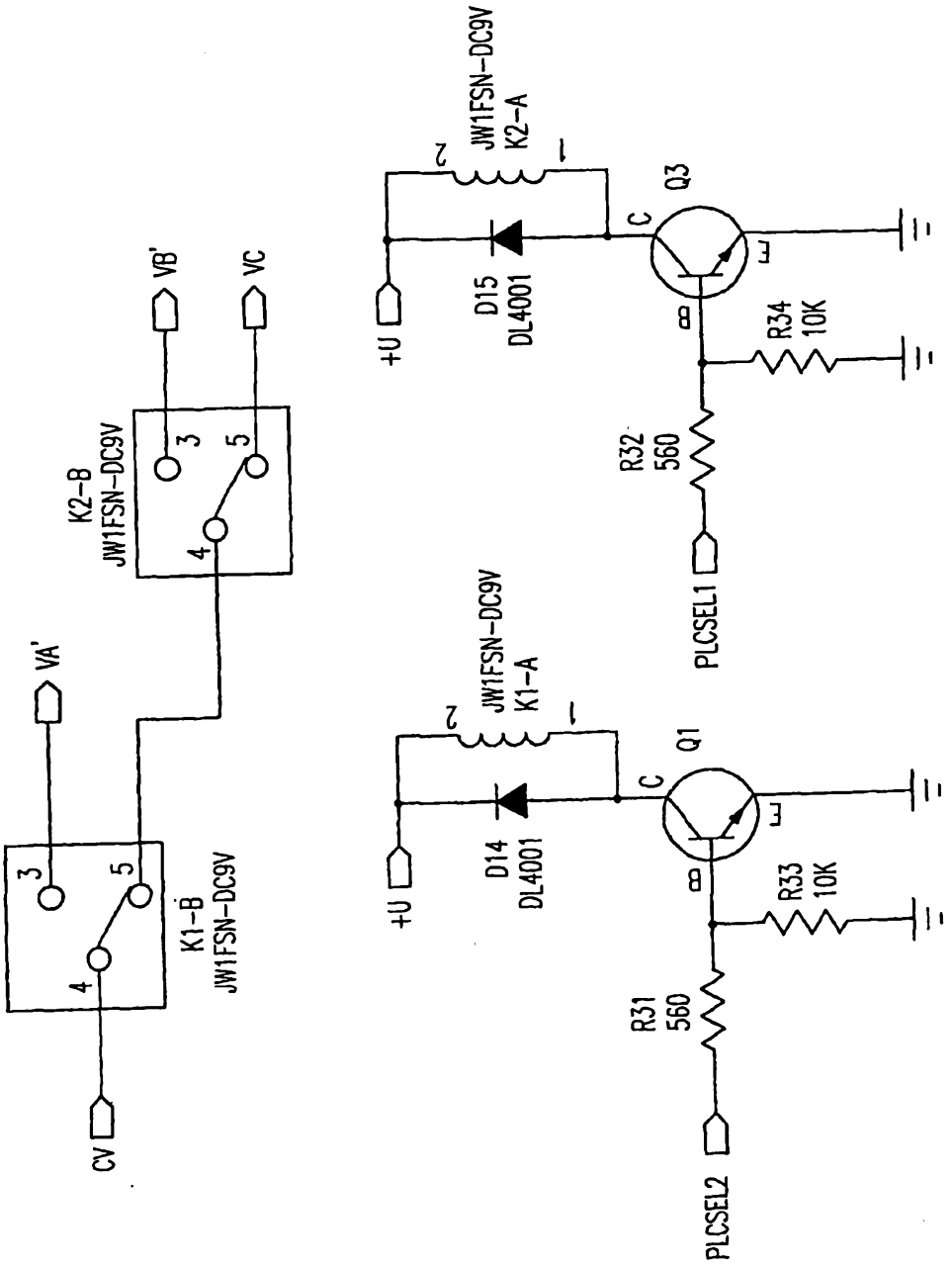

JW1FSN-DC9V
K2-A
Q3
D15
DL4001
+U
R34
10K
R32
560
PLCSEL1
JW1FSN-DC9V
K1-A
Q1
D14
DL4001
+U
R33
10K
R31
560
PLCSEL2
VB'
VC
K2-B
JW1FSN-DC9V
VA'
K1-B
JW1FSN-DC9V
CV

FIG.8A-1

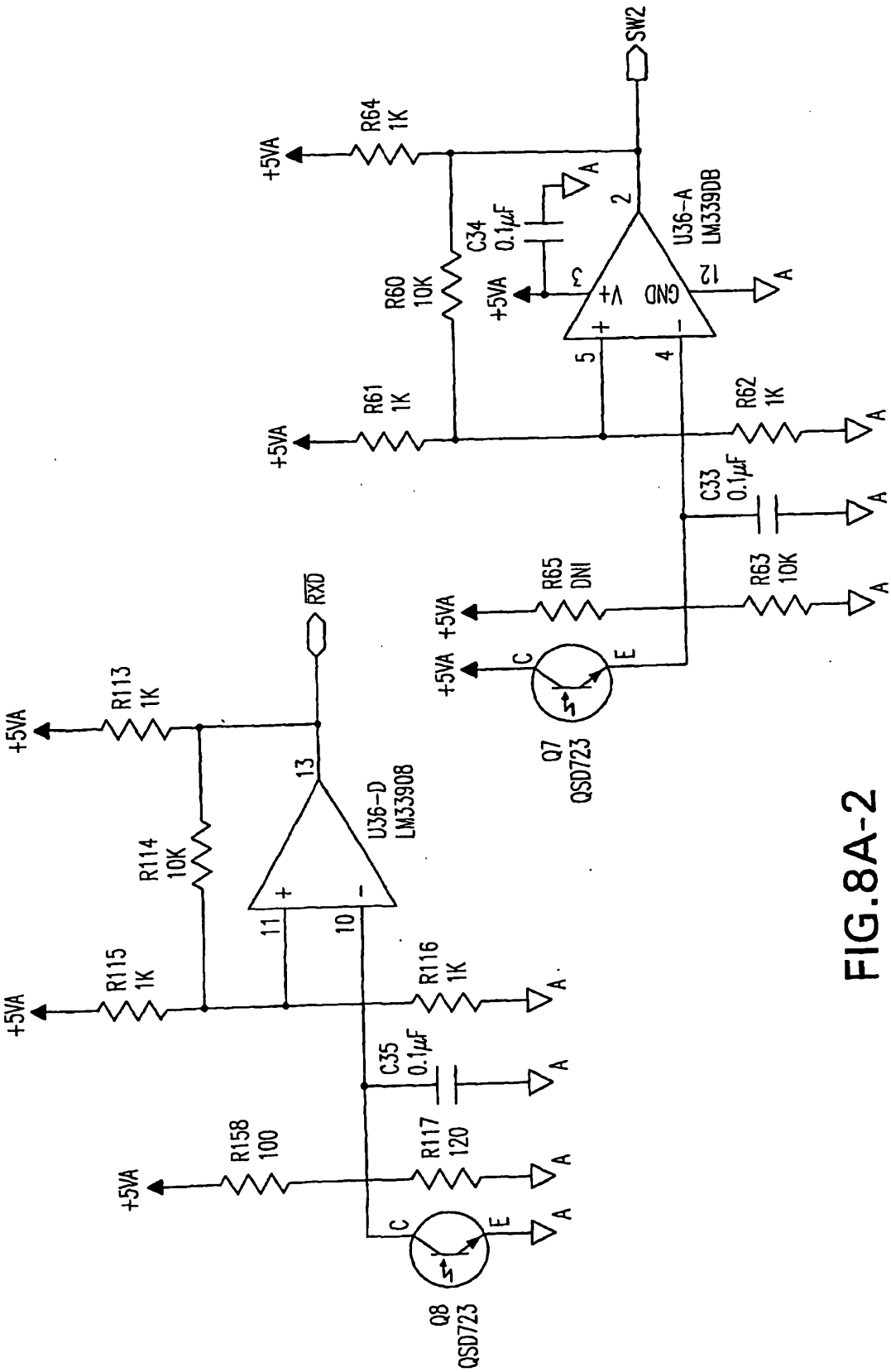

SW2
R64
1K
+5VA
C34
0.1µF
R60
10K
U36-A
LM339DB
GND
V+
R61
1K
R62
1K
C33
0.1µF
R65
DNI
R63
10K
Q7
QSD723
RXD
R113
1K
U36-D
LM33908
R114
10K
R115
1K
R116
1K
C35
0.1µF
R158
100
R117
120
Q8
QSD723

FIG.8A-2

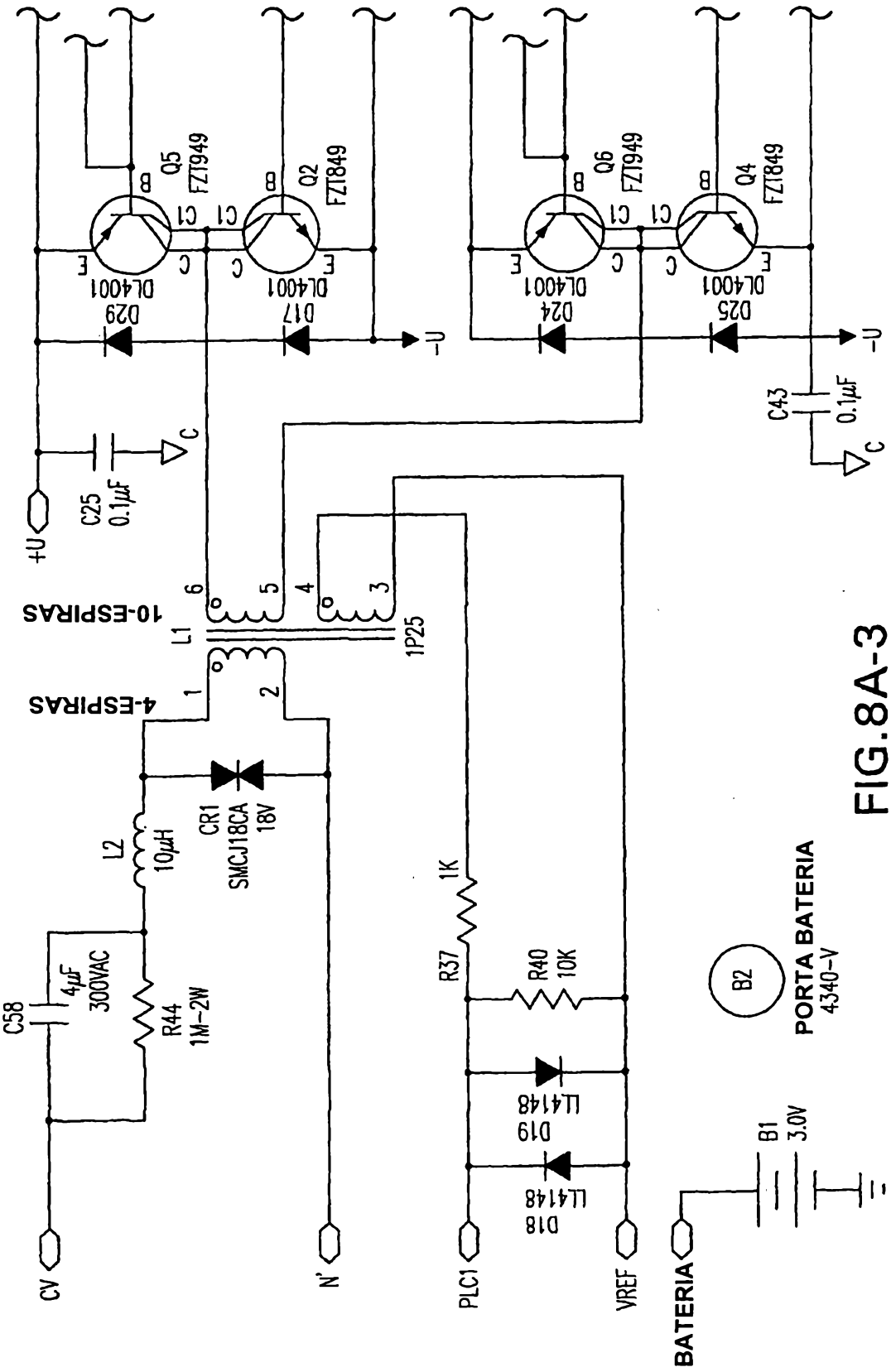

Q5 FZT949
Q2 FZT849
Q6 FZT949
Q4 FZT849
D29 DL4001
D17 DL4001
D24 DL4001
D25 DL4001
-U
+U
C25 0.1µF
C43 0.1µF
Vc
10-ESPIRAS
4-ESPIRAS
L1
1P25
CR1 SMCJ18CA 18V
L2 10µH
C58 4µF 300VAC
R44 1M-2W
R37 1K
R40 10K
D19 LL4148
D18 LL4148
CV
N'
PLC1
VREF
BATERIA
B1 3.0V
B2 PORTA BATERIA 4340-V

FIG.8A-3

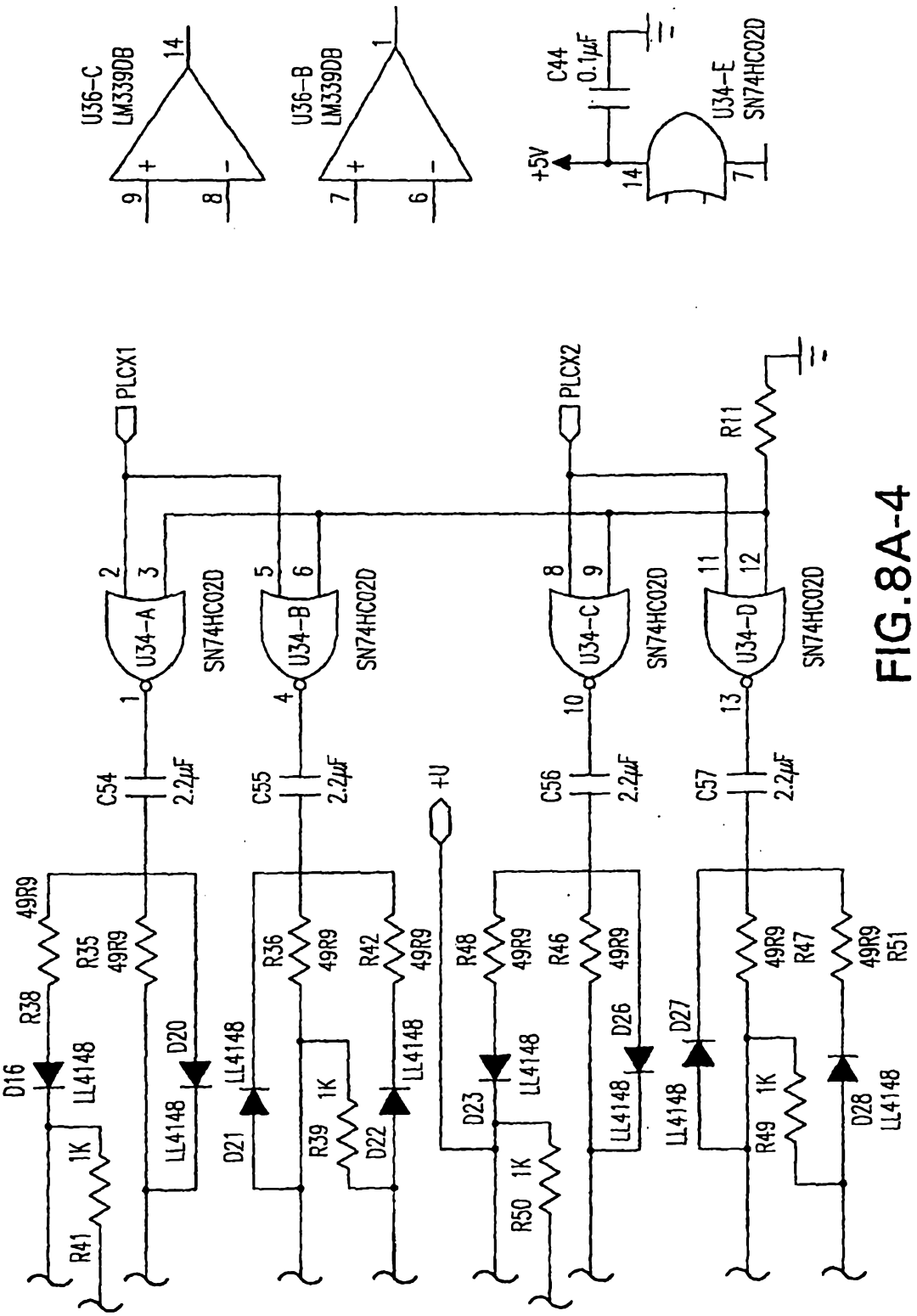

U36-C
LM339DB
U36-B
LM339DB
U34-E
SN74HC02D
C44
0.1µF
+5V
PLCX1
PLCX2
U34-A
U34-B
U34-C
U34-D
SN74HC02D
C54
C55
C56
C57
2.2µF
+U
R11
R35
R36
R42
R46
R47
R48
R51
49R9
R38
R39
R41
R49
R50
1K
D16
D20
D21
D22
D23
D26
D27
D28
LL4148

FIG.8A-4

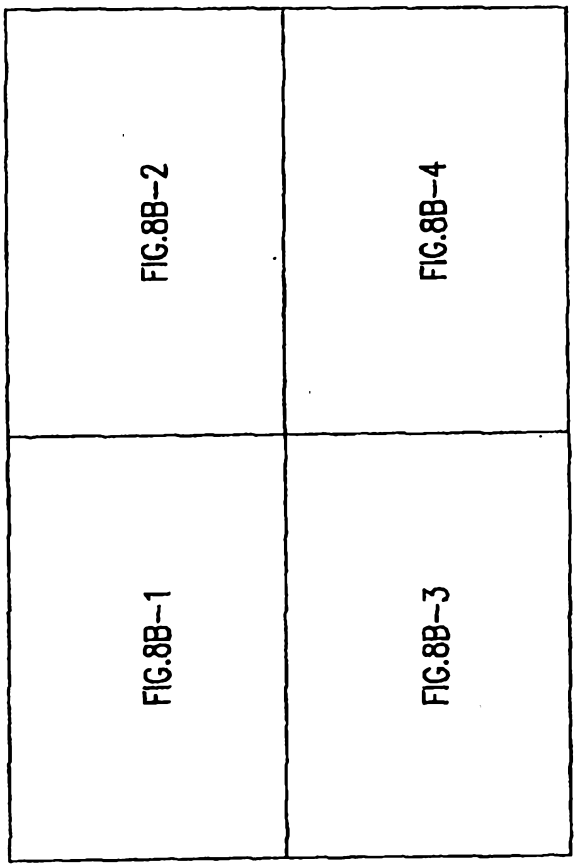


FIG.8B

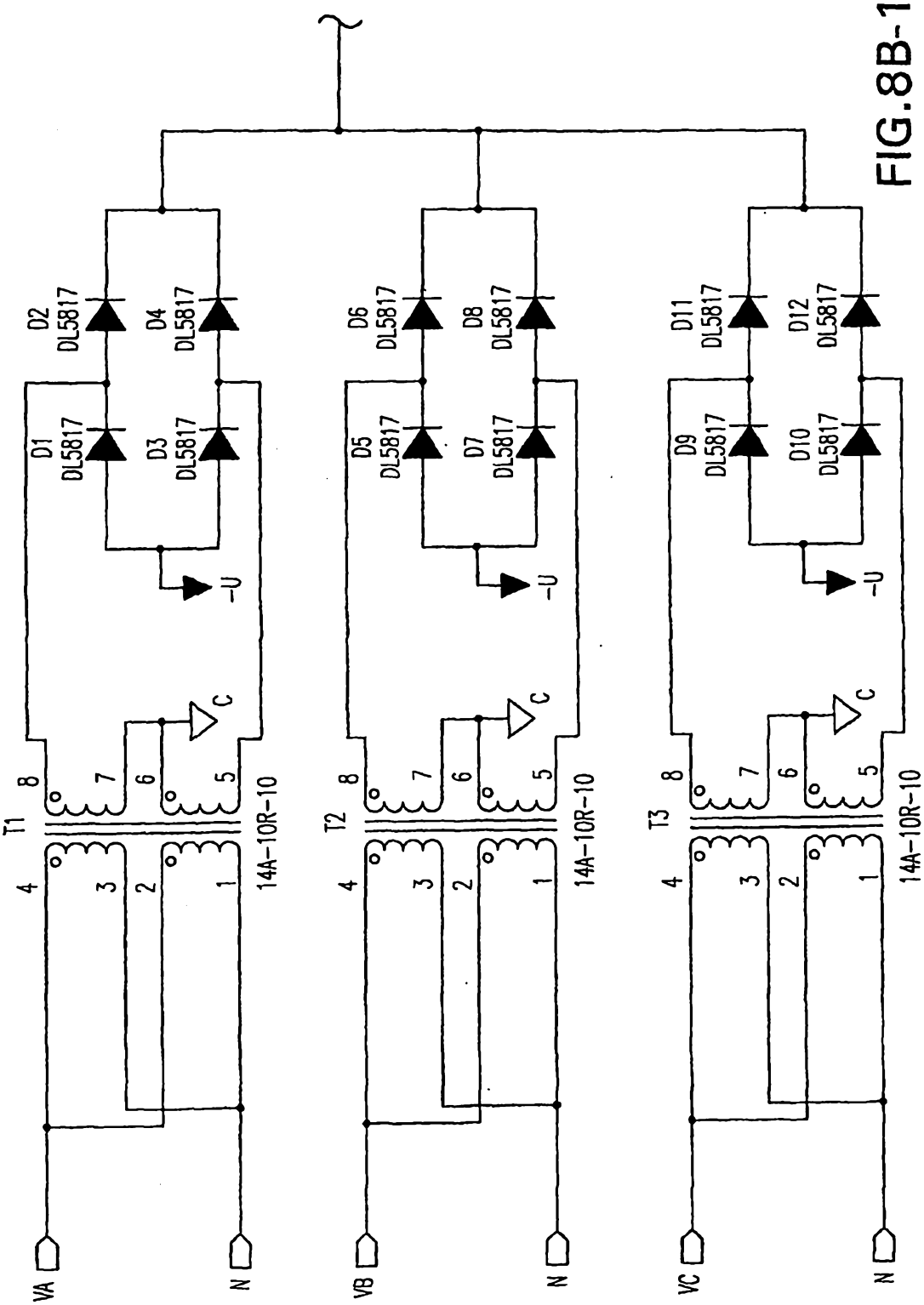

D1 DL5817
D2 DL5817
D3 DL5817
D4 DL5817
D5 DL5817
D6 DL5817
D7 DL5817
D8 DL5817
D9 DL5817
D10 DL5817
D11 DL5817
D12 DL5817
-U
C
T1
T2
T3
14A-10R-10
VA
VB
VC
N

FIG.8B-1

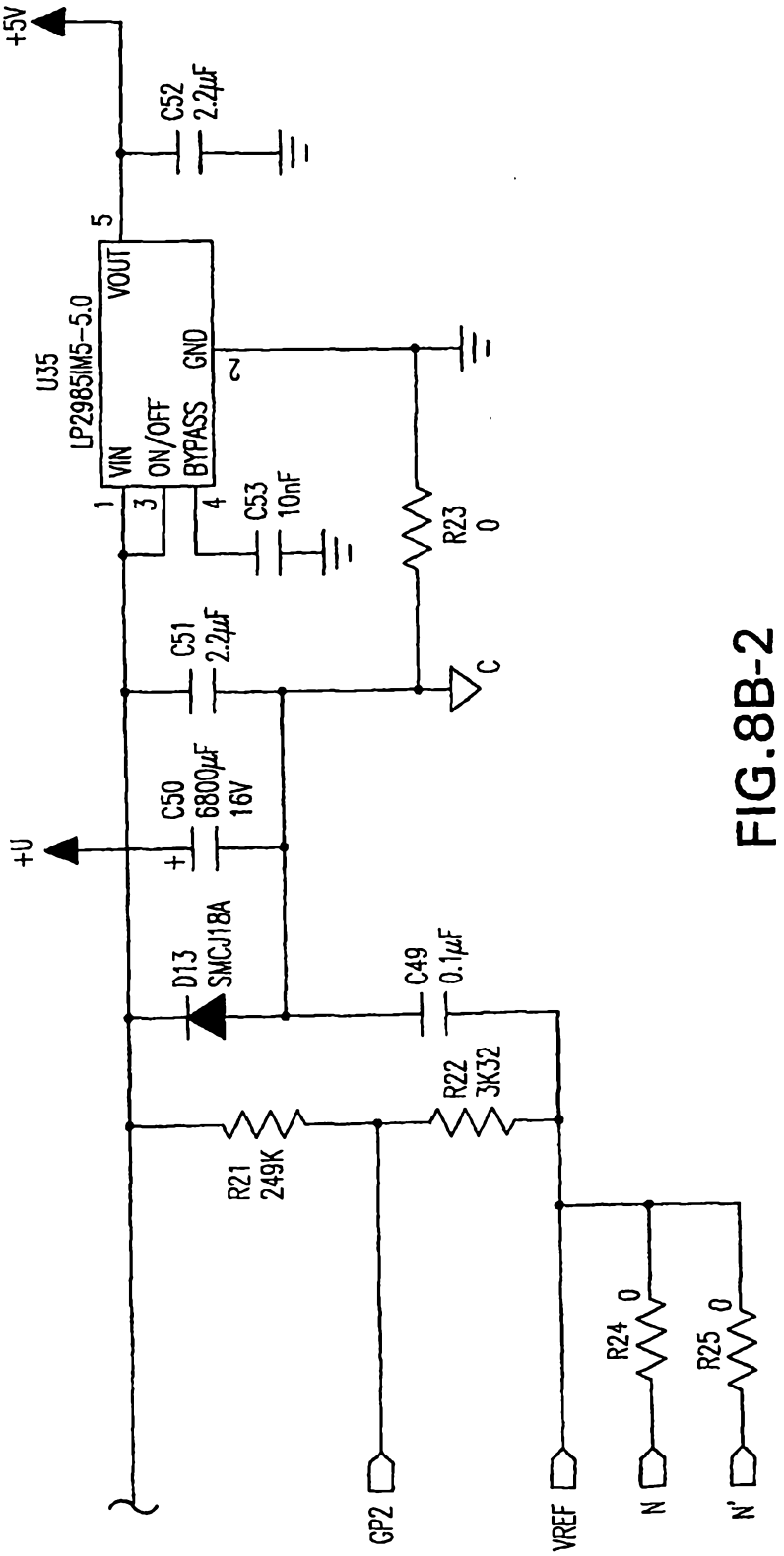

+5V
C52
2.2µF
5
VOUT
U35
LP2985IM5-5.0
GND
VIN
ON/OFF
BYPASS
1
3
4
C53
10nF
R23
0
C51
2.2µF
C
C50
6800µF
16V
+U
D13
SMCJ18A
C49
0.1µF
R22
3K32
R21
249K
GP2
VREF
R24
0
R25
0
N
N'

FIG.8B-2

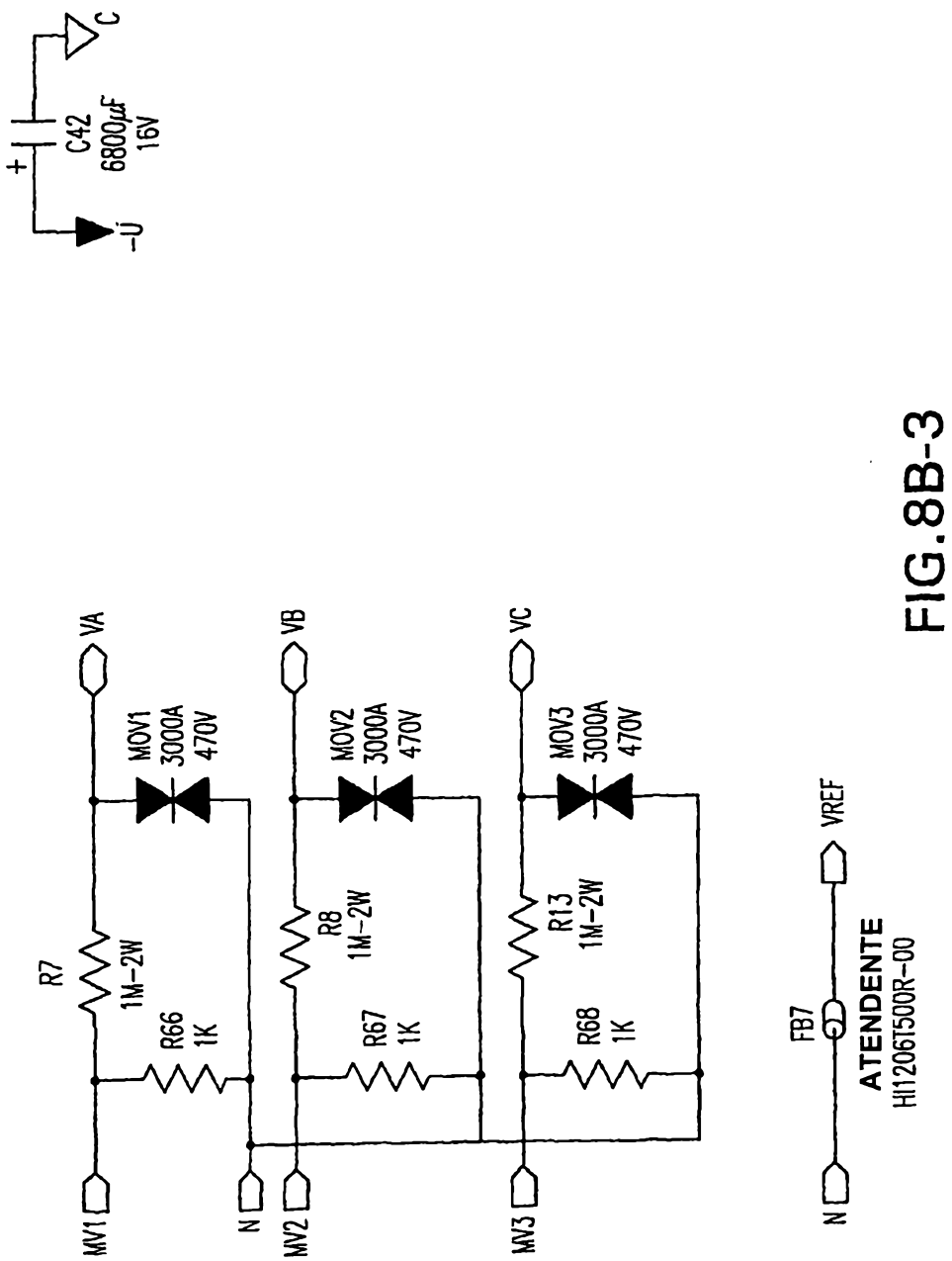
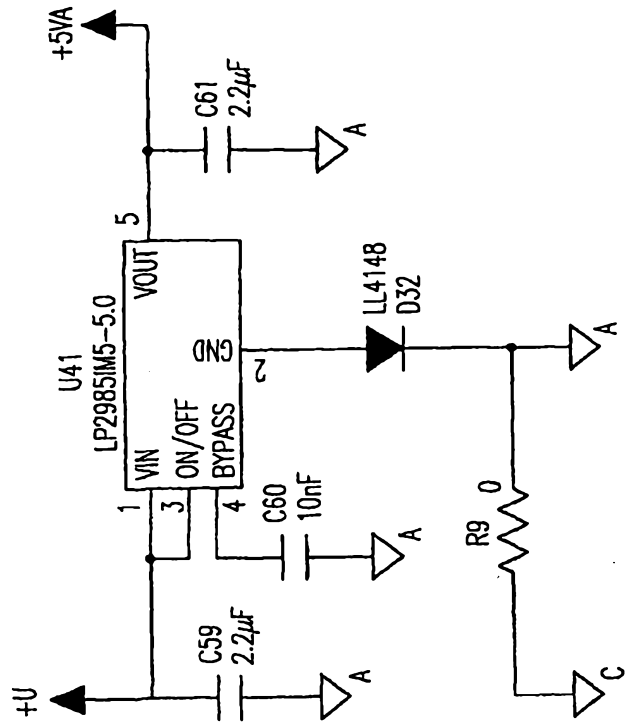

C
C42
6800μF
16V
+
-U
VA
VB
VC
MOV1
3000A
470V
MOV2
3000A
470V
MOV3
3000A
470V
R7
1M-2W
R66
1K
R8
1M-2W
R67
1K
R13
1M-2W
R68
1K
MV1
N
MV2
MV3
FB7
VREF
ATENDENTE
H11206T500R-00
N

FIG.8B-3


+5VA
C61
2.2µF
A
5
VOUT
U41
LP2985IM5-5.0
GND
2
LL414B
D32
A
VIN
ON/OFF
BYPASS
1
3
4
C60
10nF
A
R9
0
C59
2.2µF
A
+U
C

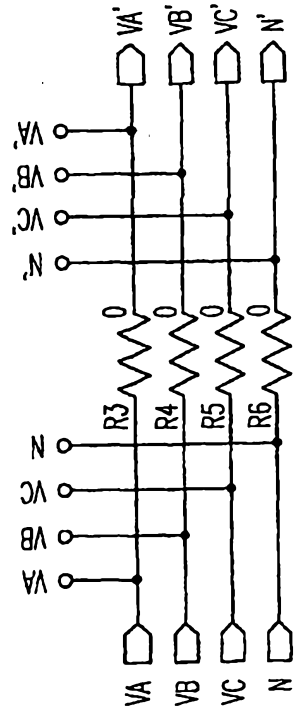

VA'
VB'
VC'
N'
VA'
VB'
VC'
N'
0
0
0
0
R3
R4
R5
R6
N
VC
VB
VA
VA
VB
VC
N

FIG.8B-4

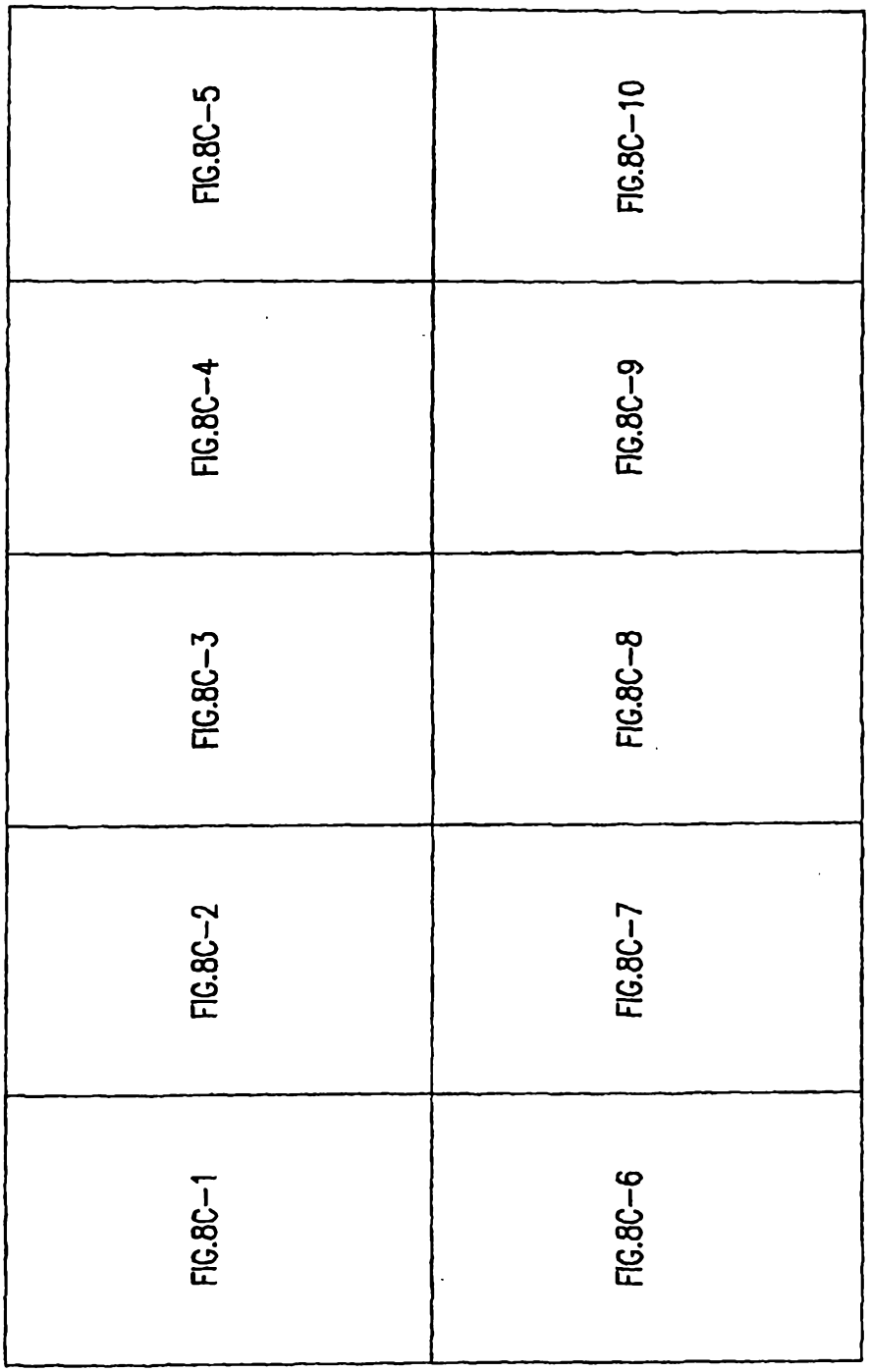


FIG.8C-1
FIG.8C-2
FIG.8C-3
FIG.8C-4
FIG.8C-5
FIG.8C-6
FIG.8C-7
FIG.8C-8
FIG.8C-9
FIG.8C-10
NOTA: PARA ENTRADAS 1 E 2 SE REFIRA A H4 DA FIGURA 3

FIG.8C

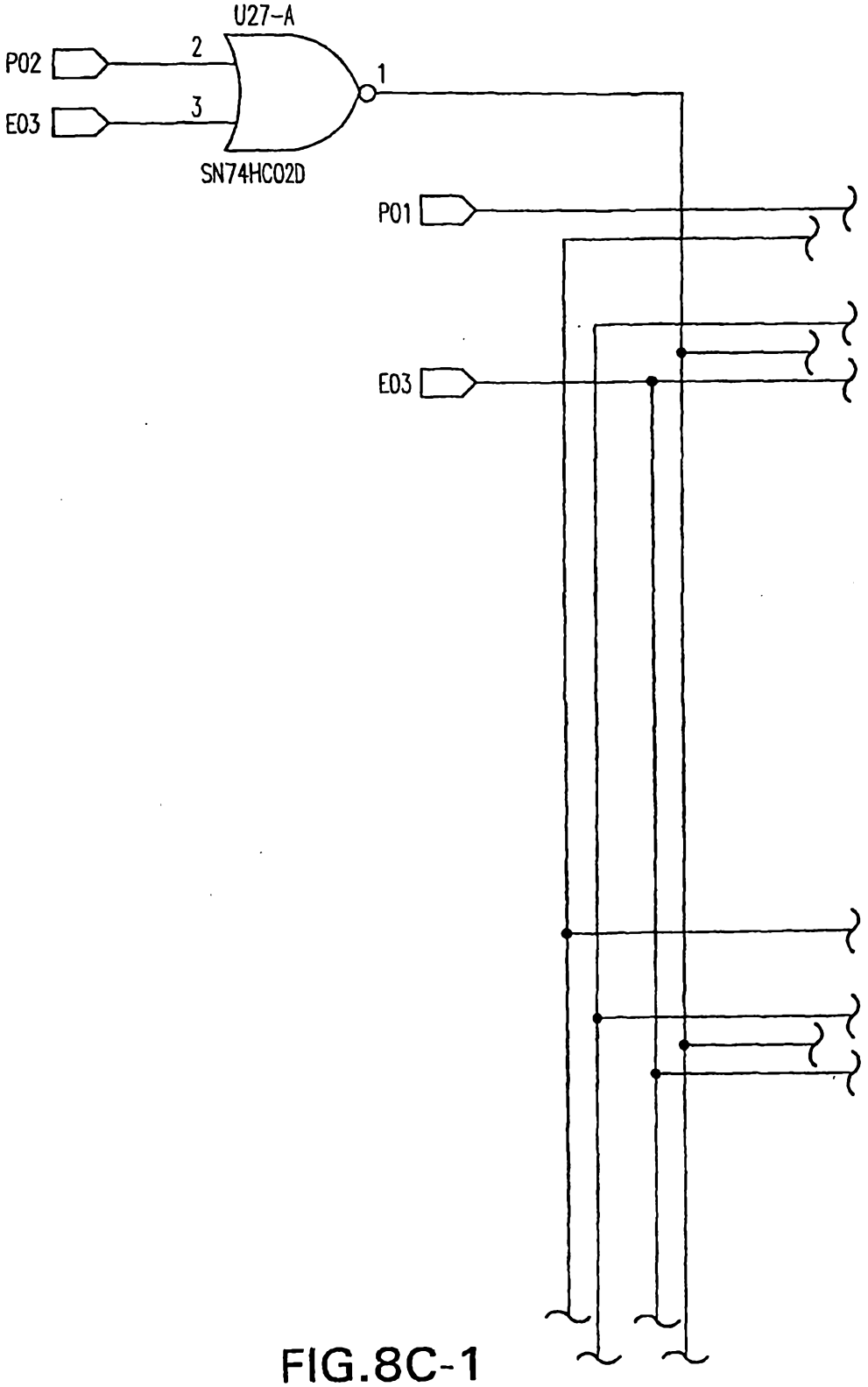


FIG.8C-1

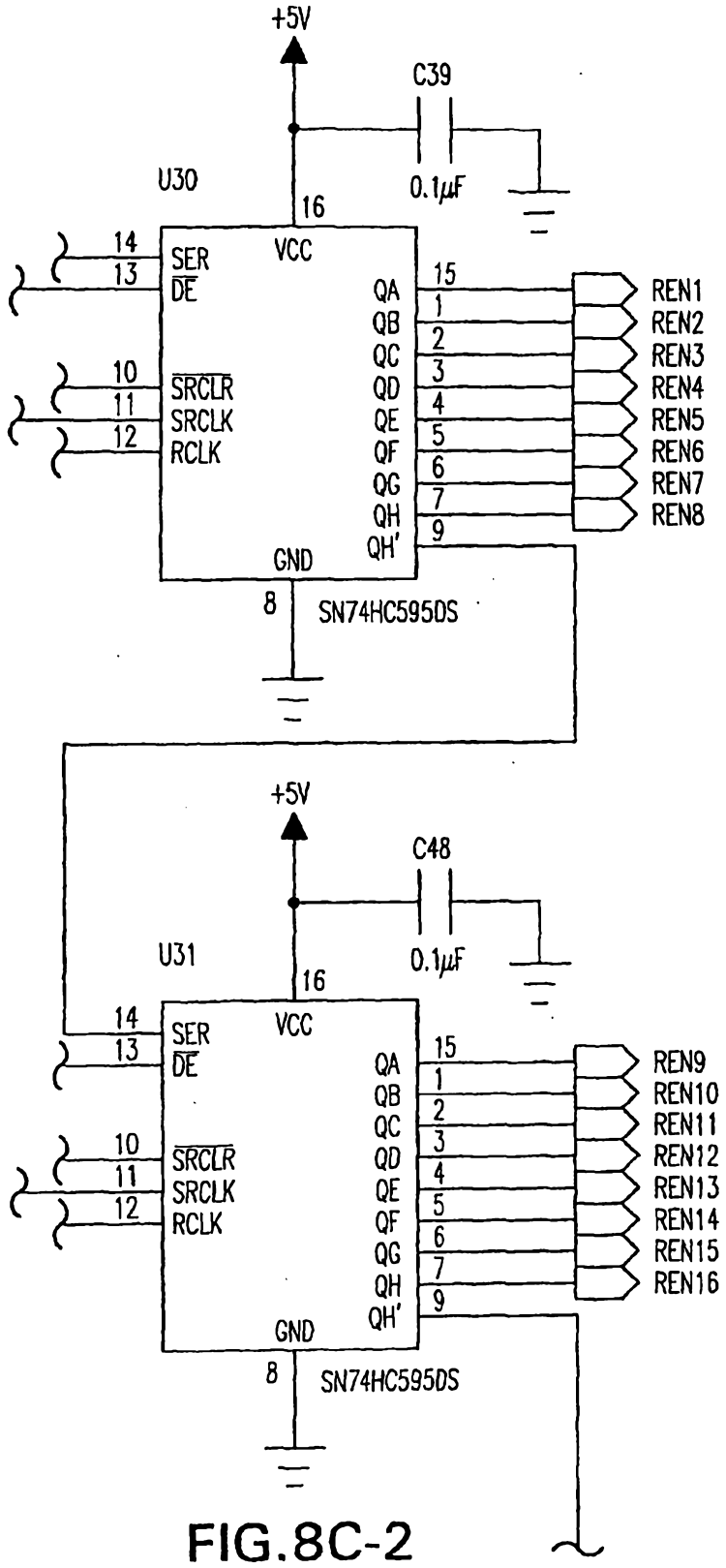

+5V
C39
0.1μF
U30
16
VCC
14
SER
13
DE
10
SRCLR
11
SRCLK
12
RCLK
QA
QB
QC
QD
QE
QF
QG
QH
QH'
15
1
2
3
4
5
6
7
9
REN1
REN2
REN3
REN4
REN5
REN6
REN7
REN8
GND
8
SN74HC595DS
+5V
C48
0.1μF
U31
16
VCC
14
SER
13
DE
10
SRCLR
11
SRCLK
12
RCLK
QA
QB
QC
QD
QE
QF
QG
QH
QH'
15
1
2
3
4
5
6
7
9
REN9
REN10
REN11
REN12
REN13
REN14
REN15
REN16
GND
8
SN74HC595DS

FIG.8C-2

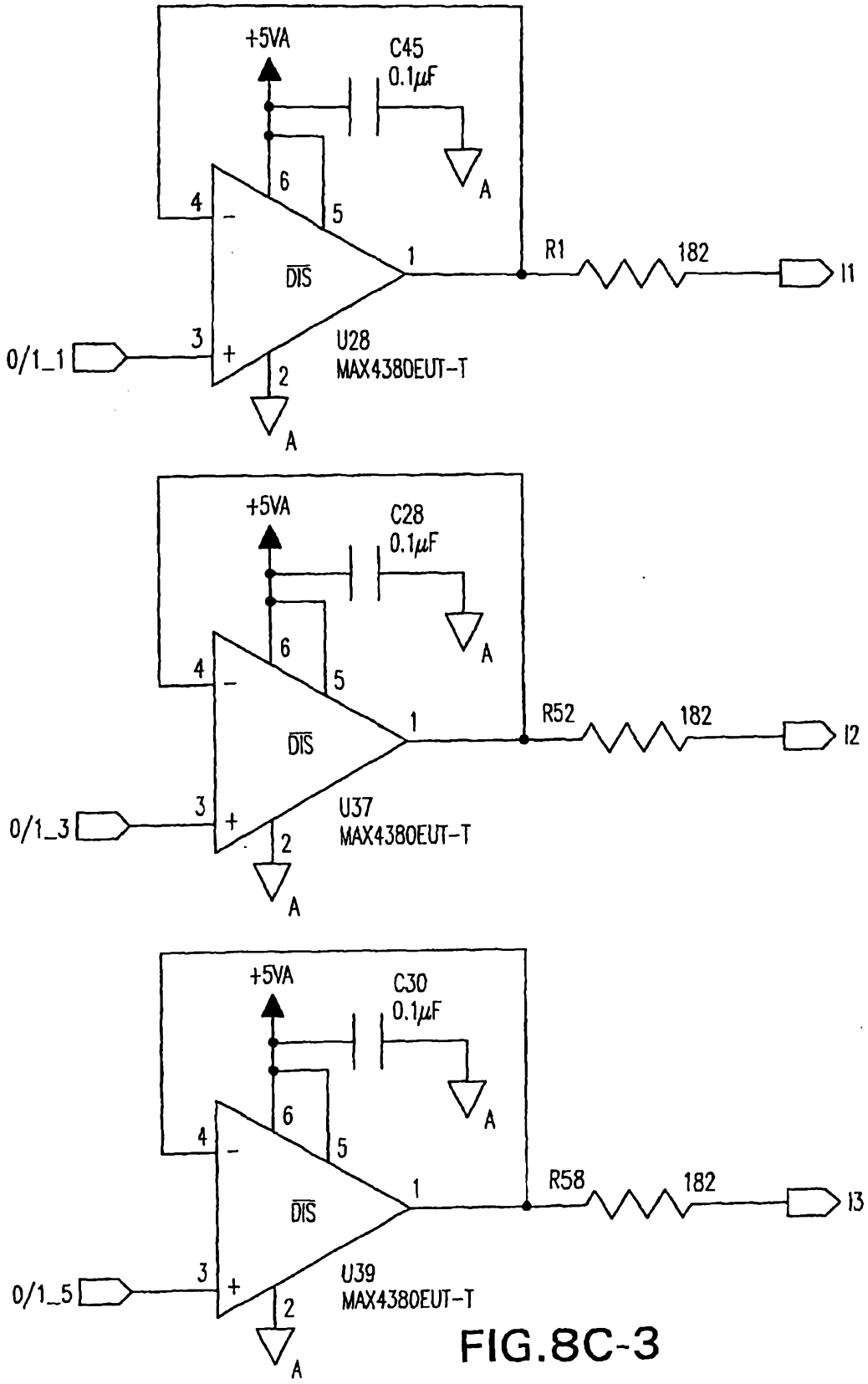

+5VA
C45
0.1µF
A
6
5
4
1
R1
182
I1
DIS
U28
MAX4380EUT-T
0/1_1
3
2
A
+5VA
C28
0.1µF
A
6
5
4
1
R52
182
I2
DIS
U37
MAX4380EUT-T
0/1_3
3
2
A
+5VA
C30
0.1µF
A
6
5
4
1
R58
182
I3
DIS
U39
MAX4380EUT-T
0/1_5
3
2
A

FIG.8C-3

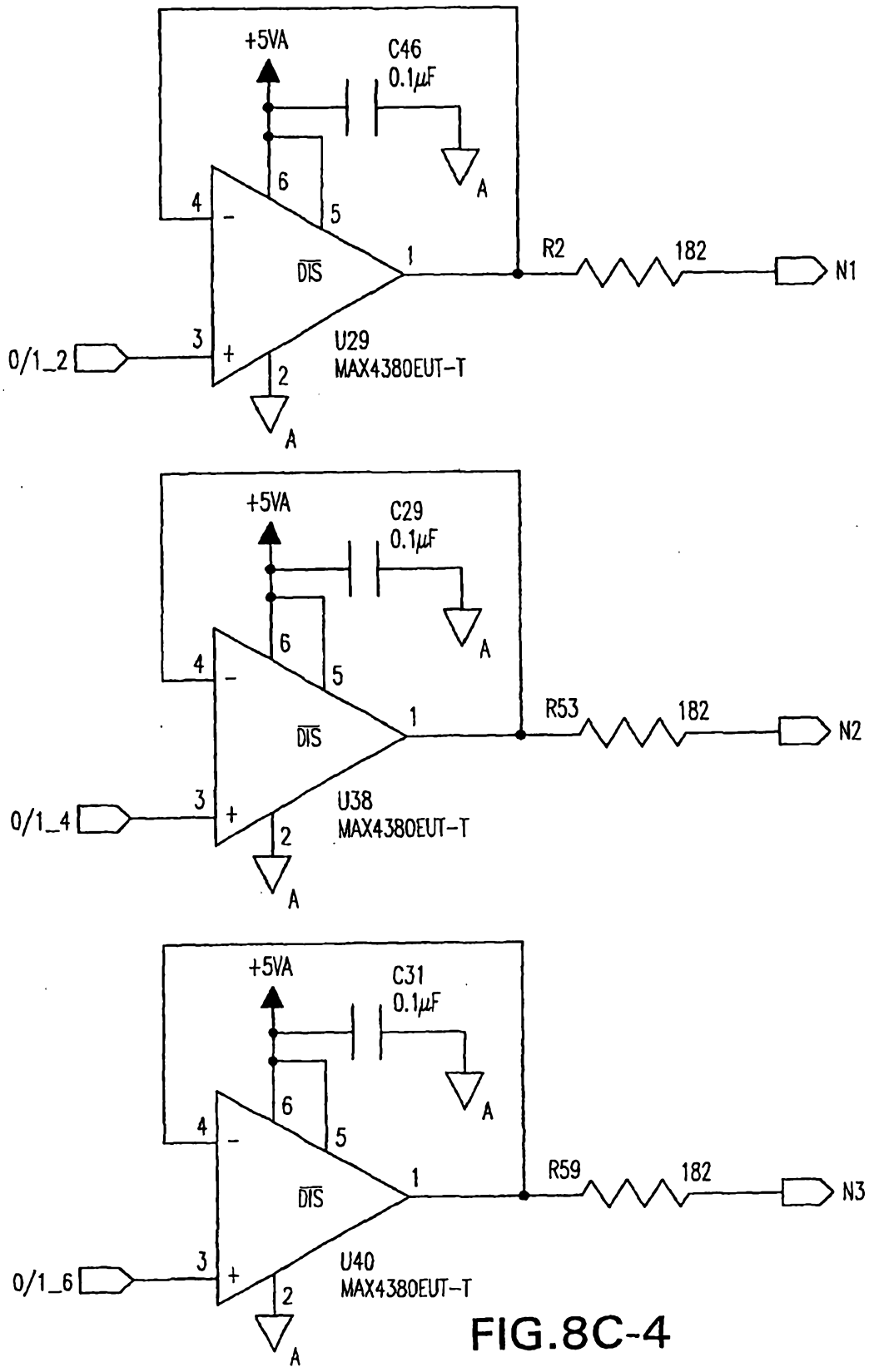

+5VA
C46
0.1μF
A
6
5
4
-
DIS
1
R2
182
N1
3
+
0/1_2
U29
MAX4380EUT-T
2
A
+5VA
C29
0.1μF
A
6
5
4
-
DIS
1
R53
182
N2
3
+
0/1_4
U38
MAX4380EUT-T
2
A
+5VA
C31
0.1μF
A
6
5
4
-
DIS
1
R59
182
N3
3
+
0/1_6
U40
MAX4380EUT-T
2
A

FIG.8C-4

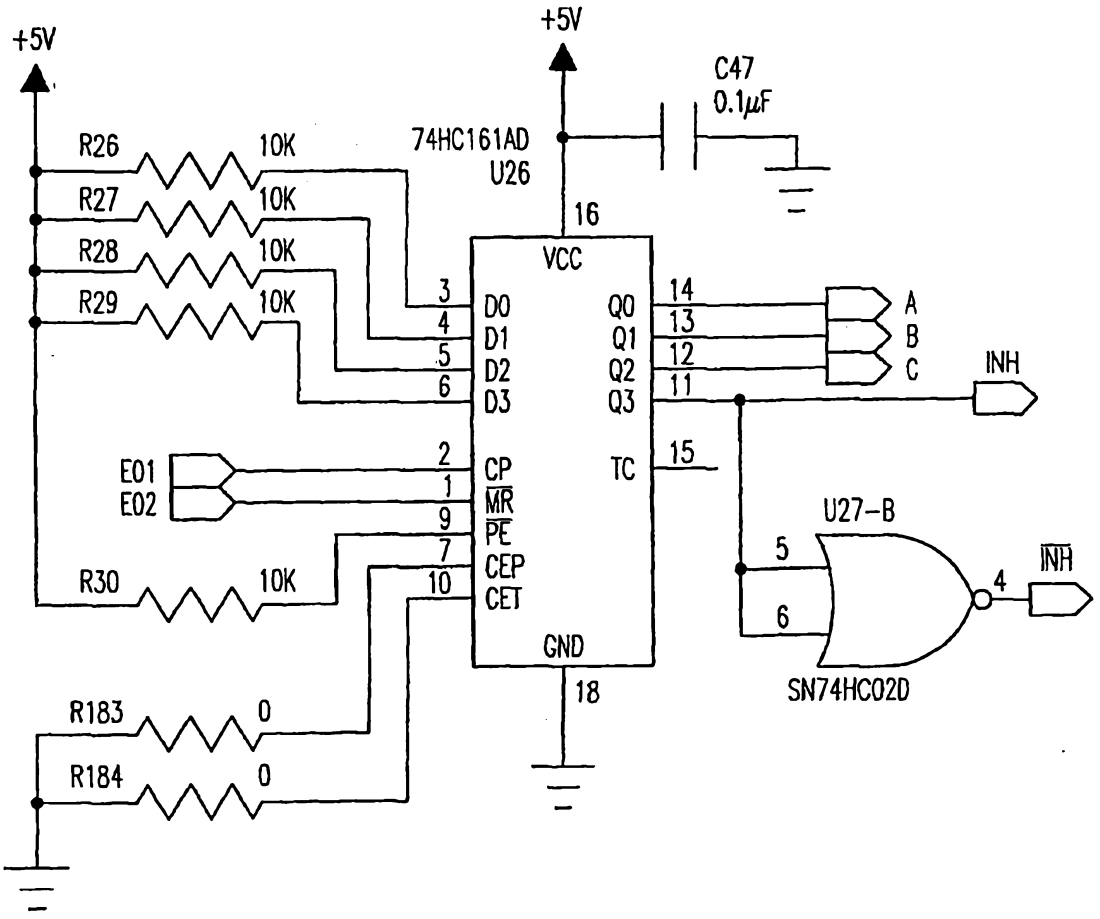

+5V
+5V
C47
0.1μF
74HC161AD
U26
R26
10K
R27
10K
R28
10K
R29
10K
16
VCC
3
D0
4
D1
5
D2
6
D3
Q0
Q1
Q2
Q3
14
13
12
11
A
B
C
INH
E01
E02
2
CP
1
MR
9
PE
7
CEP
10
CET
TC
15
U27-B
5
6
4
INH
SN74HC02D
R30
10K
R183
0
R184
0
GND
18

FIG.8C-5

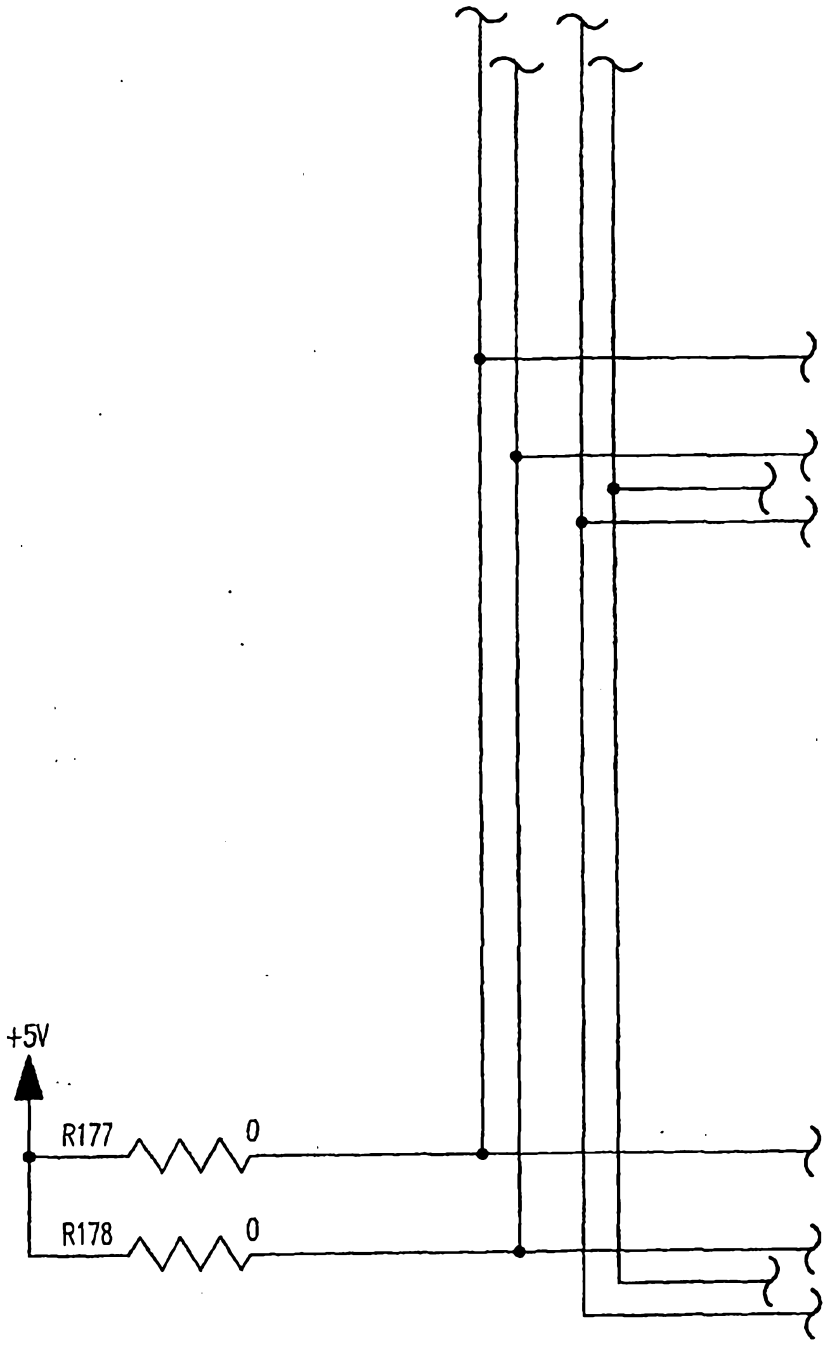


FIG.8C-6

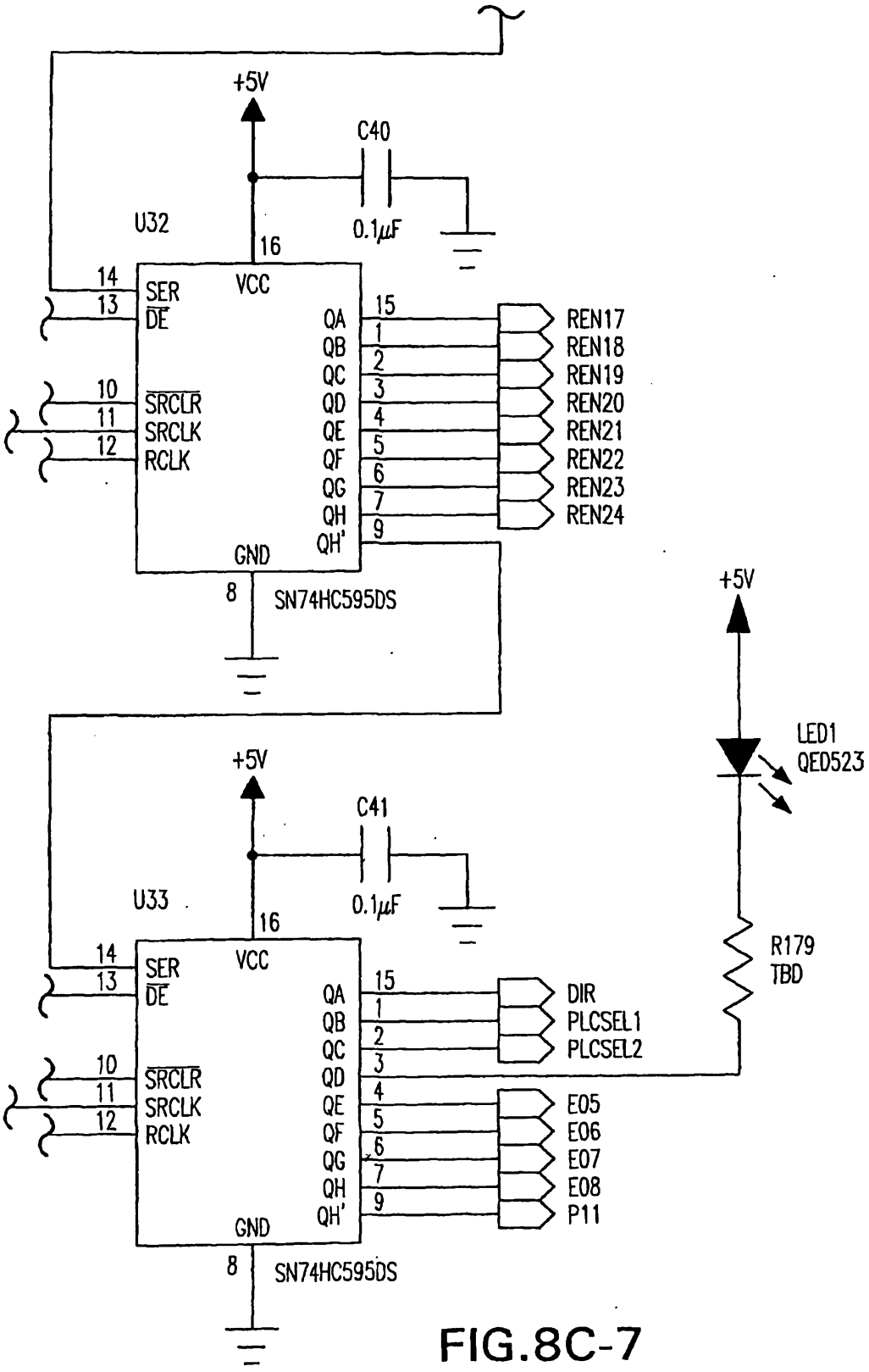

+5V
C40
0.1µF
U32
16
14
SER
13
DE
VCC
QA
QB
QC
QD
QE
QF
QG
QH
QH'
15
1
2
3
4
5
6
7
9
REN17
REN18
REN19
REN20
REN21
REN22
REN23
REN24
10
SRCLR
11
SRCLK
12
RCLK
GND
8
SN74HC595DS
+5V
LED1
QED523
+5V
C41
0.1µF
U33
16
14
SER
13
DE
VCC
QA
QB
QC
QD
QE
QF
QG
QH
QH'
15
1
2
3
4
5
6
7
9
DIR
PLCSEL1
PLCSEL2
E05
E06
E07
E08
P11
10
SRCLR
11
SRCLK
12
RCLK
GND
8
SN74HC595DS
R179
TBD

FIG.8C-7

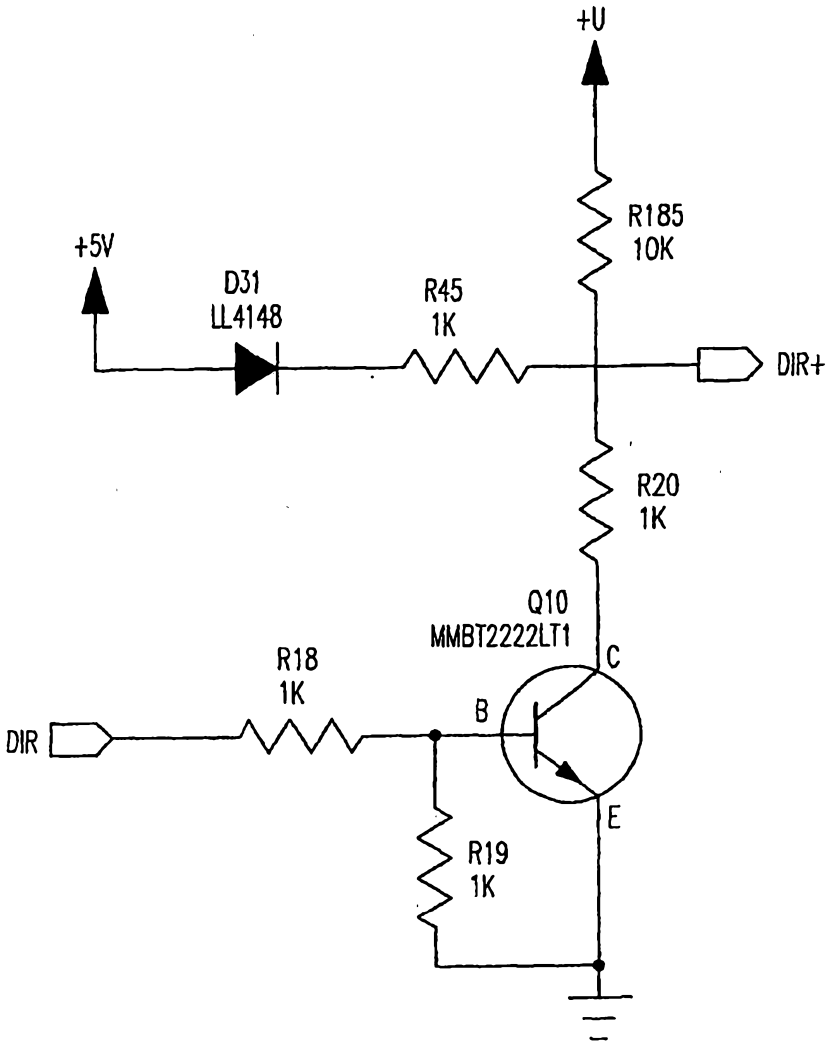

+U
R185
10K
+5V
D31
LL4148
R45
1K
DIR+
R20
1K
Q10
MMBT2222LT1
C
R18
1K
B
DIR
E
R19
1K

FIG.8C-8

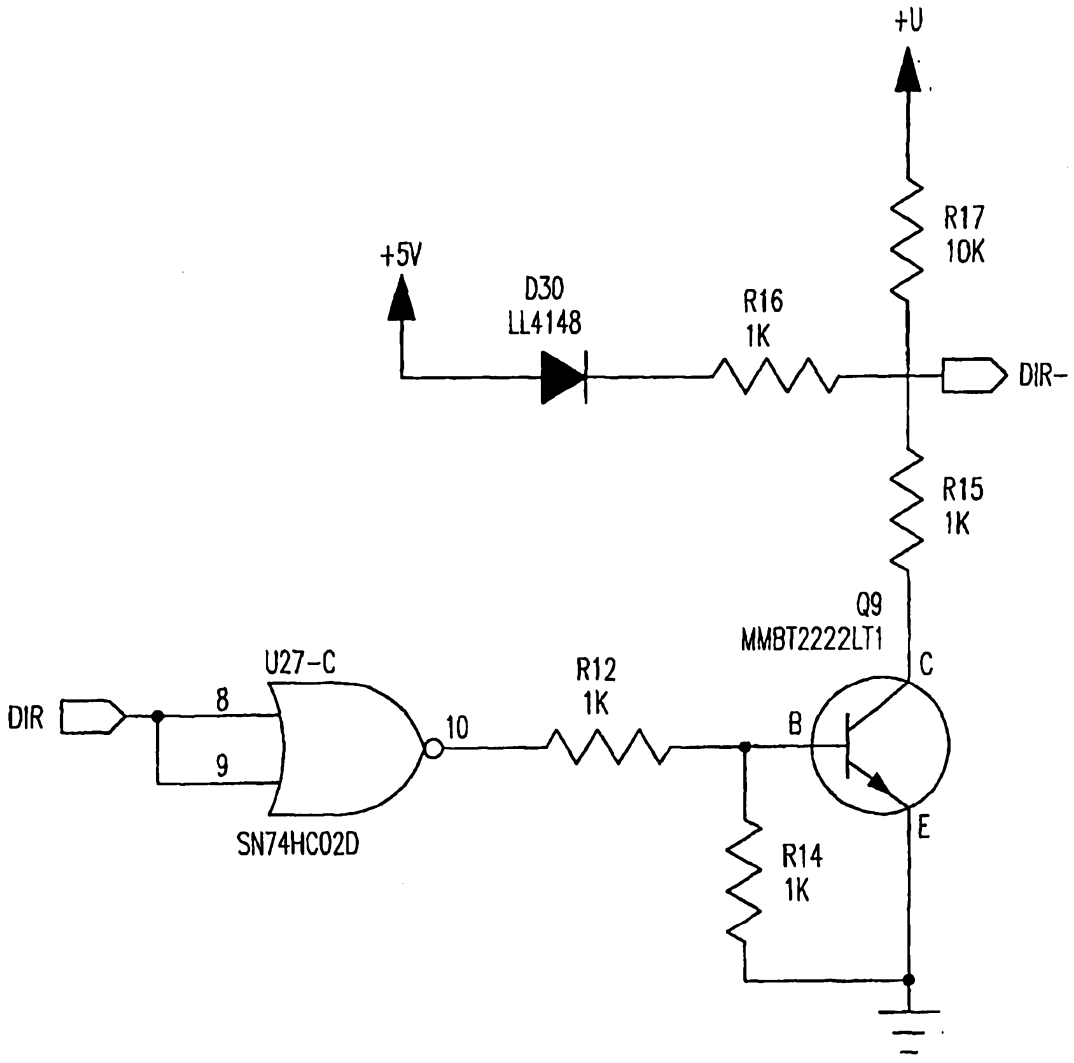

+U
R17
10K
+5V
D30
LL4148
R16
1K
DIR-
R15
1K
Q9
MMBT2222LT1
C
U27-C
R12
1K
DIR
8
9
10
B
E
SN74HC02D
R14
1K

FIG.8C-9

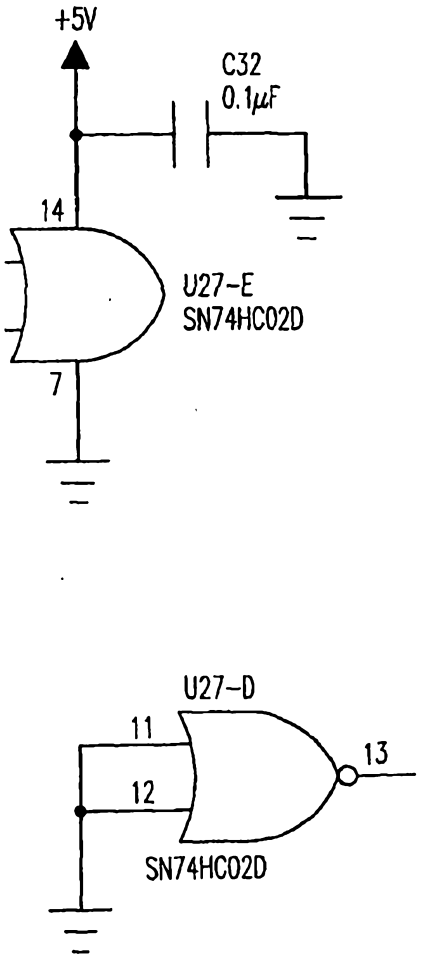


FIG.8C-10

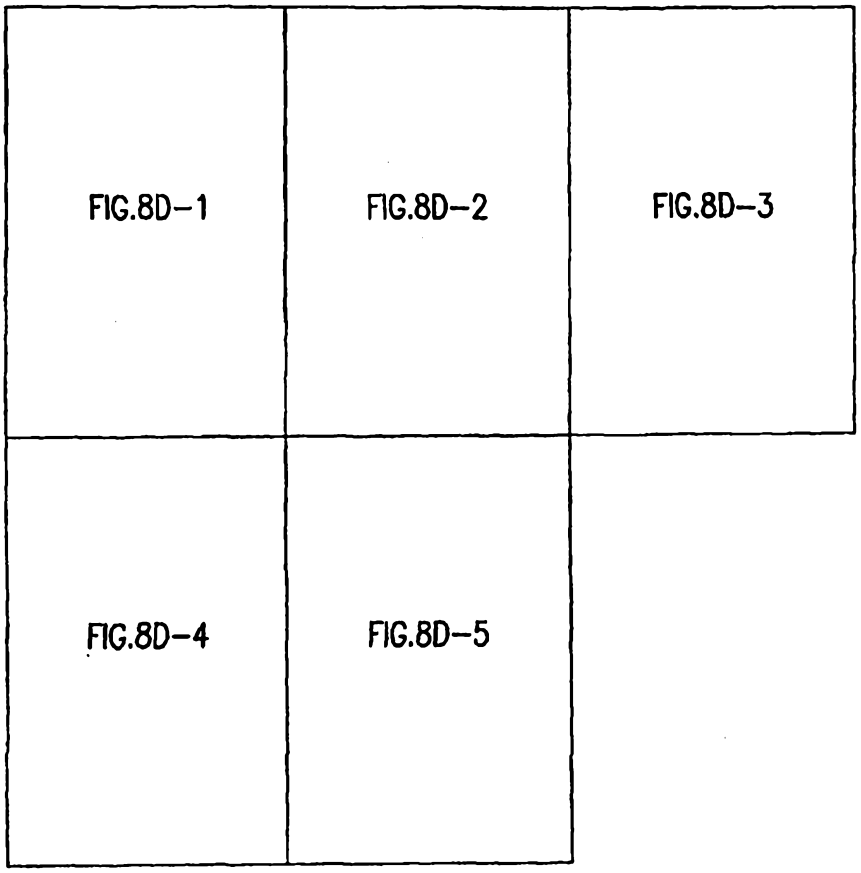
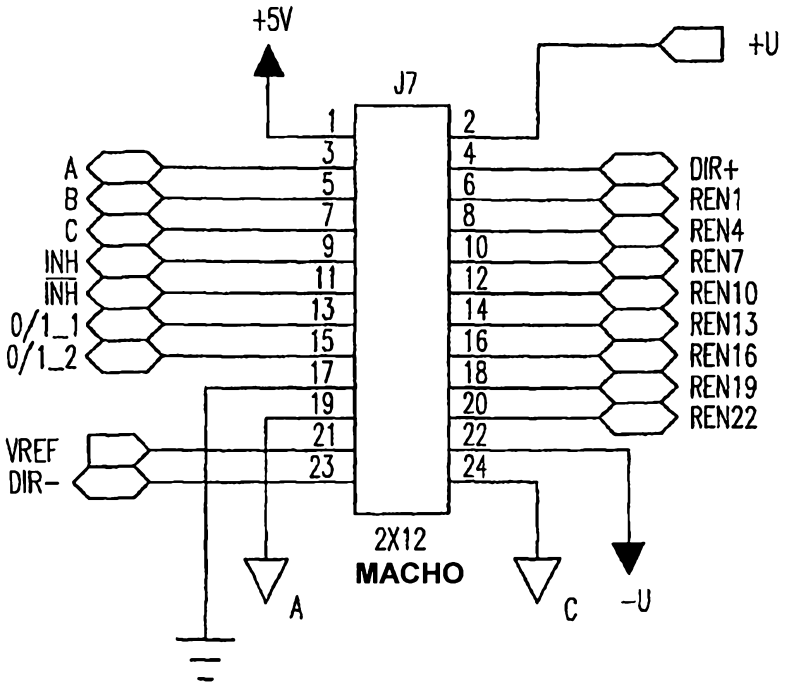


FIG.8D

FORA PARA PLACA RJ FASE A



FORA PARA PLACA RJ FASE B

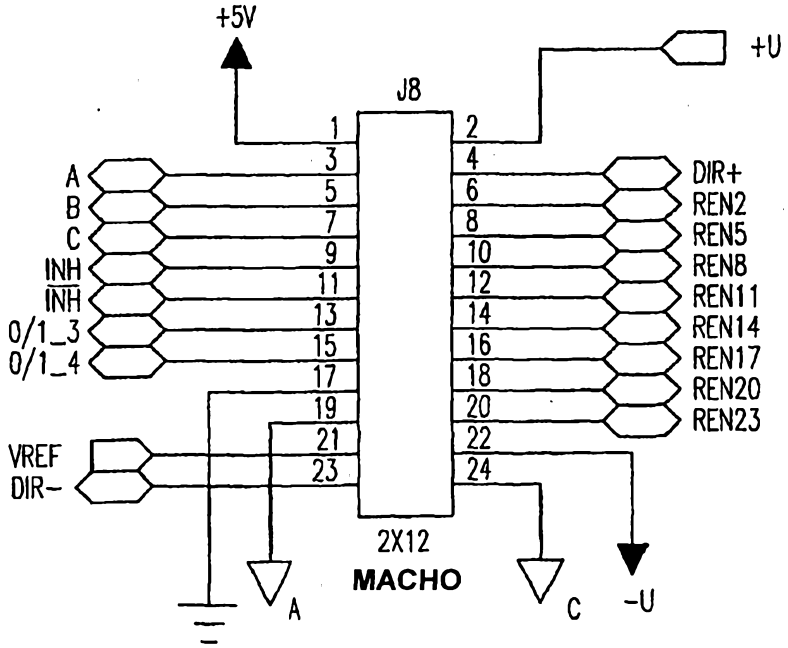
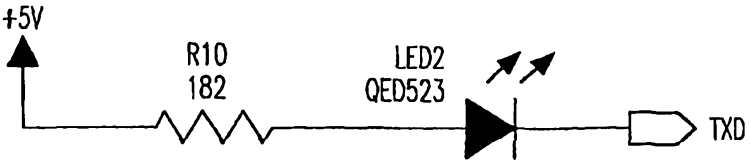
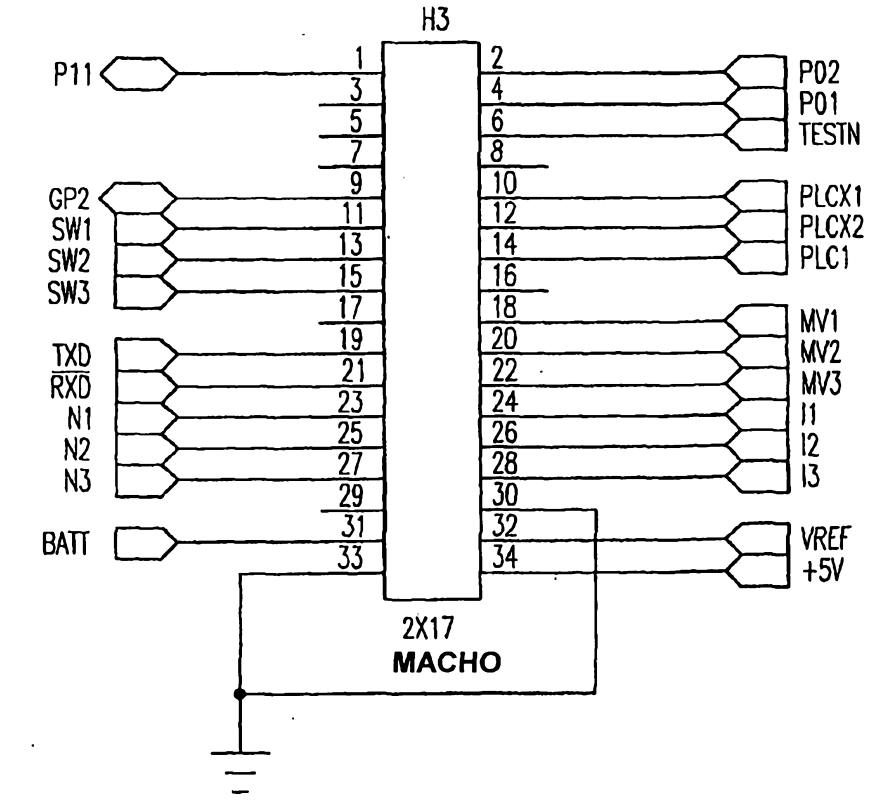


FIG.8D-1



DE PLACA DE MOSTRADOR



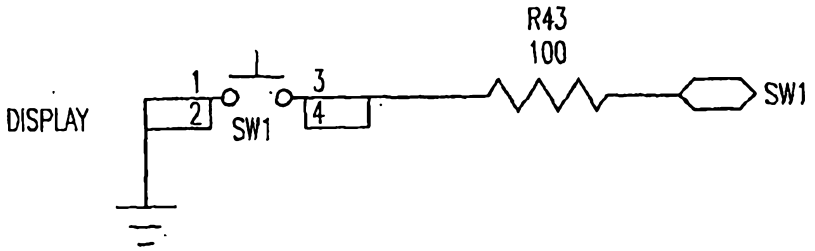


FIG.8D-2

PARA PLACA DE OPÇÃO RJ11

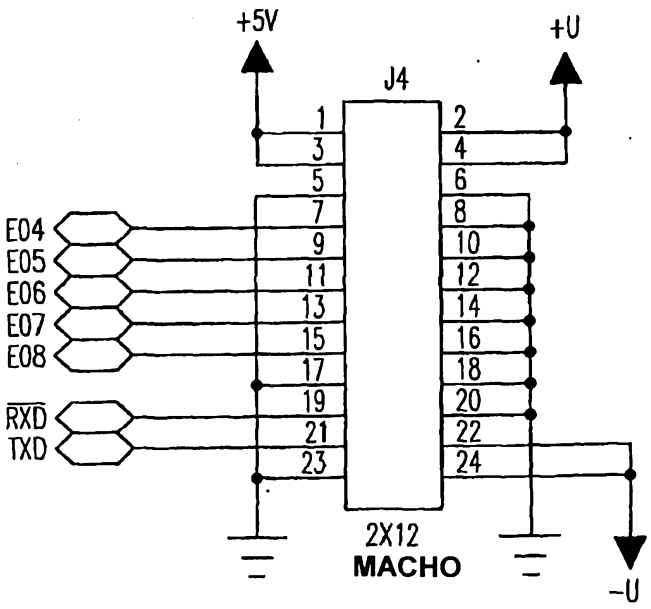


FIG.8D-3

FORA PARA PLACA RJ FASE C

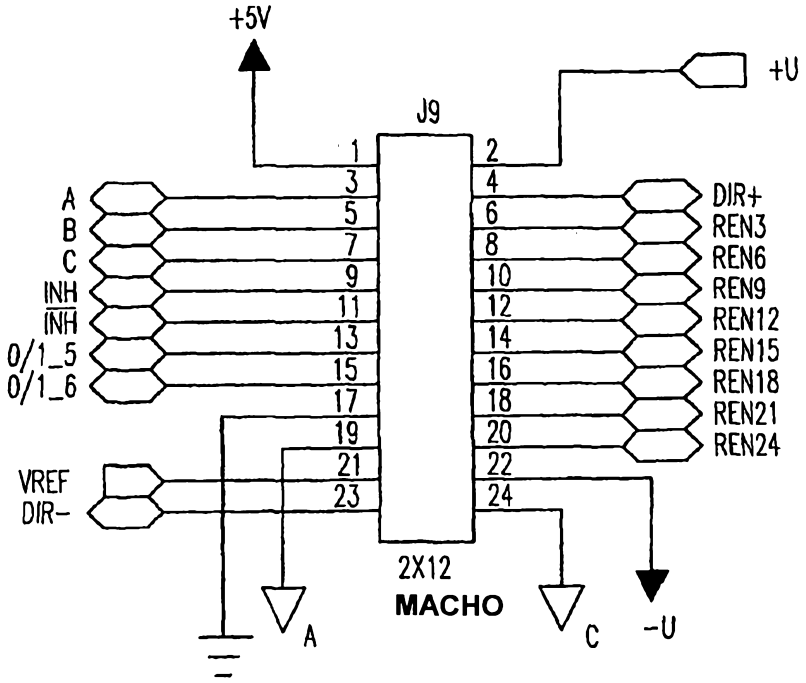


FIG.8D-4

DE PLACA DE MOSTRADOR

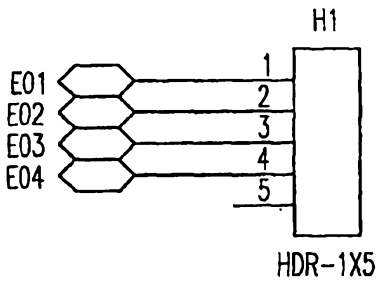


FIG.8D-5

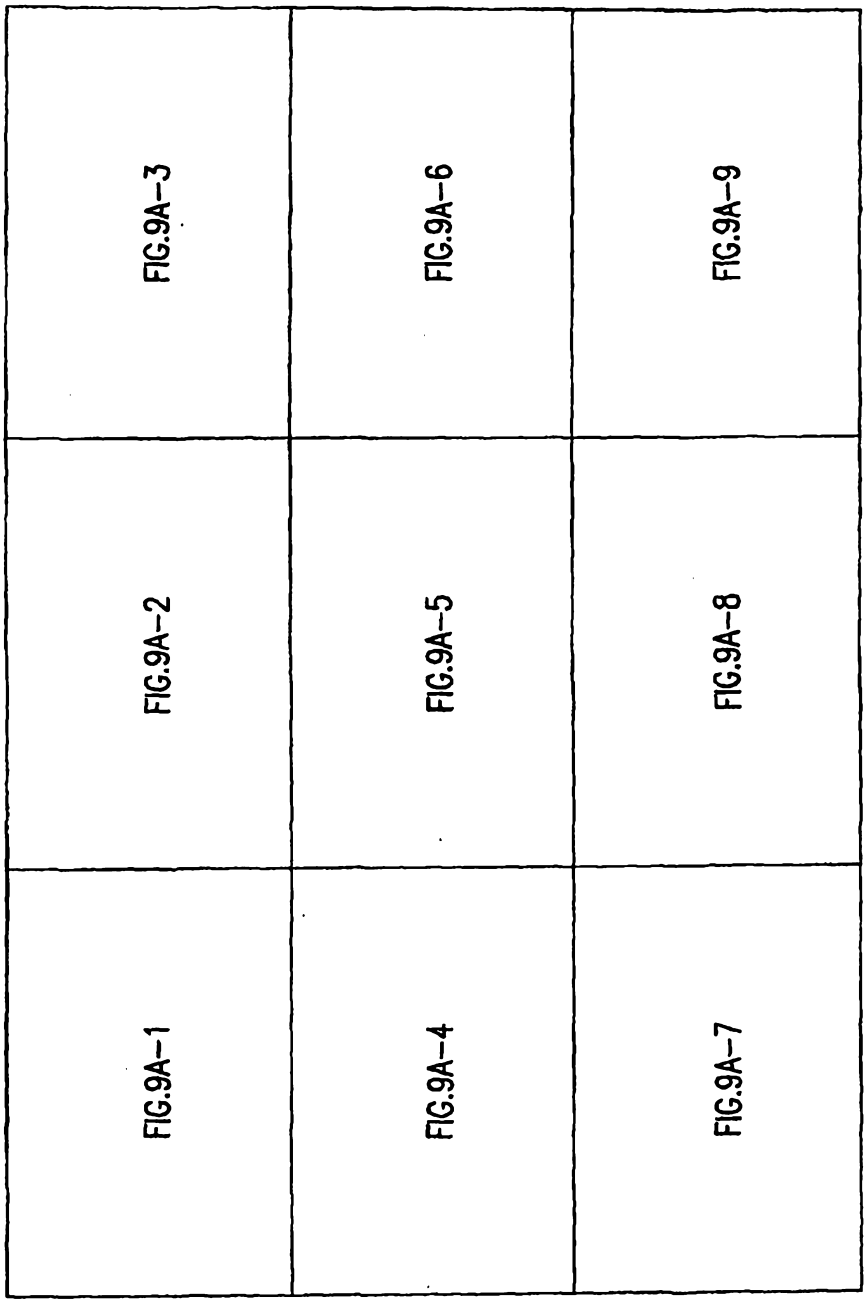


FIG.9A-3
FIG.9A-6
FIG.9A-9
FIG.9A-2
FIG.9A-5
FIG.9A-8
FIG.9A-1
FIG.9A-4
FIG.9A-7

FIG.9A

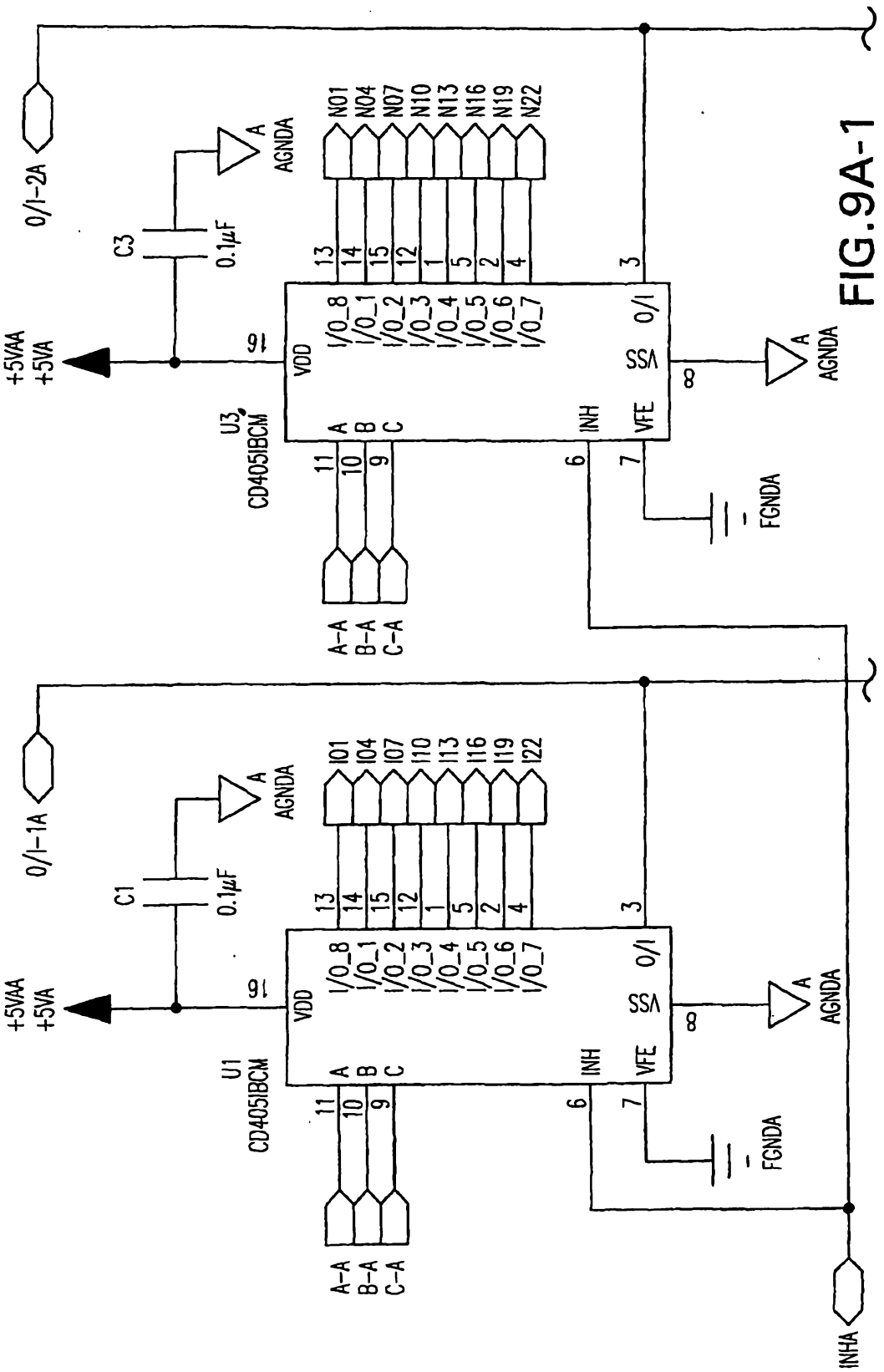

O/I-2A
C3
0.1µF
AGNDA
+5VAA
+5VA
U3
CD4051BCM
VDD
I/O_8
I/O_1
I/O_2
I/O_3
I/O_4
I/O_5
I/O_6
I/O_7
N01
N04
N07
N10
N13
N16
N19
N22
O/I
VSS
A
B
C
INH
VEE
A-A
B-A
C-A
FGNDA
O/I-1A
C1
0.1µF
I01
I04
I07
I10
I13
I16
I19
I22
U1
CD4051BCM
INHA

FIG.9A-1

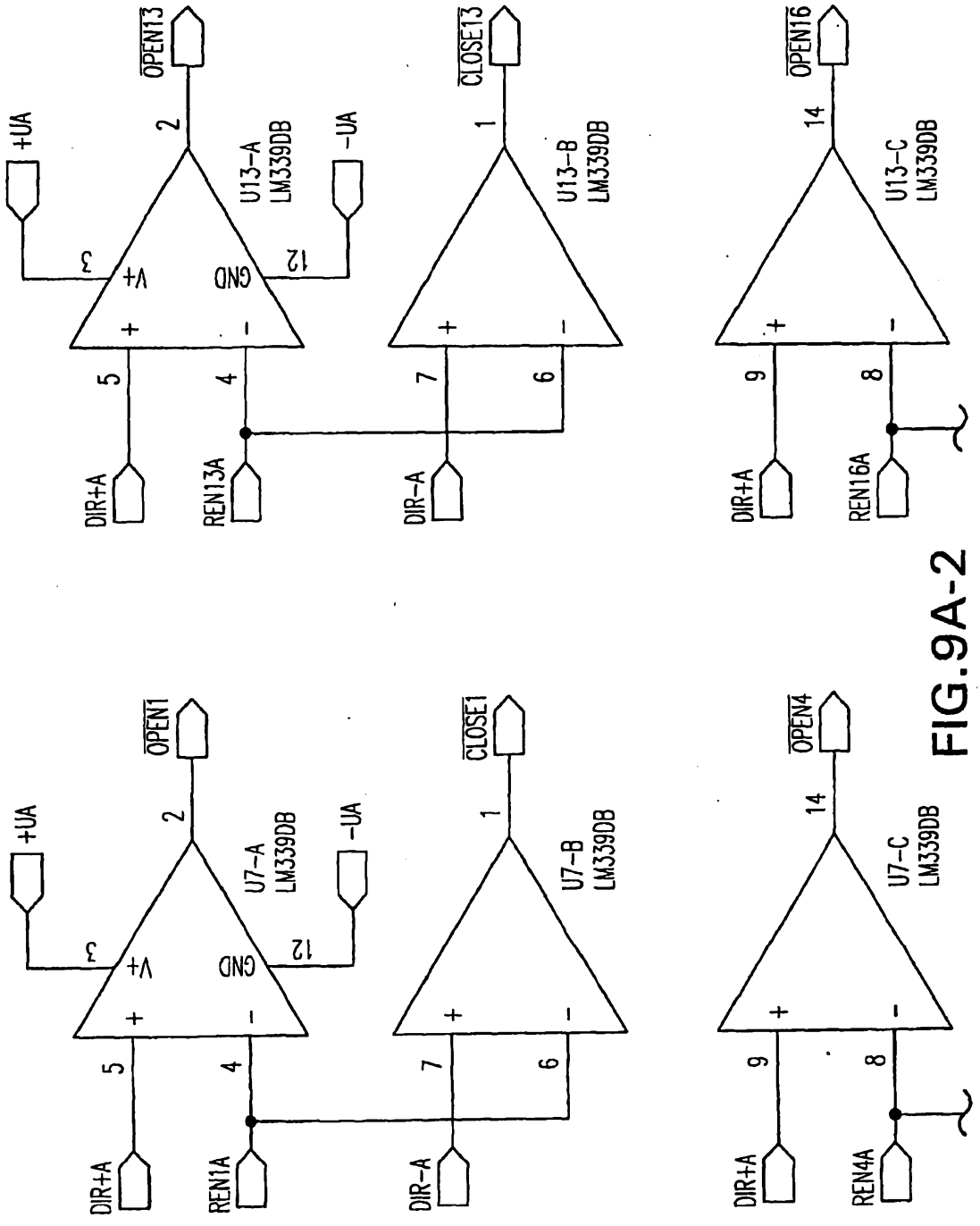

+UA
OPEN13
2
U13-A
LM339DB
-UA
3
V+
GND
12
+
-
5
4
DIR+A
REN13A
CLOSE13
1
U13-B
LM339DB
7
6
DIR-A
OPEN16
14
U13-C
LM339DB
9
8
DIR+A
REN16A
+UA
OPEN1
2
U7-A
LM339DB
-UA
3
V+
GND
12
+
-
5
4
DIR+A
REN1A
CLOSE1
1
U7-B
LM339DB
7
6
DIR-A
OPEN4
14
U7-C
LM339DB
9
8
DIR+A
REN4A

FIG.9A-2

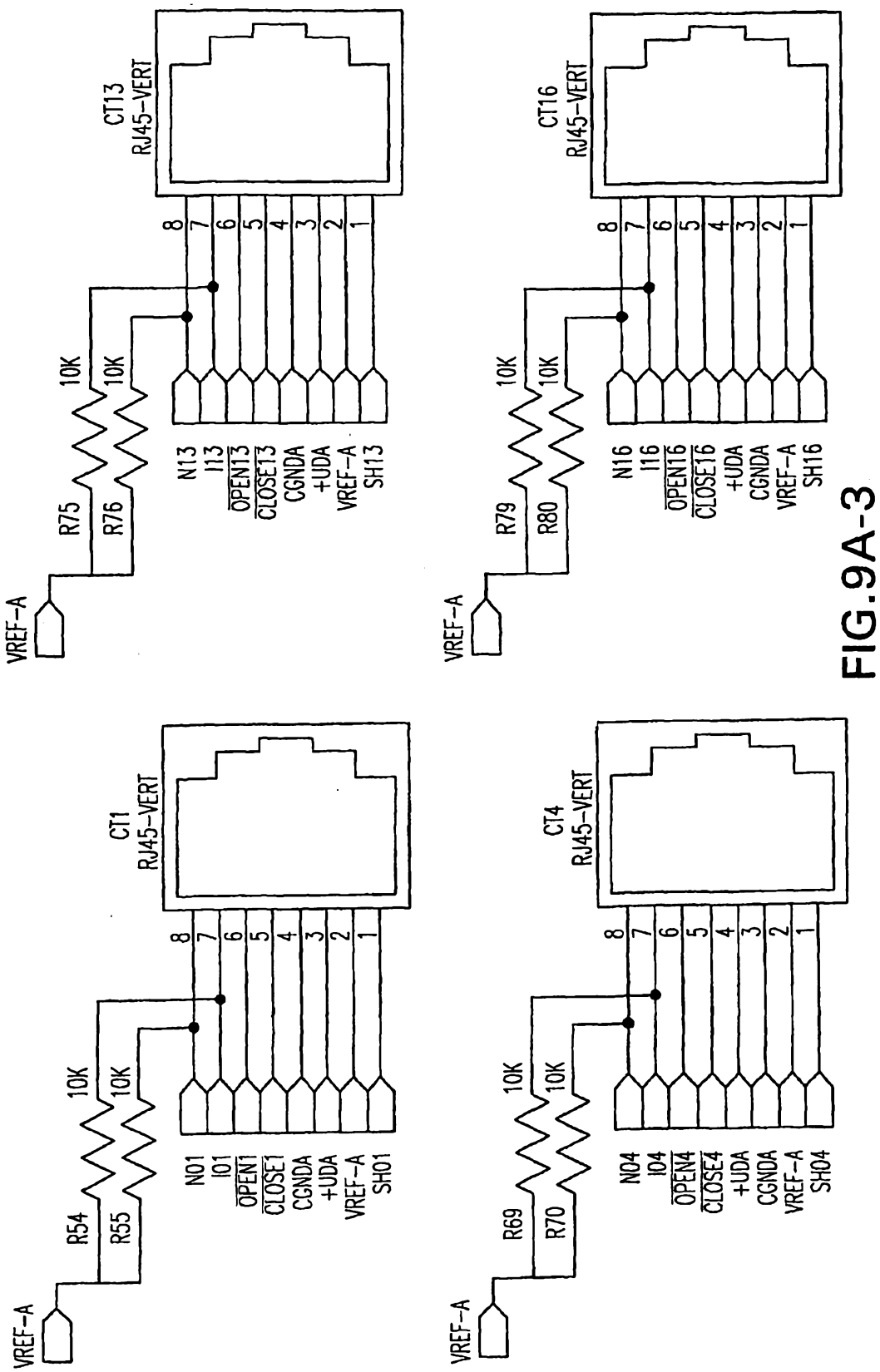

CT13
RJ45-VERT
8 7 6 5 4 3 2 1
N13
I13
OPEN13
CLOSE13
CGNDA
+UDA
VREF-A
SH13
10K
10K
R75
R76
VREF-A
CT16
RJ45-VERT
8 7 6 5 4 3 2 1
N16
I16
OPEN16
CLOSE16
+UDA
CGNDA
VREF-A
SH16
10K
10K
R79
R80
VREF-A
CT1
RJ45-VERT
8 7 6 5 4 3 2 1
N01
I01
OPEN1
CLOSE1
CGNDA
+UDA
VREF-A
SH01
10K
10K
R54
R55
VREF-A
CT4
RJ45-VERT
8 7 6 5 4 3 2 1
N04
I04
OPEN4
CLOSE4
+UDA
CGNDA
VREF-A
SH04
10K
10K
R69
R70
VREF-A

FIG.9A-3

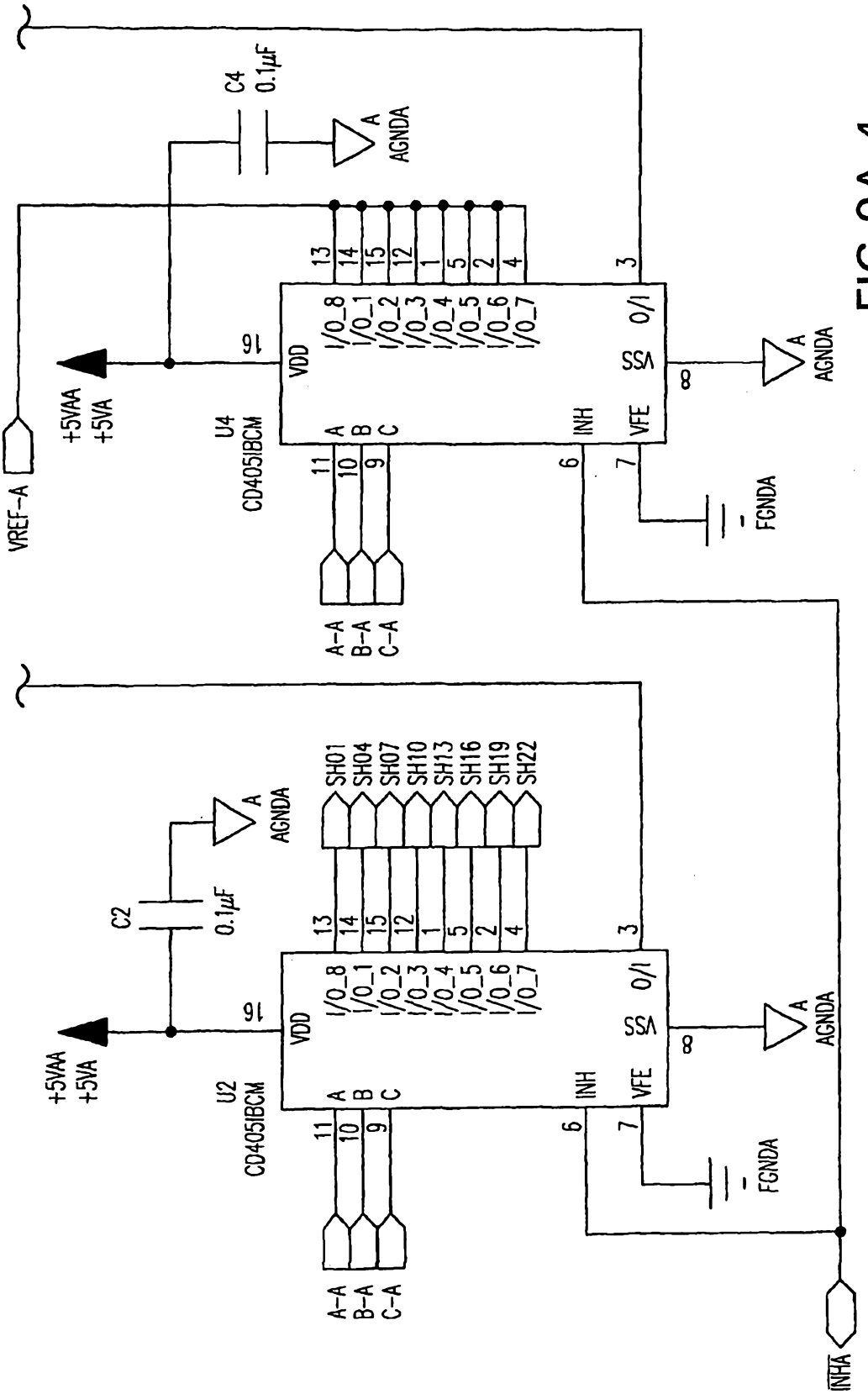

C4
0.1µF
AGNDA
VREF-A
+5VAA
+5VA
U4
CD4051BCM
VDD
I/O_8
I/O_1
I/O_2
I/O_3
I/O_4
I/O_5
I/O_6
I/O_7
I/O
VSS
INH
VEE
A
B
C
A-A
B-A
C-A
FGNDA
C2
0.1µF
U2
CD4051BCM
SH01
SH04
SH07
SH10
SH13
SH16
SH19
SH22
INHA

FIG.9A-4

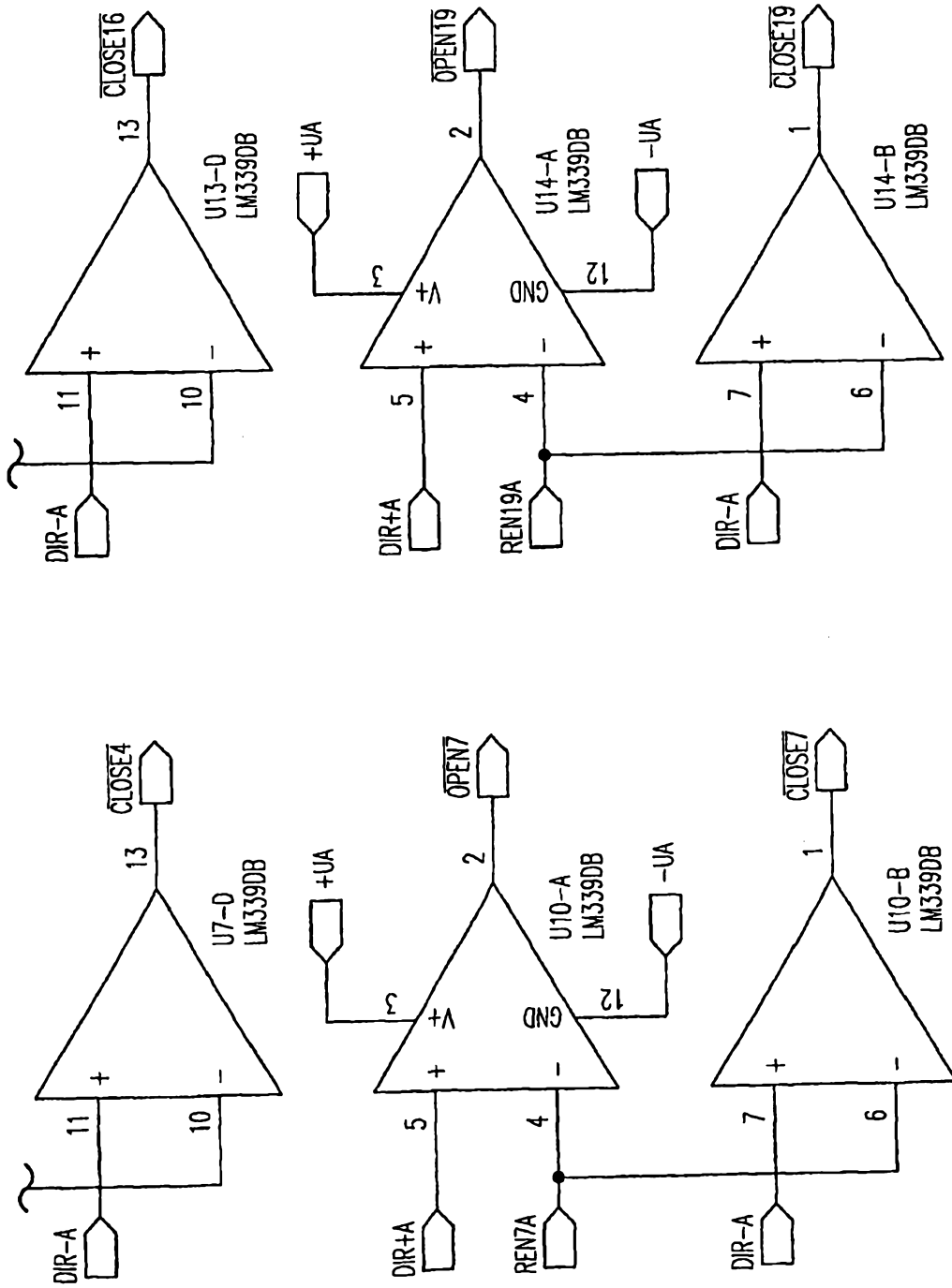

CLOSE16
13
U13-D
LM339DB
+
-
11
10
DIR-A
OPEN19
2
+UA
3
V+
GND
12
-UA
U14-A
LM339DB
5
4
DIR+A
REN19A
CLOSE19
1
U14-B
LM339DB
7
6
DIR-A
CLOSE4
13
U7-D
LM339DB
DIR-A
OPEN7
2
+UA
3
V+
GND
12
-UA
U10-A
LM339DB
5
4
DIR+A
REN7A
CLOSE7
1
U10-B
LM339DB
7
6
DIR-A

FIG.9A-5

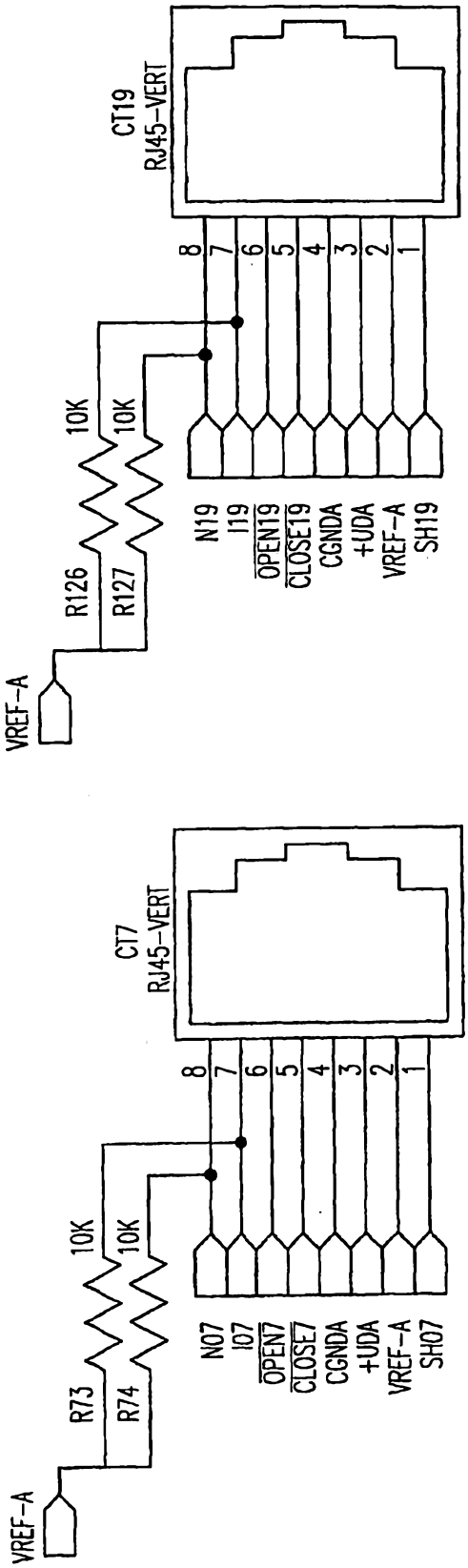

CT19
RJ45-VERT
8
7
6
5
4
3
2
1
10K
10K
R126
R127
VREF-A
N19
I19
OPEN19
CLOSE19
CGNDA
+UDA
VREF-A
SH19
CT7
RJ45-VERT
8
7
6
5
4
3
2
1
10K
10K
R73
R74
VREF-A
N07
I07
OPEN7
CLOSE7
CGNDA
+UDA
VREF-A
SH07

FIG.9A-6

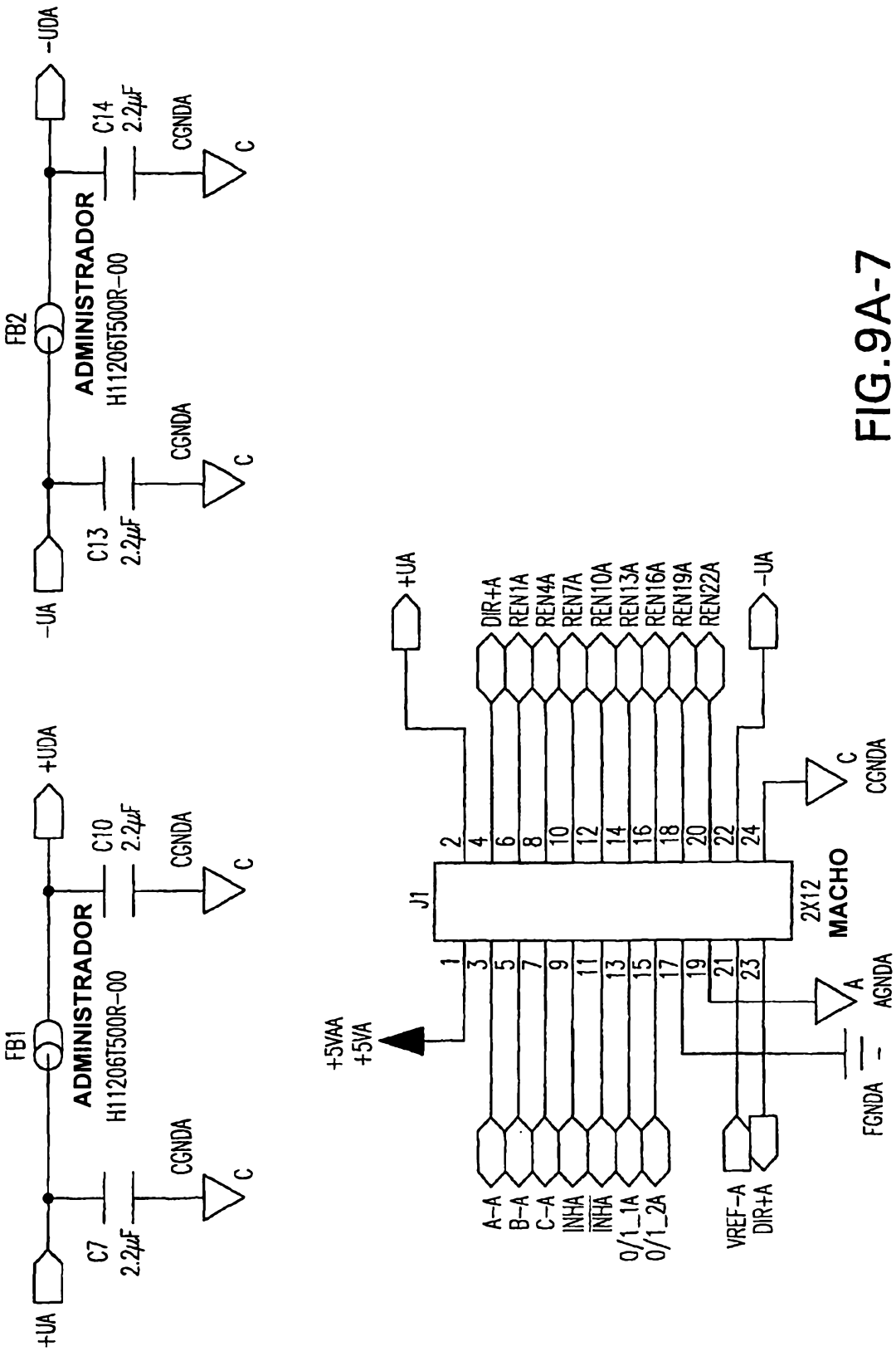

+UA
-UA
+UDA
-UDA
FB1
FB2
ADMINISTRADOR
H11206T500R-00
C7
2.2µF
C10
2.2µF
C13
2.2µF
C14
2.2µF
CGNDA
C
+5VAA
+5VA
J1
2X12
MACHO
A
AGNDA
FGNDA
DIR+A
REN1A
REN4A
REN7A
REN10A
REN13A
REN16A
REN19A
REN22A
A-A
B-A
C-A
INHA
INHA
0/1_1A
0/1_2A
VREF-A
DIR+A

FIG.9A-7

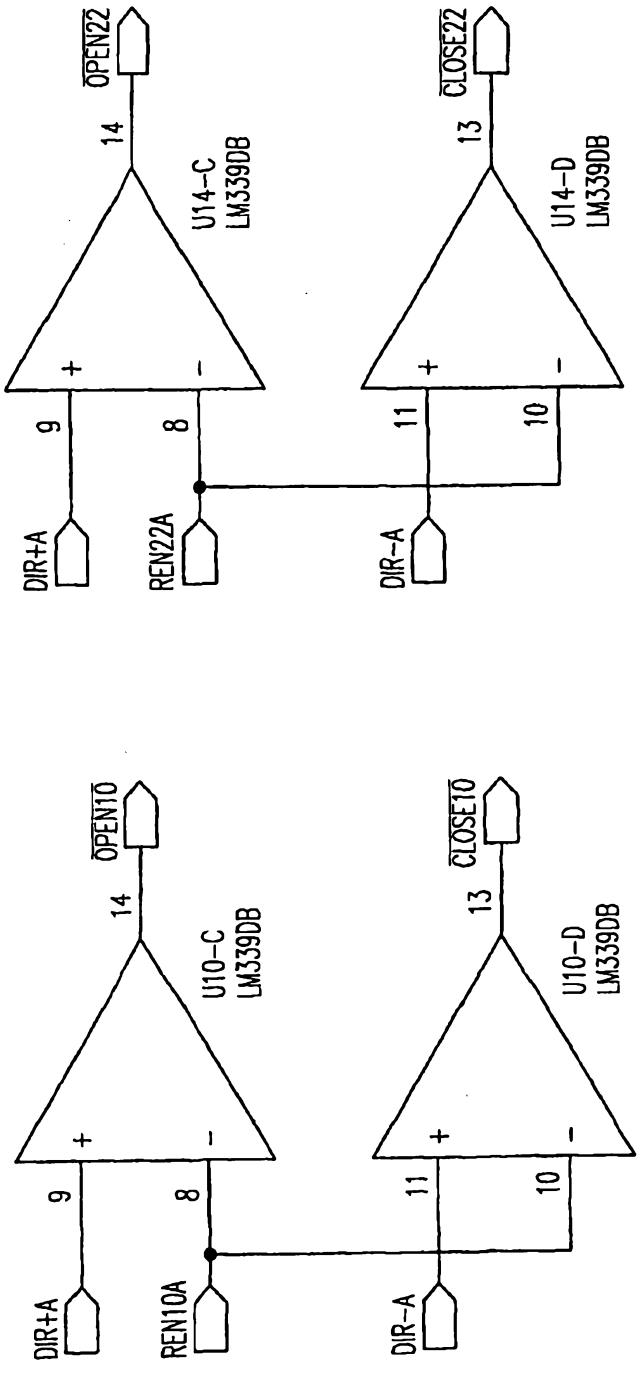

OPEN22
14
U14-C
LM339DB
+
-
9
8
DIR+A
REN22A
CLOSE22
13
U14-D
LM339DB
+
-
11
10
DIR-A
OPEN10
14
U10-C
LM339DB
+
-
9
8
DIR+A
REN10A
CLOSE10
13
U10-D
LM339DB
+
-
11
10
DIR-A

FIG.9A-8

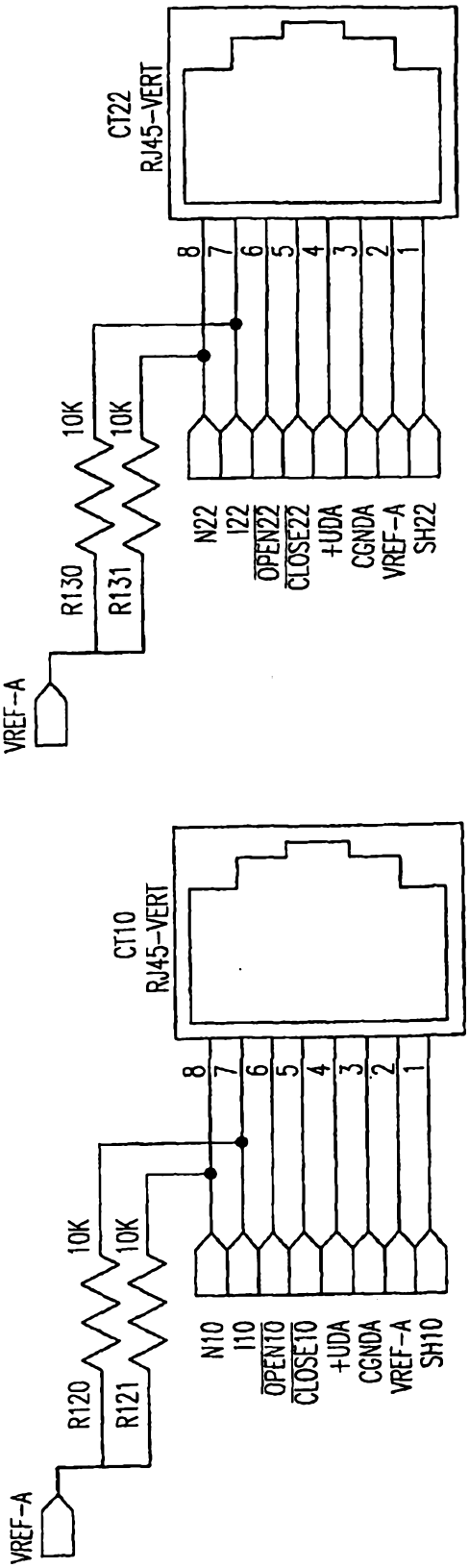

CT22
RJ45-VERT
8
7
6
5
4
3
2
1
N22
I22
OPEN22
CLOSE22
+UDA
CGNDA
VREF-A
SH22
10K
10K
R130
R131
VREF-A
CT10
RJ45-VERT
8
7
6
5
4
3
2
1
N10
I10
OPEN10
CLOSE10
+UDA
CGNDA
VREF-A
SH10
10K
10K
R120
R121
VREF-A

FIG.9A-9

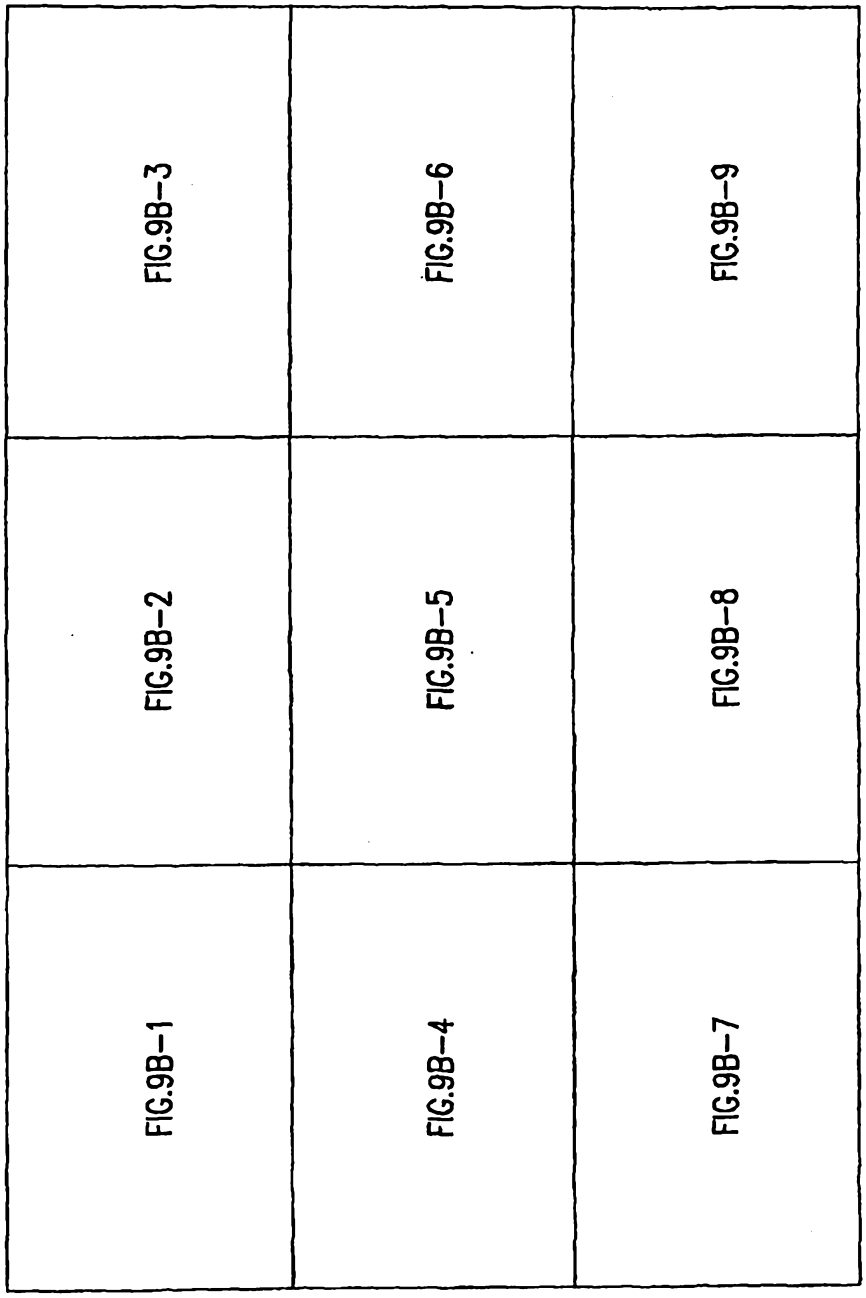


FIG.9B-3
FIG.9B-6
FIG.9B-9
FIG.9B-2
FIG.9B-5
FIG.9B-8
FIG.9B-1
FIG.9B-4
FIG.9B-7

FIG.9B

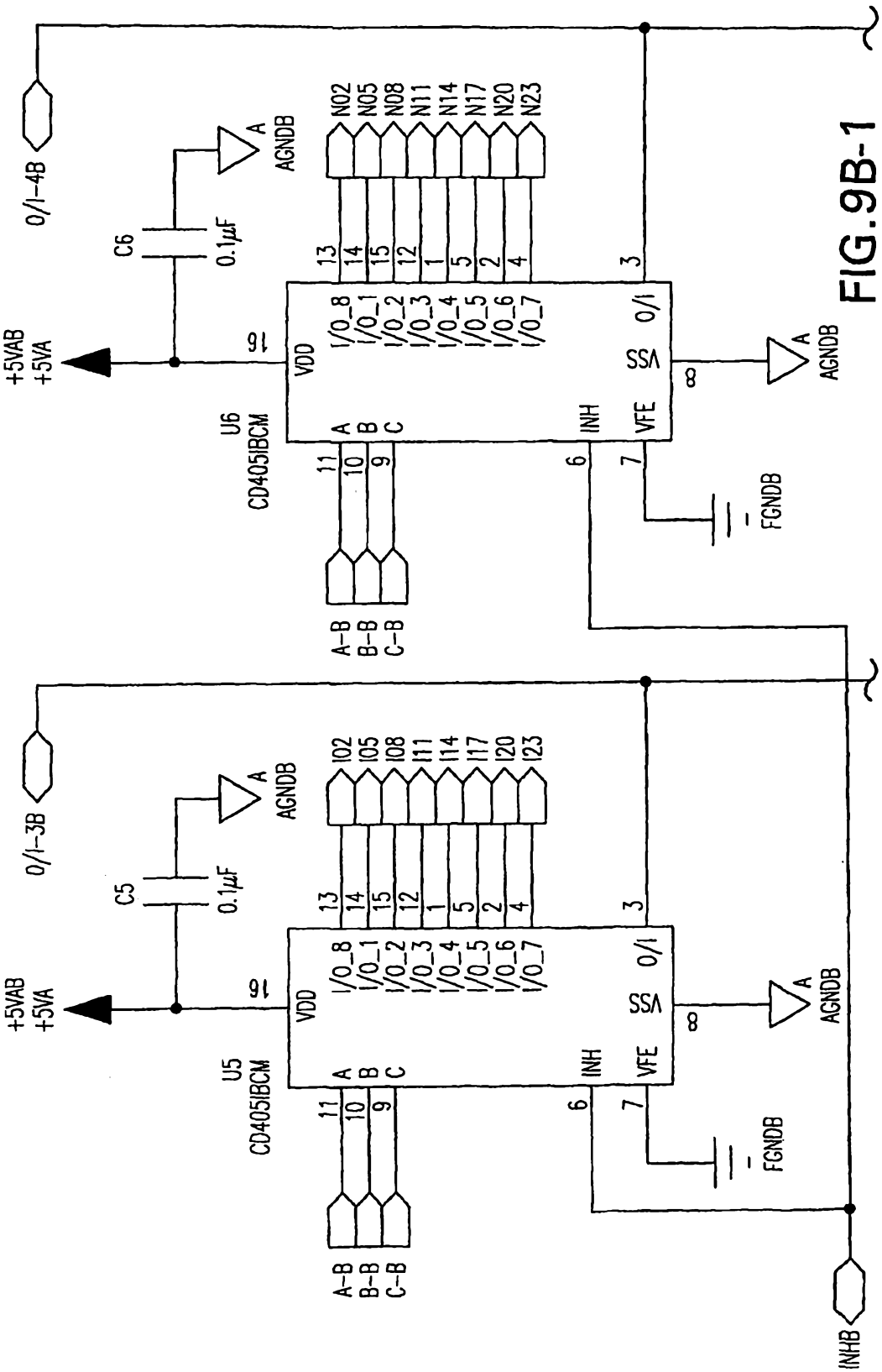

O/I-4B
C6
0.1µF
AGNDB
N02
N05
N08
N11
N14
N17
N20
N23
+5VAB
+5VA
VDD
I/O_8
I/O_1
I/O_2
I/O_3
I/O_4
I/O_5
I/O_6
I/O_7
O/I
SSA
U6
CD4051BCM
A
B
C
INH
VEE
A-B
B-B
C-B
FGNDB
O/I-3B
C5
0.1µF
AGNDB
I02
I05
I08
I11
I14
I17
I20
I23
+5VAB
+5VA
U5
CD4051BCM
FGNDB
INHB

FIG.9B-1

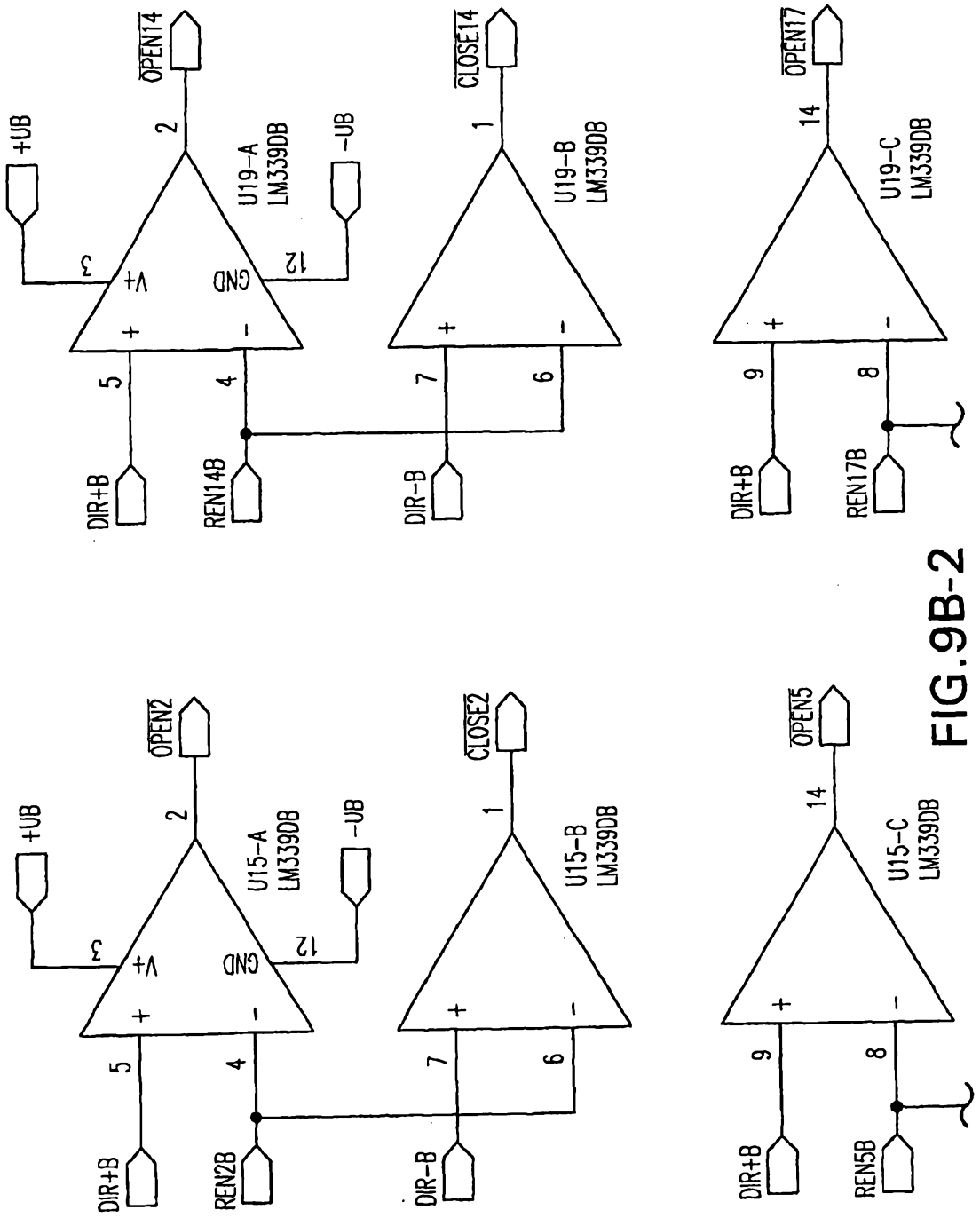

+UB
OPEN14
2
U19-A
LM339DB
-UB
3
V+
GND
12
+
-
5
4
DIR+B
REN14B
CLOSE14
1
U19-B
LM339DB
7
6
DIR-B
OPEN17
14
U19-C
LM339DB
9
8
DIR+B
REN17B
+UB
OPEN2
2
U15-A
LM339DB
-UB
3
V+
GND
12
+
-
5
4
DIR+B
REN2B
CLOSE2
1
U15-B
LM339DB
7
6
DIR-B
OPEN5
14
U15-C
LM339DB
9
8
DIR+B
REN5B

FIG.9B-2

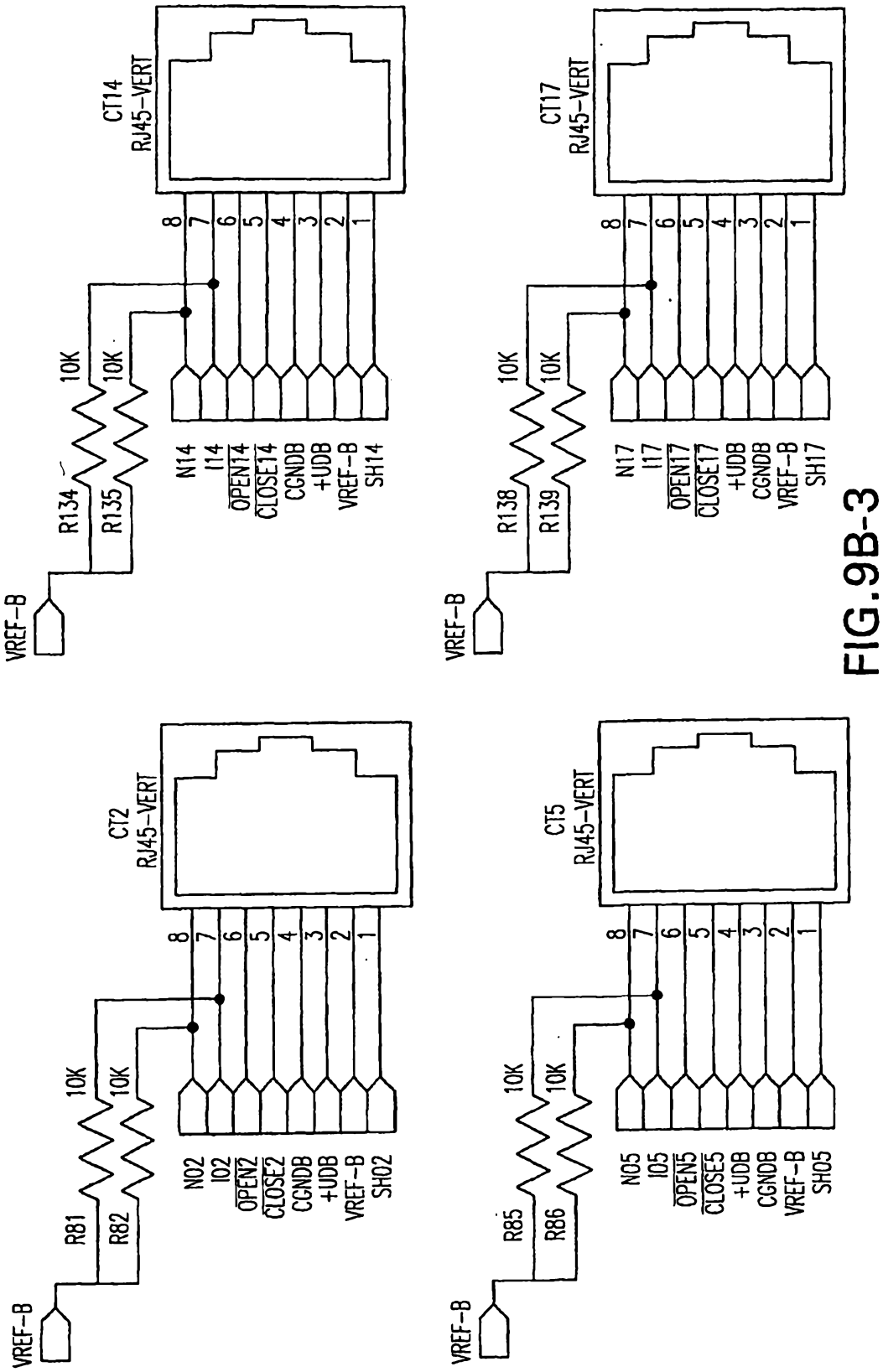

CT14
RJ45-VERT
CT17
RJ45-VERT
CT2
RJ45-VERT
CT5
RJ45-VERT
10K
R134
R135
R138
R139
R81
R82
R85
R86
VREF-B
N14
I14
OPEN14
CLOSE14
CGNDB
+UDB
VREF-B
SH14
N17
I17
OPEN17
CLOSE17
+UDB
CGNDB
VREF-B
SH17
N02
I02
OPEN2
CLOSE2
CGNDB
+UDB
VREF-B
SH02
N05
I05
OPEN5
CLOSE5
+UDB
CGNDB
VREF-B
SH05

FIG.9B-3

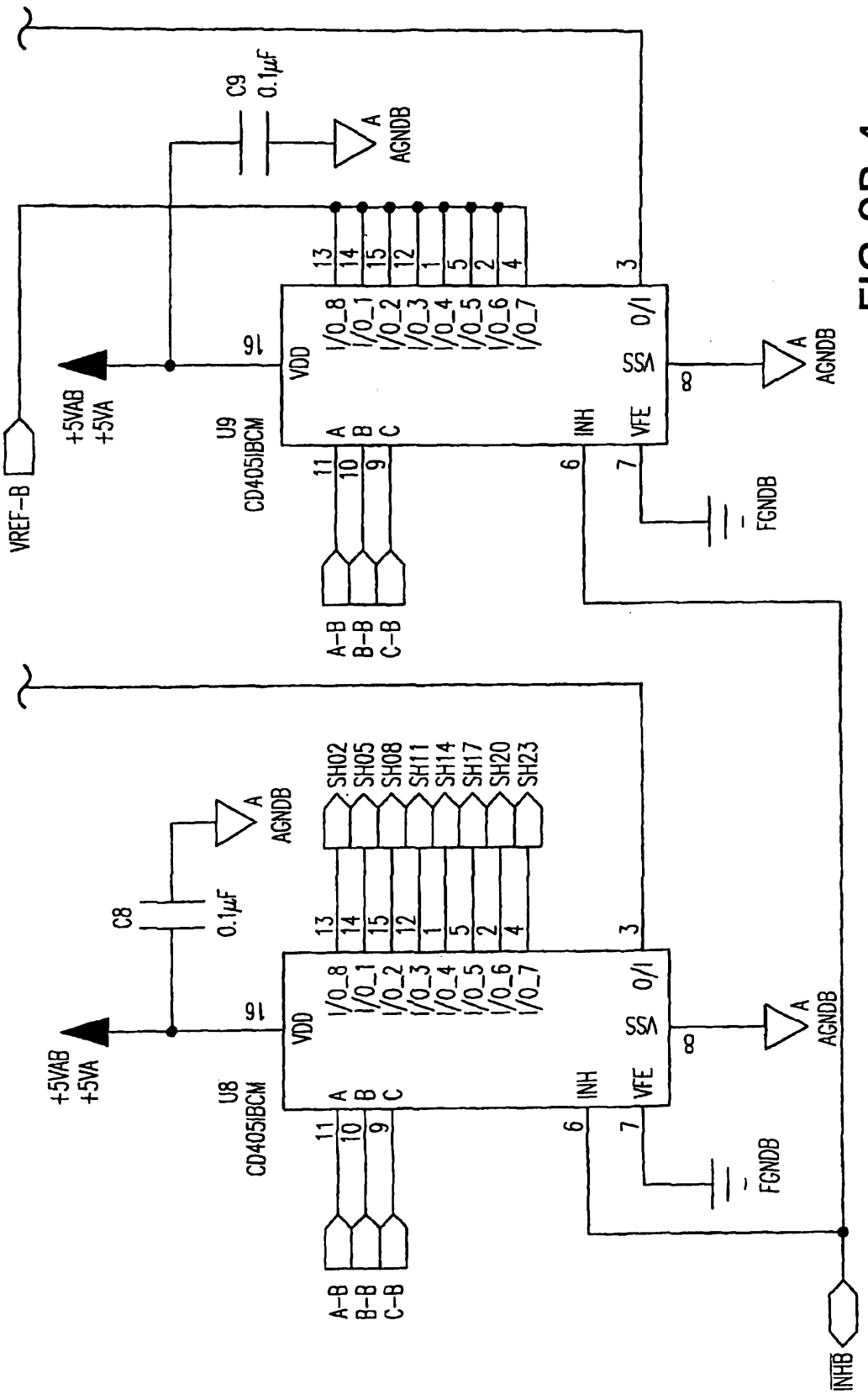

C9
0.1µF
AGNDB
VREF-B
+5VAB
+5VA
U9
CD4051BCM
VDD
I/O_8
I/O_1
I/O_2
I/O_3
I/O_4
I/O_5
I/O_6
I/O_7
COM
VSS
A
B
C
INH
VEE
A-B
B-B
C-B
FGNDB
C8
0.1µF
U8
SH02
SH05
SH08
SH11
SH14
SH17
SH20
SH23
INHB

FIG.9B-4

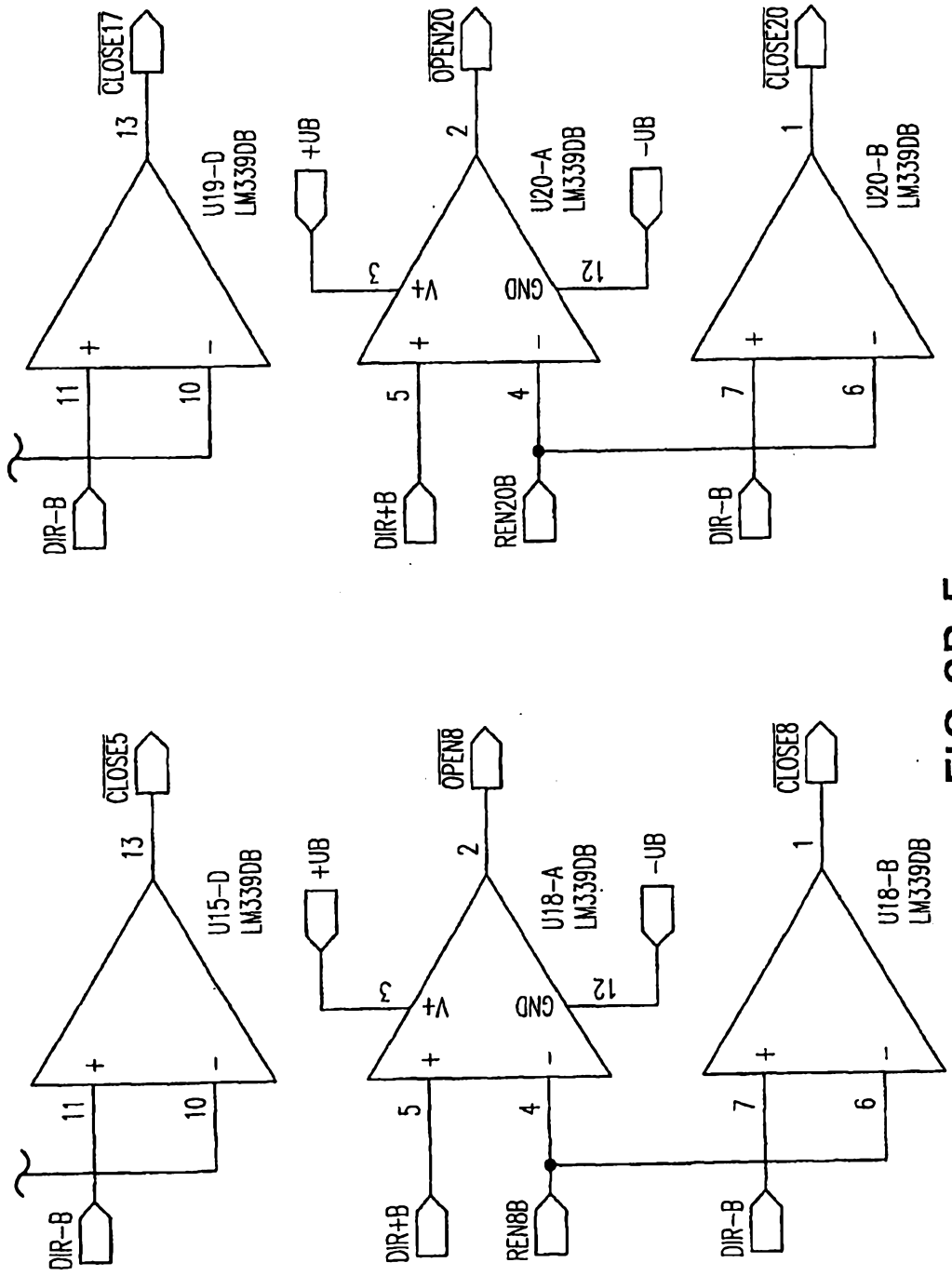

CLOSE17
13
U19-D
LM339DB
11
10
DIR-B
+UB
OPEN20
2
U20-A
LM339DB
-UB
3
V+
GND
12
5
4
DIR+B
REN20B
CLOSE20
1
U20-B
LM339DB
7
6
DIR-B
CLOSE5
13
U15-D
LM339DB
11
10
DIR-B
+UB
OPEN8
2
U18-A
LM339DB
-UB
3
V+
GND
12
5
4
DIR+B
REN8B
CLOSE8
1
U18-B
LM339DB
7
6
DIR-B

FIG.9B-5

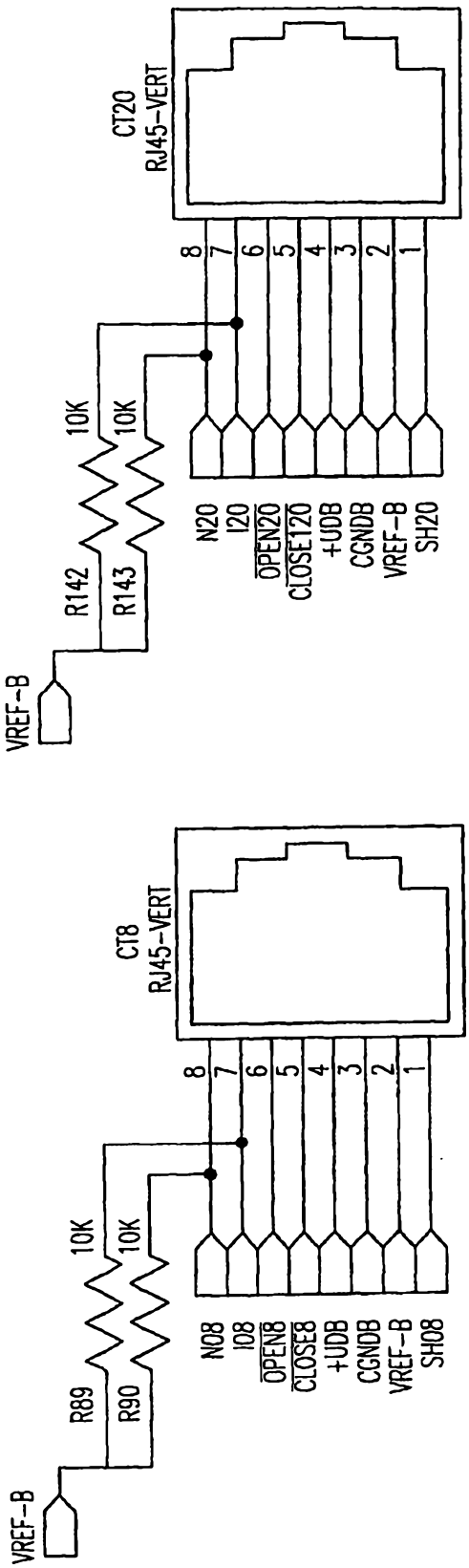

CT20
RJ45-VERT
8
7
6
5
4
3
2
1
N20
I20
OPEN20
CLOSE120
+UDB
CGNDB
VREF-B
SH20
10K
10K
R142
R143
VREF-B
CT8
RJ45-VERT
8
7
6
5
4
3
2
1
N08
I08
OPEN8
CLOSE8
+UDB
CGNDB
VREF-B
SH08
10K
10K
R89
R90
VREF-B

FIG.9B-6

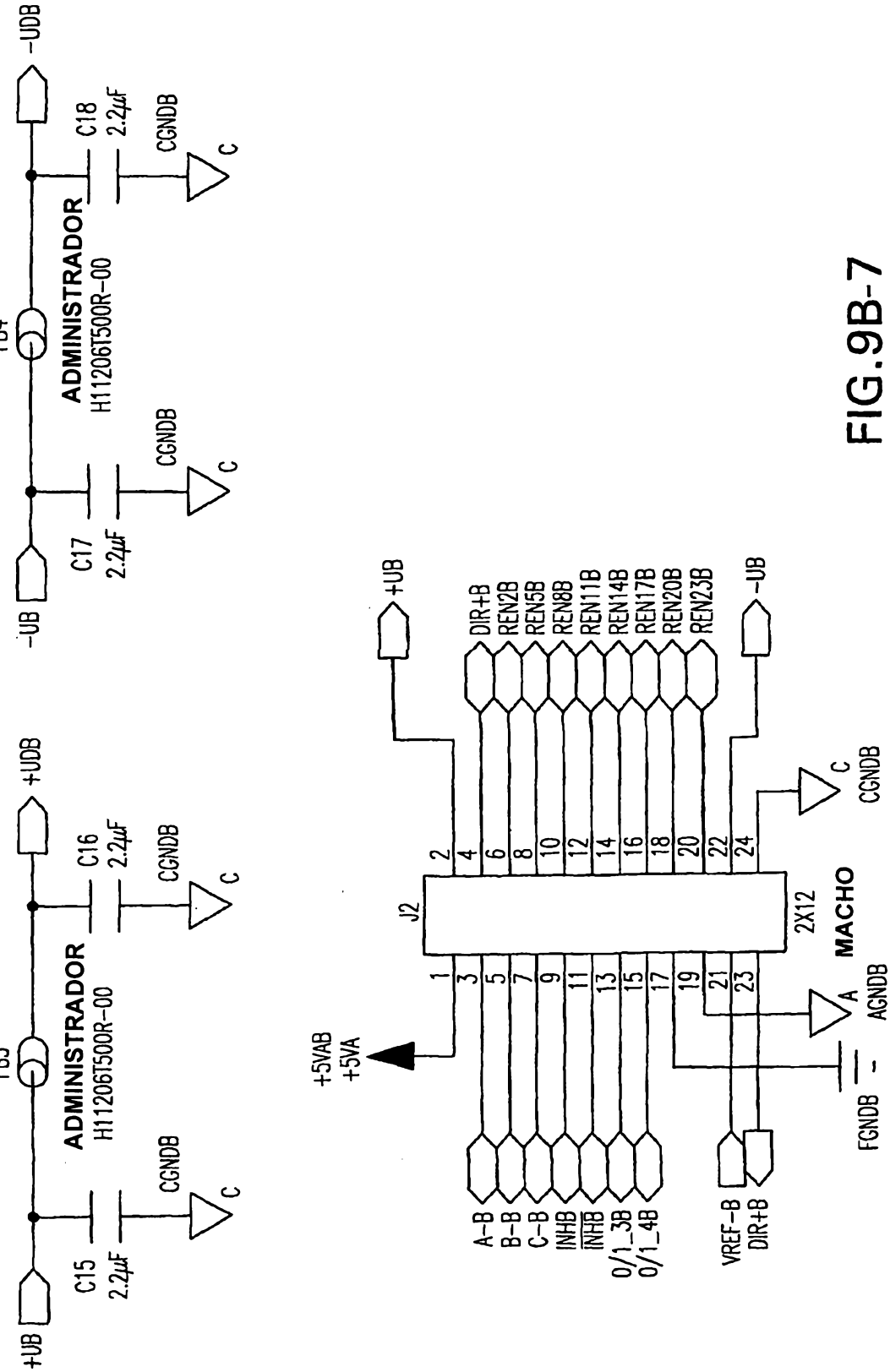

+UB
C15
2.2µF
CGNDB
FB3
ADMINISTRADOR
H11206T500R-00
C16
2.2µF
CGNDB
+UDB
-UB
C17
2.2µF
CGNDB
FB4
ADMINISTRADOR
H11206T500R-00
C18
2.2µF
CGNDB
-UDB
+5VAB
+5VA
J2
2X12
MACHO
+UB
-UB
DIR+B
REN2B
REN5B
REN8B
REN11B
REN14B
REN17B
REN20B
REN23B
A-B
B-B
C-B
INHB
INHB
0/1_3B
0/1_4B
VREF-B
DIR+B
FGNDB
AGNDB
CGNDB

FIG.9B-7

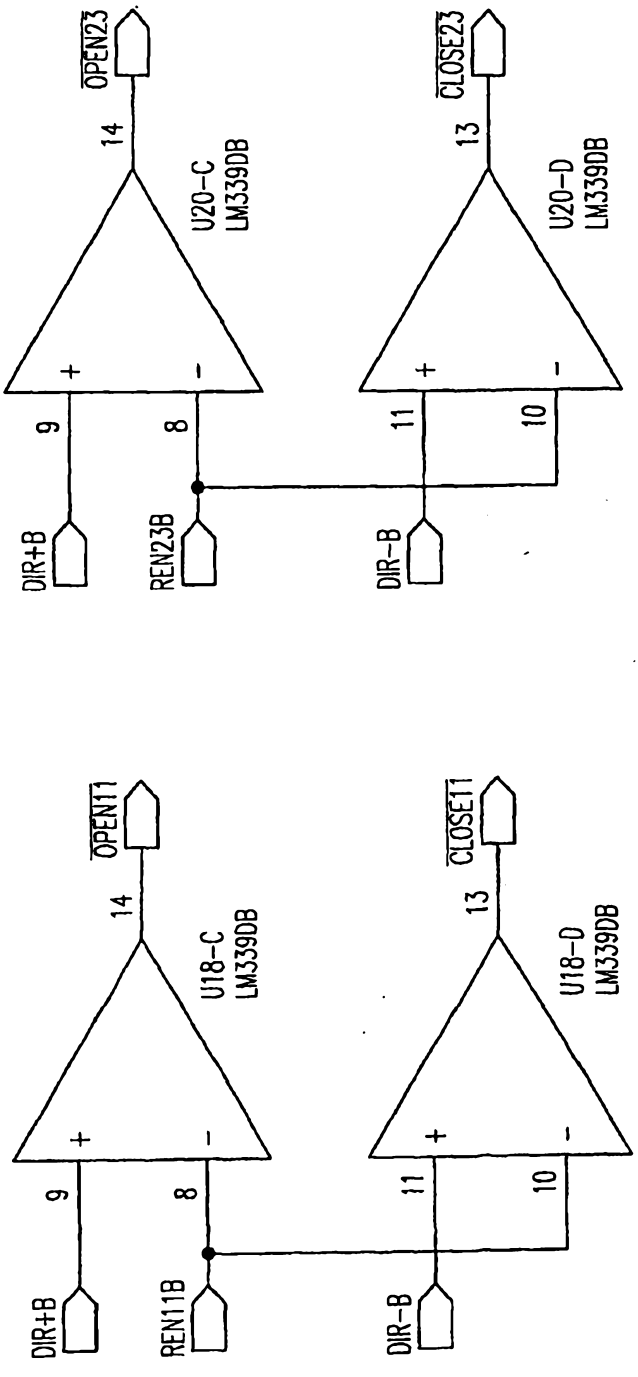

OPEN23
14
U20-C
LM339DB
+
-
9
8
DIR+B
REN23B
CLOSE23
13
U20-D
LM339DB
+
-
11
10
DIR-B
OPEN11
14
U18-C
LM339DB
+
-
9
8
DIR+B
REN11B
CLOSE11
13
U18-D
LM339DB
+
-
11
10
DIR-B

FIG.9B-8

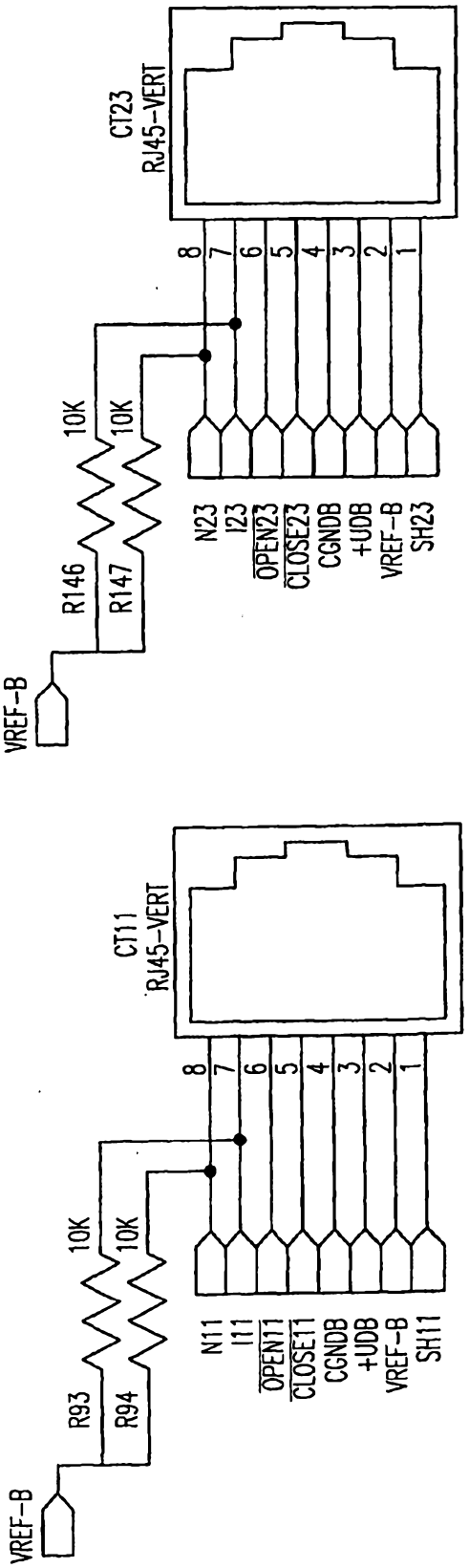

CT23
RJ45-VERT
8
7
6
5
4
3
2
1
N23
I23
OPEN23
CLOSE23
CGNDB
+UDB
VREF-B
SH23
10K
10K
R146
R147
VREF-B
CT11
RJ45-VERT
8
7
6
5
4
3
2
1
N11
I11
OPEN11
CLOSE11
CGNDB
+UDB
VREF-B
SH11
10K
10K
R93
R94
VREF-B

FIG.9B-9

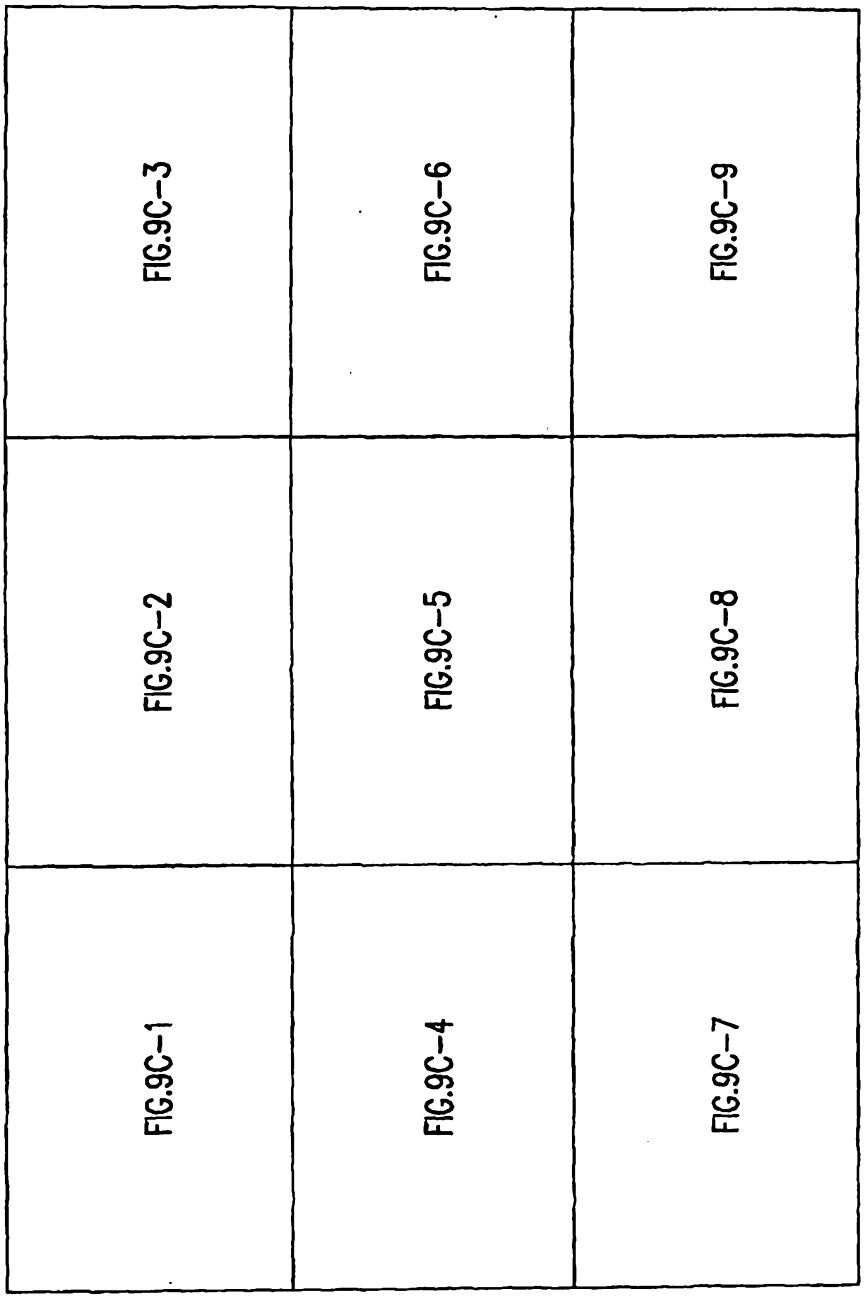


FIG.9C-1
FIG.9C-2
FIG.9C-3
FIG.9C-4
FIG.9C-5
FIG.9C-6
FIG.9C-7
FIG.9C-8
FIG.9C-9

FIG.9C

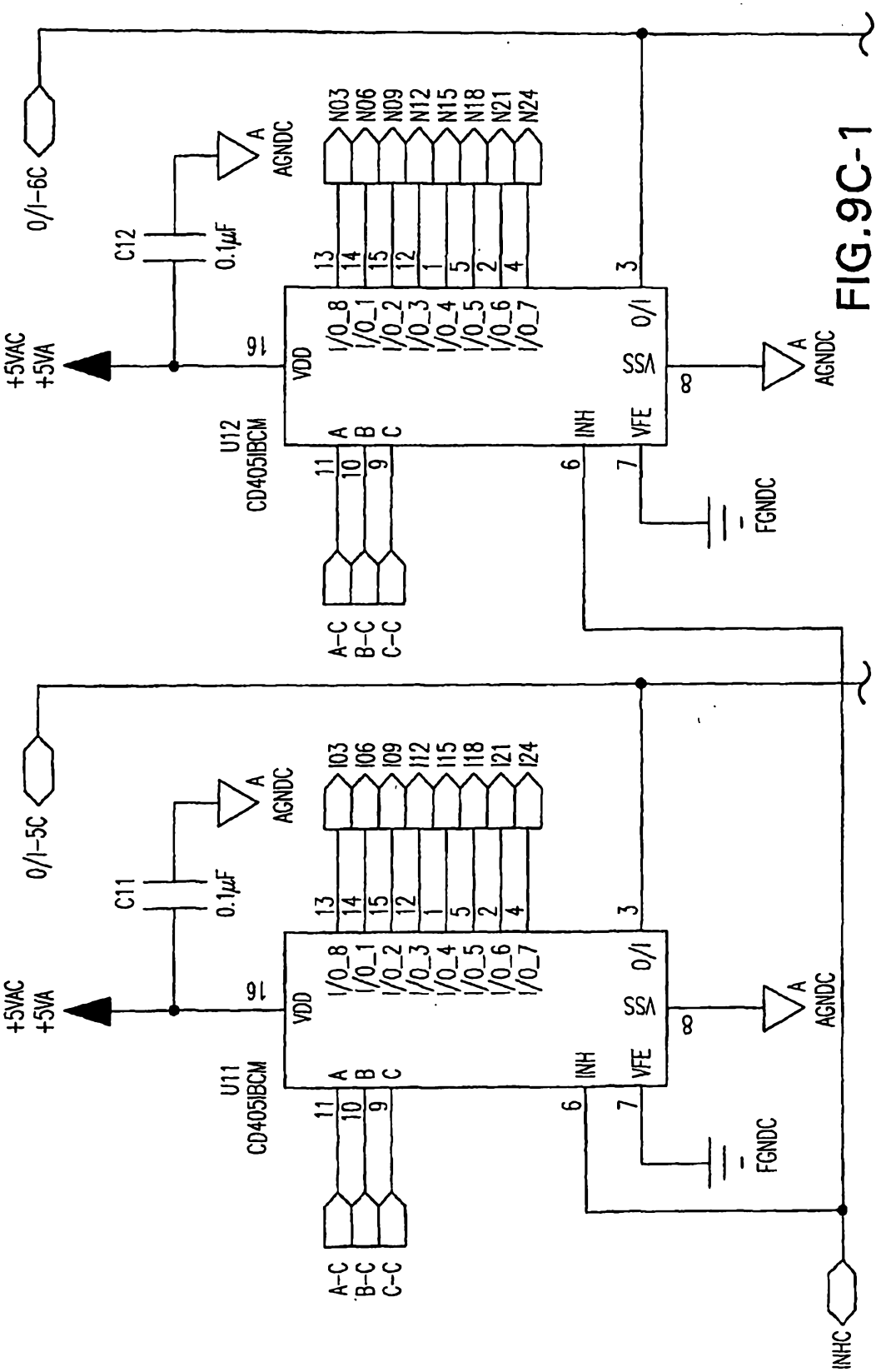

O/I-6C
C12
0.1µF
AGNDC
N03
N06
N09
N12
N15
N18
N21
N24
+5VAC
+5VA
U12
CD4051BCM
VDD
I/O_8
I/O_1
I/O_2
I/O_3
I/O_4
I/O_5
I/O_6
I/O_7
O/I
VSS
VEE
INH
A
B
C
A-C
B-C
C-C
FGNDC
O/I-5C
C11
I03
I06
I09
I12
I15
I18
I21
I24
U11
INHC

FIG.9C-1

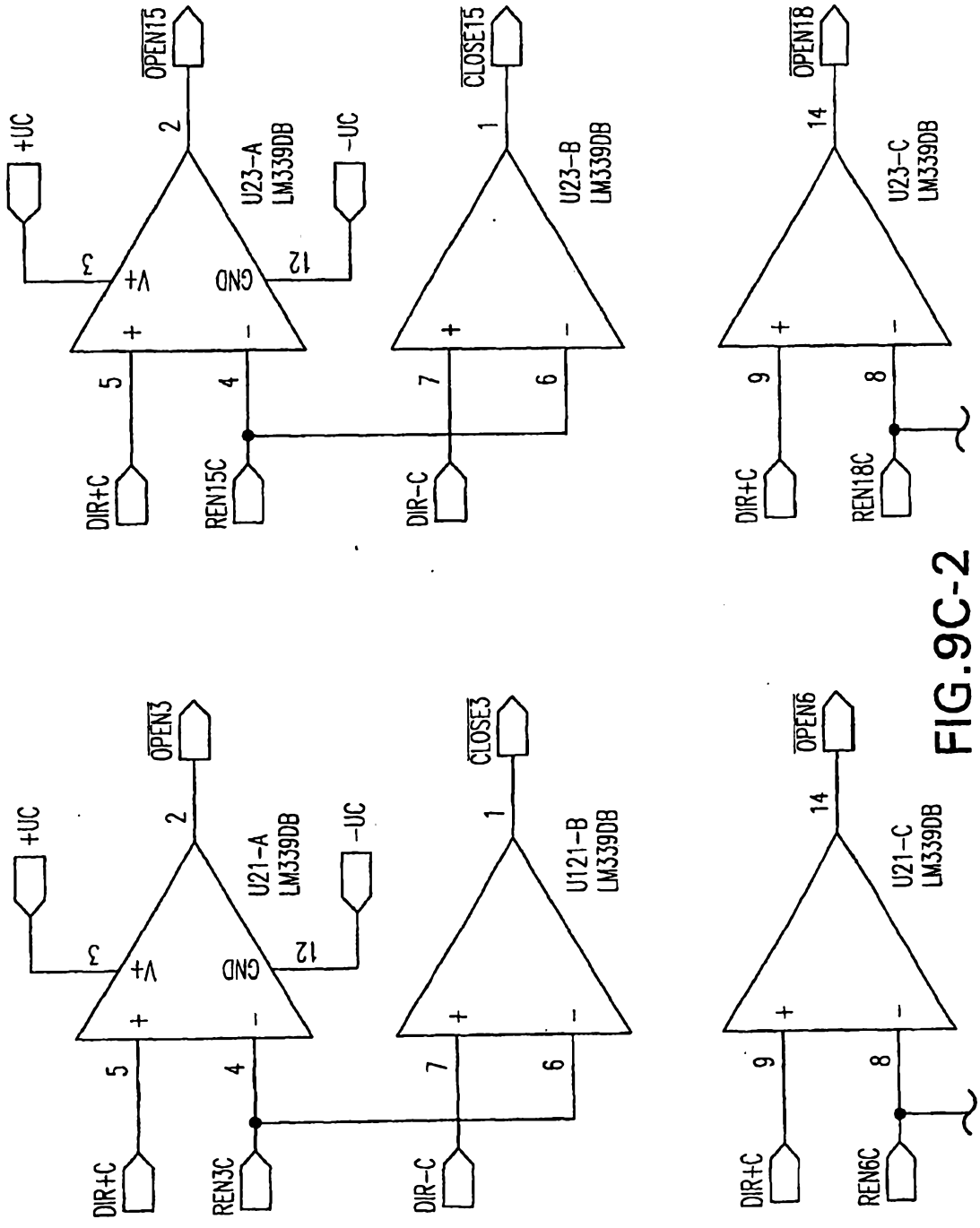

+UC
OPEN15
2
U23-A
LM339DB
-UC
3
V+
GND
12
+
-
5
4
DIR+C
REN15C
CLOSE15
1
U23-B
LM339DB
7
6
DIR-C
OPEN18
14
U23-C
LM339DB
9
8
DIR+C
REN18C
+UC
OPEN3
2
U21-A
LM339DB
-UC
3
V+
GND
12
+
-
5
4
DIR+C
REN3C
CLOSE3
1
U121-B
LM339DB
7
6
DIR-C
OPEN6
14
U21-C
LM339DB
9
8
DIR+C
REN6C

FIG.9C-2

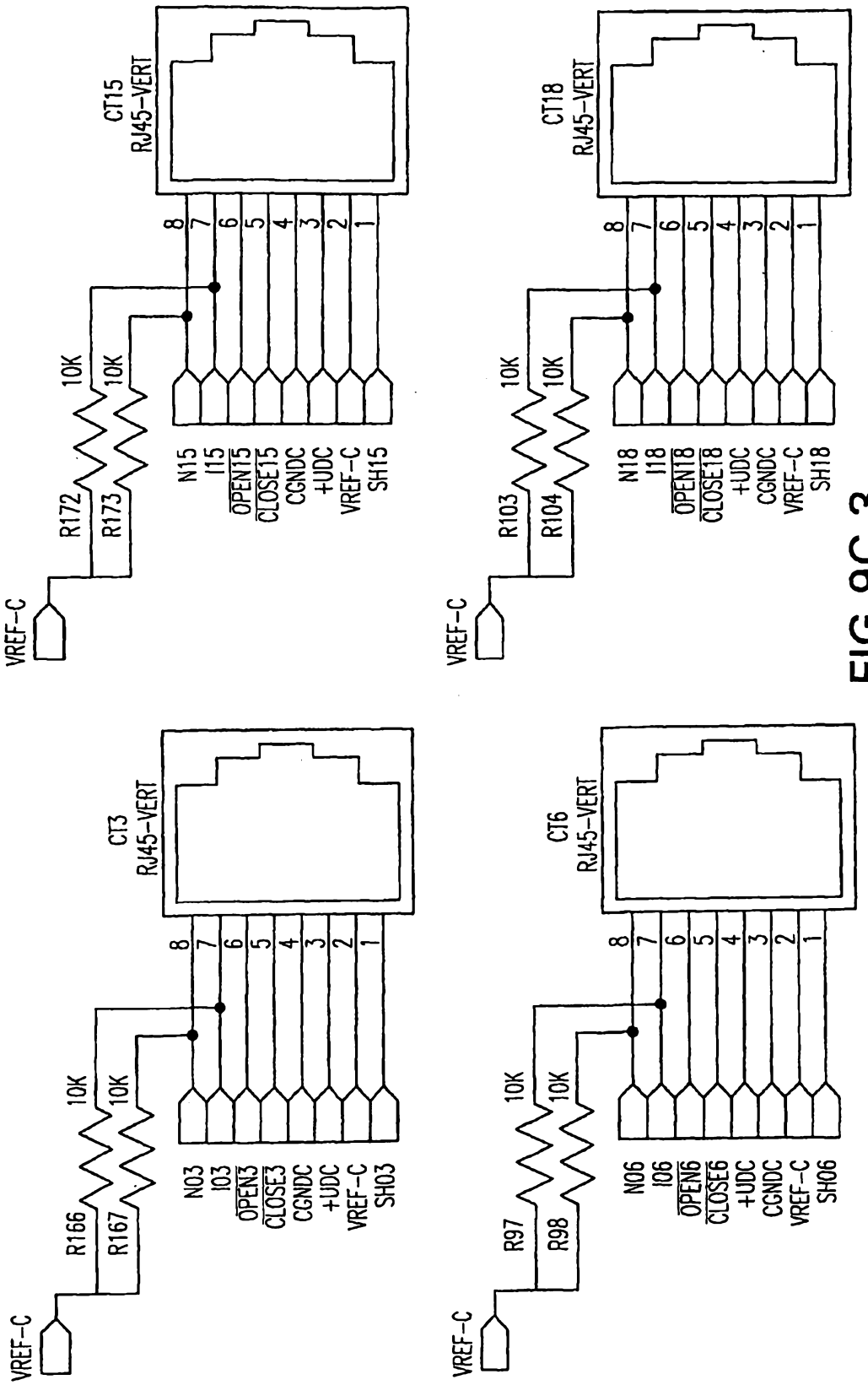

CT15
RJ45-VERT
8 7 6 5 4 3 2 1
N15
I15
OPEN15
CLOSE15
CGNDC
+UDC
VREF-C
SH15
10K
10K
R172
R173
VREF-C
CT18
RJ45-VERT
8 7 6 5 4 3 2 1
N18
I18
OPEN18
CLOSE18
+UDC
CGNDC
VREF-C
SH18
10K
10K
R103
R104
VREF-C
CT3
RJ45-VERT
8 7 6 5 4 3 2 1
N03
I03
OPEN3
CLOSE3
CGNDC
+UDC
VREF-C
SH03
10K
10K
R166
R167
VREF-C
CT6
RJ45-VERT
8 7 6 5 4 3 2 1
N06
I06
OPEN6
CLOSE6
+UDC
CGNDC
VREF-C
SH06
10K
10K
R97
R98
VREF-C

FIG.9C-3

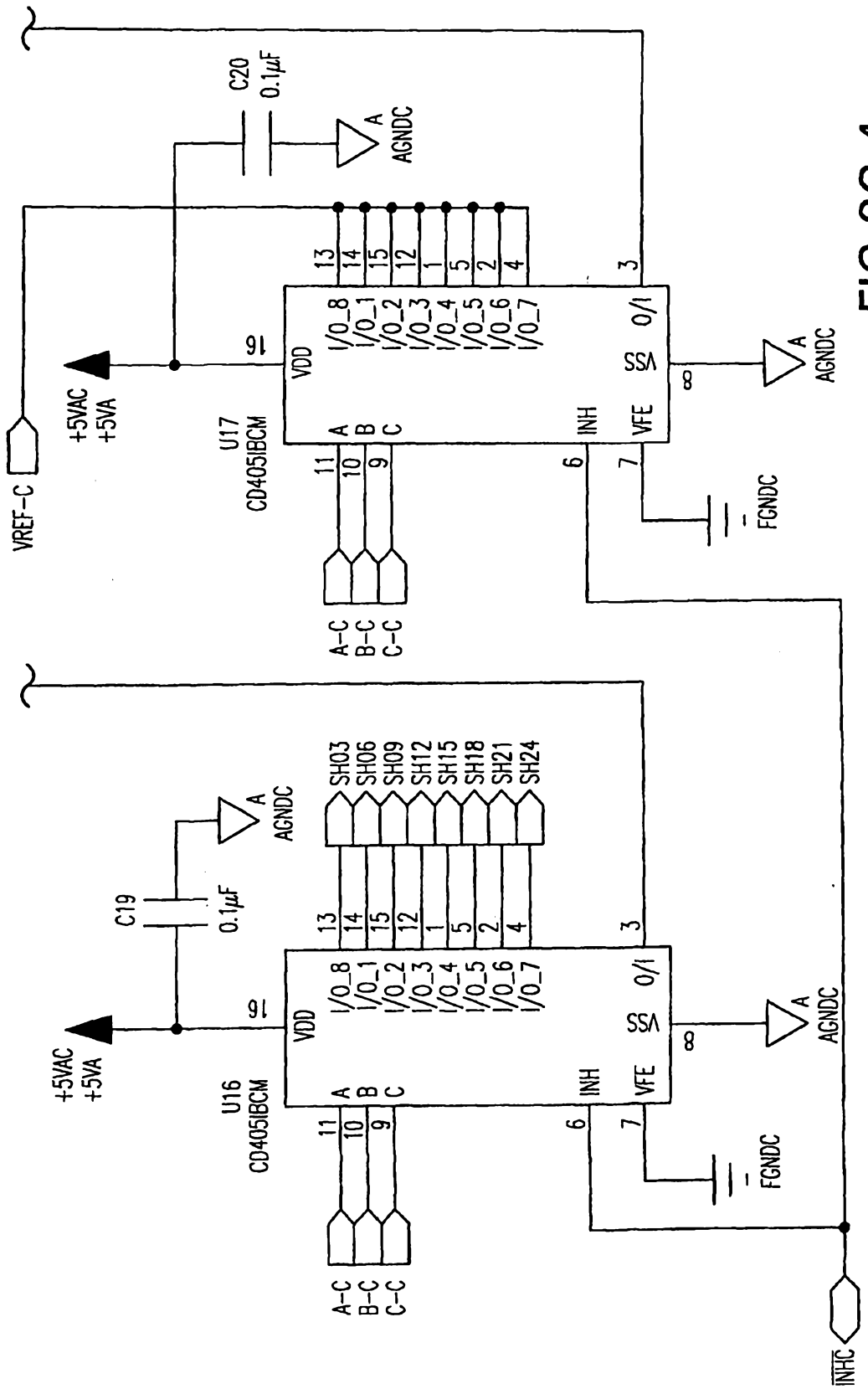

C20
0.1µF
AGNDC
VREF-C
+5VAC
+5VA
U17
CD4051BCM
VDD
VSS
INH
VEE
I/O_8
I/O_1
I/O_2
I/O_3
I/O_4
I/O_5
I/O_6
I/O_7
O/I
A
B
C
A-C
B-C
C-C
FGNDC
C19
0.1µF
U16
CD4051BCM
SH03
SH06
SH09
SH12
SH15
SH18
SH21
SH24
INHC

FIG.9C-4

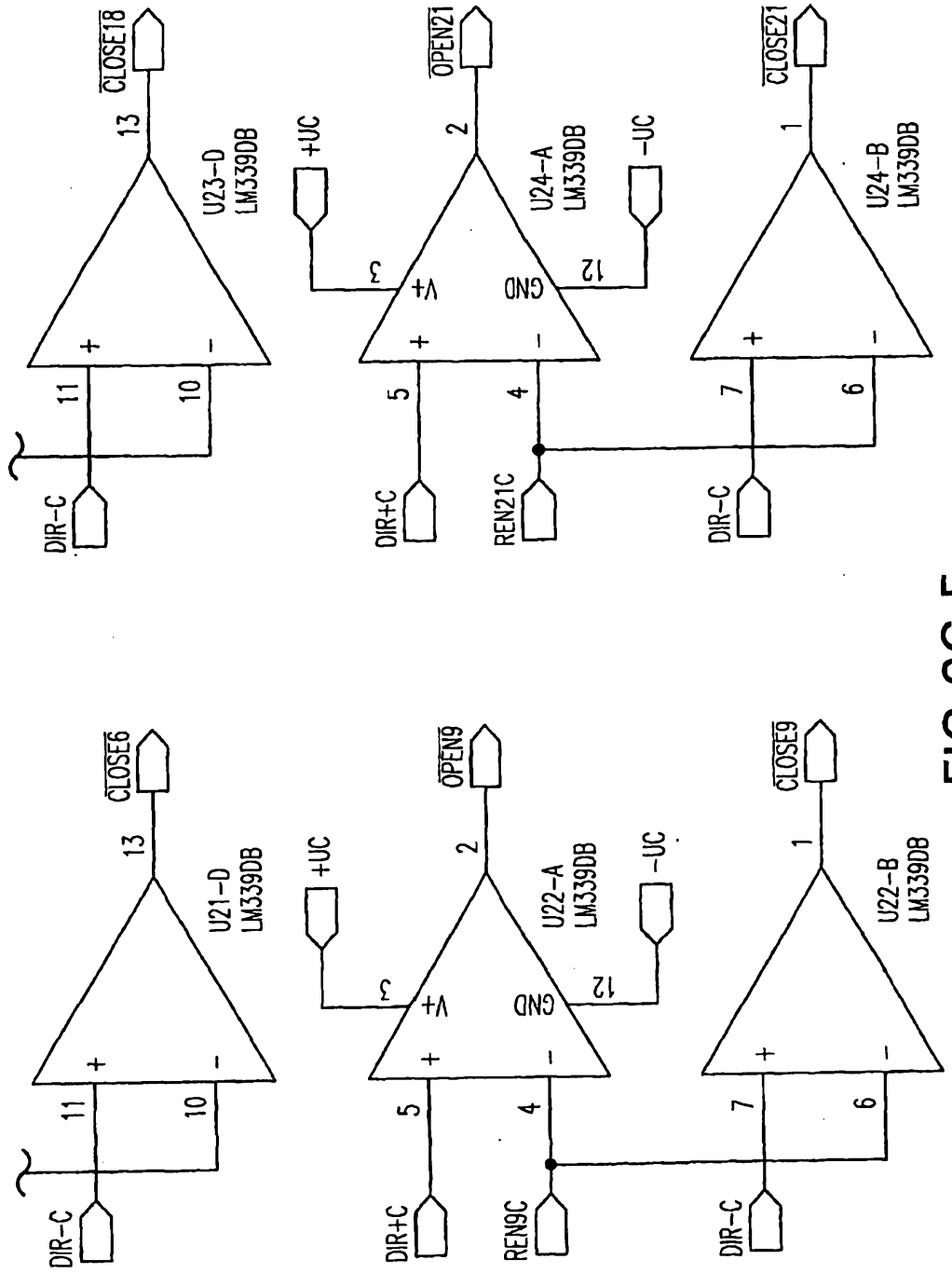

CLOSE18
13
U23-D
LM339DB
11
10
DIR-C
OPEN21
2
+UC
3
V+
U24-A
LM339DB
GND
12
-UC
5
4
DIR+C
REN21C
CLOSE21
1
U24-B
LM339DB
7
6
DIR-C
CLOSE6
13
U21-D
LM339DB
11
10
DIR-C
OPEN9
2
+UC
3
V+
U22-A
LM339DB
GND
12
-UC
5
4
DIR+C
REN9C
CLOSE9
1
U22-B
LM339DB
7
6
DIR-C

FIG.9C-5

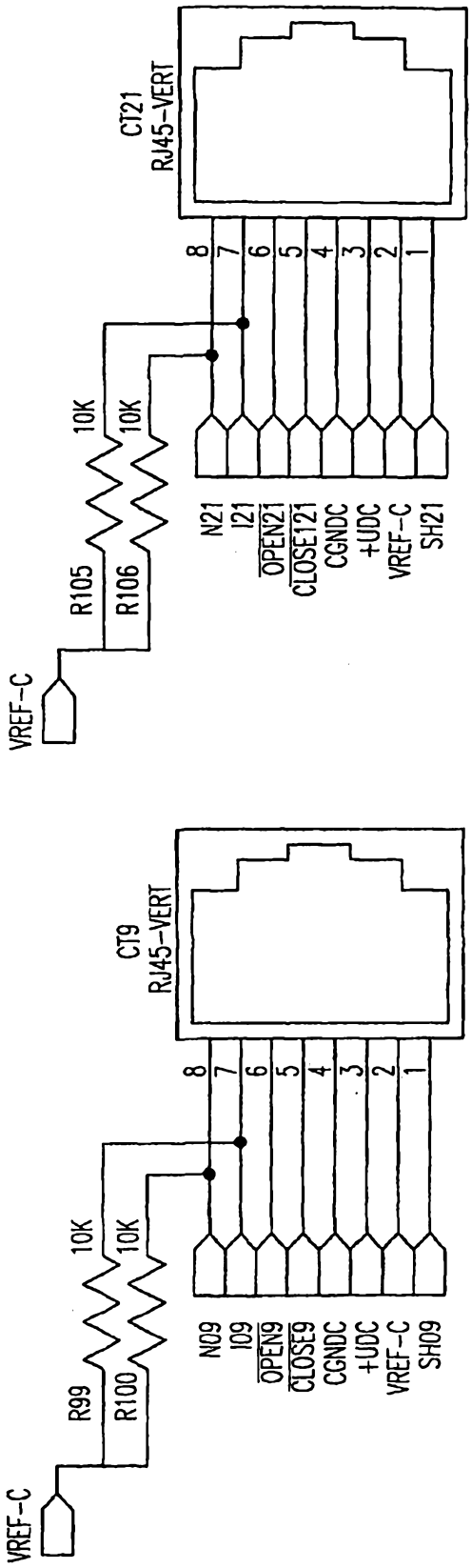

CT21
RJ45-VERT
8
7
6
5
4
3
2
1
N21
I21
OPEN21
CLOSE121
CGNDC
+UDC
VREF-C
SH21
10K
10K
R105
R106
VREF-C
CT9
RJ45-VERT
8
7
6
5
4
3
2
1
N09
I09
OPEN9
CLOSE9
CGNDC
+UDC
VREF-C
SH09
10K
10K
R99
R100
VREF-C

FIG.9C-6

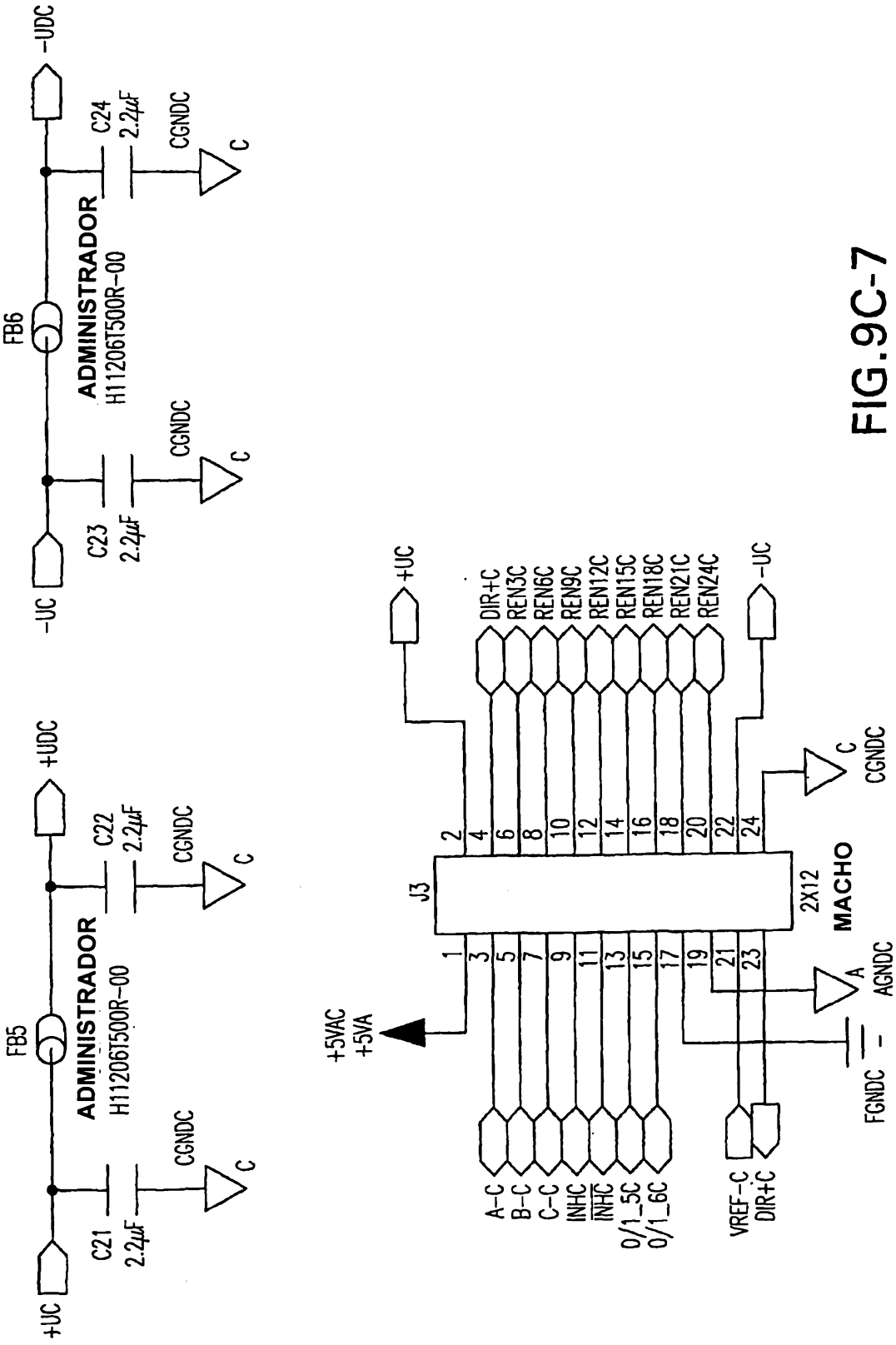

+UC
-UC
+UDC
-UDC
FB5
FB6
ADMINISTRADOR
H11206T500R-00
C21
2.2µF
C22
2.2µF
C23
2.2µF
C24
2.2µF
CGNDC
C
+5VAC
+5VA
J3
2X12
MACHO
A
AGNDC
FGNDC
DIR+C
REN3C
REN6C
REN9C
REN12C
REN15C
REN18C
REN21C
REN24C
A-C
B-C
C-C
INHC
INHC
0/1_5C
0/1_6C
VREF-C
DIR+C

FIG.9C-7

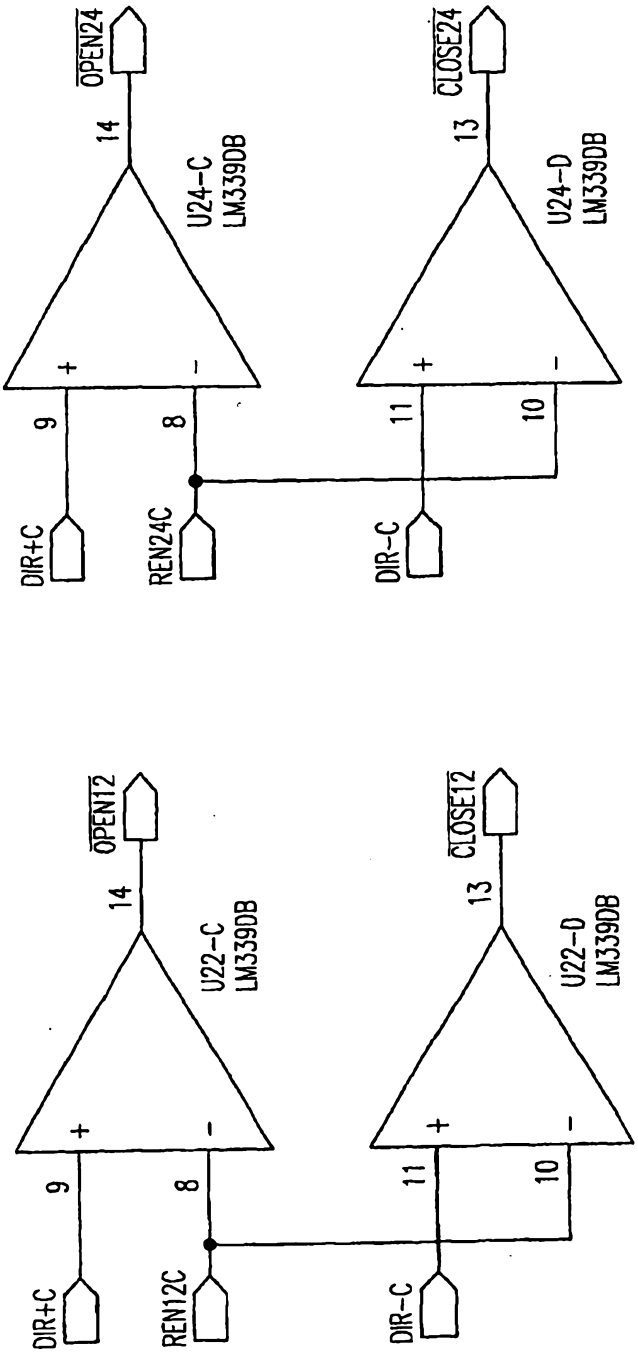

OPEN24
14
U24-C
LM339DB
9
8
DIR+C
REN24C
CLOSE24
13
U24-D
LM339DB
11
10
DIR-C
OPEN12
14
U22-C
LM339DB
9
8
DIR+C
REN12C
CLOSE12
13
U22-D
LM339DB
11
10
DIR-C

FIG.9C-8

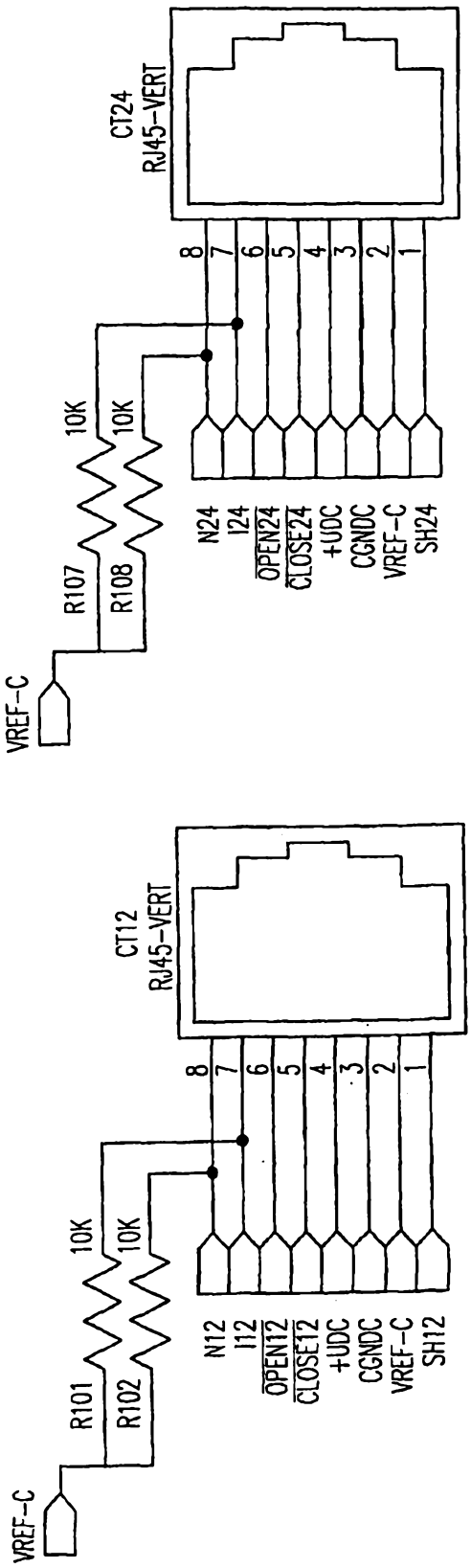

CT24
RJ45-VERT
8
7
6
5
4
3
2
1
N24
I24
OPEN24
CLOSE24
+UDC
CGNDC
VREF-C
SH24
10K
10K
R107
R108
VREF-C
CT12
RJ45-VERT
8
7
6
5
4
3
2
1
N12
I12
OPEN12
CLOSE12
+UDC
CGNDC
VREF-C
SH12
10K
10K
R101
R102
VREF-C

FIG.9C-9

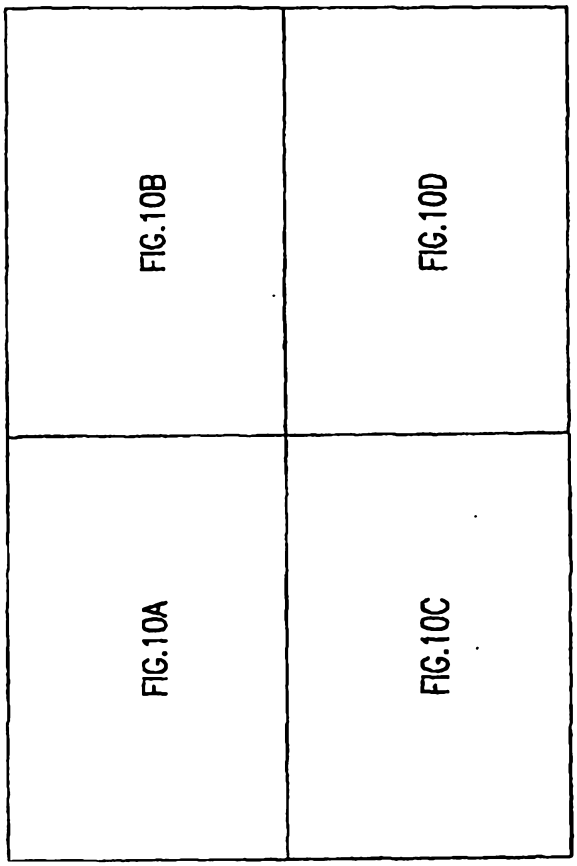


FIG.10

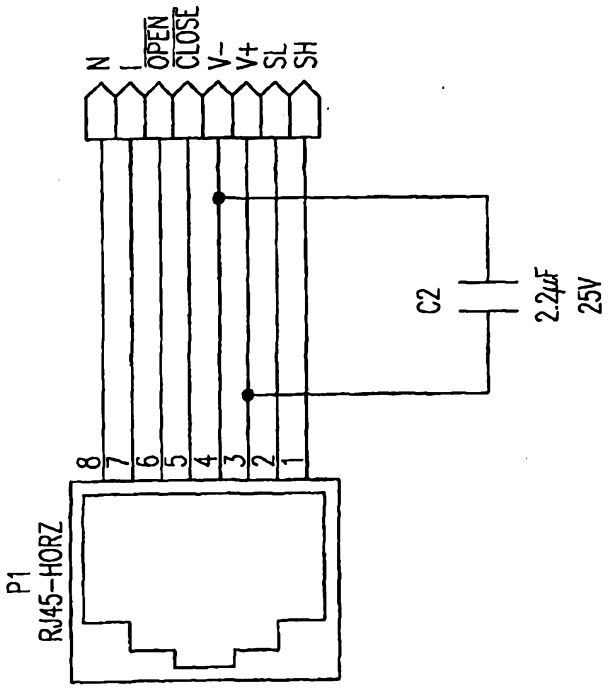


FIG.10A

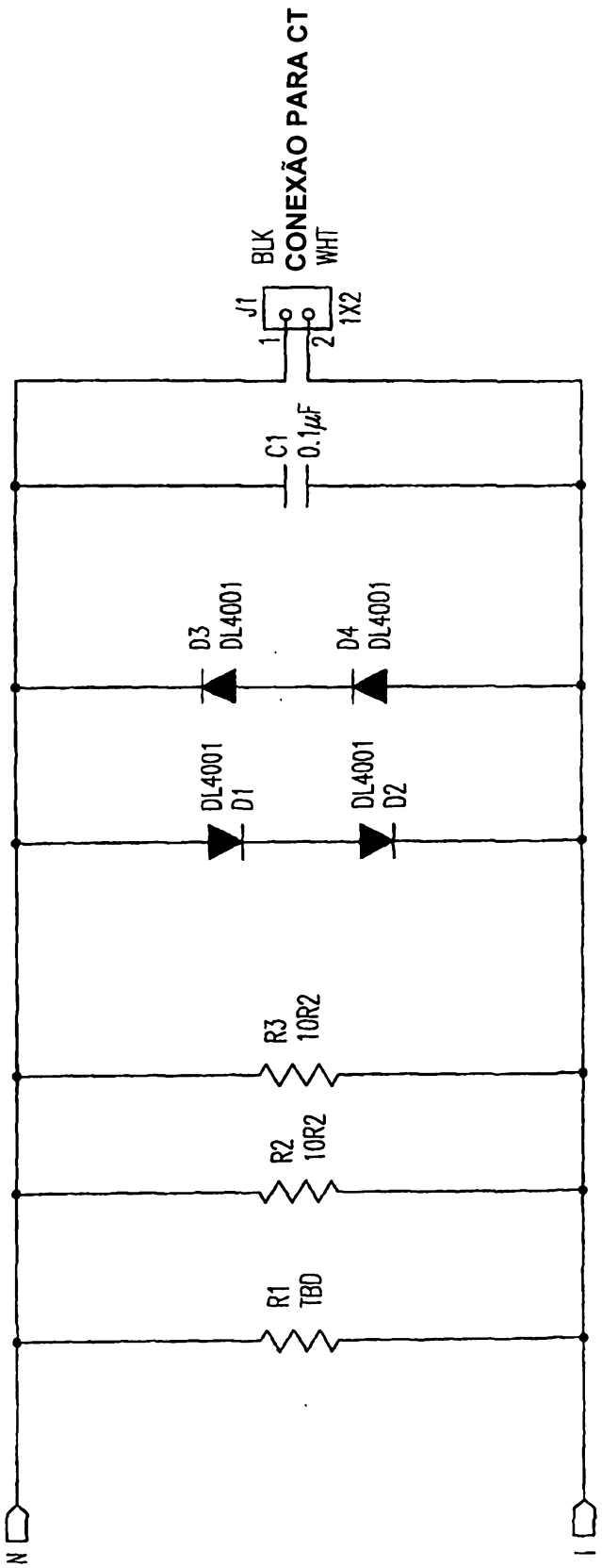

J1
1
2
1X2
BLK
WHT
CONEXÃO PARA CT
C1
0.1µF
D3
DL4001
D4
DL4001
DL4001
D1
DL4001
D2
R3
10R2
R2
10R2
R1
TBD
N
L

FIG.10B

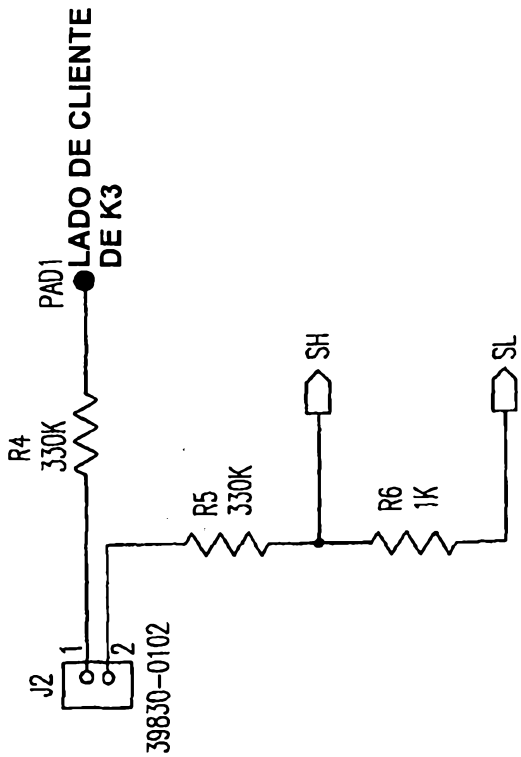

LADO DE CLIENTE
DE K3
PAD1
R4
330K
R5
330K
R6
1K
SH
SL
J2
1
2
39830-0102

FIG.10C

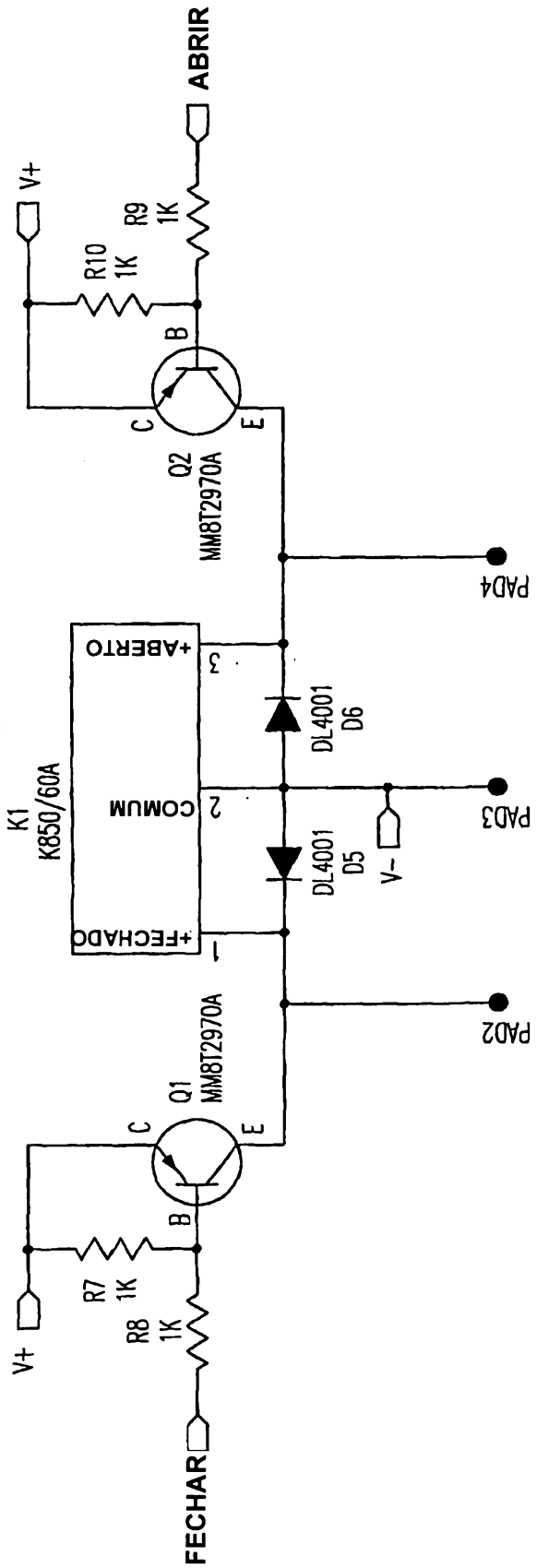

ABRIR
V+
R9
1K
R10
1K
B
C
E
Q2
MM8T2970A
PAD4
+ABERTO
3
DL4001
D6
K1
K850/60A
COMUM
2
PAD3
DL4001
D5
V-
+FECHADO
1
PAD2
Q1
MM8T2970A
C
E
B
R7
1K
V+
R8
1K
FECHAR

FIG.10D

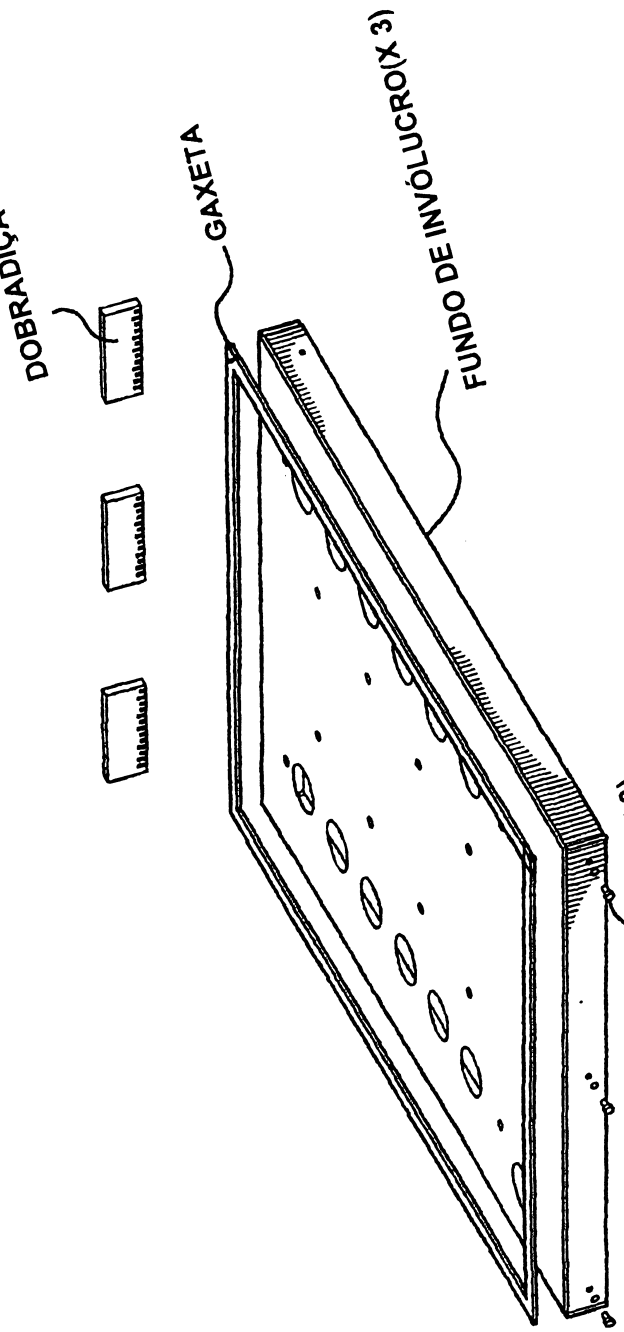

DOBRADIÇA
GAXETA
FUNDO DE INVÓLUCRO(X 3)
PARAFUSO E ARRUELA DE RETENÇÃO (X 3)

FIG. 11

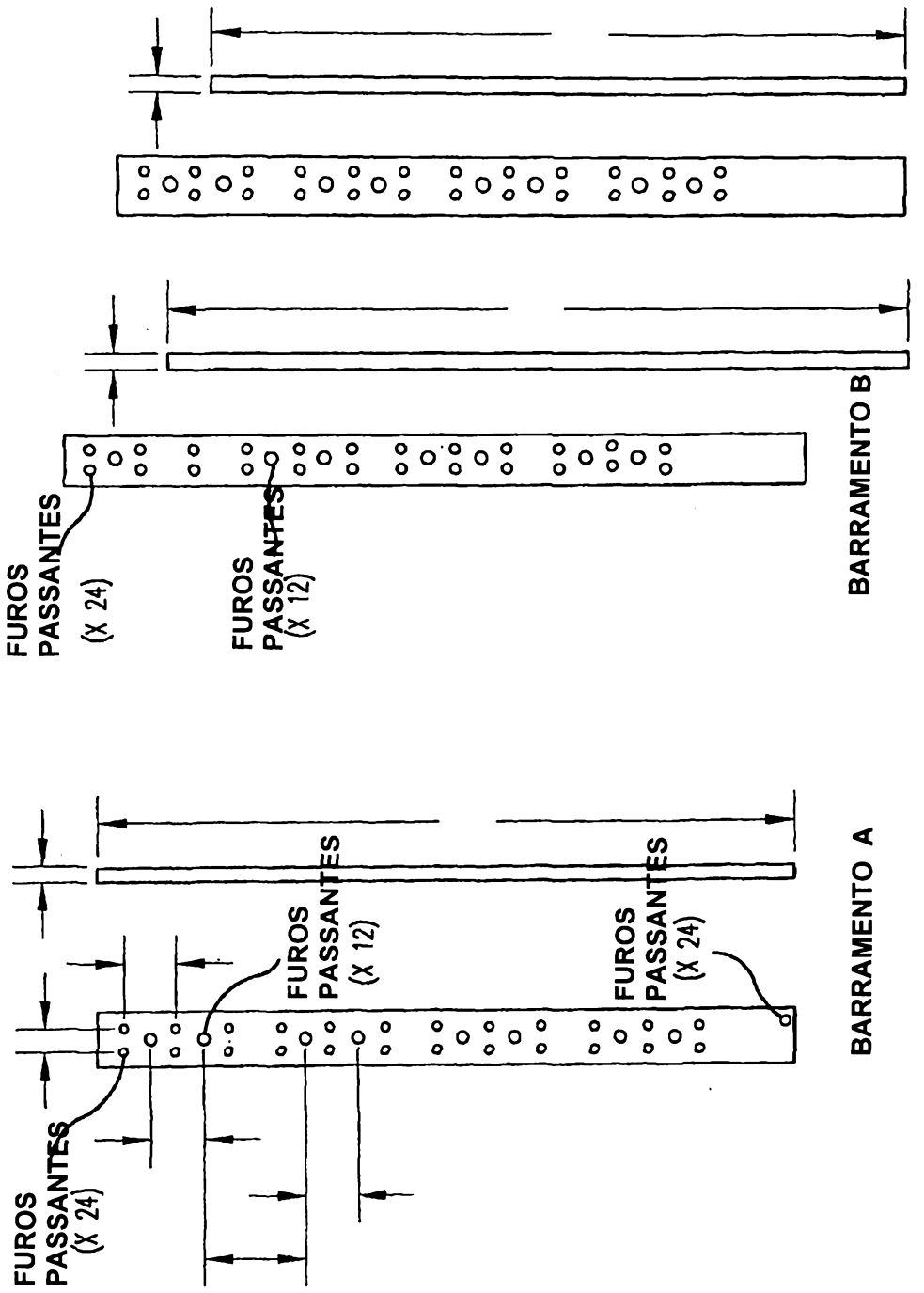

FUROS PASSANTES (X 24)
FUROS PASSANTES (X 12)
BARRAMENTO B
FUROS PASSANTES (X 12)
FUROS PASSANTES (X 24)
FUROS PASSANTES (X 24)
BARRAMENTO A

FIG.12

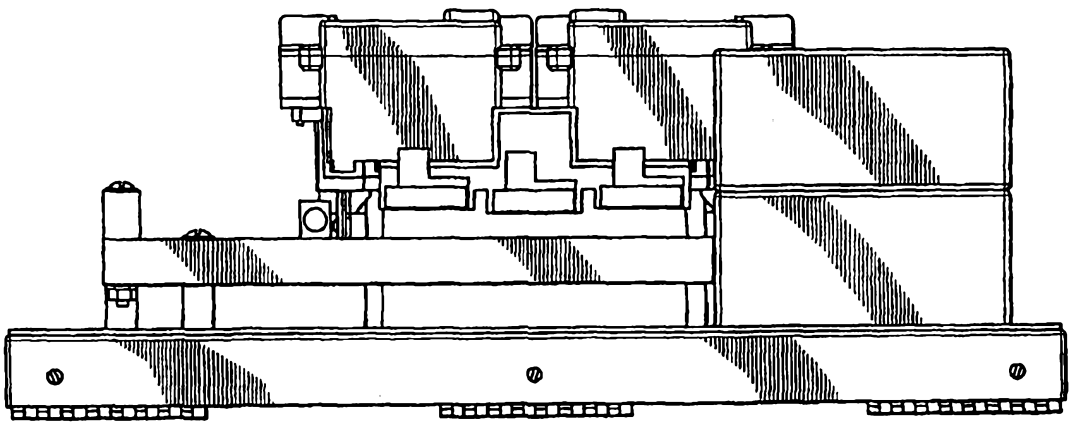


FIG.13

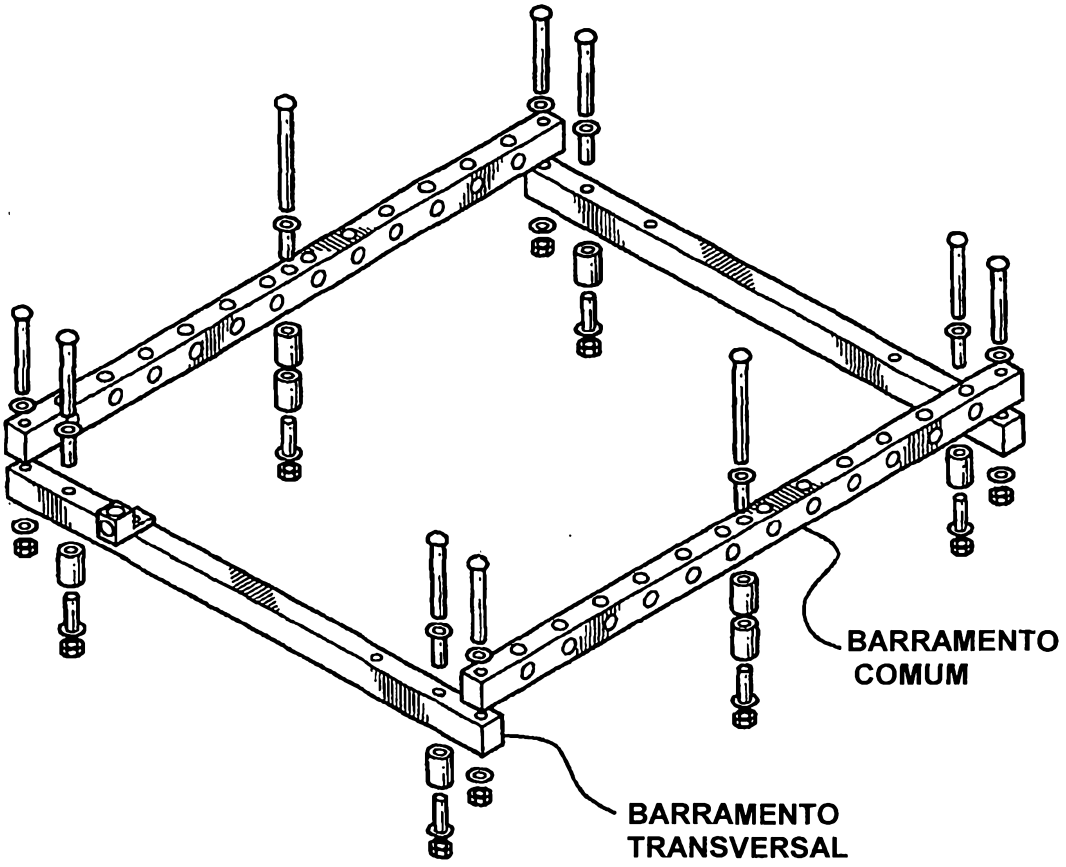


FIG.14

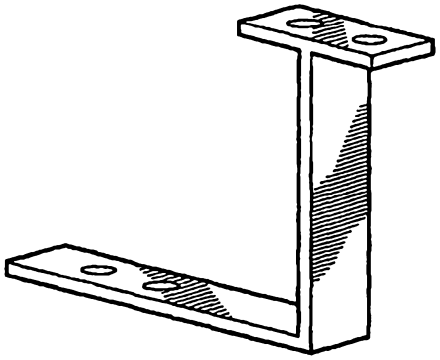


FIG.15A

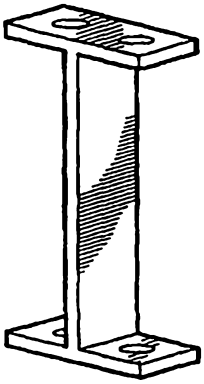


FIG.15B

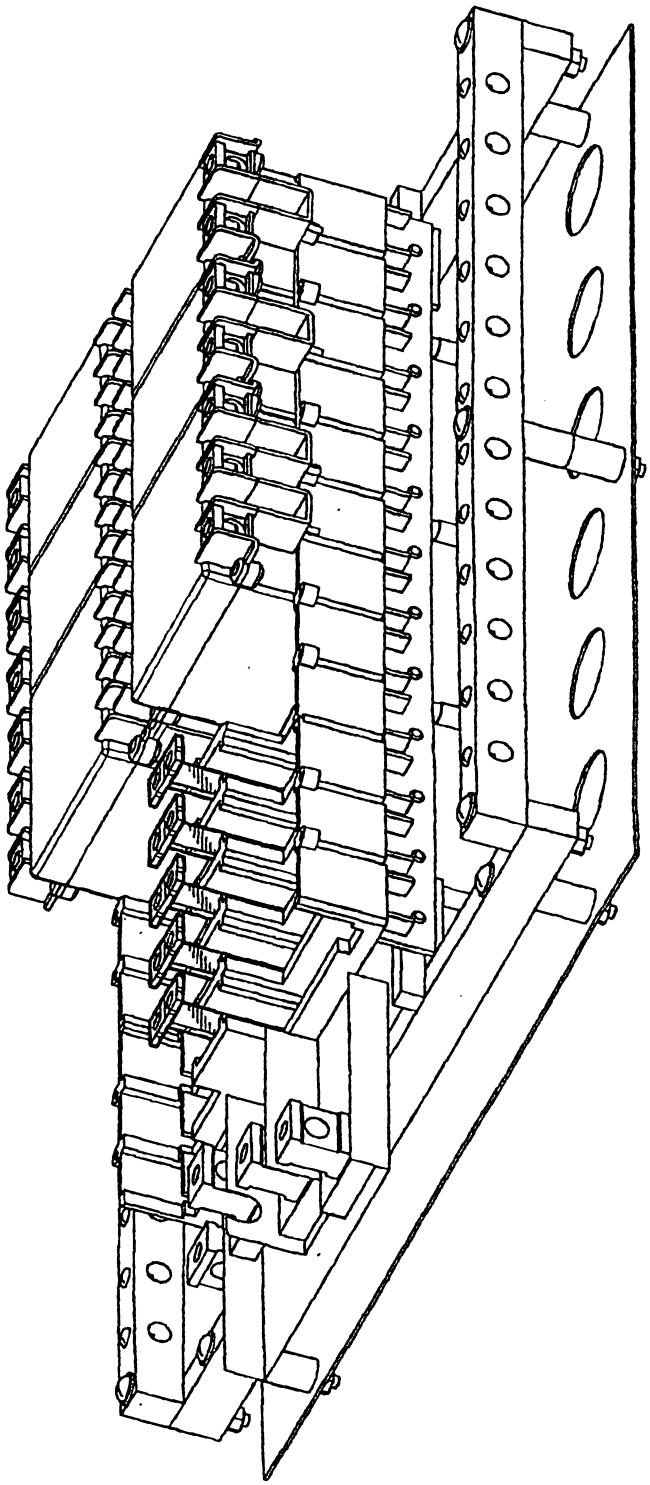


FIG.16

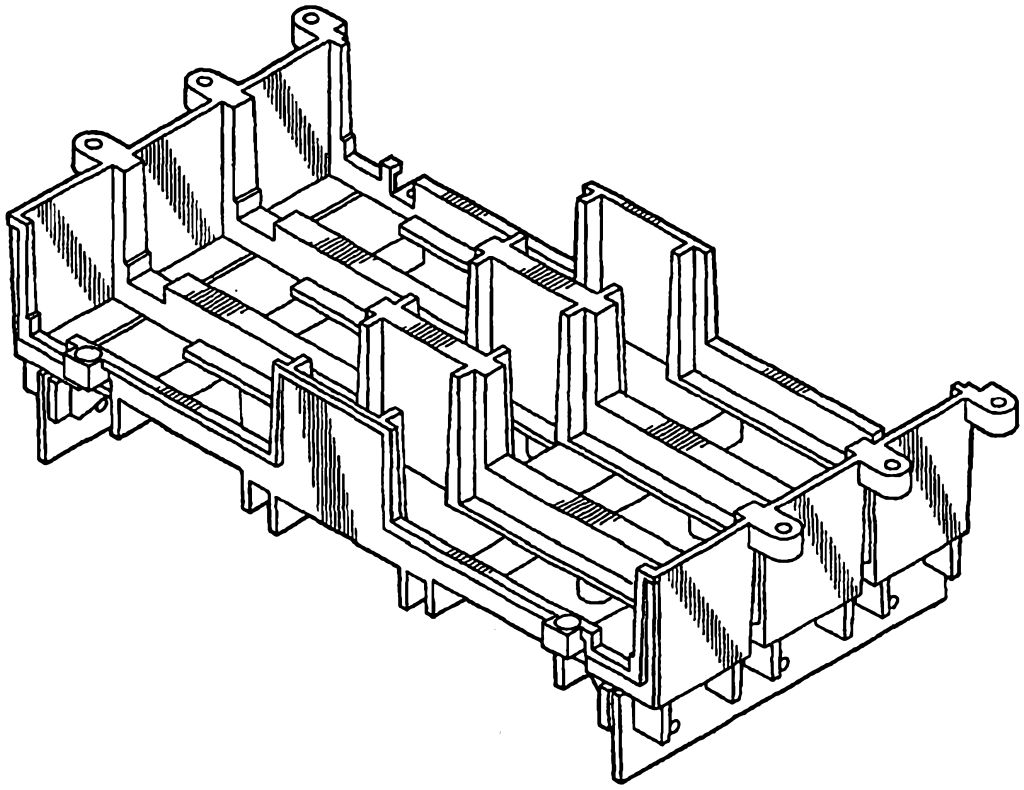


FIG.17

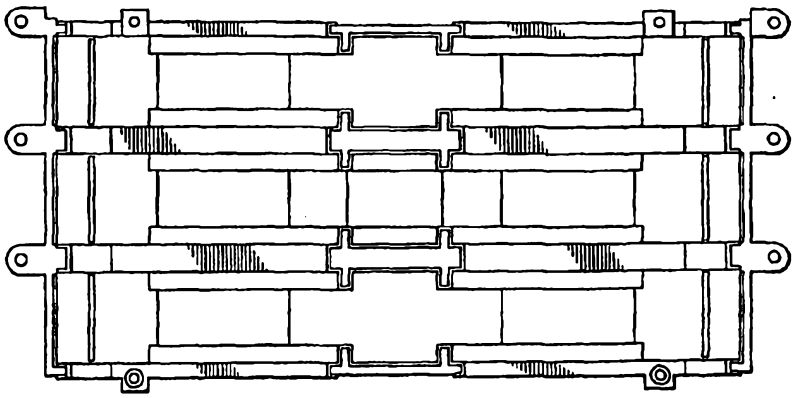


FIG.18A

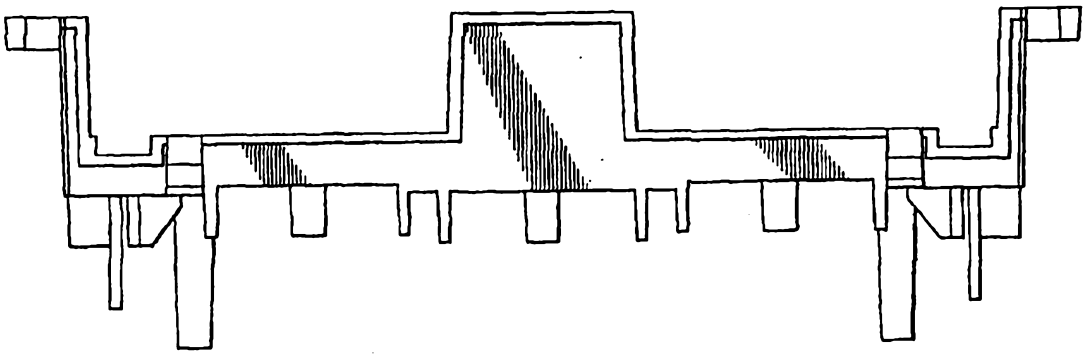


FIG.18B

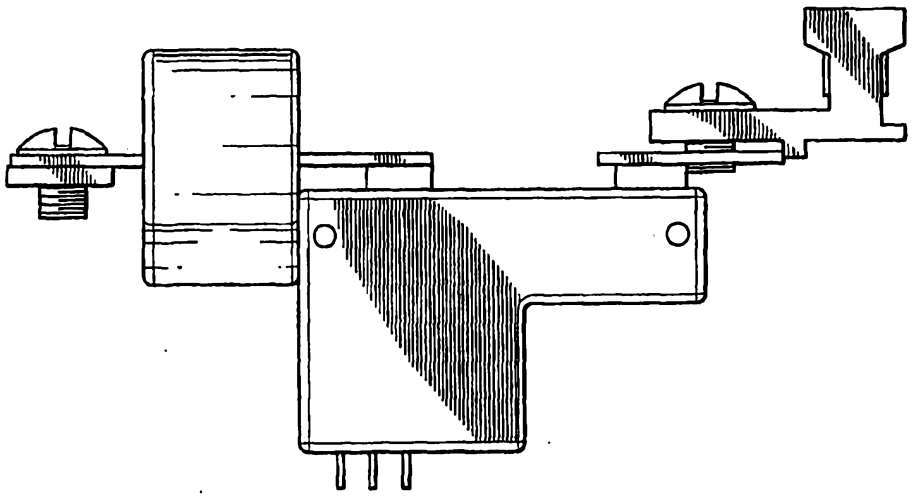


FIG.19

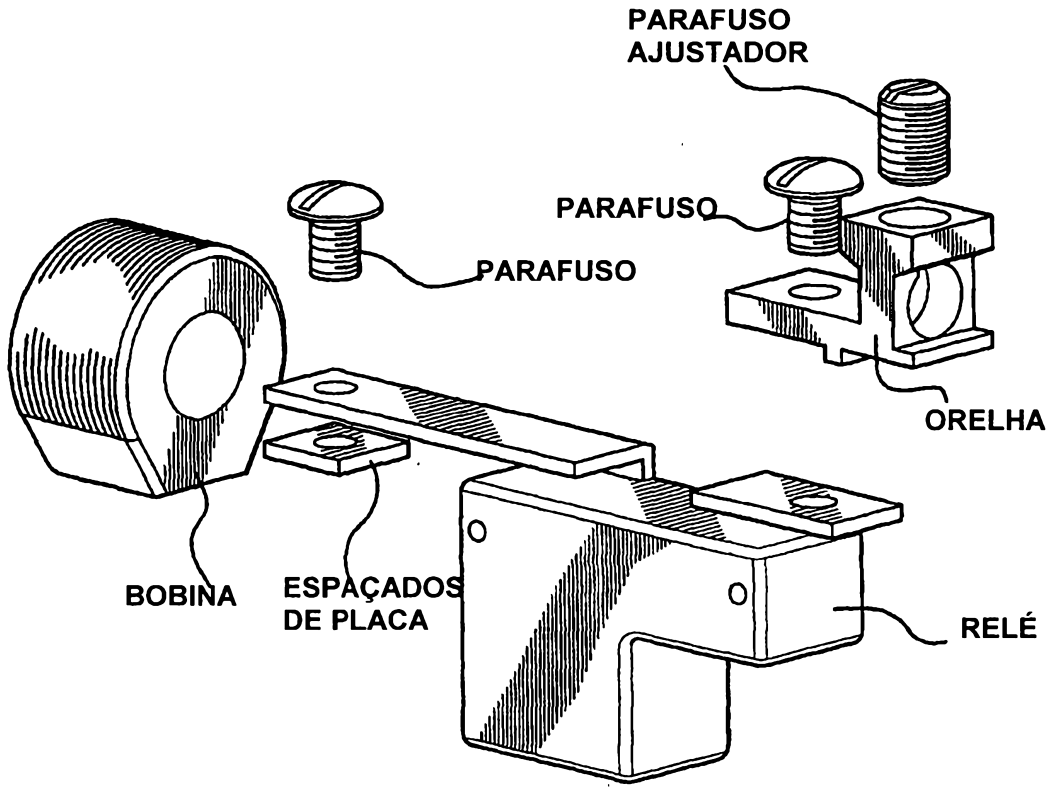

PARAFUSO
AJUSTADOR
PARAFUSO
PARAFUSO
ORELHA
BOBINA
ESPAÇADOS
DE PLACA
RELÉ

FIG.20

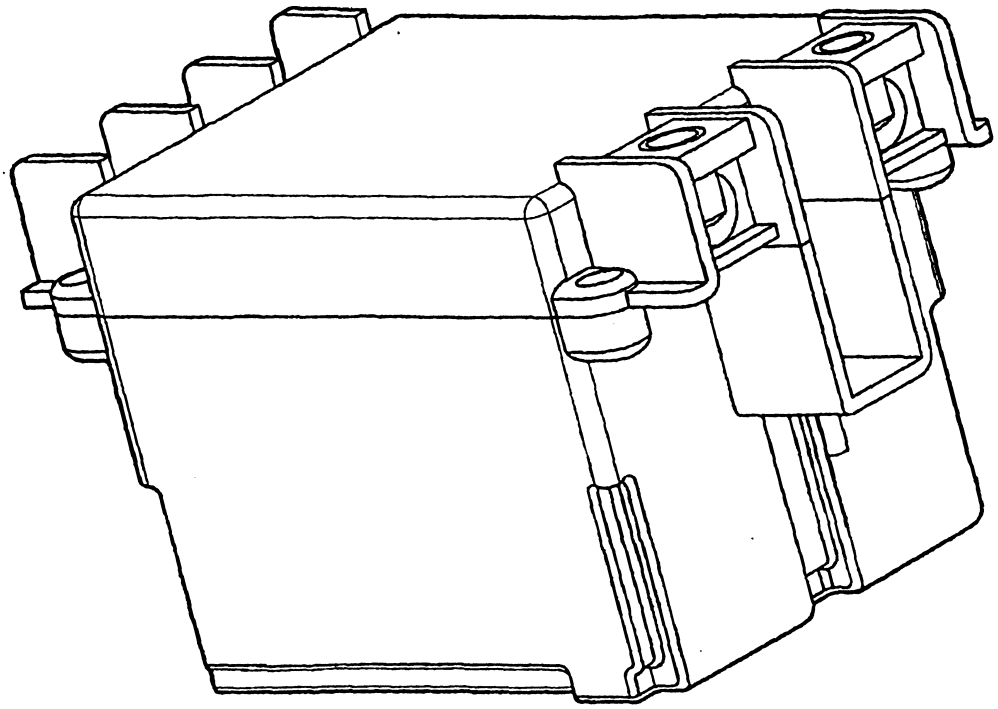


FIG.21A

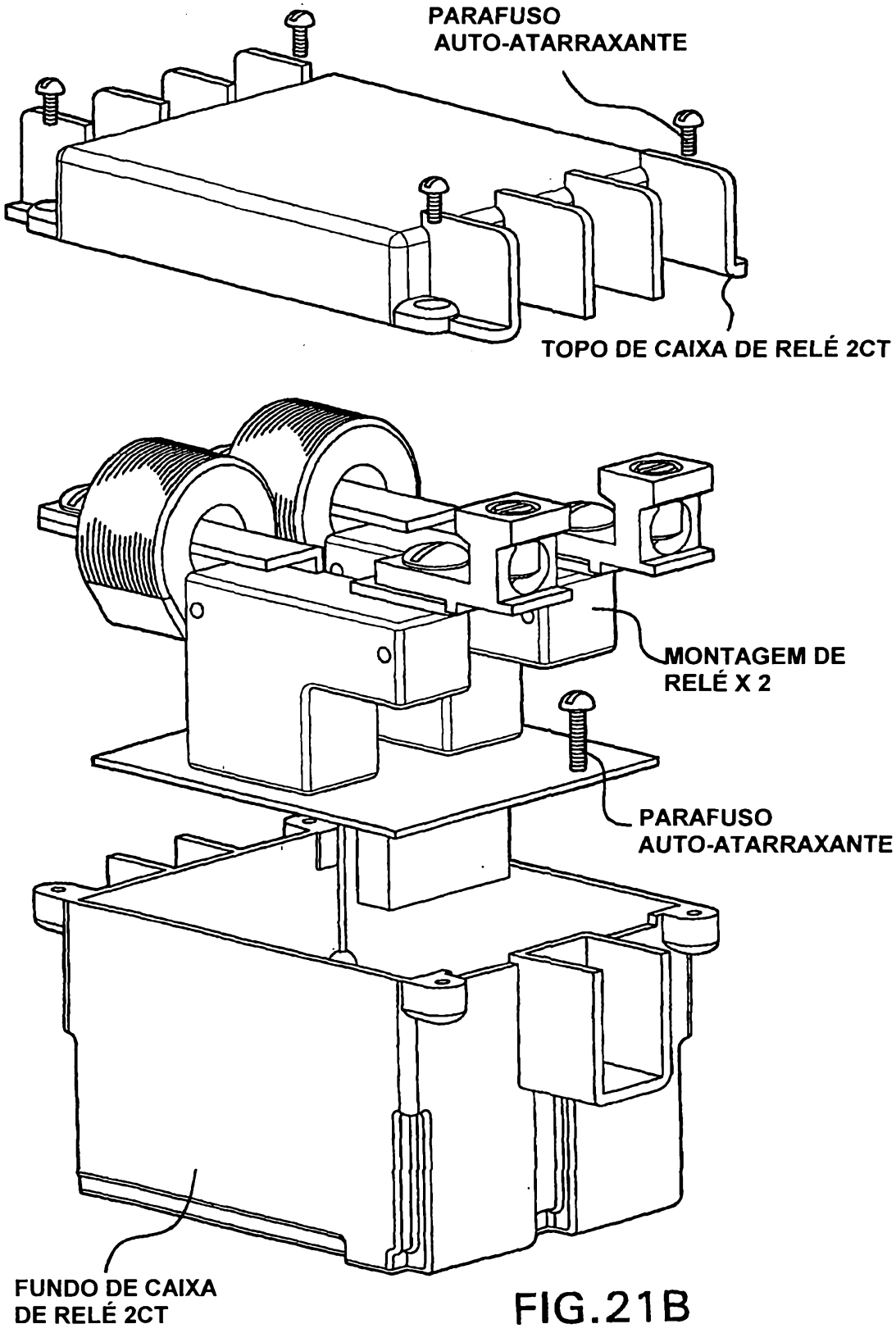

PARAFUSO
AUTO-ATARRAXANTE
TOPO DE CAIXA DE RELÉ 2CT
MONTAGEM DE
RELÉ X 2
PARAFUSO
AUTO-ATARRAXANTE
FUNDO DE CAIXA
DE RELÉ 2CT

FIG.21B

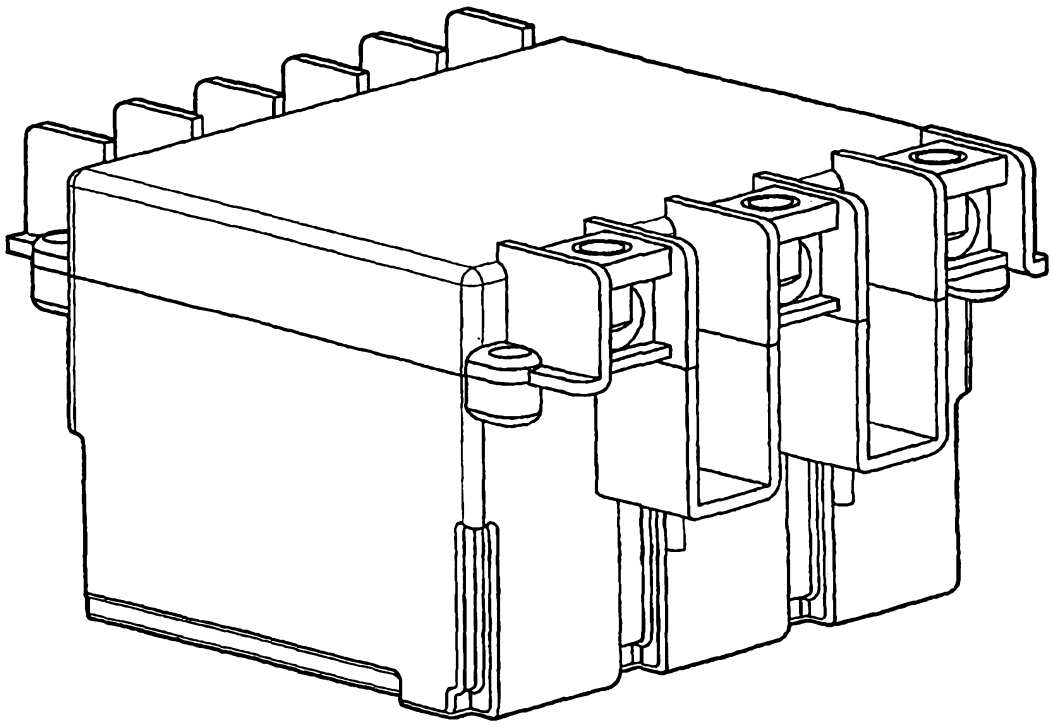


FIG.21C

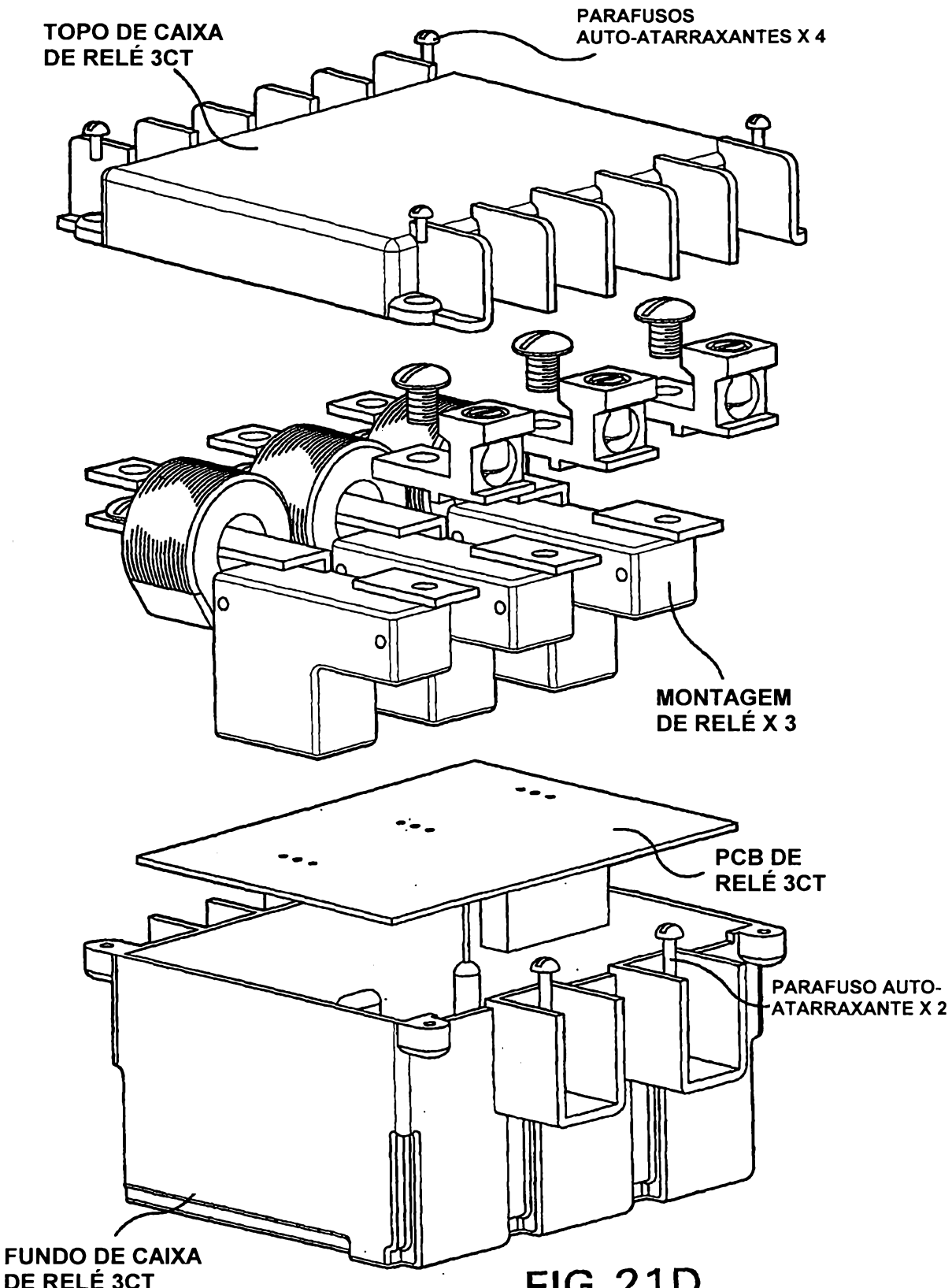

TOPO DE CAIXA
DE RELÉ 3CT
PARAFUSOS
AUTO-ATARRAXANTES X 4
MONTAGEM
DE RELÉ X 3
PCB DE
RELÉ 3CT
PARAFUSO AUTO-
ATARRAXANTE X 2
FUNDO DE CAIXA
DE RELÉ 3CT

FIG.21D

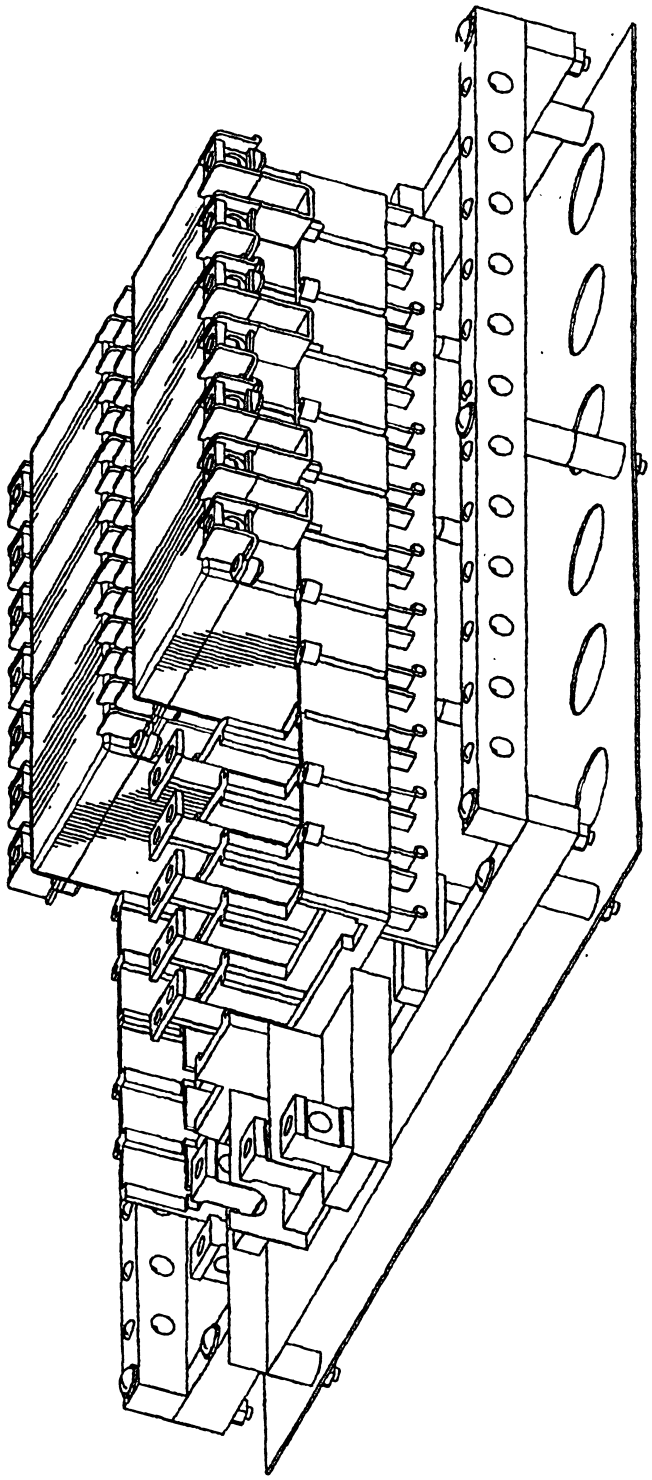


FIG.22

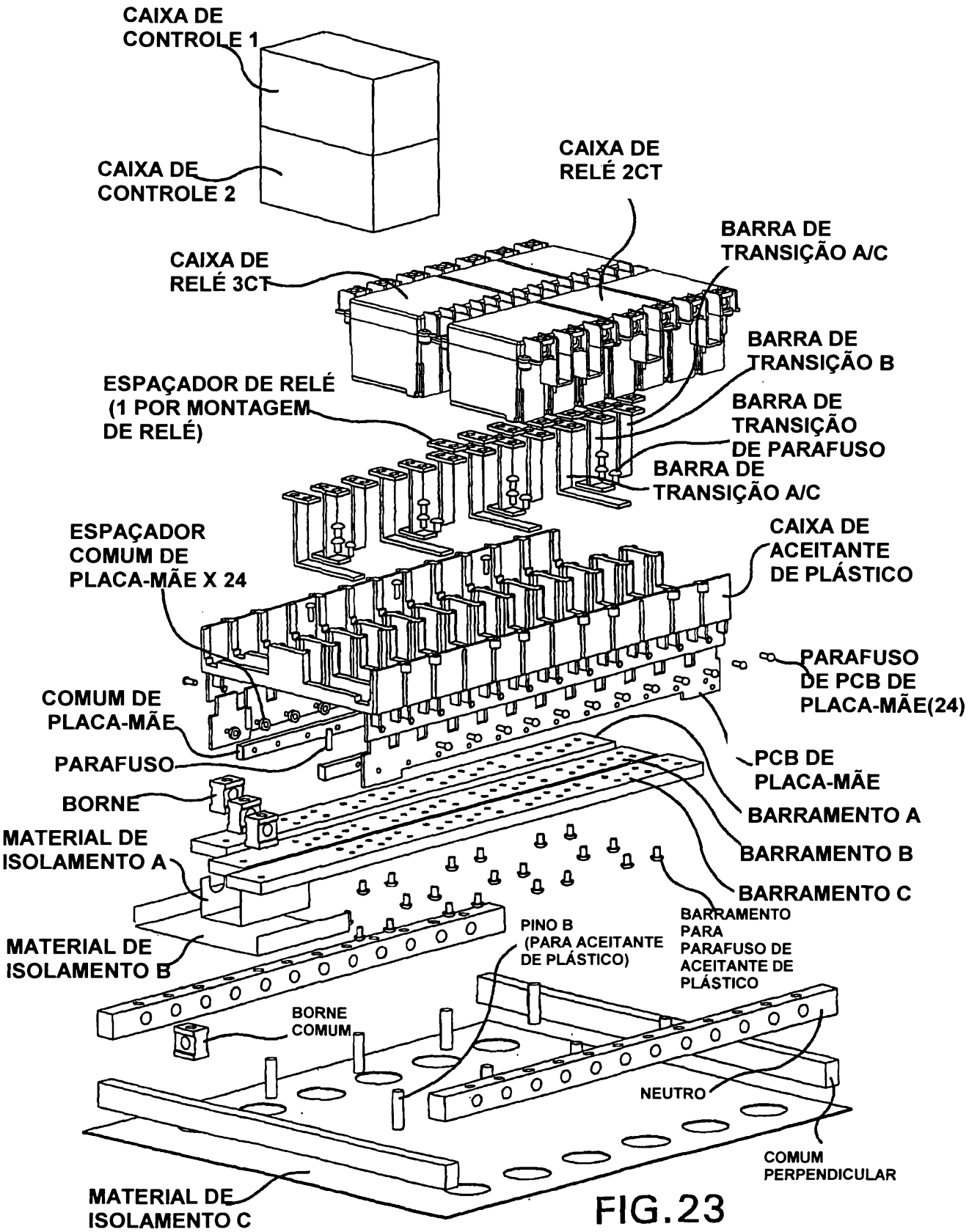

CAIXA DE CONTROLE 1
CAIXA DE CONTROLE 2
CAIXA DE RELÉ 2CT
BARRA DE TRANSIÇÃO A/C
CAIXA DE RELÉ 3CT
BARRA DE TRANSIÇÃO B
ESPAÇADOR DE RELÉ (1 POR MONTAGEM DE RELÉ)
BARRA DE TRANSIÇÃO DE PARAFUSO
BARRA DE TRANSIÇÃO A/C
ESPAÇADOR COMUM DE PLACA-MÃE X 24
CAIXA DE ACEITANTE DE PLÁSTICO
PARAFUSO DE PCB DE PLACA-MÃE(24)
COMUM DE PLACA-MÃE
PARAFUSO
PCB DE PLACA-MÃE
BORNE
BARRAMENTO A
MATERIAL DE ISOLAMENTO A
BARRAMENTO B
BARRAMENTO C
BARRAMENTO PARA PARAFUSO DE ACEITANTE DE PLÁSTICO
PINO B (PARA ACEITANTE DE PLÁSTICO)
MATERIAL DE ISOLAMENTO B
BORNE COMUM
NEUTRO
COMUM PERPENDICULAR
MATERIAL DE ISOLAMENTO C

FIG.23

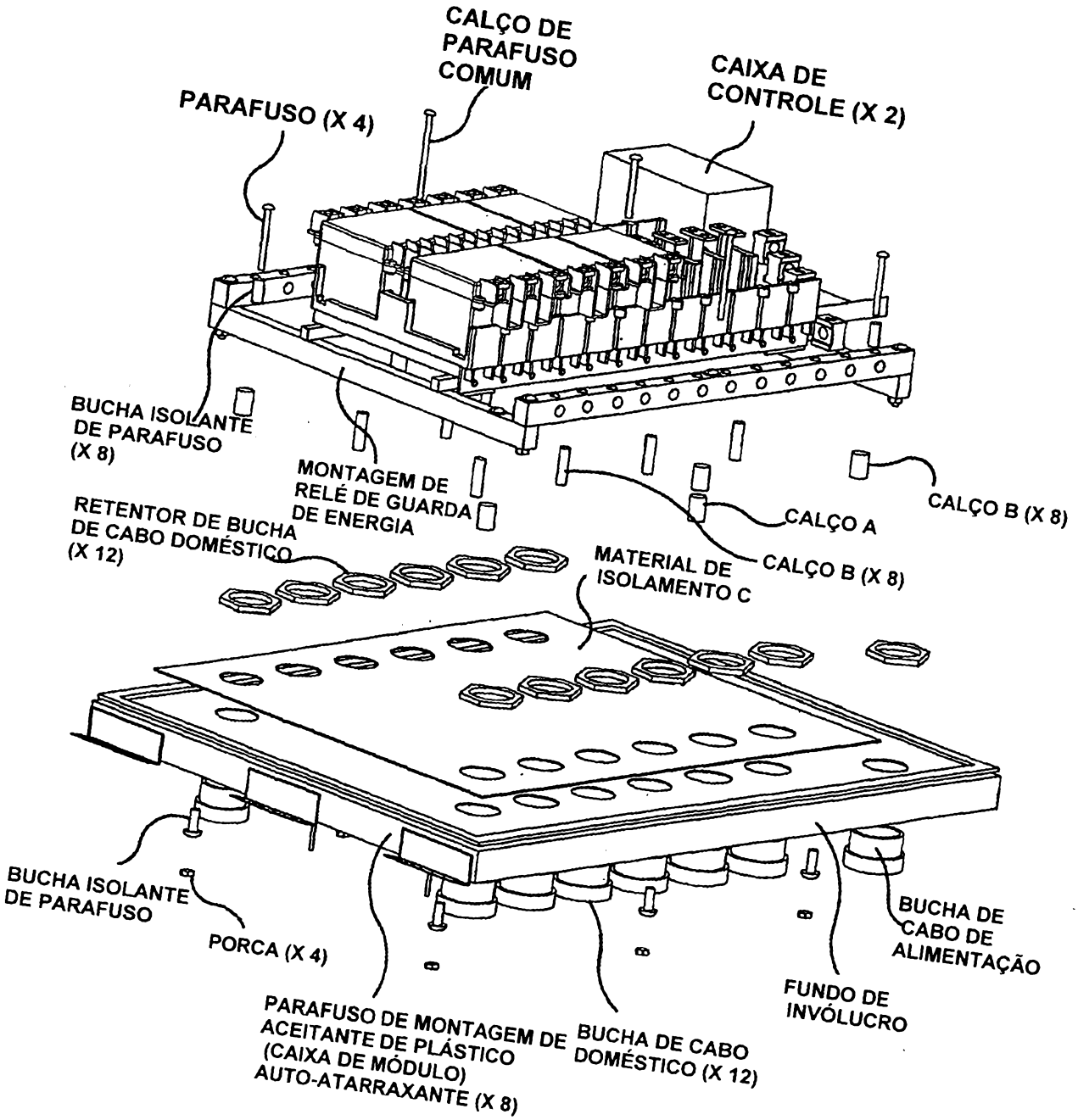

CALÇO DE PARAFUSO COMUM
PARAFUSO (X 4)
CAIXA DE CONTROLE (X 2)
BUCHA ISOLANTE DE PARAFUSO (X 8)
MONTAGEM DE RELÉ DE GUARDA DE ENERGIA
CALÇO A
CALÇO B (X 8)
RETENTOR DE BUCHA DE CABO DOMÉSTICO (X 12)
CALÇO B (X 8)
MATERIAL DE ISOLAMENTO C
BUCHA ISOLANTE DE PARAFUSO
PORCA (X 4)
PARAFUSO DE MONTAGEM DE ACEITANTE DE PLÁSTICO (CAIXA DE MÓDULO) AUTO-ATARRAXANTE (X 8)
BUCHA DE CABO DOMÉSTICO (X 12)
FUNDO DE INVÓLUCRO
BUCHA DE CABO DE ALIMENTAÇÃO

FIG.24

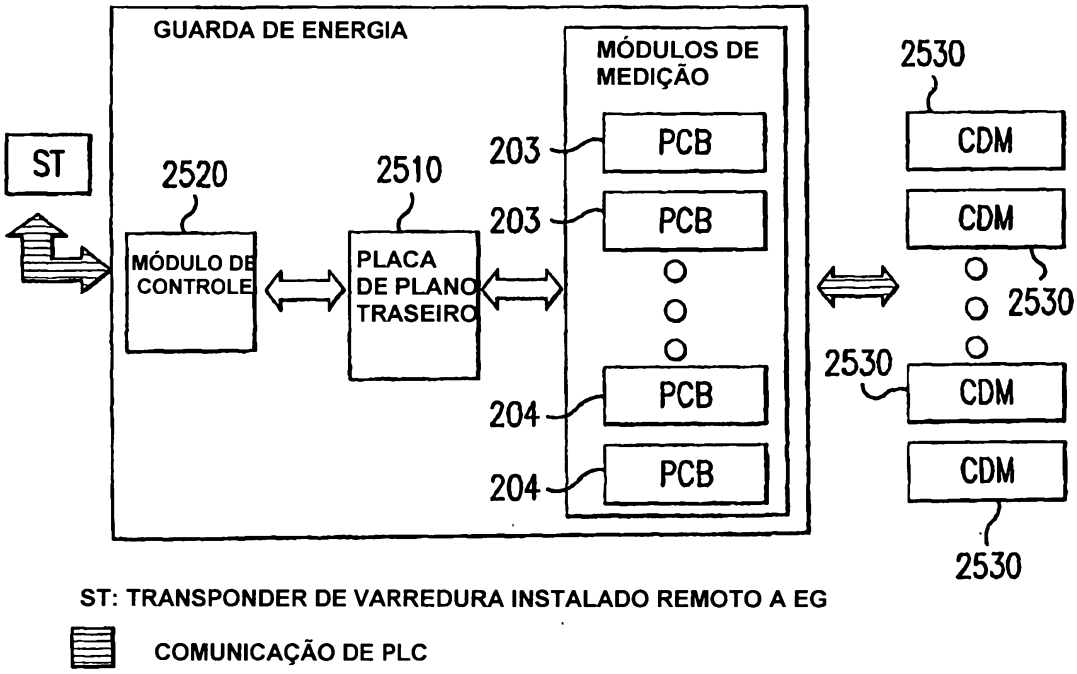

GUARDA DE ENERGIA
MÓDULOS DE MEDIÇÃO
ST
2520
2510
203
203
204
204
MÓDULO DE CONTROLE
PLACA DE PLANO TRASEIRO
PCB
PCB
PCB
PCB
2530
CDM
CDM
2530
2530
CDM
CDM
2530
ST: TRANSPONDER DE VARREDURA INSTALADO REMOTO A EG
COMUNICAÇÃO DE PLC

FIG.25

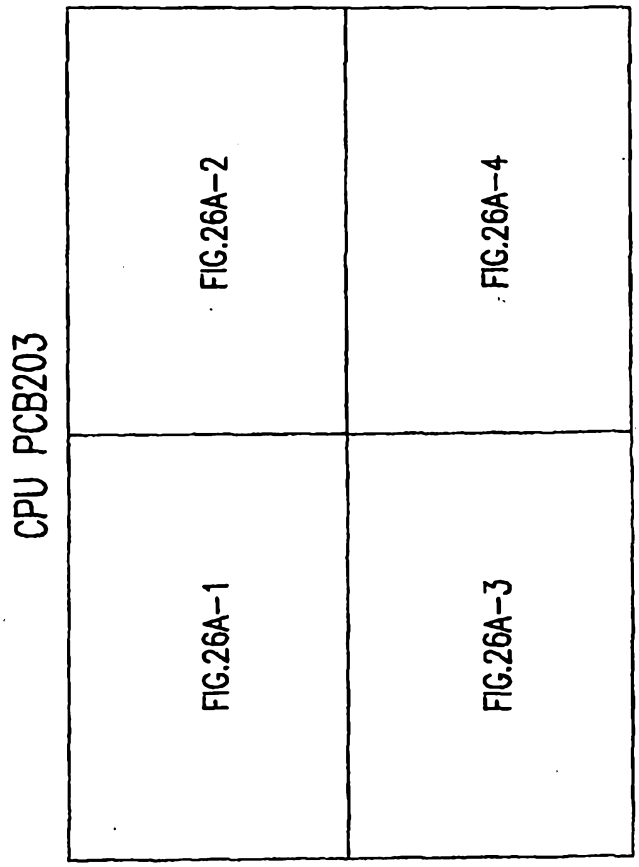


FIG.26A

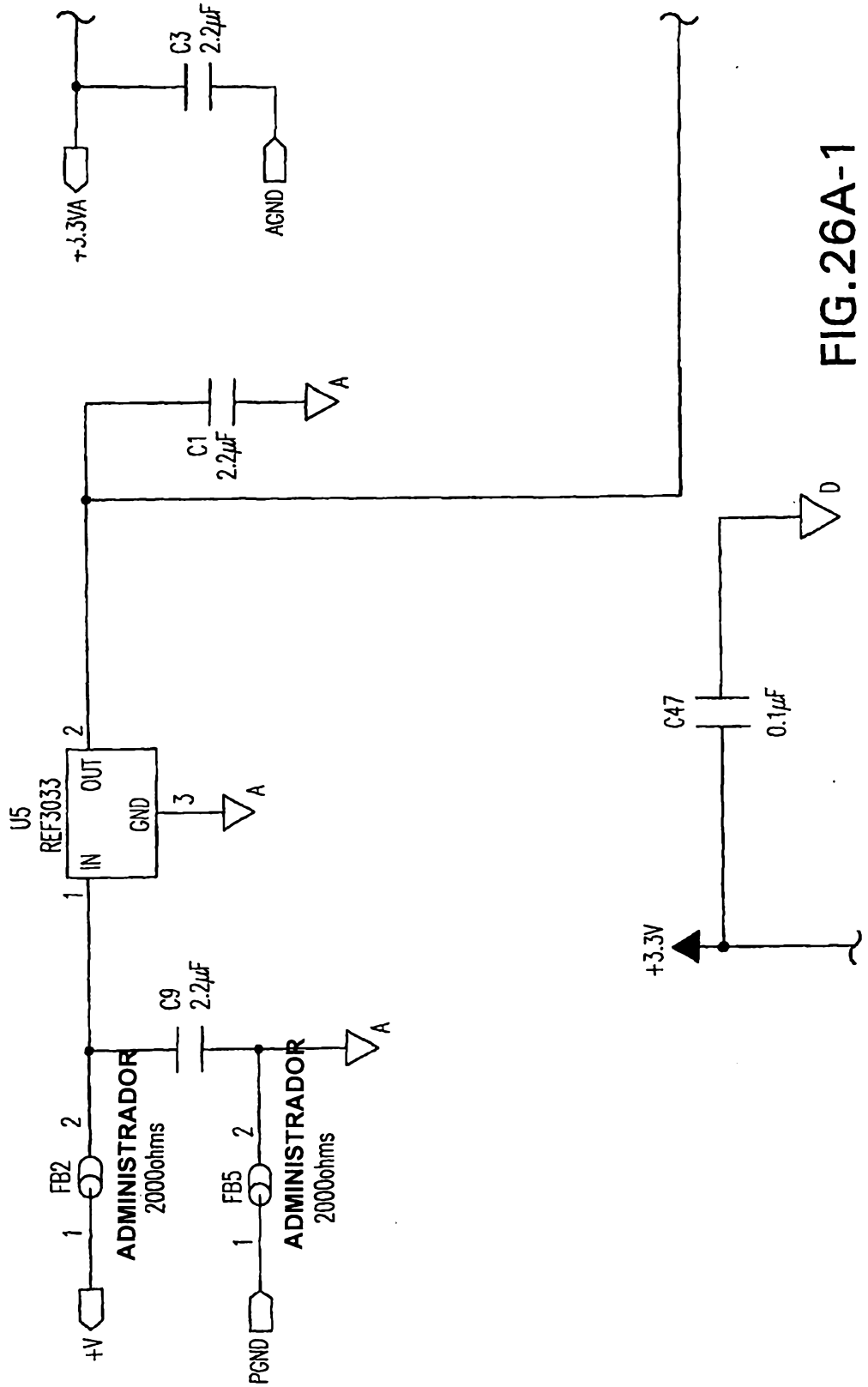

+3.3VA
C3
2.2µF
AGND
C1
2.2µF
A
U5
REF3033
IN
OUT
GND
2
3
1
A
C9
2.2µF
A
FB2
2
1
ADMINISTRADOR
2000ohms
+V
FB5
2
1
ADMINISTRADOR
2000ohms
PGND
D
C47
0.1µF
+3.3V

FIG.26A-1

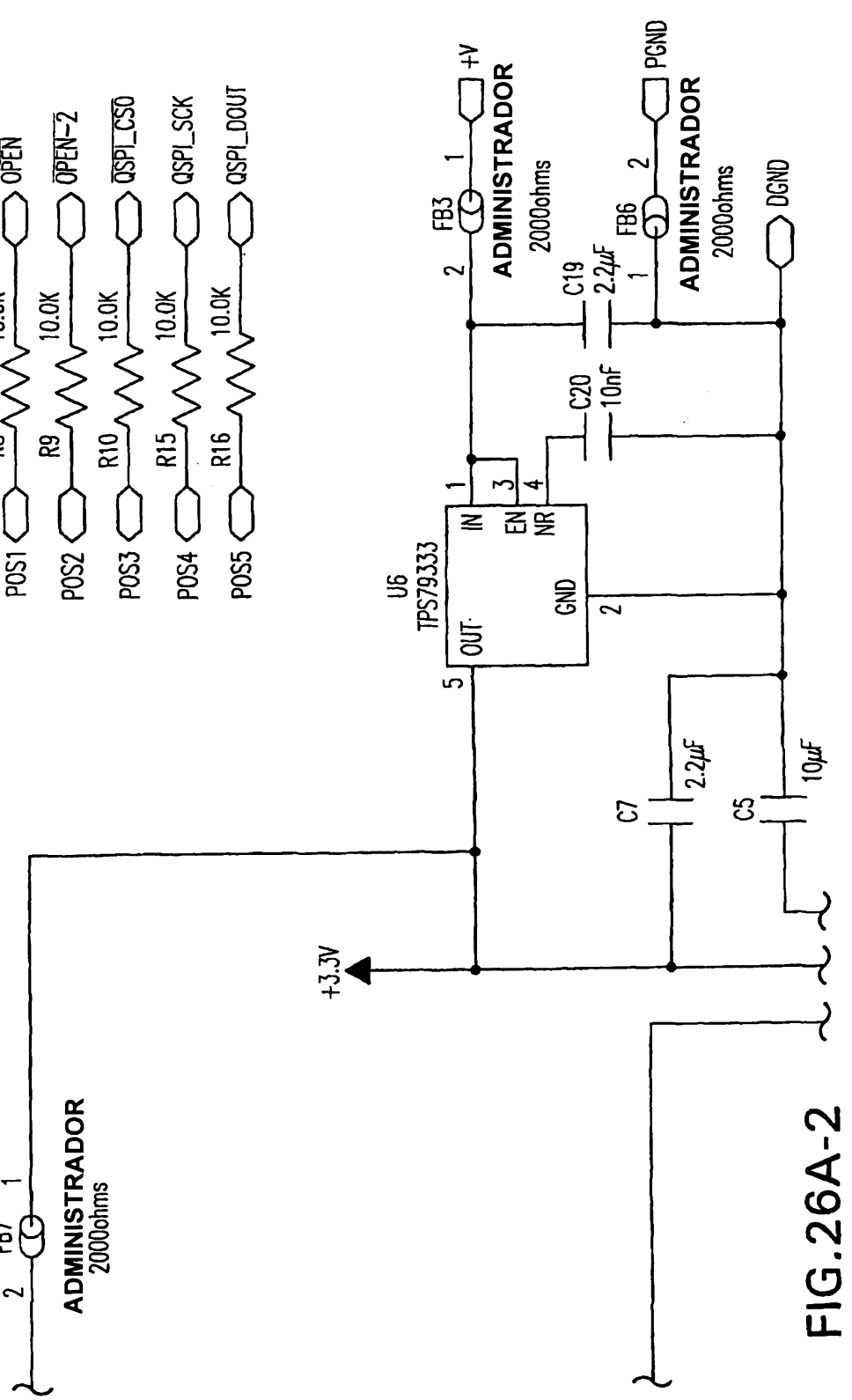

POS0
POS1
POS2
POS3
POS4
POS5
R7
R8
R9
R10
R15
R16
10.0K
CLOSE
OPEN
OPEN-2
QSPI_CS0
QSPI_SCK
QSPI_DOUT
FB7
ADMINISTRADOR
2000ohms
+3.3V
U6
TPS79333
IN
EN
NR
GND
OUT
FB3
+V
FB6
PGND
DGND
C19
2.2µF
C20
10nF
C7
2.2µF
C5
10µF

FIG.26A-2

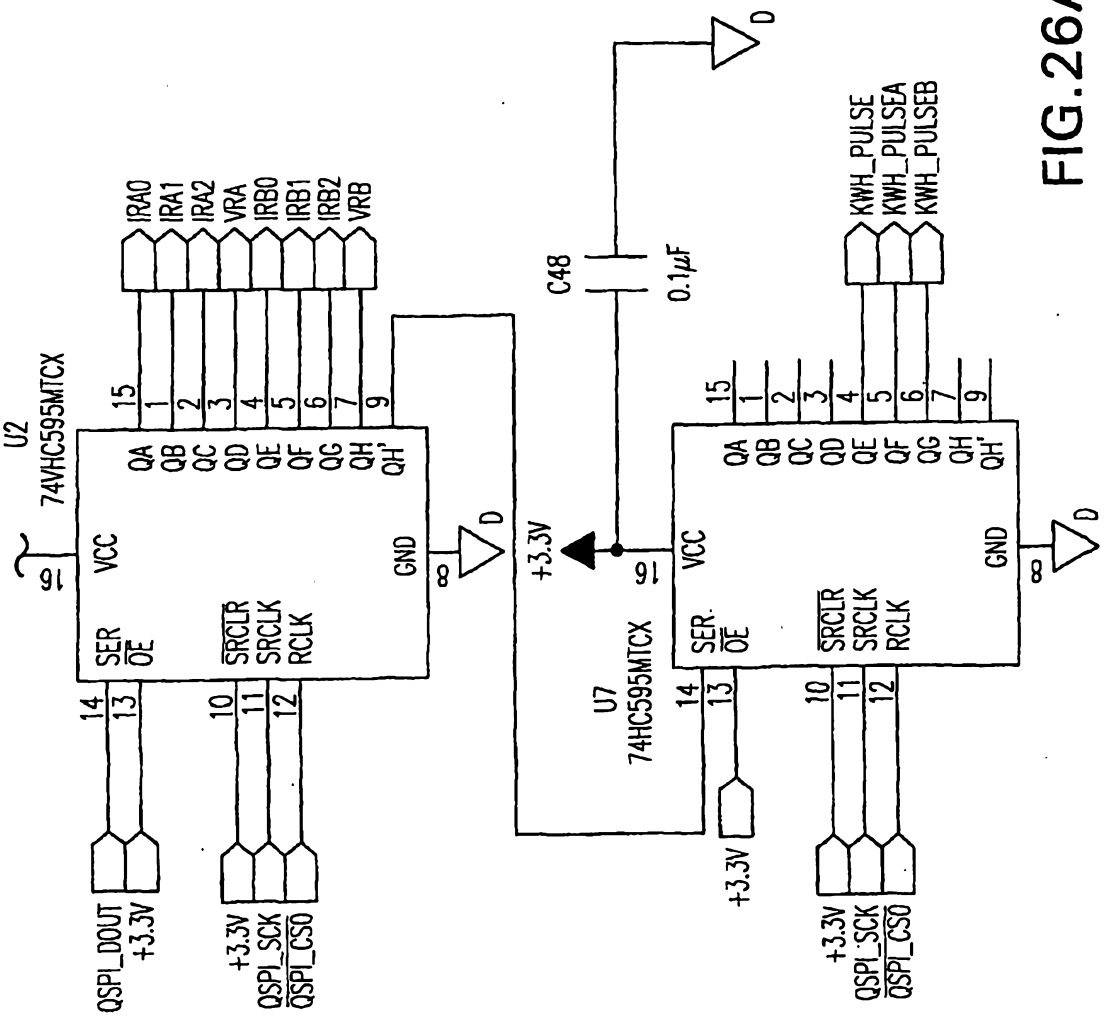

U2
74VHC595MTCX
U7
74HC595MTCX
VCC
GND
SER
OE
SRCLR
SRCLK
RCLK
QA
QB
QC
QD
QE
QF
QG
QH
QH'
IRA0
IRA1
IRA2
VRA
IRB0
IRB1
IRB2
VRB
C48
0.1μF
+3.3V
QSPI_DOUT
QSPI_SCK
QSPI_CS0
KWH_PULSE
KWH_PULSEA
KWH_PULSEB

FIG.26A-3

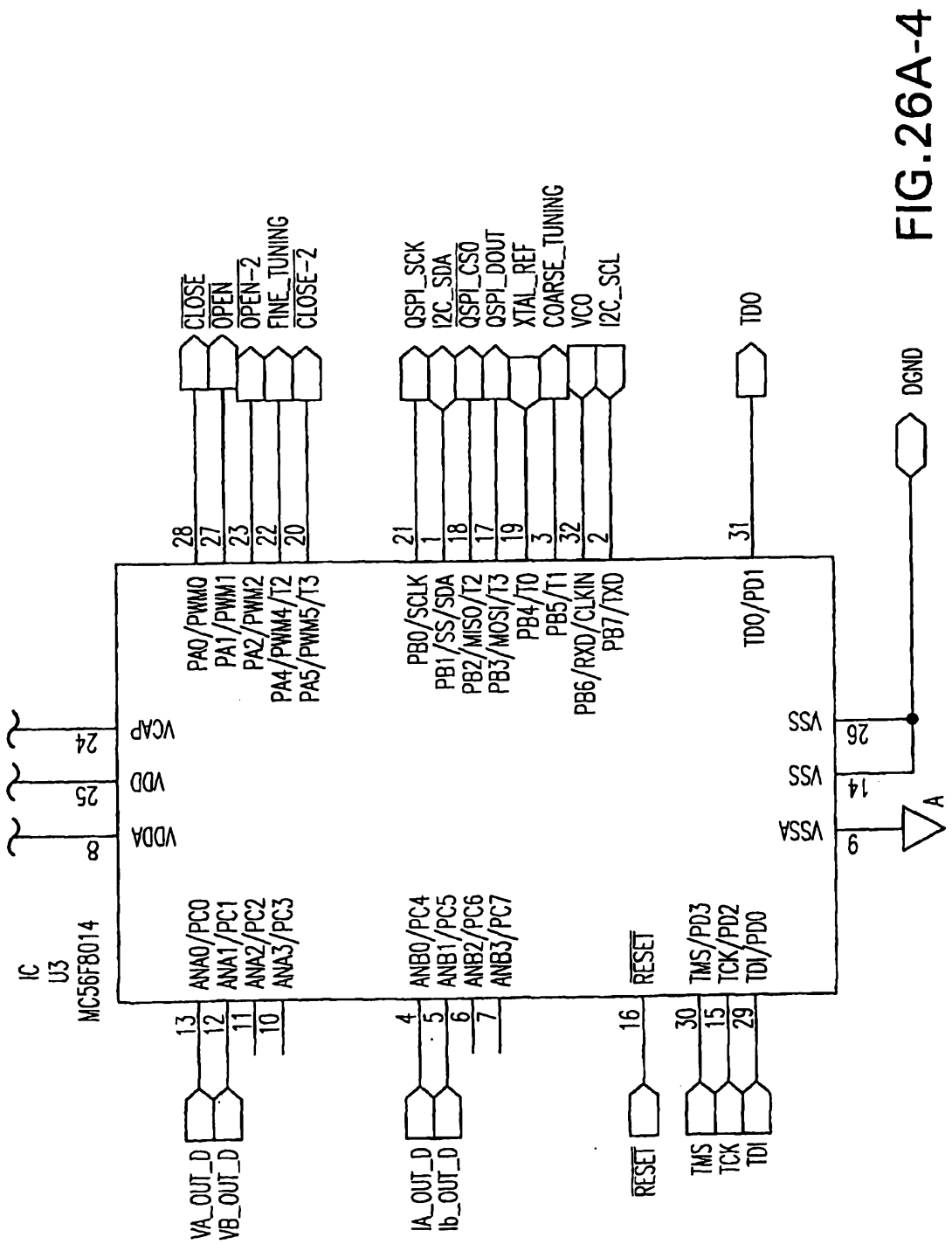

IC
U3
MC56F8014
VCAP
VDD
VDDA
VSS
VSS
VSSA
PA0/PWM0
PA1/PWM1
PA2/PWM2
PA4/PWM4/T2
PA5/PWM5/T3
PB0/SCLK
PB1/SS/SDA
PB2/MISO/T2
PB3/MOSI/T3
PB4/T0
PB5/T1
PB6/RXD/CLKIN
PB7/TXD
TDO/PD1
ANA0/PC0
ANA1/PC1
ANA2/PC2
ANA3/PC3
ANB0/PC4
ANB1/PC5
ANB2/PC6
ANB3/PC7
RESET
TMS/PD3
TCK/PD2
TDI/PD0
CLOSE
OPEN
OPEN-2
FINE_TUNING
CLOSE-2
QSPI_SCK
I2C_SDA
QSPI_CS0
QSPI_DOUT
XTAL_REF
COARSE_TUNING
VCO
I2C_SCL
TDO
DGND
A
VA_OUT_D
VB_OUT_D
IA_OUT_D
Ib_OUT_D
RESET
TMS
TCK
TDI

FIG.26A-4

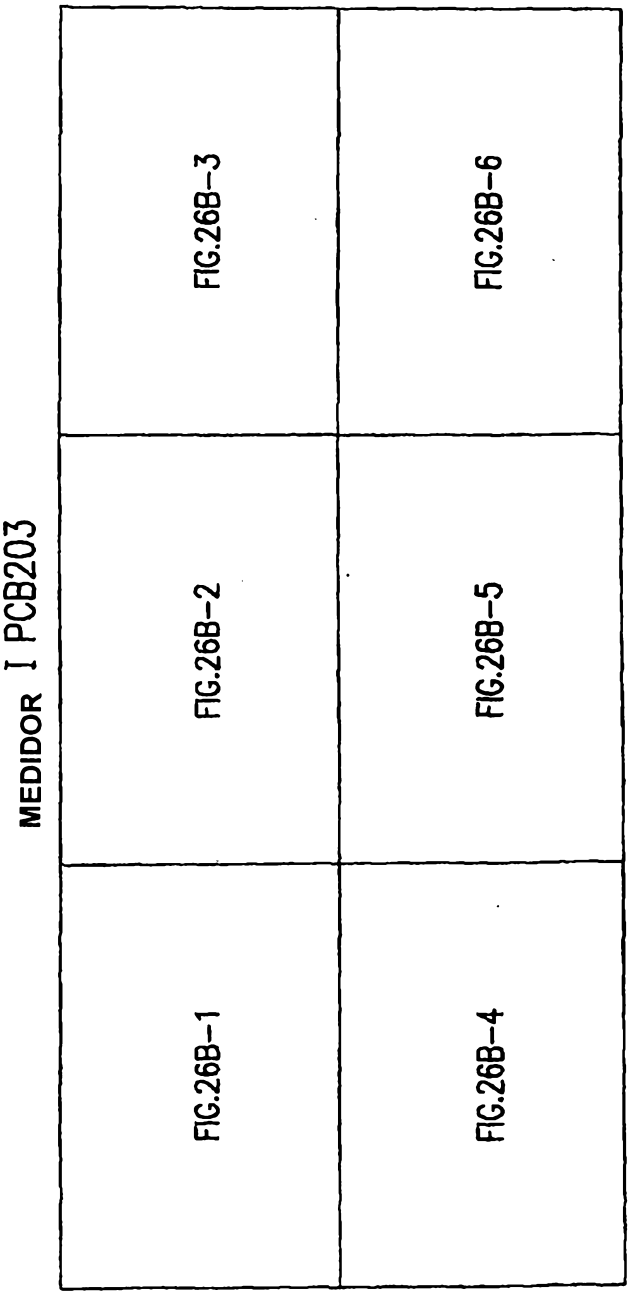

MEDIDOR I PCB203
FIG.26B-1
FIG.26B-2
FIG.26B-3
FIG.26B-4
FIG.26B-5
FIG.26B-6

FIG.26B

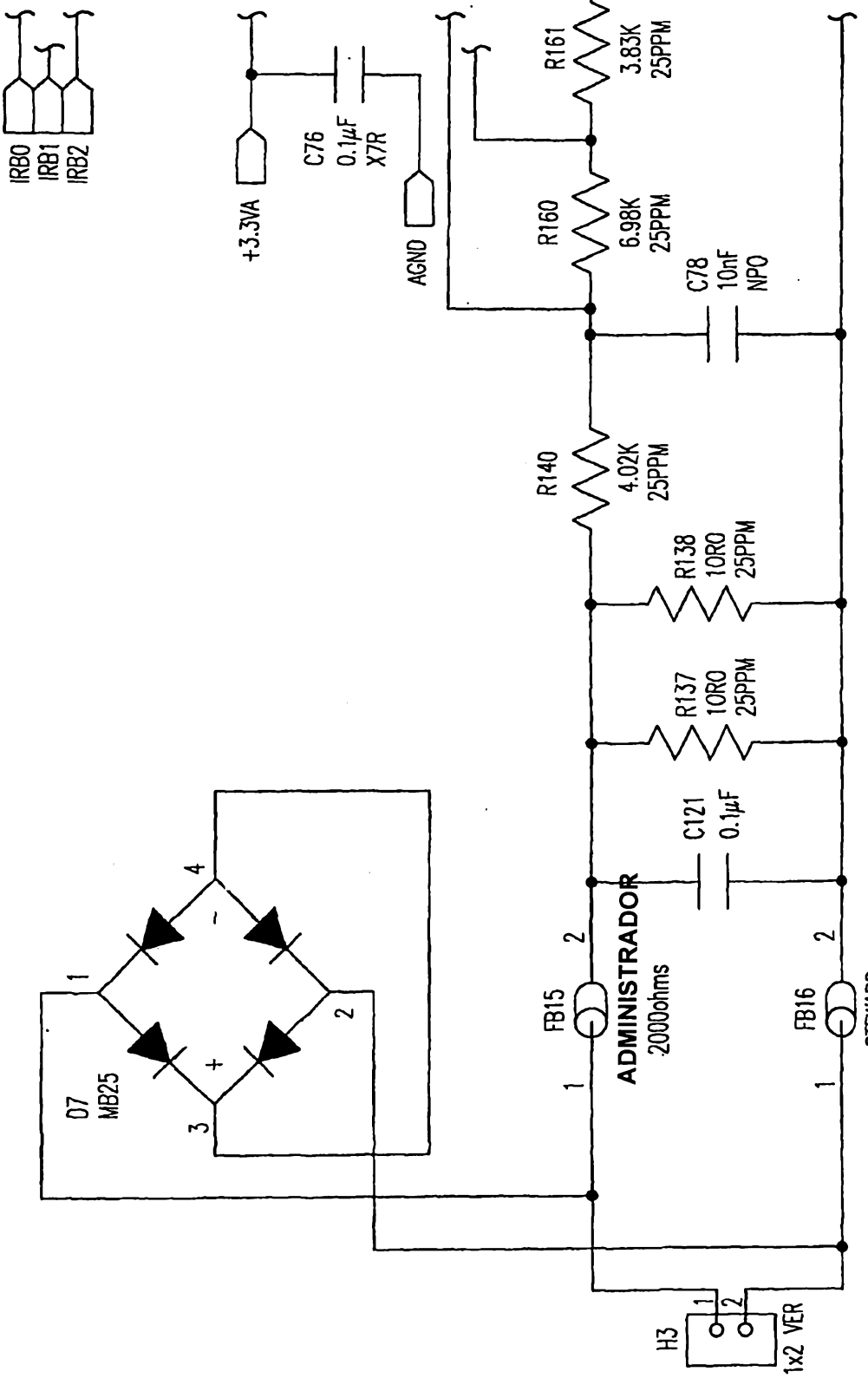

IRB0
IRB1
IRB2
+3.3VA
C76
0.1µF
X7R
AGND
R161
3.83K
25PPM
R160
6.98K
25PPM
C78
10nF
NPO
R140
4.02K
25PPM
R138
10R0
25PPM
R137
10R0
25PPM
C121
0.1µF
D7
MB25
FB15
FB16
ADMINISTRADOR
2000ohms
H3
1x2 VER

FIG.26B-1

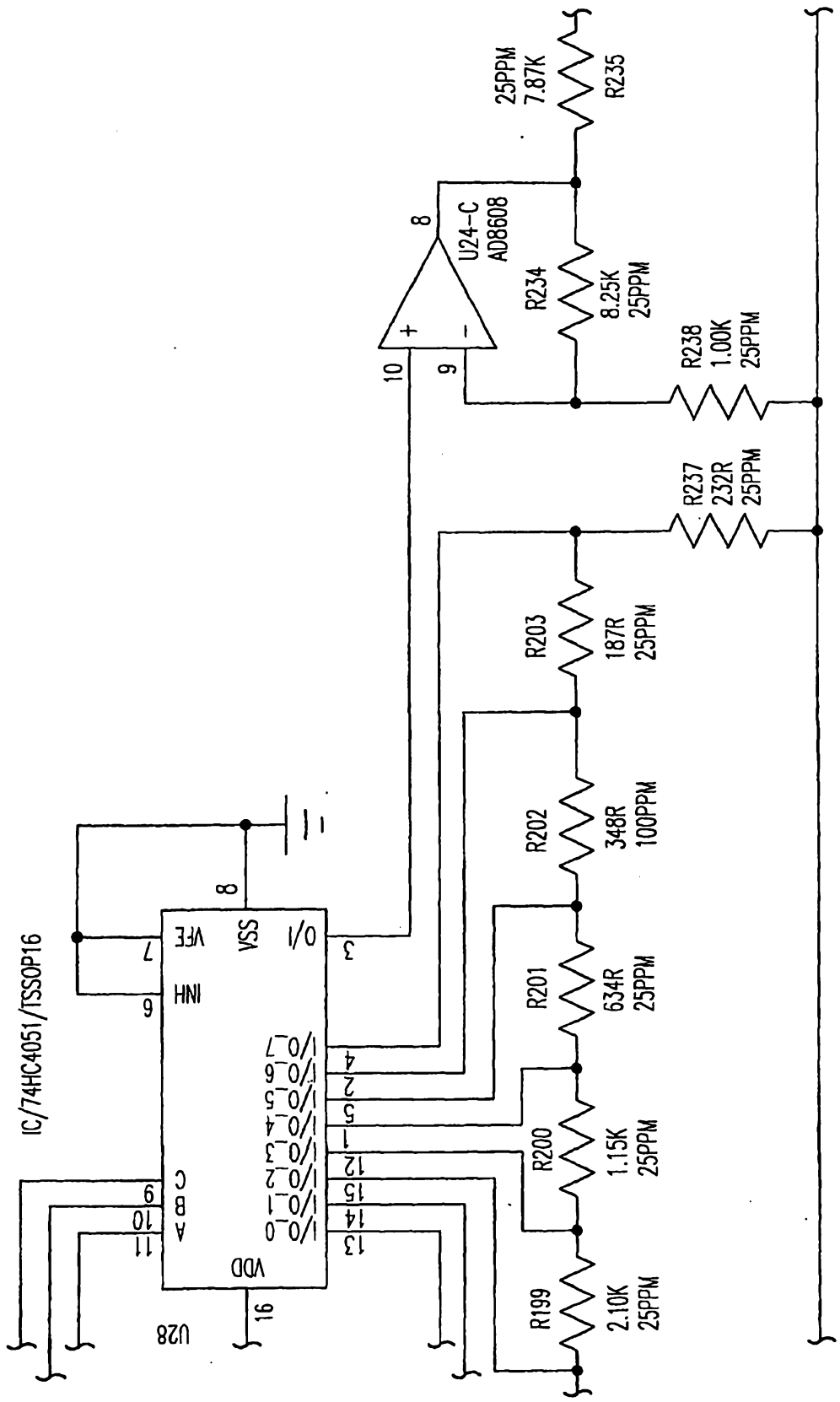

IC/74HC4051/TSSOP16
U28
VDD
VSS
VEE
INH
A
B
C
I/O
I/O_0
I/O_1
I/O_2
I/O_3
I/O_4
I/O_5
I/O_6
I/O_7
U24-C
AD8608
R199
2.10K
25PPM
R200
1.15K
25PPM
R201
634R
25PPM
R202
348R
100PPM
R203
187R
25PPM
R237
232R
25PPM
R238
1.00K
25PPM
R234
8.25K
25PPM
R235
7.87K
25PPM

FIG.26B-2

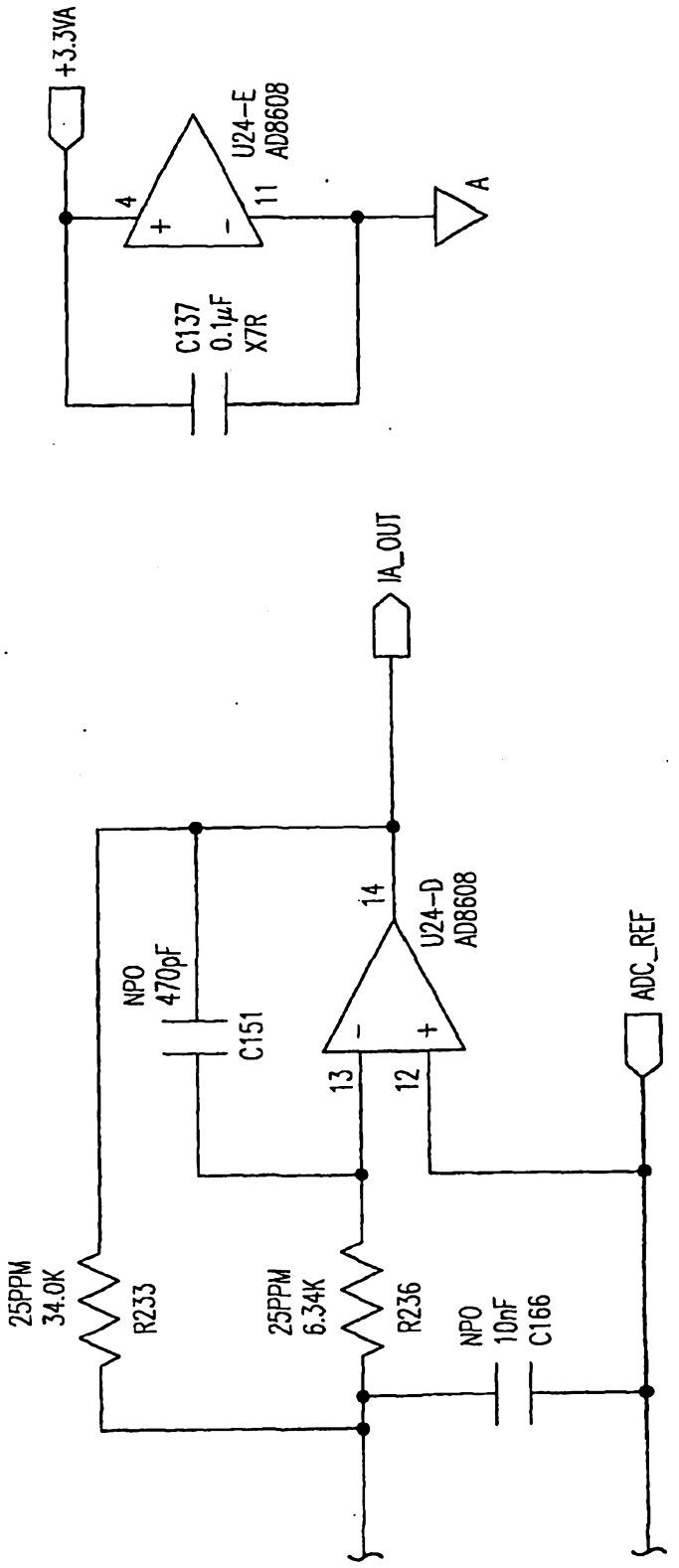

+3.3VA
U24-E
AD8608
4
11
A
C137
0.1µF
X7R
IA_OUT
14
U24-D
AD8608
NPO
470pF
C151
13
12
ADC_REF
25PPM
34.0K
R233
25PPM
6.34K
R236
NPO
10nF
C166

FIG.26B-3

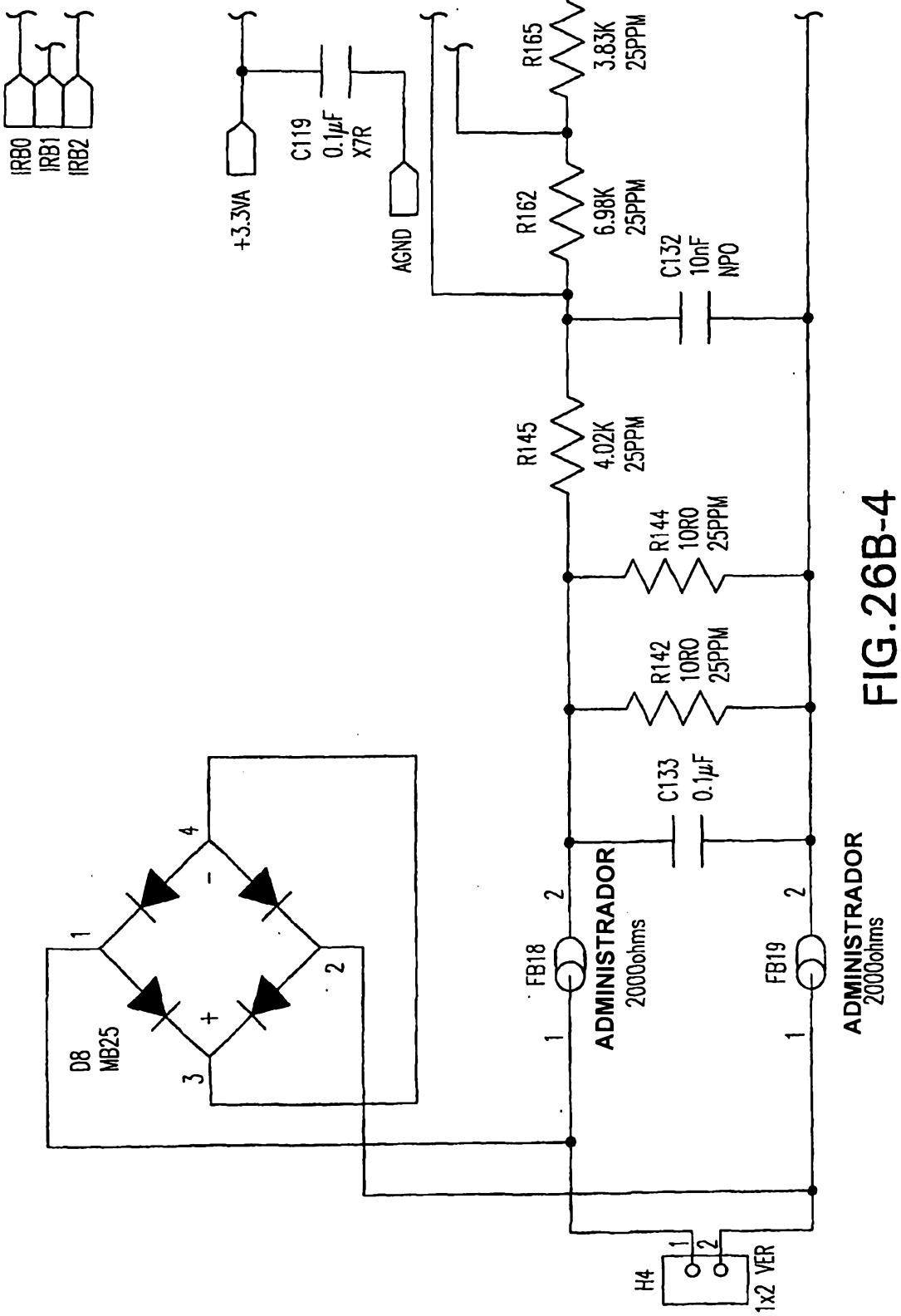

IRB0
IRB1
IRB2
+3.3VA
C119
0.1µF
X7R
AGND
R165
3.83K
25PPM
R162
6.98K
25PPM
C132
10nF
NPO
R145
4.02K
25PPM
R144
10RO
25PPM
R142
10RO
25PPM
C133
0.1µF
D8
MB25
FB18
FB19
ADMINISTRADOR
2000ohms
ADMINISTRADOR
2000ohms
H4
1x2 VER

FIG.26B-4

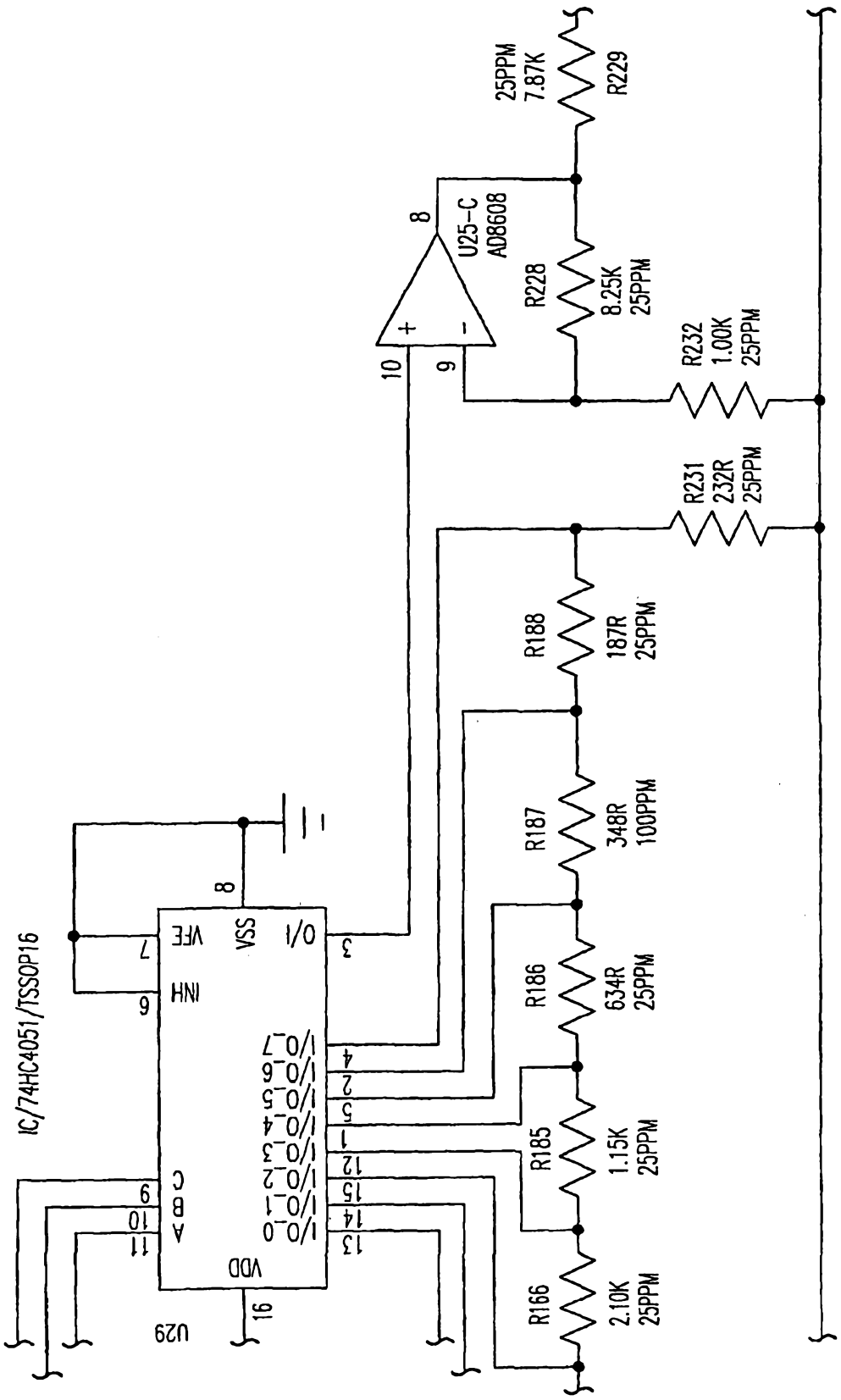

IC/74HC4051/TSSOP16
U29
VDD
VSS
VEE
INH
A
B
C
I/O
I/O_0
I/O_1
I/O_2
I/O_3
I/O_4
I/O_5
I/O_6
I/O_7
U25-C
AD8608
R166
2.10K
25PPM
R185
1.15K
25PPM
R186
634R
25PPM
R187
348R
100PPM
R188
187R
25PPM
R231
232R
25PPM
R232
1.00K
25PPM
R228
8.25K
25PPM
R229
25PPM
7.87K

FIG.26B-5

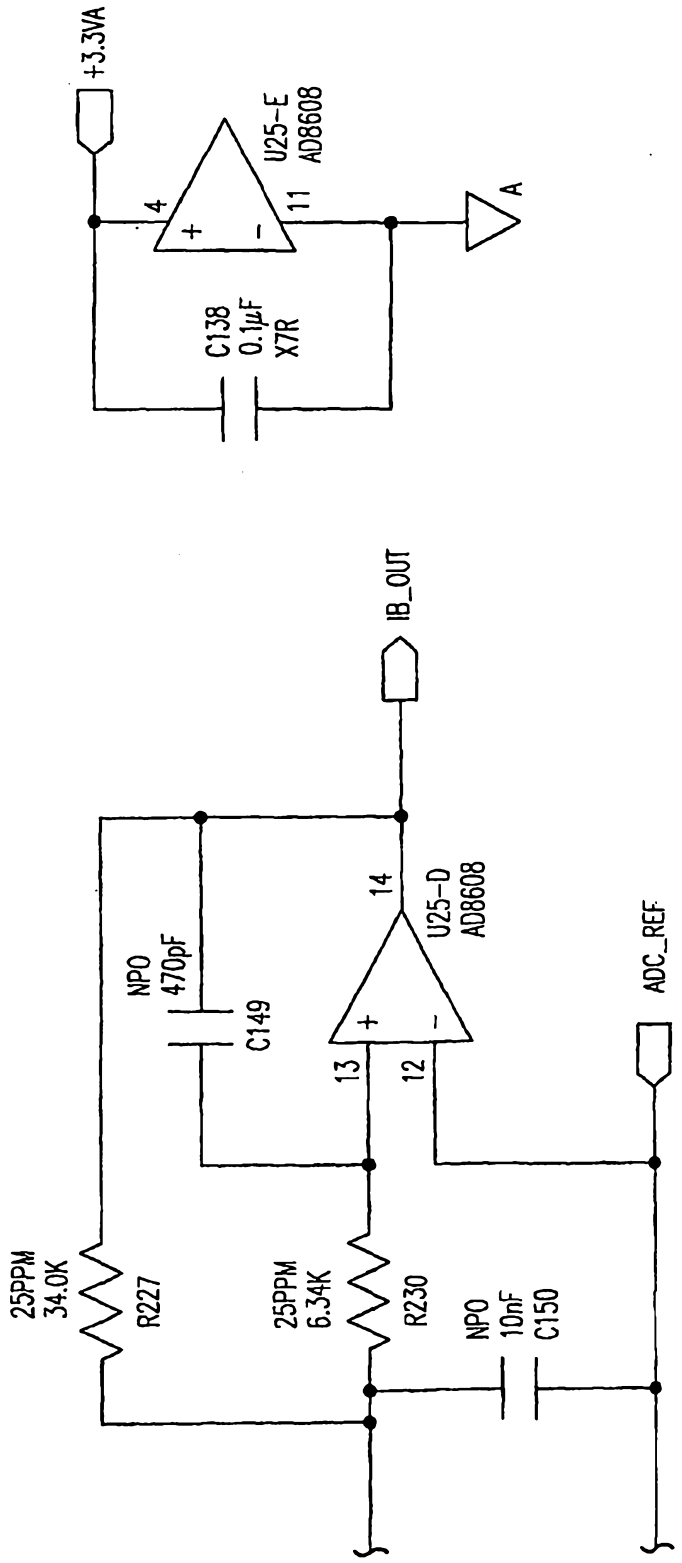

+3.3VA
U25-E
AD8608
4
11
A
C138
0.1µF
X7R
IB_OUT
14
U25-D
AD8608
NPO
470pF
C149
13
12
ADC_REF
25PPM
34.0K
R227
25PPM
6.34K
R230
NPO
10nF
C150

FIG.26B-6

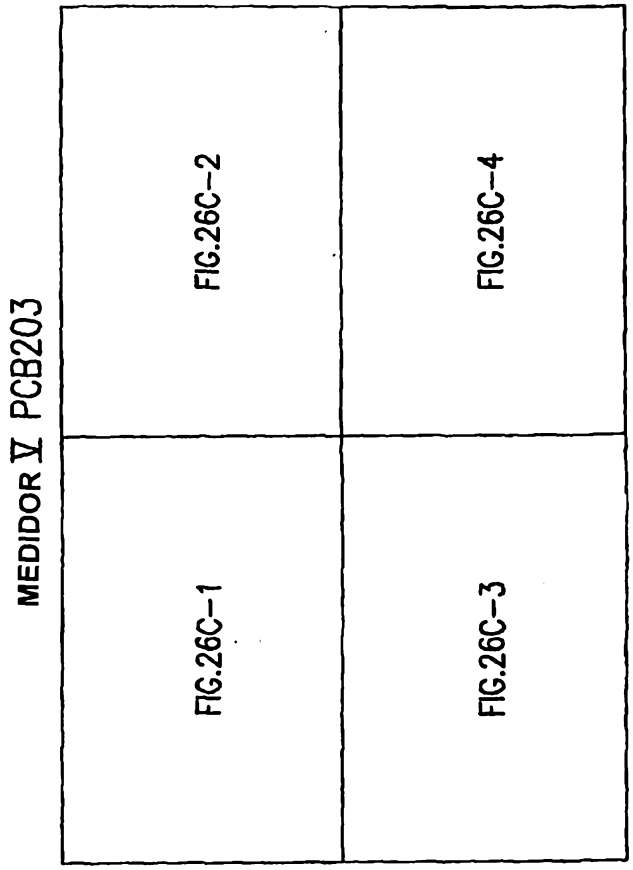


FIG.26C

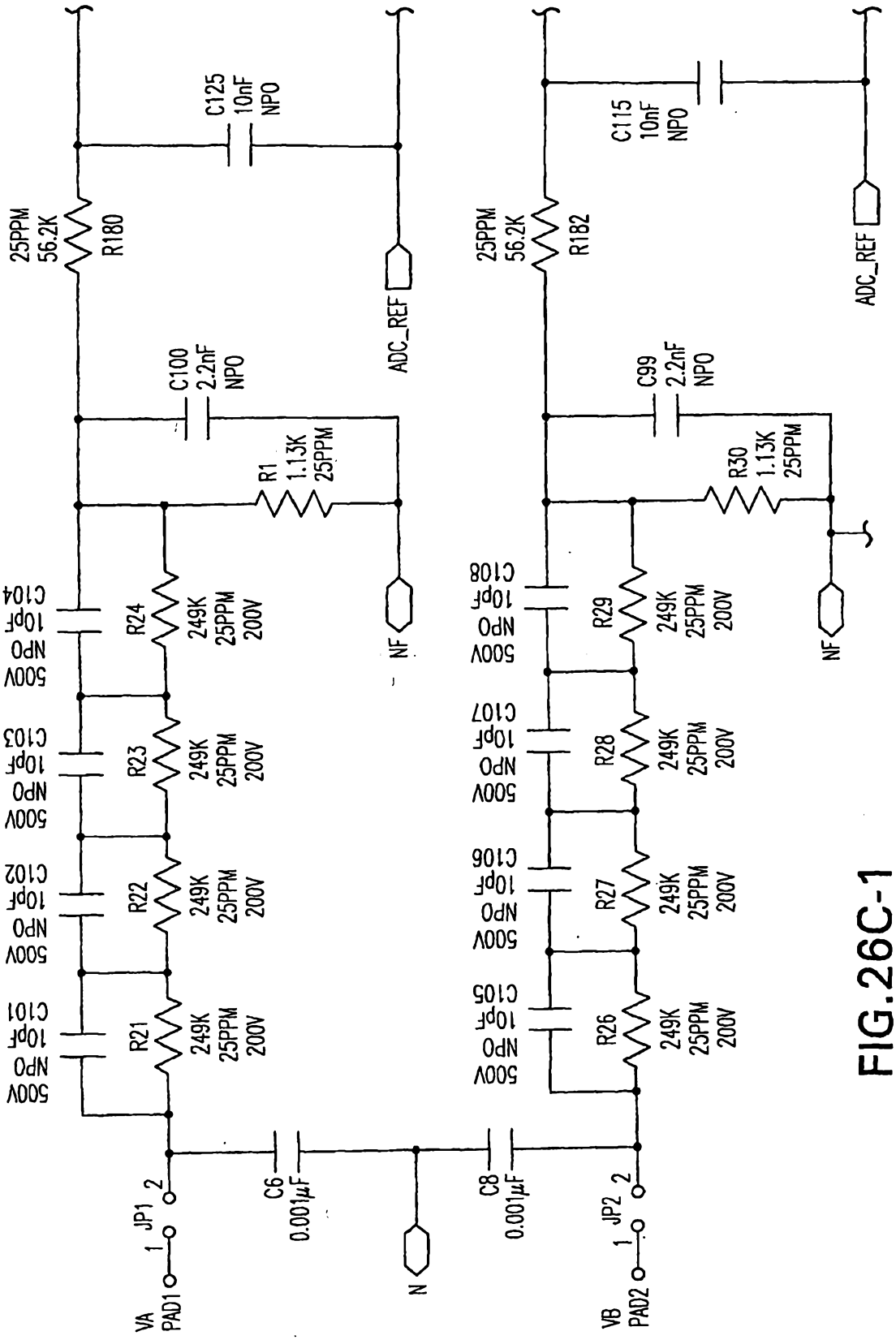

VA
PAD1
JP1
1
2
C6
0.001µF
N
C8
0.001µF
VB
PAD2
JP2
1
2
R21
249K
25PPM
200V
C101
10pF
NPO
500V
R22
249K
25PPM
200V
C102
10pF
NPO
500V
R23
249K
25PPM
200V
C103
10pF
NPO
500V
R24
249K
25PPM
200V
C104
10pF
NPO
500V
R1
1.13K
25PPM
NF
C100
2.2nF
NPO
25PPM
56.2K
R180
C125
10nF
NPO
ADC_REF
R26
249K
25PPM
200V
C105
10pF
NPO
500V
R27
249K
25PPM
200V
C106
10pF
NPO
500V
R28
249K
25PPM
200V
C107
10pF
NPO
500V
R29
249K
25PPM
200V
C108
10pF
NPO
500V
R30
1.13K
25PPM
NF
C99
2.2nF
NPO
25PPM
56.2K
R182
C115
10nF
NPO
ADC_REF

FIG.26C-1

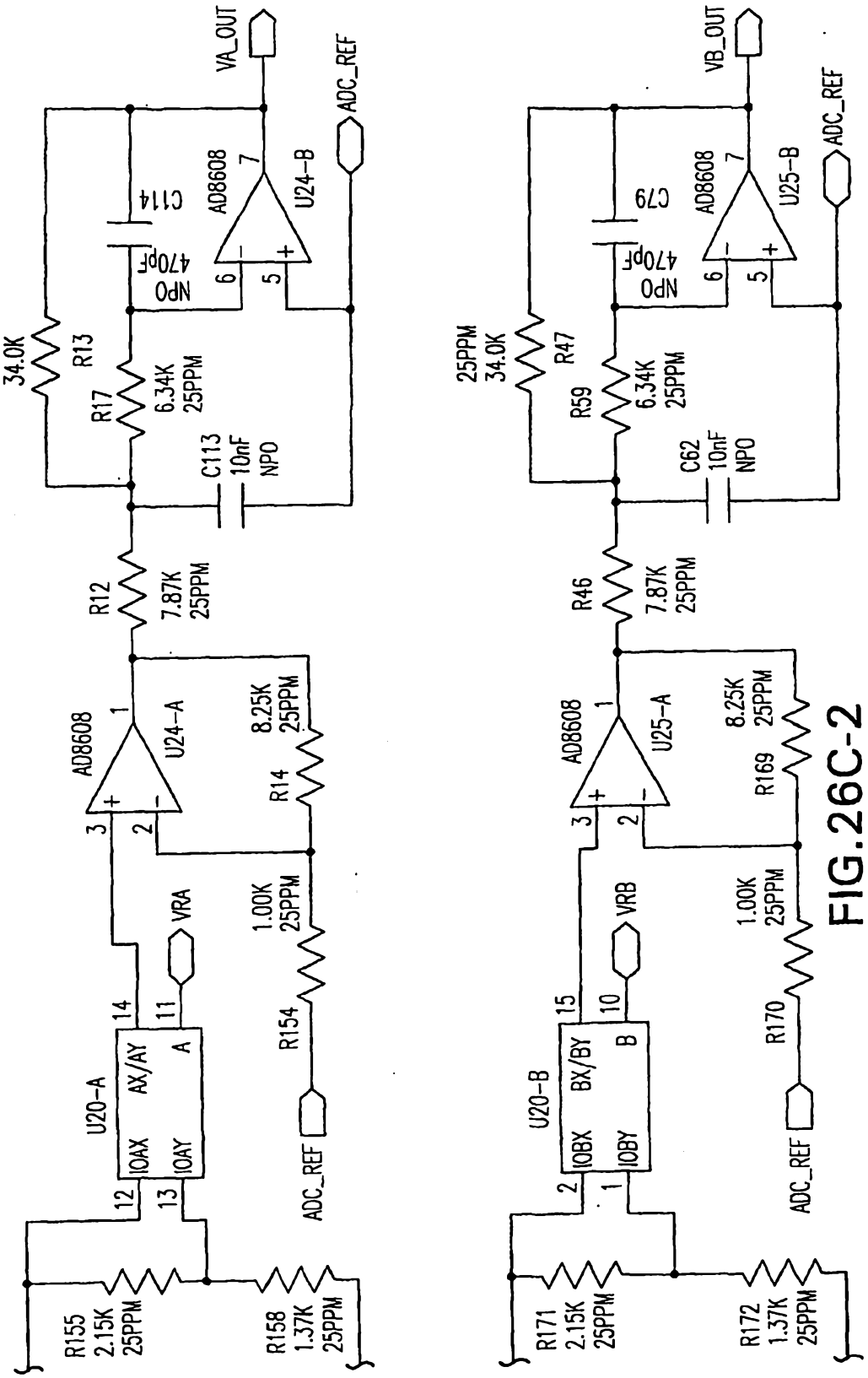

VA_OUT
ADC_REF
AD8608
U24-B
C114
470pF
NPO
25PPM
34.0K
R13
R17
6.34K
25PPM
C113
10nF
NPO
R12
7.87K
25PPM
U24-A
8.25K
25PPM
R14
1.00K
25PPM
R154
VRA
U20-A
AX/AY
A
IOAX
IOAY
R155
2.15K
25PPM
R158
1.37K
25PPM
VB_OUT
C79
R47
R59
C62
R46
R169
R170
U25-A
U25-B
U20-B
BX/BY
B
IOBX
IOBY
VRB
R171
R172

FIG.26C-2

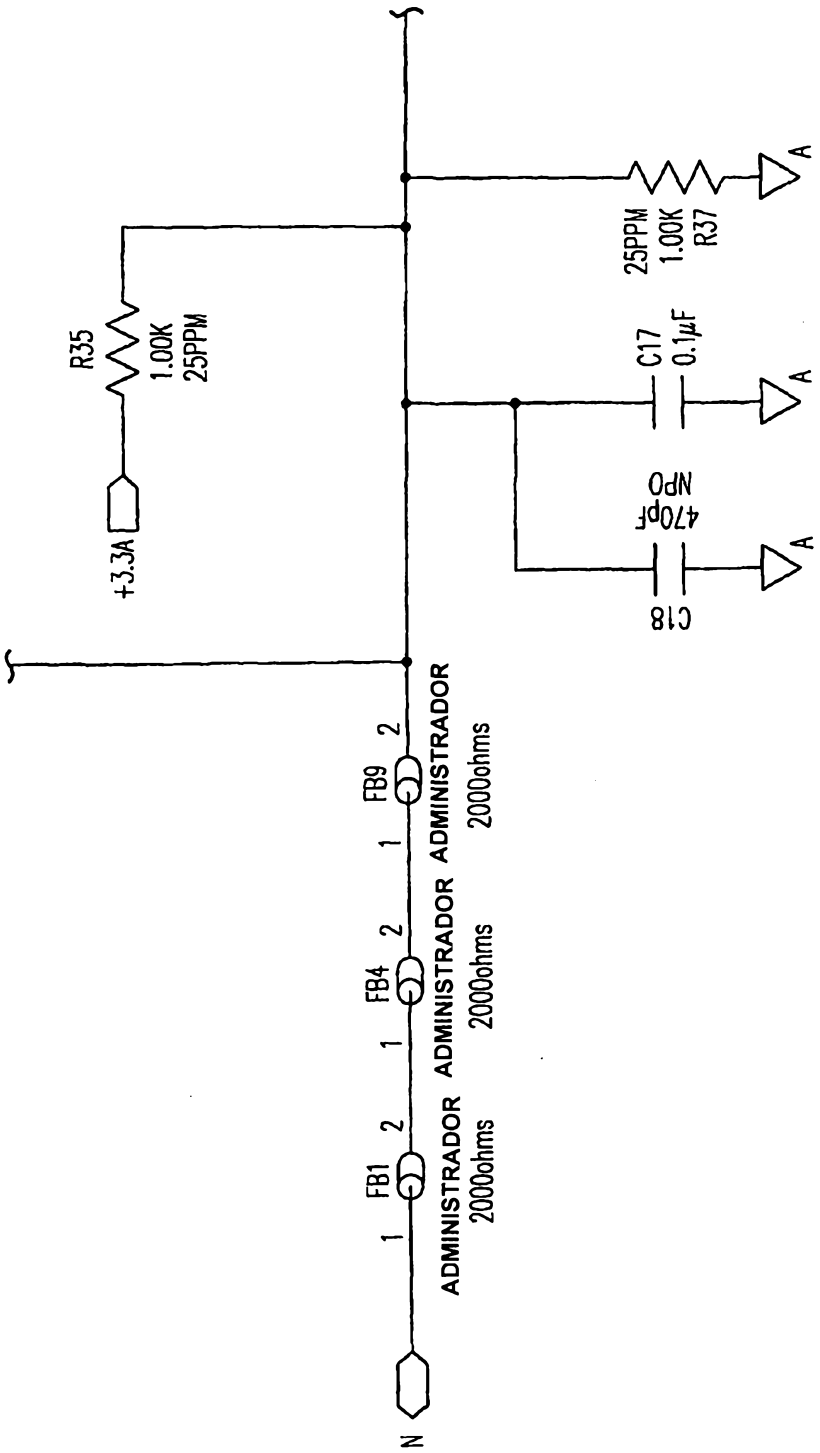

R35
1.00K
25PPM
+3.3A
25PPM
1.00K
R37
C17
0.1µF
470pF
NPO
C18
A
A
A
FB9
FB4
FB1
ADMINISTRADOR
2000ohms
ADMINISTRADOR
2000ohms
ADMINISTRADOR
2000ohms
N

FIG.26C-3

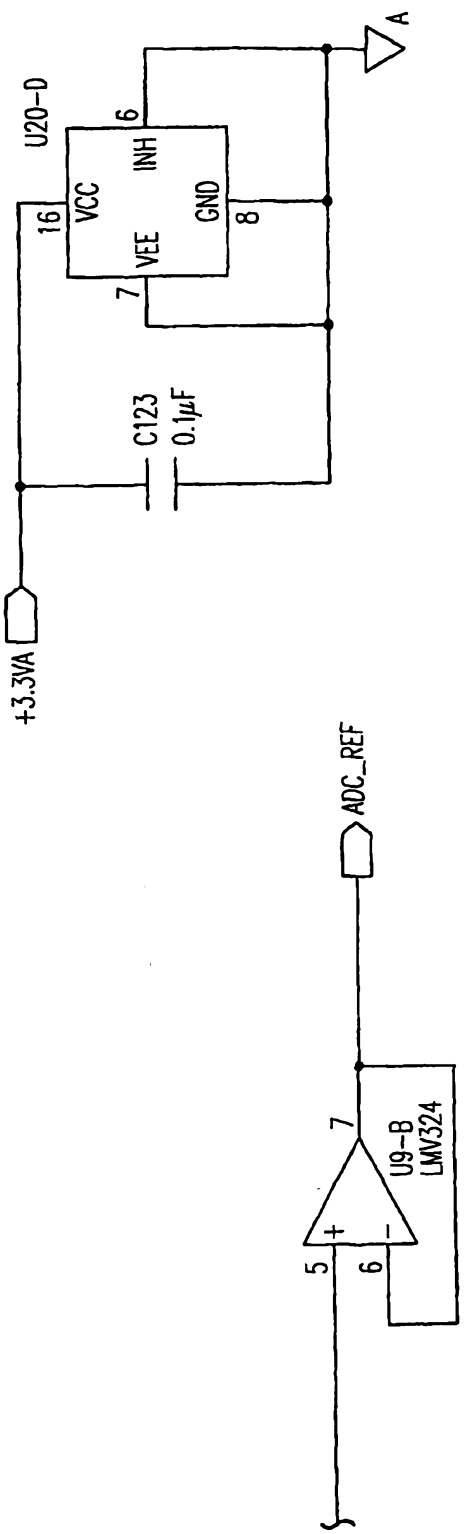

A
U20-D
6
INH
16
VCC
GND
8
VEE
7
C123
0.1µF
+3.3VA
ADC_REF
7
U9-B
LMV324
+
−
5
6

FIG.26C-4

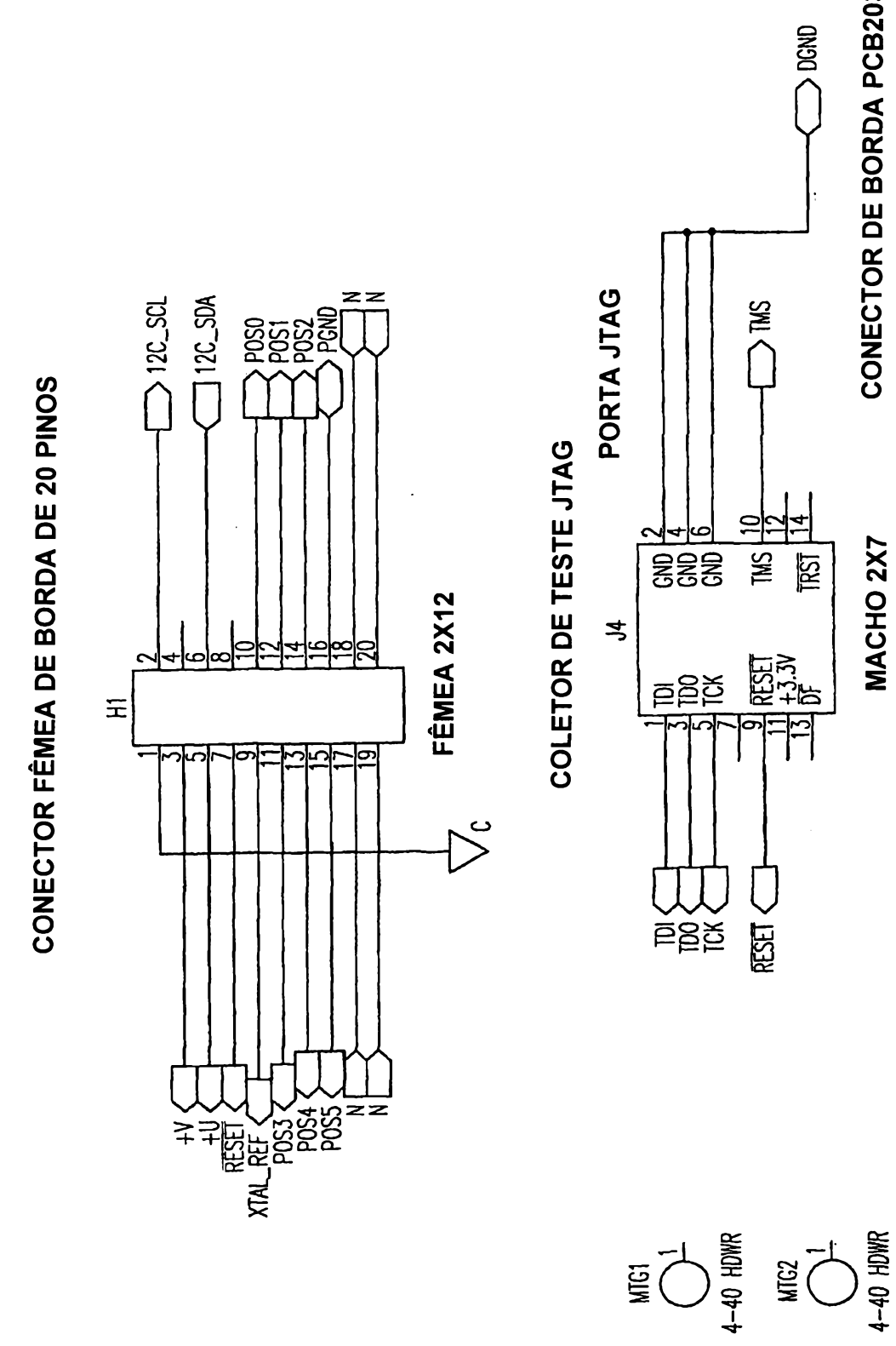

CONECTOR FÊMEA DE BORDA DE 20 PINOS
H1
FÊMEA 2X12
12C_SCL
12C_SDA
POS0
POS1
POS2
PGND
N
N
+V
+U
RESET
REF
XTAL_
POS3
POS4
POS5
N
N
C
COLETOR DE TESTE JTAG
PORTA JTAG
J4
MACHO 2X7
TDI
TDO
TCK
RESET
+3.3V
DF
GND
GND
GND
TMS
TRST
TMS
DGND
CONECTOR DE BORDA PCB203
MTG1
4-40 HDWR
MTG2
4-40 HDWR

FIG.26D

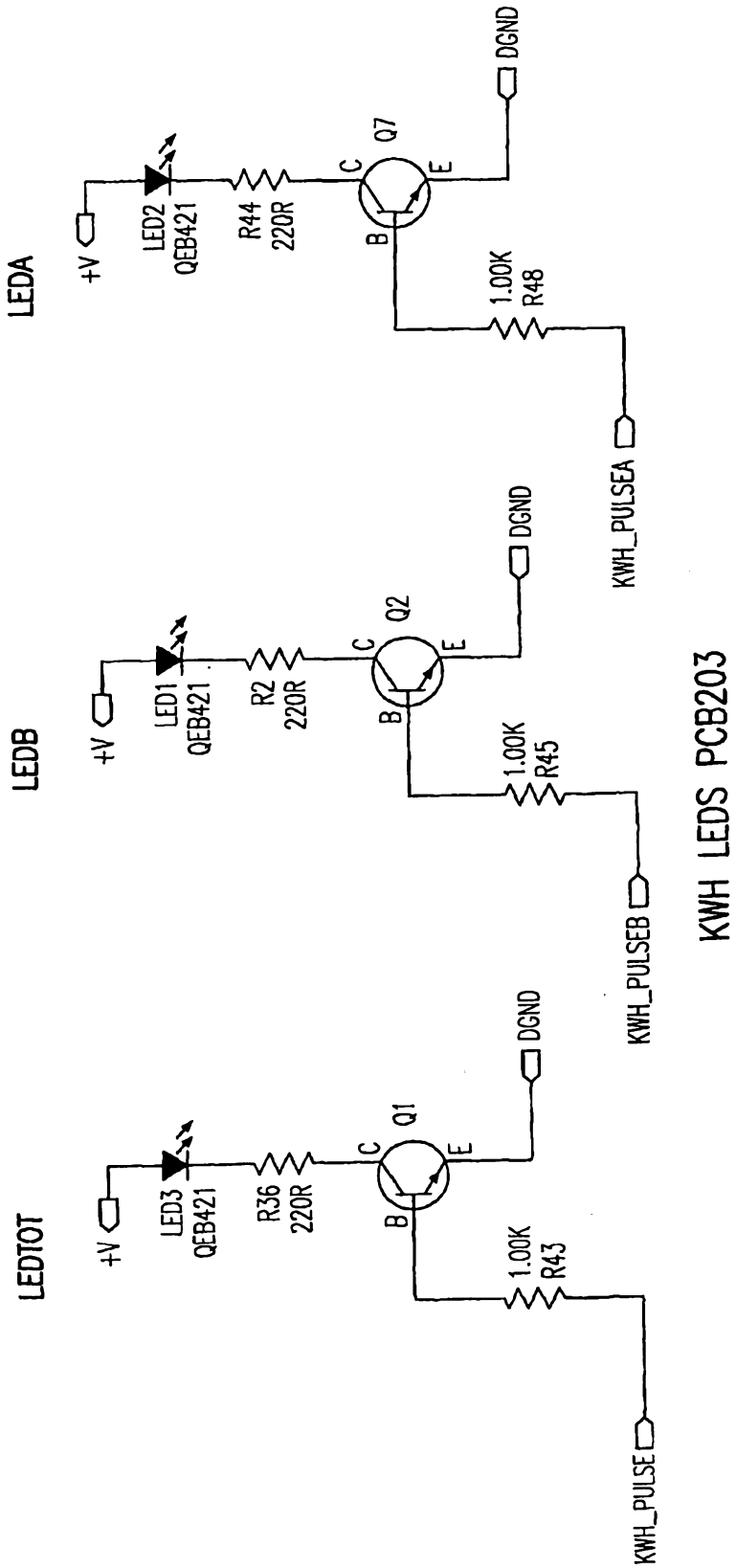

LEDA
LED2
QEB421
+V
R44
220R
Q7
C
B
E
DGND
1.00K
R48
KWH_PULSEA
LEDB
LED1
QEB421
+V
R2
220R
Q2
C
B
E
DGND
1.00K
R45
KWH_PULSEB
LEDTOT
LED3
QEB421
+V
R36
220R
Q1
C
B
E
DGND
1.00K
R43
KWH_PULSE
KWH LEDS PCB203

FIG.26E

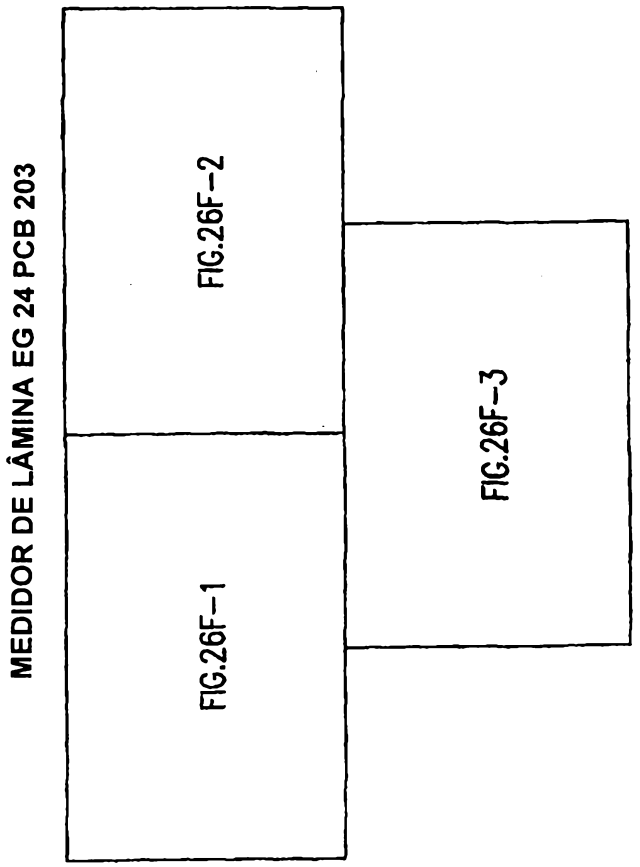


FIG.26F

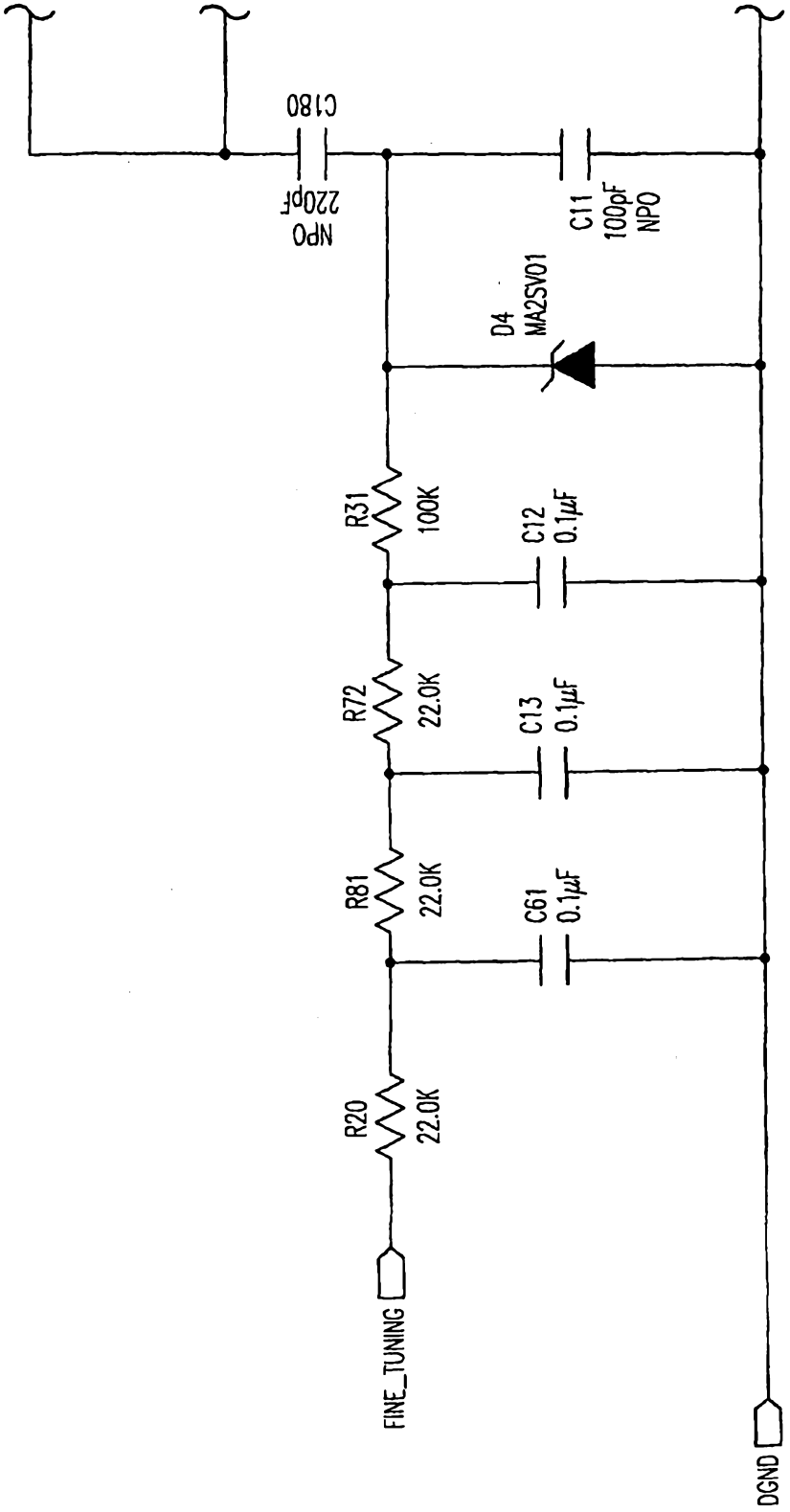

C180
220pF
NPO
C11
100pF
NPO
D4
MA2SV01
R31
100K
C12
0.1μF
R72
22.0K
C13
0.1μF
R81
22.0K
C61
0.1μF
R20
22.0K
FINE_TUNING
DGND

FIG.26F-1

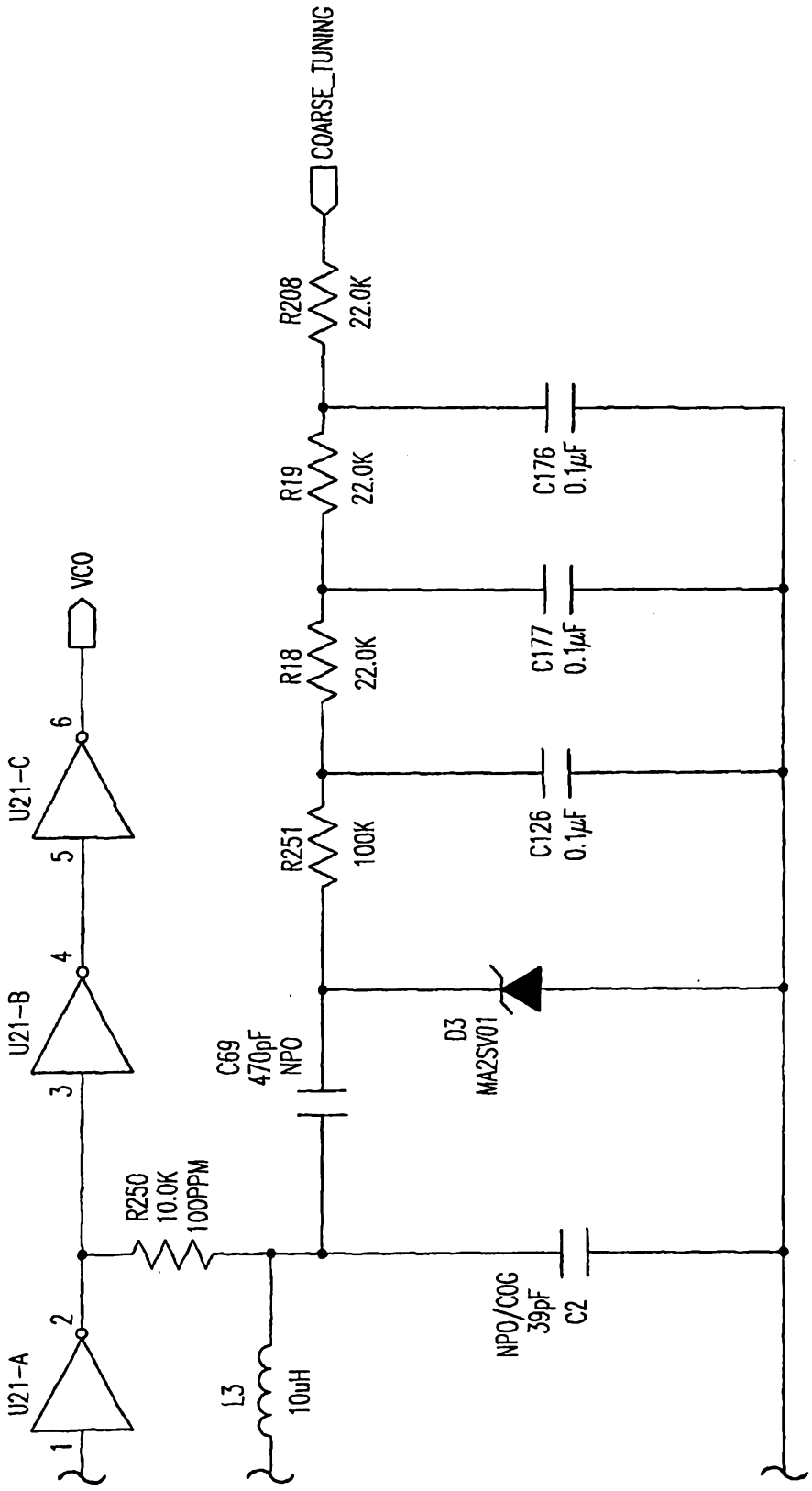

COARSE_TUNING
R208
22.0K
R19
22.0K
C176
0.1µF
VCO
R18
22.0K
C177
0.1µF
U21-C
6
5
R251
100K
C126
0.1µF
U21-B
4
3
C69
470pF
NPO
D3
MA2SV01
R250
10.0K
100PPM
U21-A
2
1
L3
10uH
NPO/COG
39pF
C2

FIG.26F-2

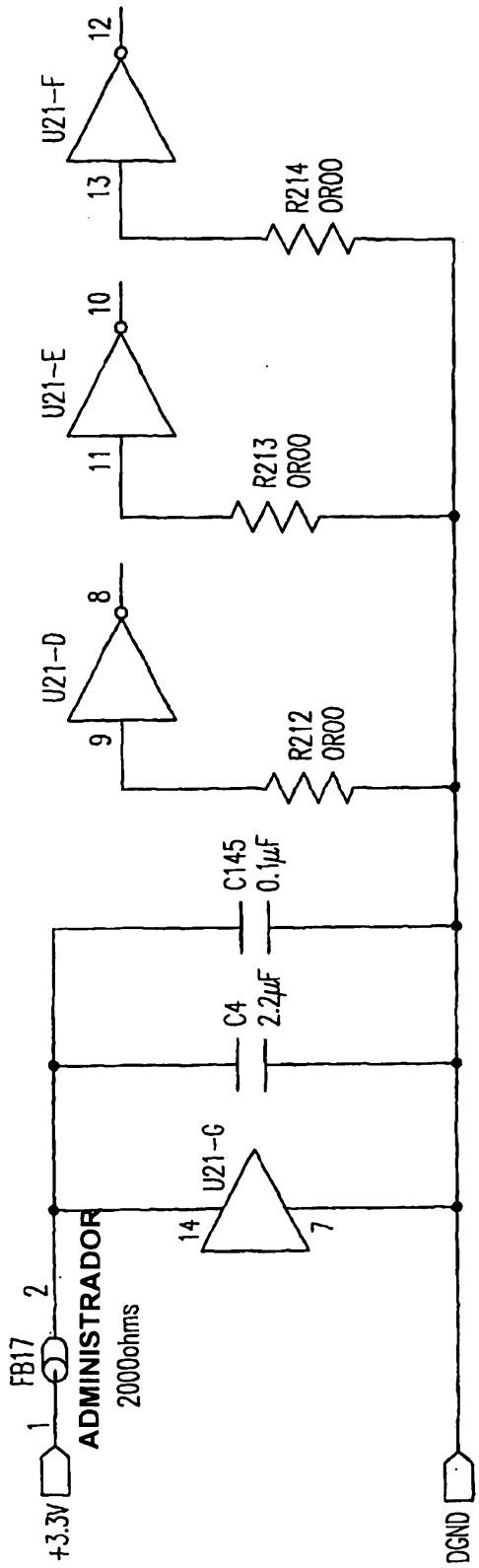

U21-F
12
13
R214
0R00
U21-E
10
11
R213
0R00
U21-D
8
9
R212
0R00
C145
0.1µF
C4
2.2µF
U21-G
14
7
FB17
1
2
ADMINISTRADOR
2000ohms
+3.3V
DGND

FIG.26F-3

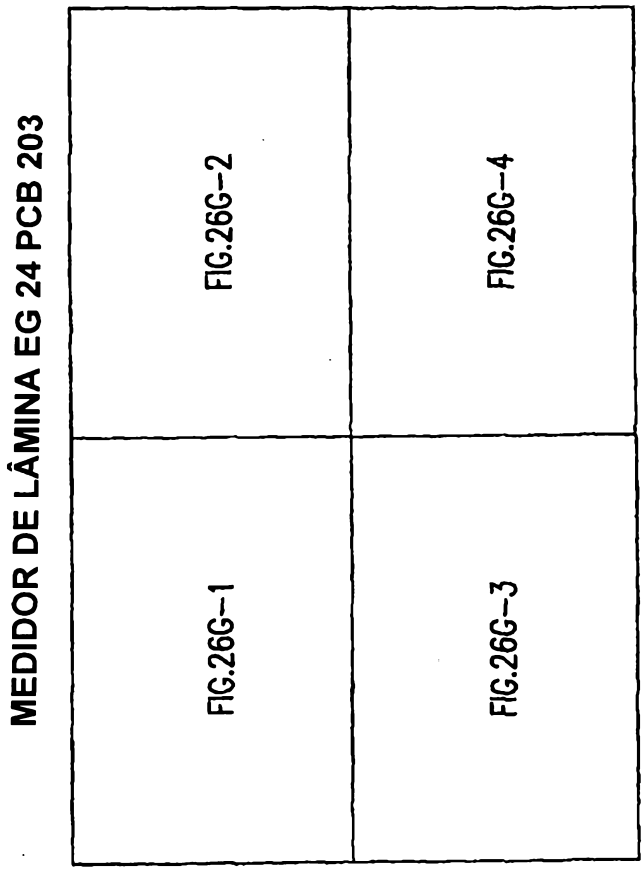

MEDIDOR DE LÂMINA EG 24 PCB 203
FIG.26G-1
FIG.26G-2
FIG.26G-3
FIG.26G-4

FIG.26G

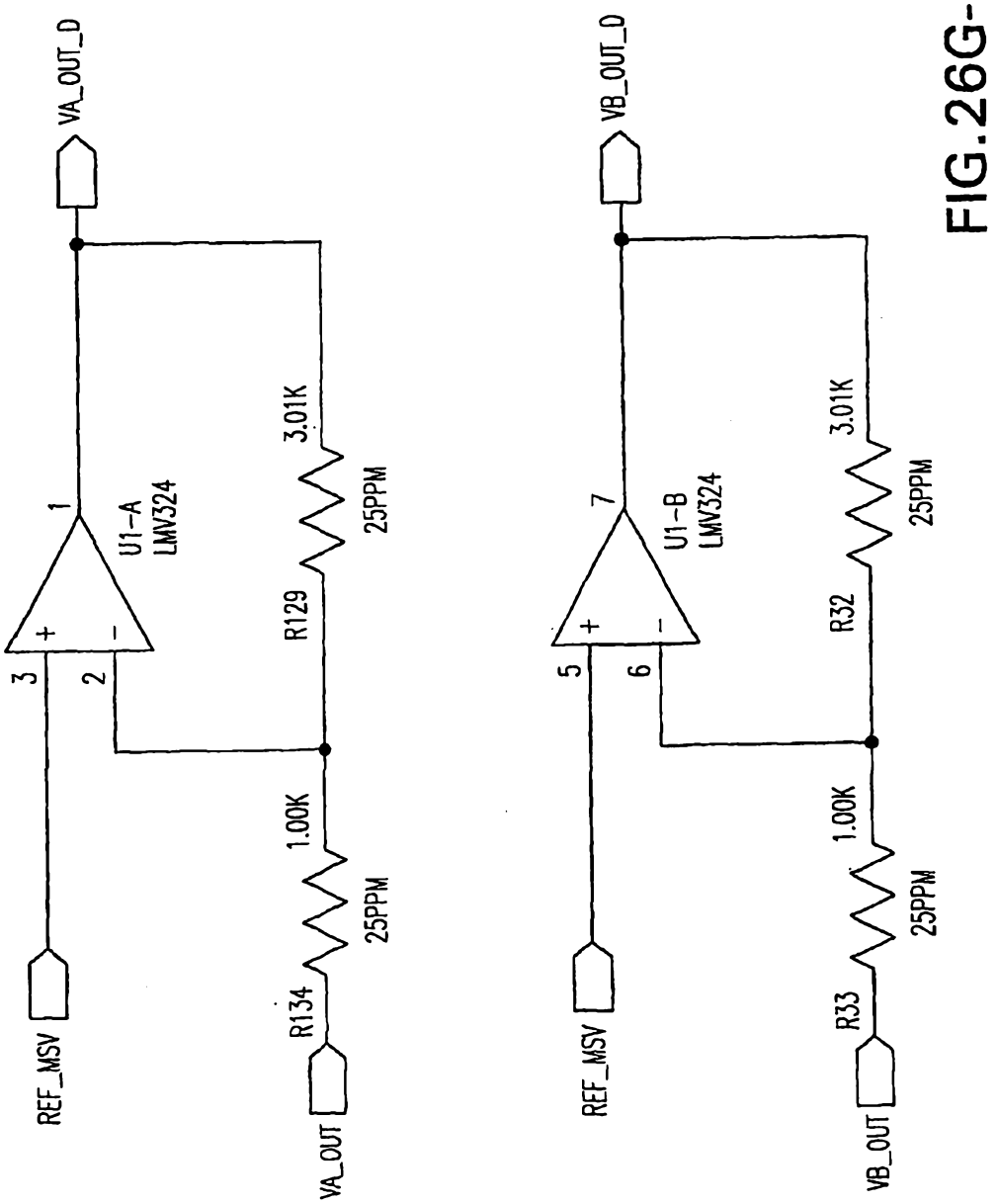

VA_OUT_D
U1-A
LMV324
1
2
3
+
-
R129
3.01K
25PPM
1.00K
25PPM
R134
REF_MSV
VA_OUT
VB_OUT_D
U1-B
LMV324
7
6
5
+
-
R32
3.01K
25PPM
1.00K
25PPM
R33
REF_MSV
VB_OUT

FIG.26G-1

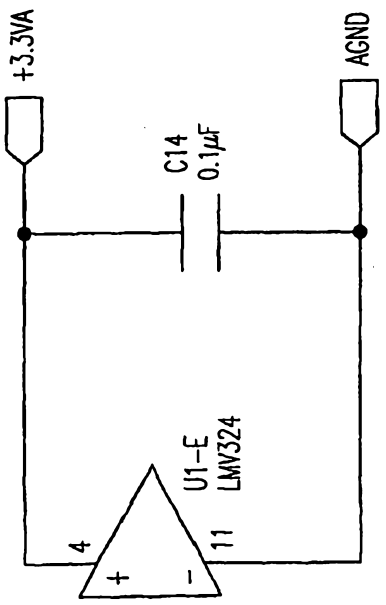


FIG.26G-2

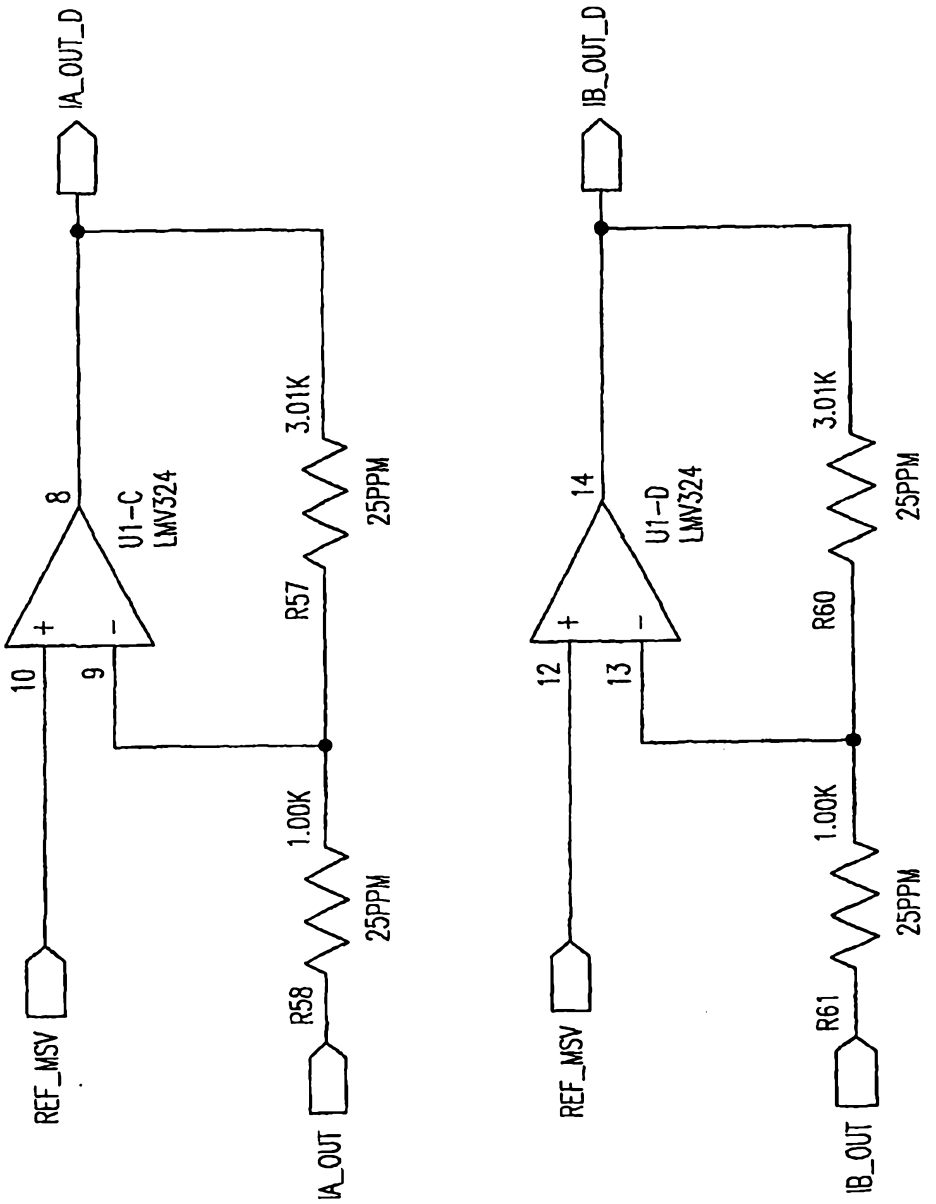

IA_OUT_D
IB_OUT_D
U1-C
LMV324
U1-D
LMV324
8
14
10
9
12
13
+
-
3.01K
25PPM
R57
R60
1.00K
25PPM
R58
R61
REF_MSV
IA_OUT
IB_OUT

FIG.26G-3

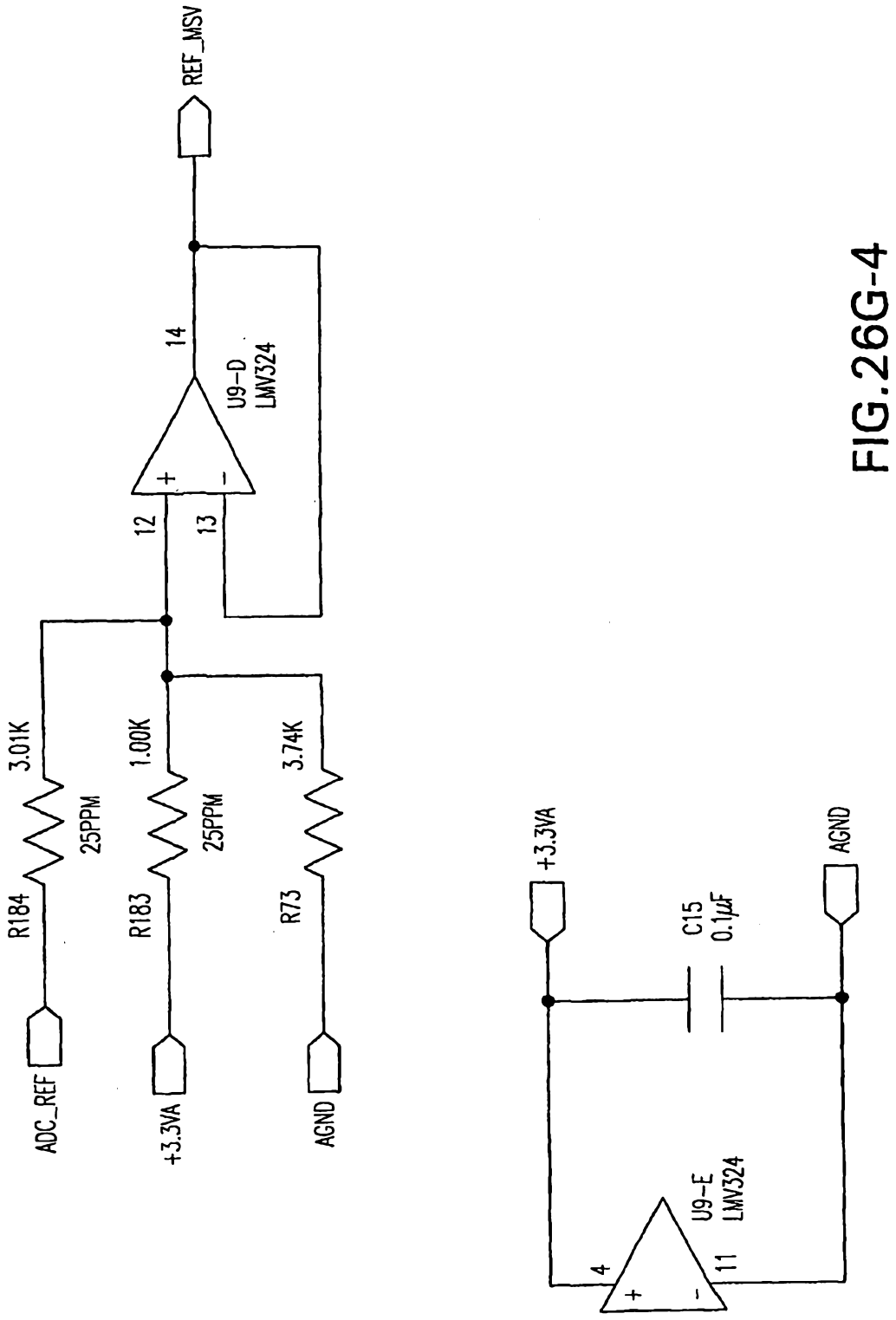

REF_MSV
14
U9-D
LMV324
12
13
+
-
3.01K
1.00K
3.74K
25PPM
25PPM
R184
R183
R73
ADC_REF
+3.3VA
AGND
+3.3VA
C15
0.1μF
AGND
U9-E
LMV324
4
11
+
-

FIG.26G-4

MEDIDOR DE LÂMINA EG 24 PCB 203

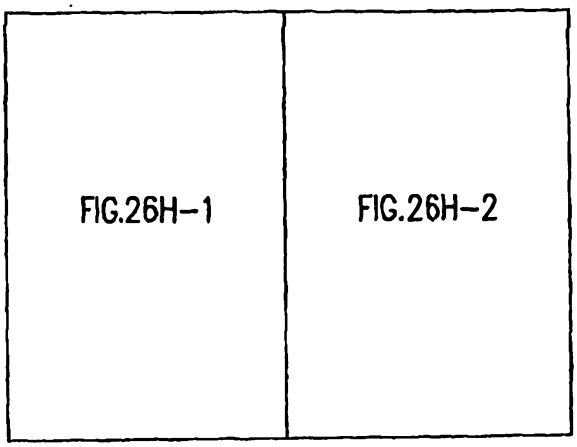


FIG.26H

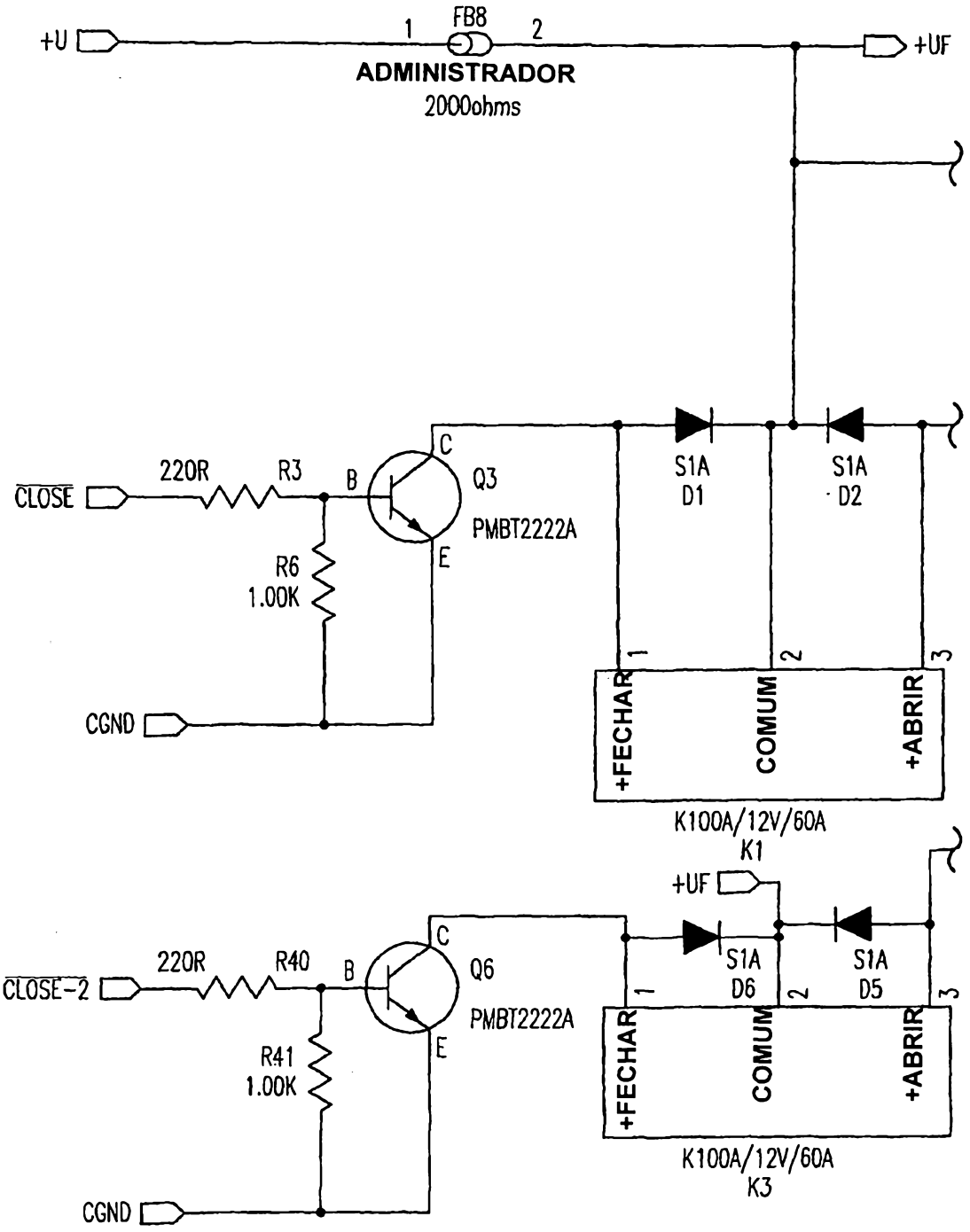

+U
FB8
1
2
+UF
ADMINISTRADOR
2000ohms
220R
R3
B
C
Q3
PMBT2222A
E
CLOSE
R6
1.00K
CGND
S1A
D1
S1A
D2
1
2
3
+FECHAR
COMUM
+ABRIR
K100A/12V/60A
K1
+UF
220R
R40
B
C
Q6
PMBT2222A
E
CLOSE-2
R41
1.00K
CGND
S1A
D6
S1A
D5
1
2
3
+FECHAR
COMUM
+ABRIR
K100A/12V/60A
K3

FIG.26H-1

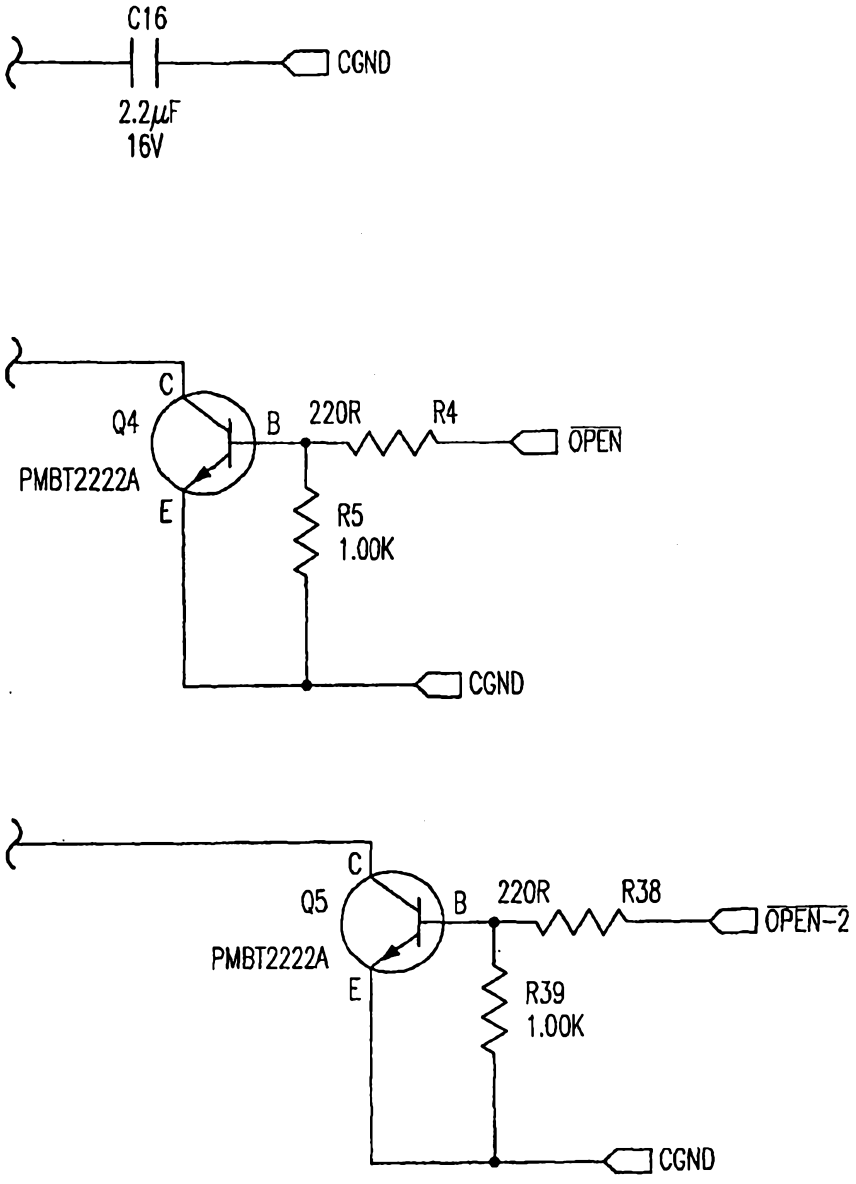

C16
2.2μF
16V
CGND
C
Q4
B
220R
R4
OPEN
PMBT2222A
E
R5
1.00K
CGND
C
Q5
B
220R
R38
OPEN-2
PMBT2222A
E
R39
1.00K
CGND

FIG.26H-2

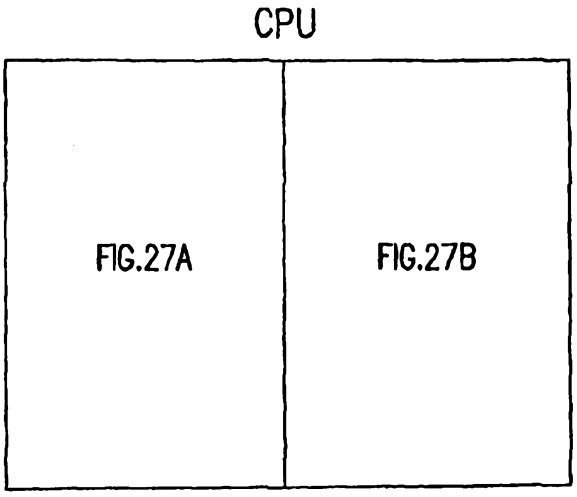


FIG.27

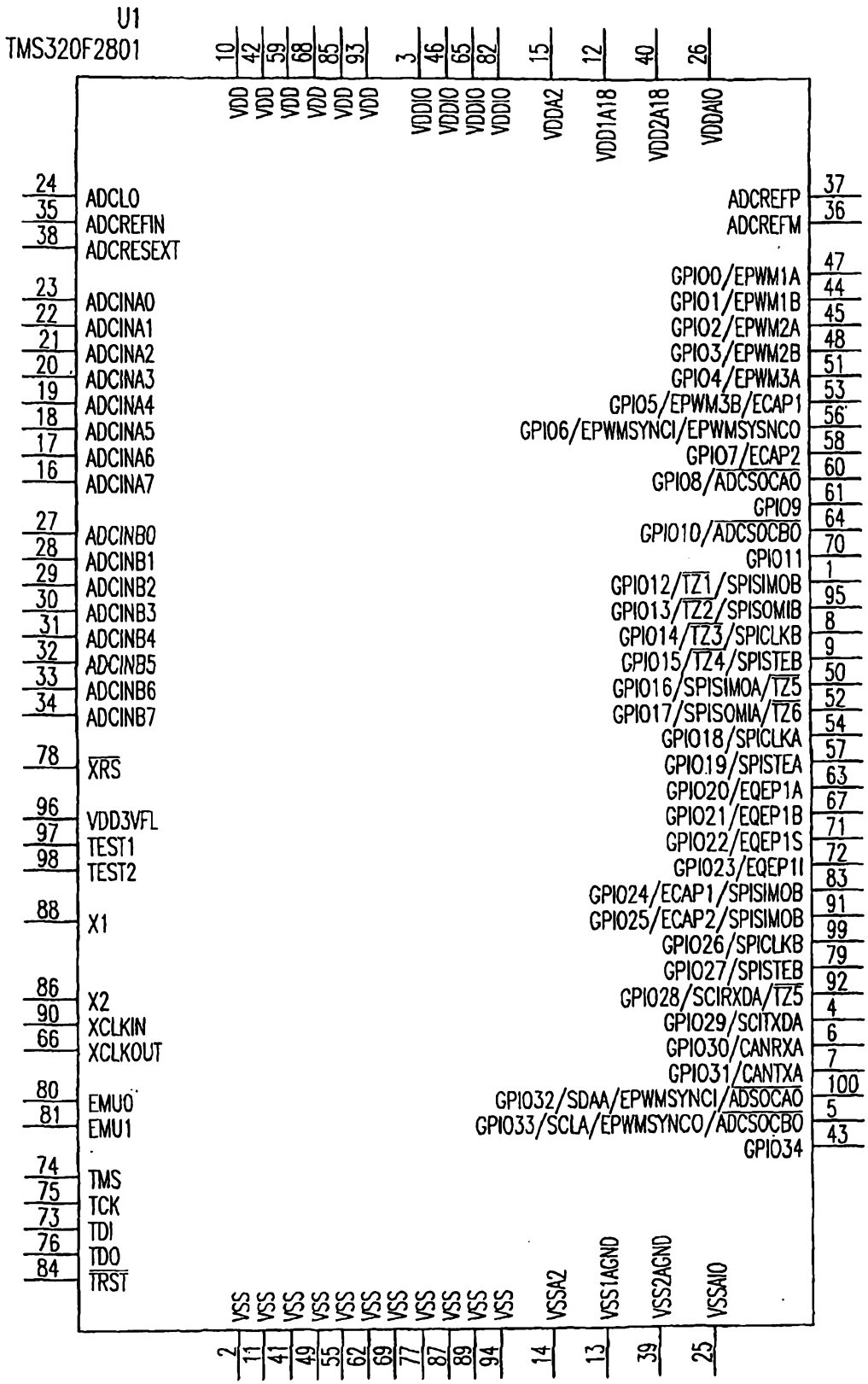

U1
TMS320F2801
10 VDD
42 VDD
59 VDD
68 VDD
85 VDD
93 VDD
3 VDDIO
46 VDDIO
65 VDDIO
82 VDDIO
15 VDDA2
12 VDD1A18
40 VDD2A18
26 VDDAIO
24 ADCLO
35 ADCREFIN
38 ADCRESEXT
23 ADCINA0
22 ADCINA1
21 ADCINA2
20 ADCINA3
19 ADCINA4
18 ADCINA5
17 ADCINA6
16 ADCINA7
27 ADCINB0
28 ADCINB1
29 ADCINB2
30 ADCINB3
31 ADCINB4
32 ADCINB5
33 ADCINB6
34 ADCINB7
78 XRS
96 VDD3VFL
97 TEST1
98 TEST2
88 X1
86 X2
90 XCLKIN
66 XCLKOUT
80 EMU0
81 EMU1
74 TMS
75 TCK
73 TDI
76 TDO
84 TRST
37 ADCREFP
36 ADCREFM
47 GPIO0/EPWM1A
44 GPIO1/EPWM1B
45 GPIO2/EPWM2A
48 GPIO3/EPWM2B
51 GPIO4/EPWM3A
53 GPIO5/EPWM3B/ECAP1
56 GPIO6/EPWMSYNCI/EPWMSYNCO
58 GPIO7/ECAP2
60 GPIO8/ADCSOCAO
61 GPIO9
64 GPIO10/ADCSOCBO
70 GPIO11
1 GPIO12/TZ1/SPISIMOB
95 GPIO13/TZ2/SPISOMIB
8 GPIO14/TZ3/SPICLKB
9 GPIO15/TZ4/SPISTEB
50 GPIO16/SPISIMOA/TZ5
52 GPIO17/SPISOMIA/TZ6
54 GPIO18/SPICLKA
57 GPIO19/SPISTEA
63 GPIO20/EQEP1A
67 GPIO21/EQEP1B
71 GPIO22/EQEP1S
72 GPIO23/EQEP1I
83 GPIO24/ECAP1/SPISIMOB
91 GPIO25/ECAP2/SPISIMOB
99 GPIO26/SPICLKB
79 GPIO27/SPISTEB
92 GPIO28/SCIRXDA/TZ5
4 GPIO29/SCITXDA
6 GPIO30/CANRXA
7 GPIO31/CANTXA
100 GPIO32/SDAA/EPWMSYNCI/ADSOCAO
5 GPIO33/SCLA/EPWMSYNCO/ADCSOCBO
43 GPIO34
2 VSS
11 VSS
41 VSS
49 VSS
55 VSS
62 VSS
69 VSS
77 VSS
87 VSS
89 VSS
94 VSS
14 VSSA2
13 VSS1AGND
39 VSS2AGND
25 VSSAIO

FIG.27A

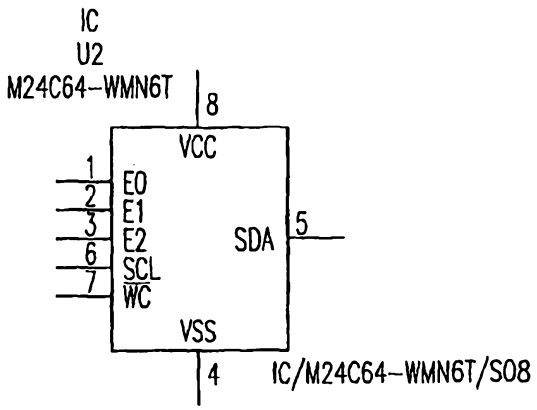

IC
U2
M24C64-WMN6T
8
VCC
1
E0
2
E1
3
E2
6
SCL
7
WC
SDA
5
VSS
4
IC/M24C64-WMN6T/SO8

FIG.27B

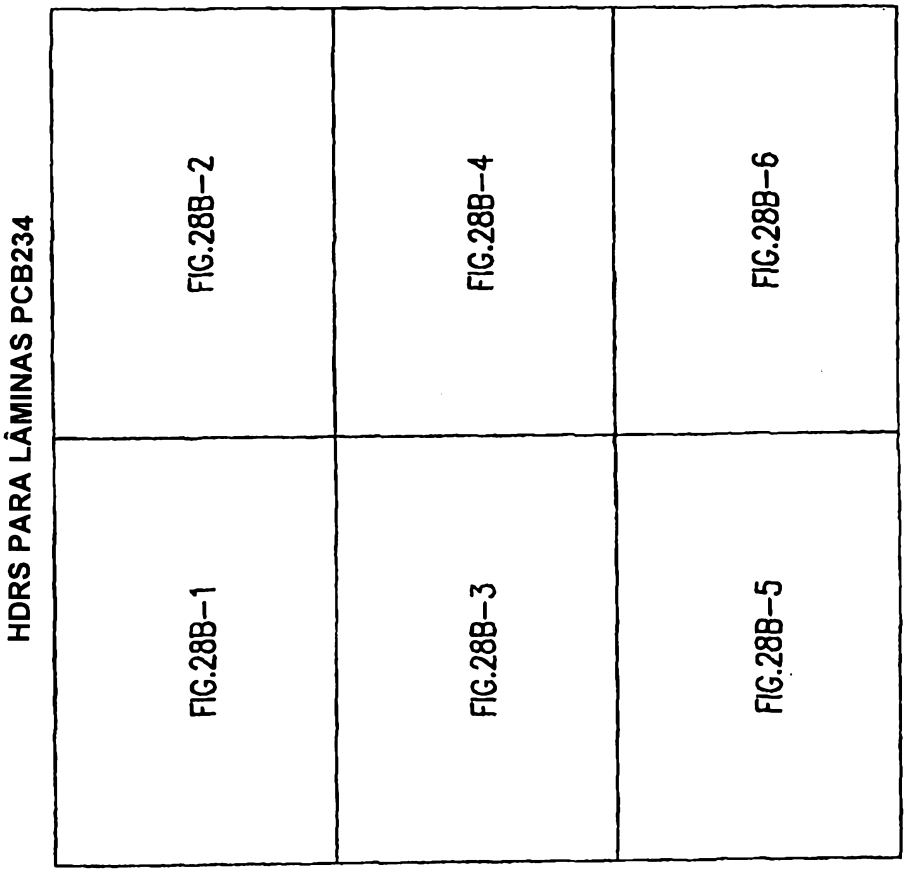

HDRS PARA LÂMINAS PCB234
FIG.28B-1
FIG.28B-2
FIG.28B-3
FIG.28B-4
FIG.28B-5
FIG.28B-6

FIG.28B

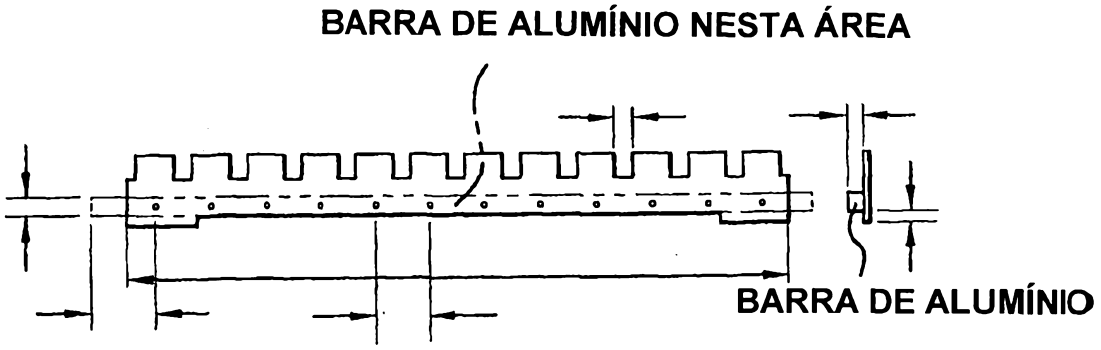


FIG.28A

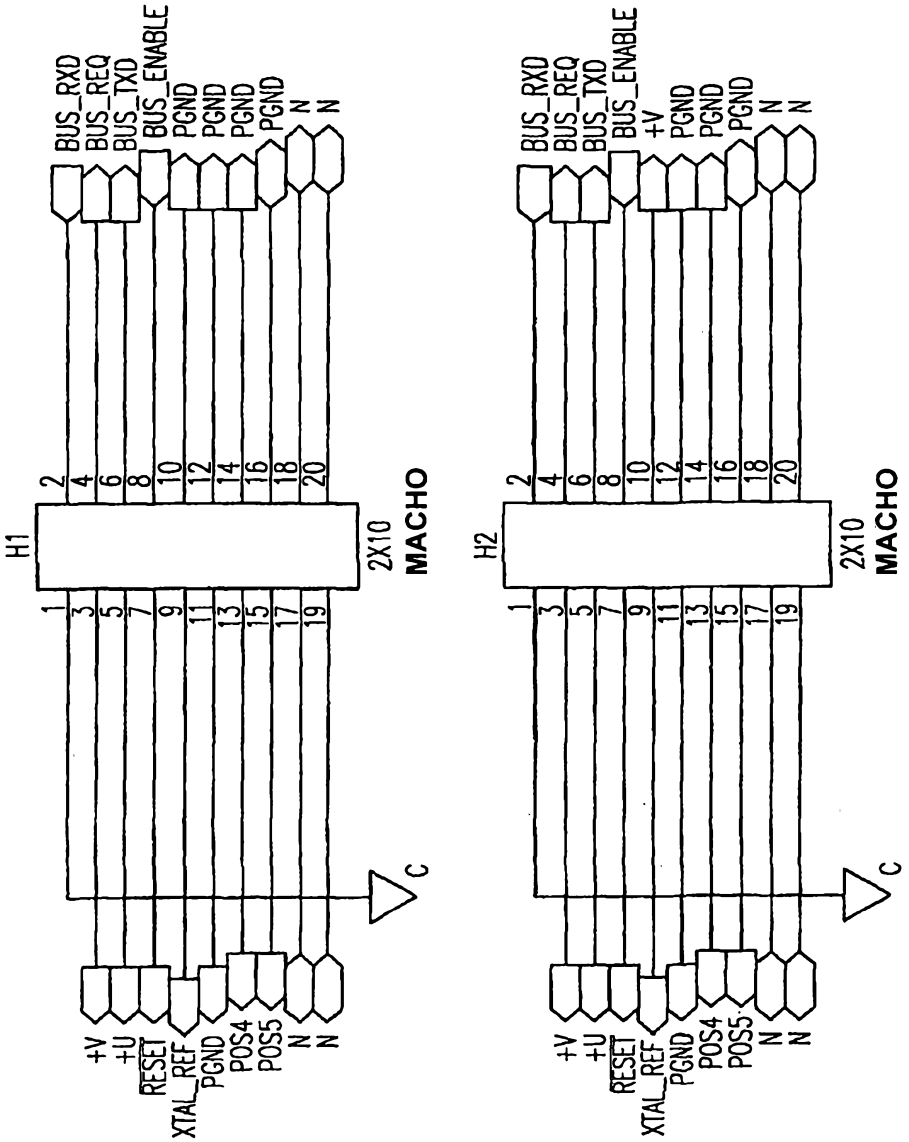

H1
2X10
MACHO
BUS_RXD
BUS_REQ
BUS_TXD
BUS_ENABLE
PGND
PGND
PGND
PGND
N
N
+V
+U
RESET
XTAL_REF
PGND
POS4
POS5
N
N
C
H2
2X10
MACHO
BUS_RXD
BUS_REQ
BUS_TXD
BUS_ENABLE
+V
PGND
PGND
PGND
N
N
+V
+U
RESET
XTAL_REF
PGND
POS4
POS5
N
N
C

FIG.28B-1

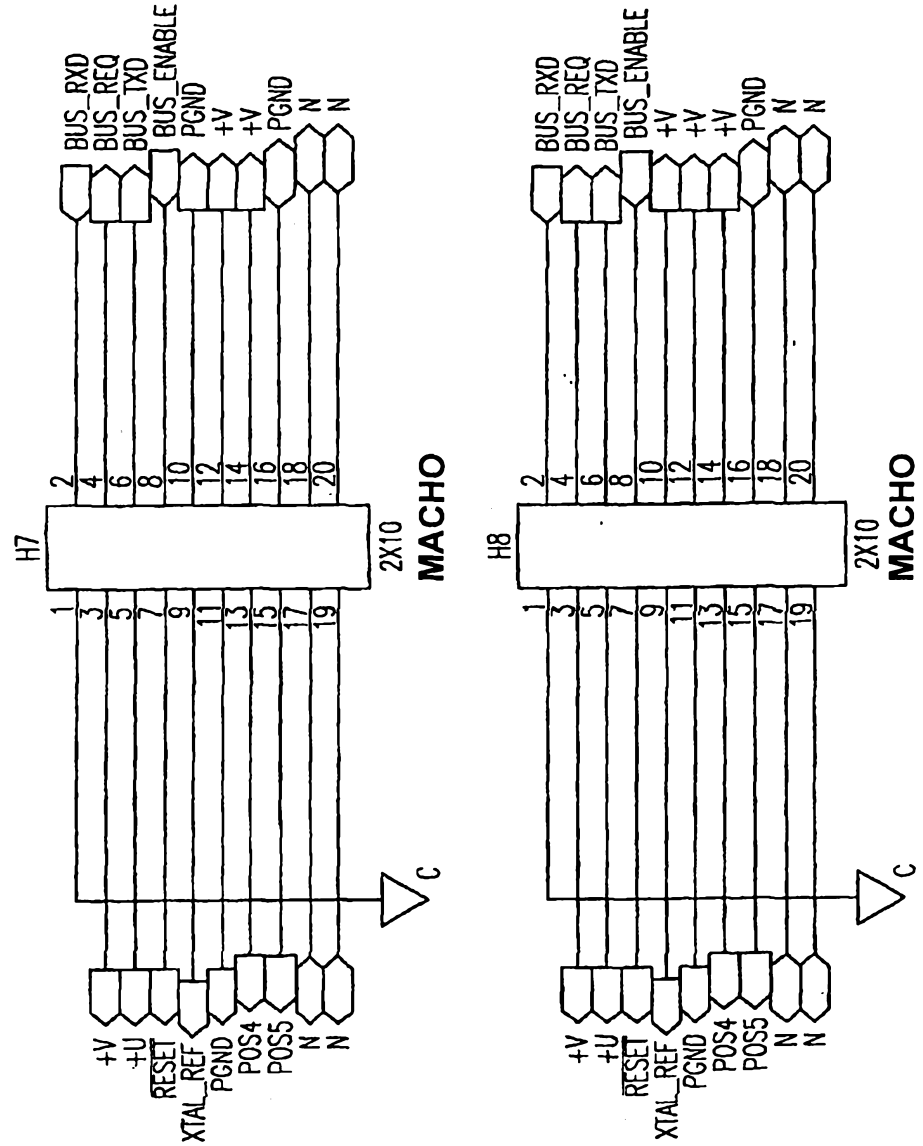

H7
2X10
MACHO
BUS_RXD
BUS_REQ
BUS_TXD
BUS_ENABLE
PGND
+V
+V
PGND
N
N
+V
+U
RESET
XTAL_REF
PGND
POS4
POS5
N
N
C
H8
2X10
MACHO
BUS_RXD
BUS_REQ
BUS_TXD
BUS_ENABLE
+V
+V
+V
PGND
N
N
+V
+U
RESET
XTAL_REF
PGND
POS4
POS5
N
N
C

FIG.28B-2

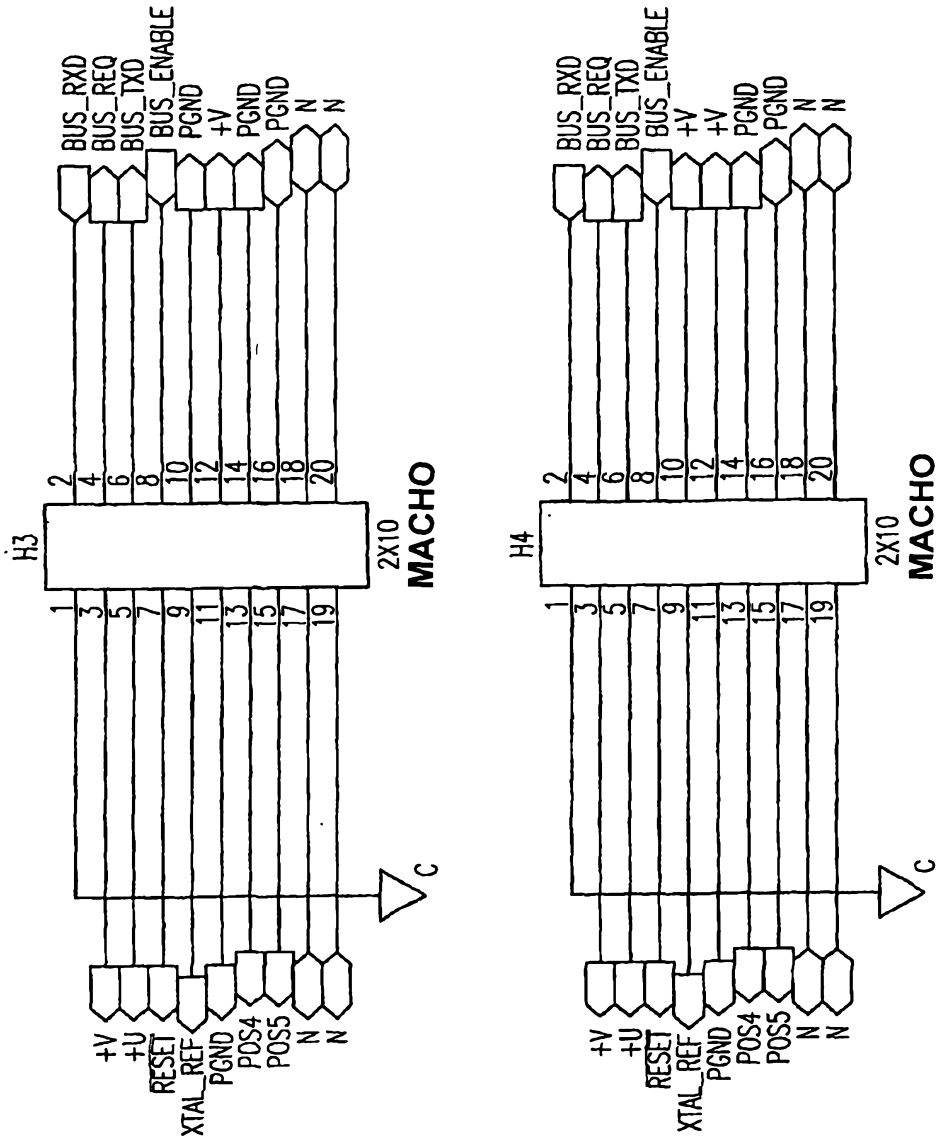

H3
2X10
MACHO
BUS_RXD
BUS_REQ
BUS_TXD
BUS_ENABLE
PGND
+V
PGND
PGND
N
N
+V
+U
RESET
XTAL_REF
PGND
POS4
POS5
N
N
H4
2X10
MACHO
BUS_RXD
BUS_REQ
BUS_TXD
BUS_ENABLE
+V
+V
PGND
PGND
N
N
+V
+U
RESET
XTAL_REF
PGND
POS4
POS5
N
N

FIG.28B-3

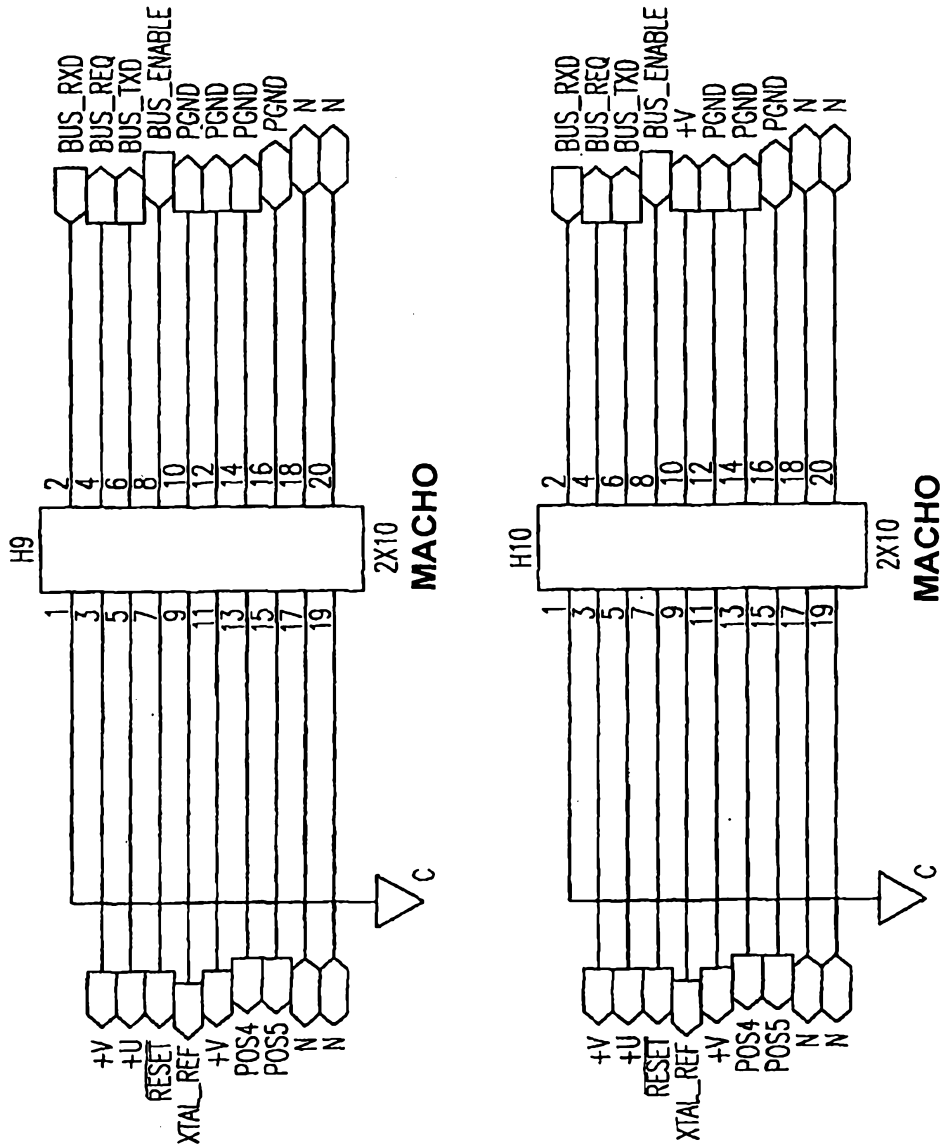

H9
2X10
MACHO
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
BUS_RXD
BUS_REQ
BUS_TXD
BUS_ENABLE
PGND
PGND
PGND
PGND
N
N
+V
+U
RESET
XTAL_REF
+V
POS4
POS5
N
N
C
H10
2X10
MACHO
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
BUS_RXD
BUS_REQ
BUS_TXD
BUS_ENABLE
+V
PGND
PGND
PGND
N
N
+V
+U
RESET
XTAL_REF
+V
POS4
POS5
N
N
C

FIG.28B-4

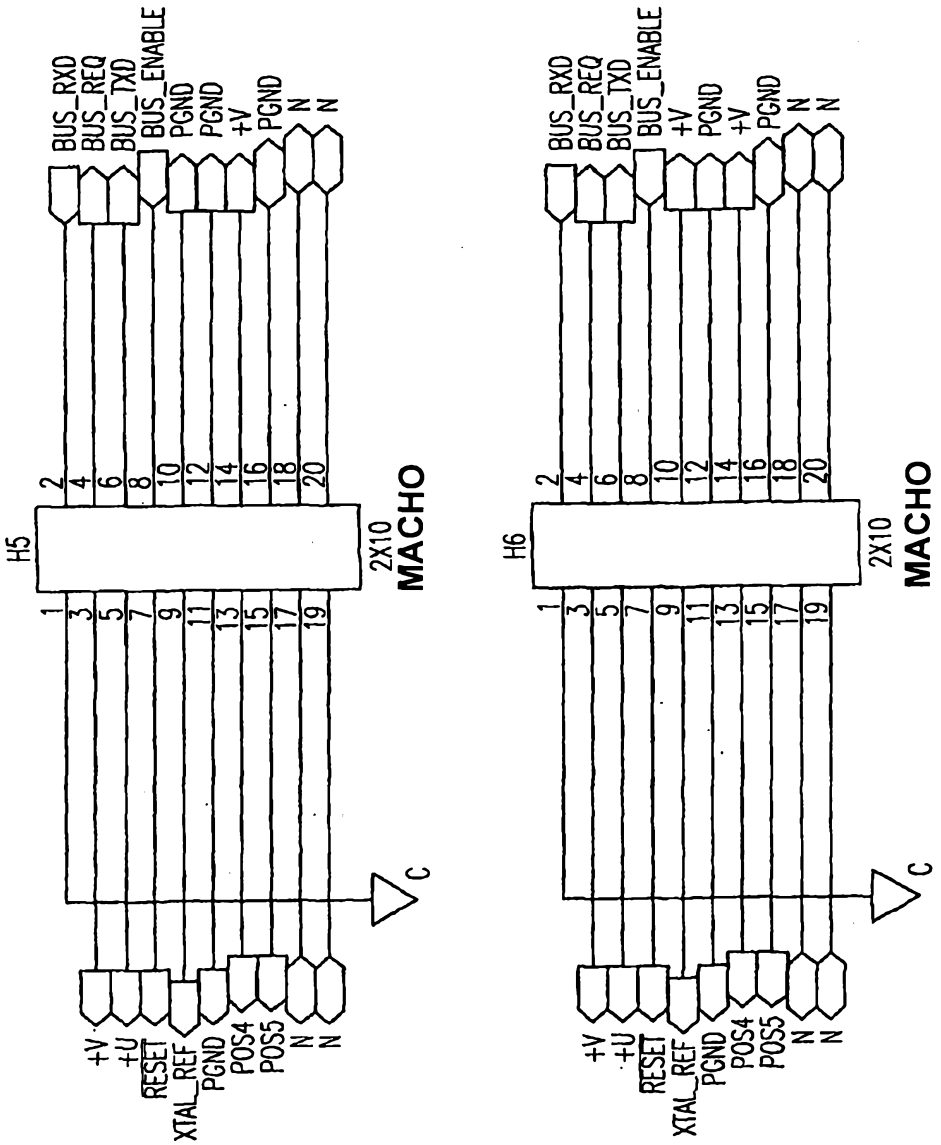

H5
2X10
MACHO
1
3
5
7
9
11
13
15
17
19
2
4
6
8
10
12
14
16
18
20
+V
+U
RESET
XTAL_REF
PGND
POS4
POS5
N
N
BUS_RXD
BUS_REQ
BUS_TXD
BUS_ENABLE
PGND
PGND
+V
PGND
N
N
C
H6
2X10
MACHO
1
3
5
7
9
11
13
15
17
19
2
4
6
8
10
12
14
16
18
20
+V
+U
RESET
XTAL_REF
PGND
POS4
POS5
N
N
BUS_RXD
BUS_REQ
BUS_TXD
BUS_ENABLE
+V
PGND
+V
PGND
N
N
C

FIG.28B-5

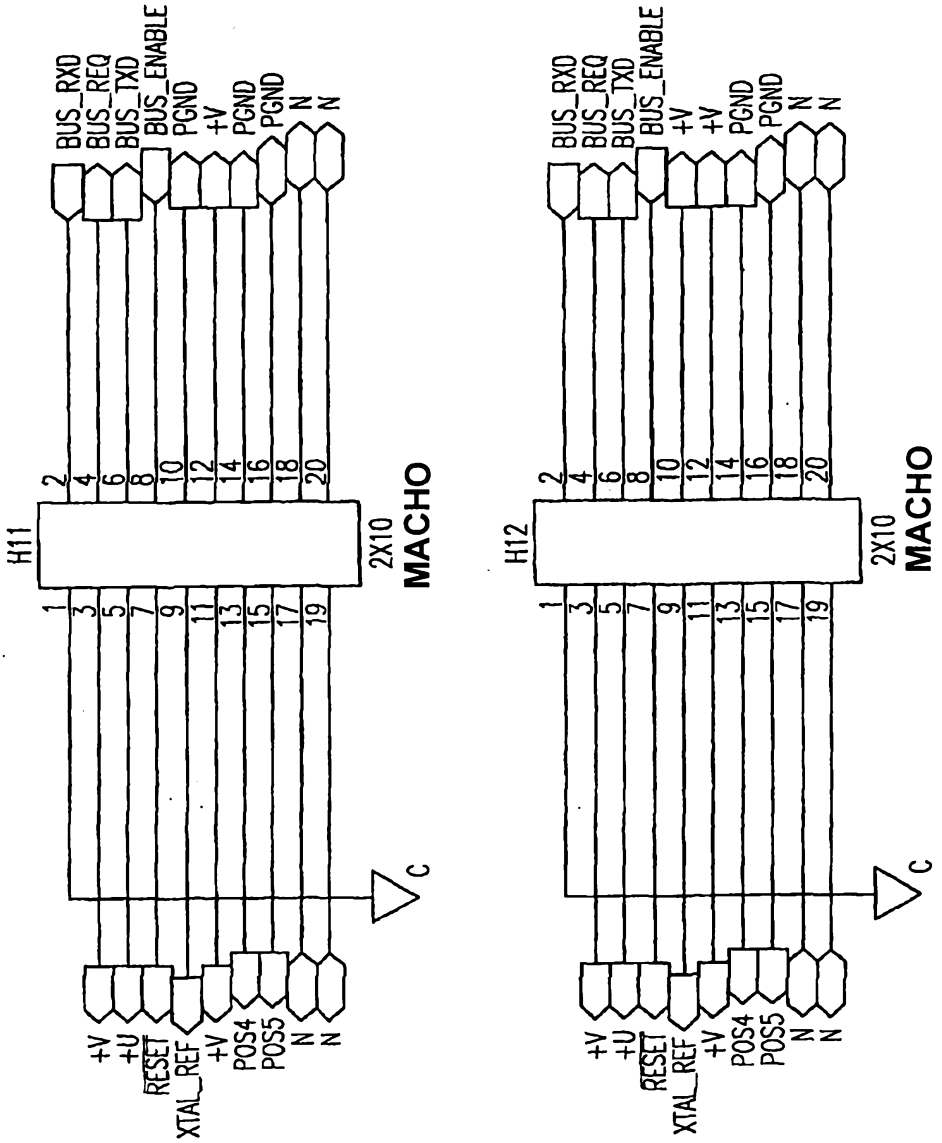

H11
2X10
MACHO
BUS_RXD
BUS_REQ
BUS_TXD
BUS_ENABLE
PGND
+V
PGND
PGND
N
N
+V
+U
RESET
XTAL_REF
+V
POS4
POS5
N
N
C
H12
2X10
MACHO
BUS_RXD
BUS_REQ
BUS_TXD
BUS_ENABLE
+V
+V
PGND
PGND
N
N
+V
+U
RESET
XTAL_REF
+V
POS4
POS5
N
N
C

FIG.28B-6

COLETORES PCB234

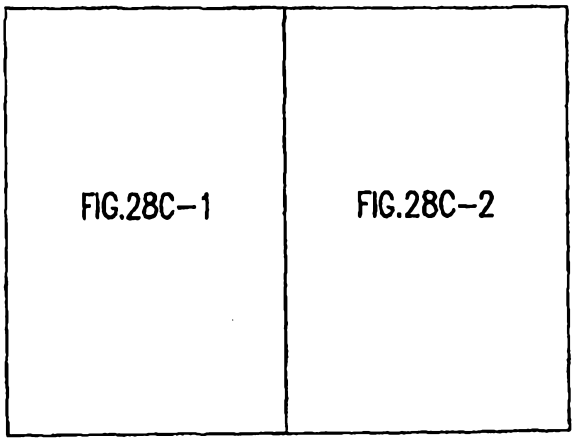
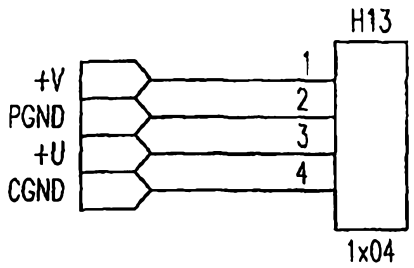
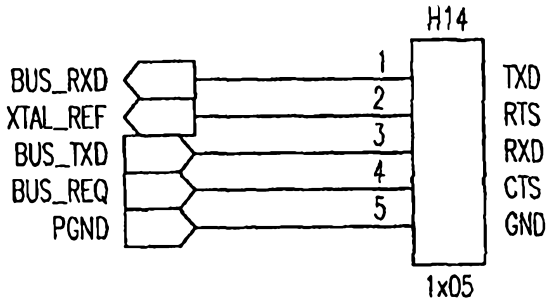


FIG.28C

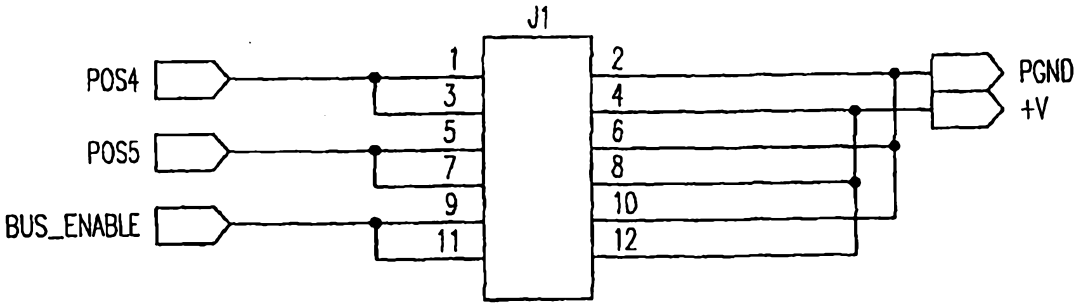
COLETOR DE POTÊNCIA



COLETOR RS232



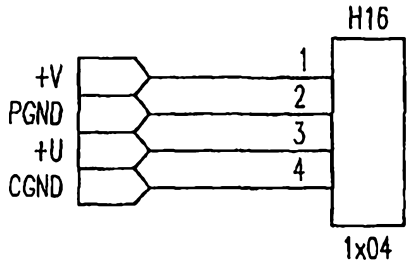
PONTES



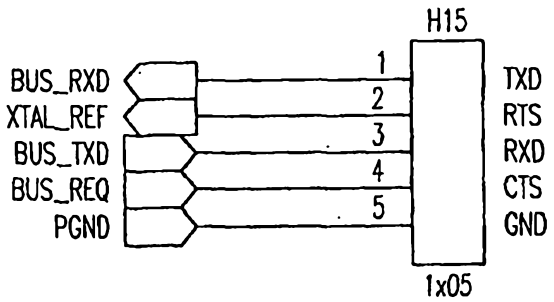
MACHO 2X6 VERTICAL

FIG.28C-1

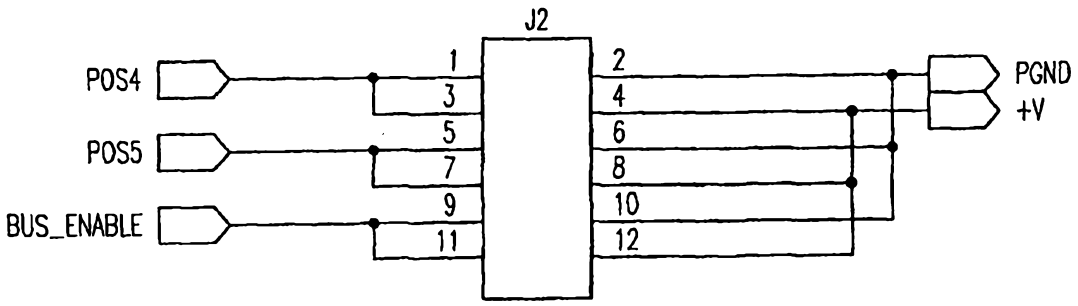
COLETOR DE POTÊNCIA



COLETOR RS232



PONTES



MACHO 2X6 VERTICAL

FIG.28C-2

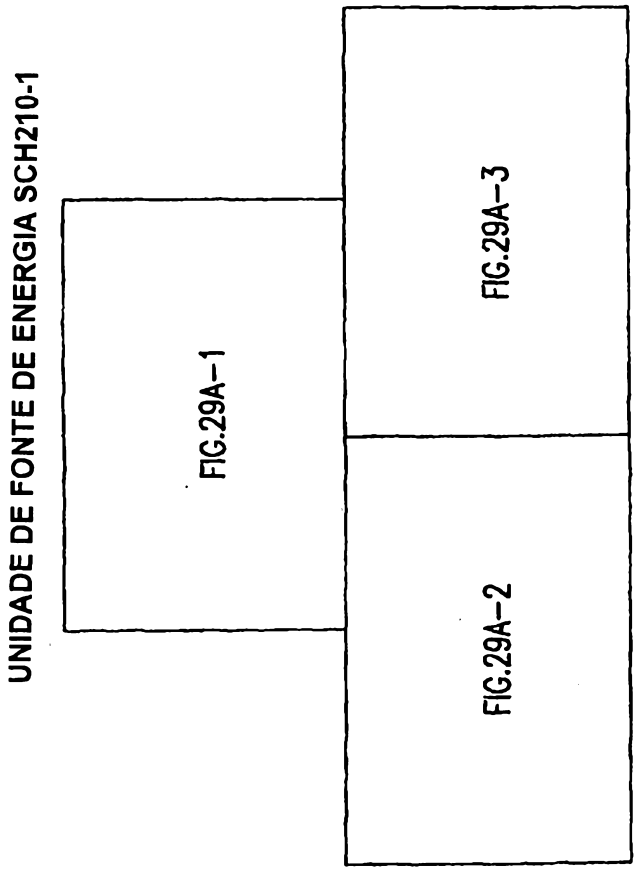

UNIDADE DE FONTE DE ENERGIA SCH210-1
FIG.29A-1
FIG.29A-2
FIG.29A-3

FIG.29A

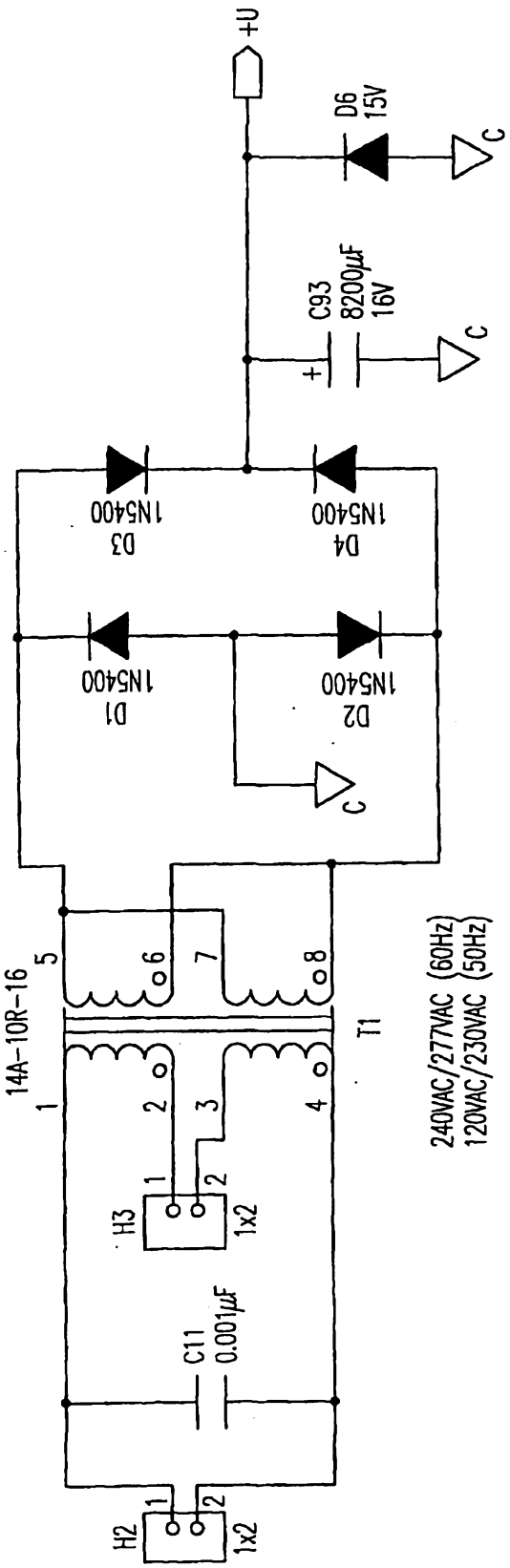

+U
D6
15V
C
C93
8200µF
16V
C
+
D3
1N5400
D4
1N5400
D1
1N5400
D2
1N5400
C
14A-10R-16
1
2
3
4
5
6
7
8
T1
240VAC/277VAC (60Hz)
120VAC/230VAC (50Hz)
H3
1x2
C11
0.001µF
H2
1x2

FIG.29A-1

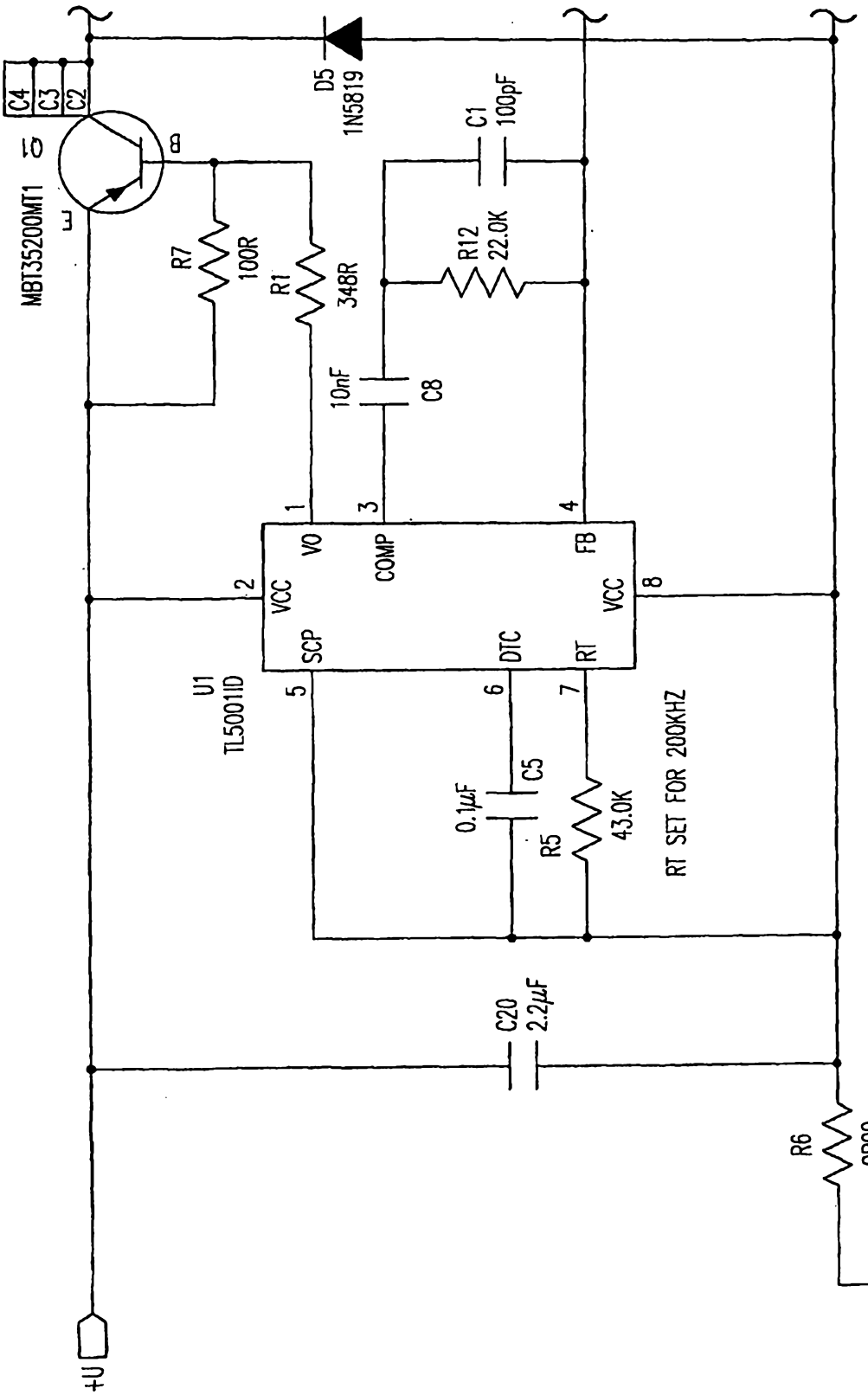

C4
C3
C2
MBT35200MT1
D5
1N5819
C1
100pF
R7
100R
R1
348R
R12
22.0K
10nF
C8
U1
TL5001ID
VO
COMP
FB
VCC
SCP
DTC
RT
VCC
0.1µF
C5
R5
43.0K
RT SET FOR 200KHZ
C20
2.2µF
R6
0R00
+U
Vc

FIG.29A-2

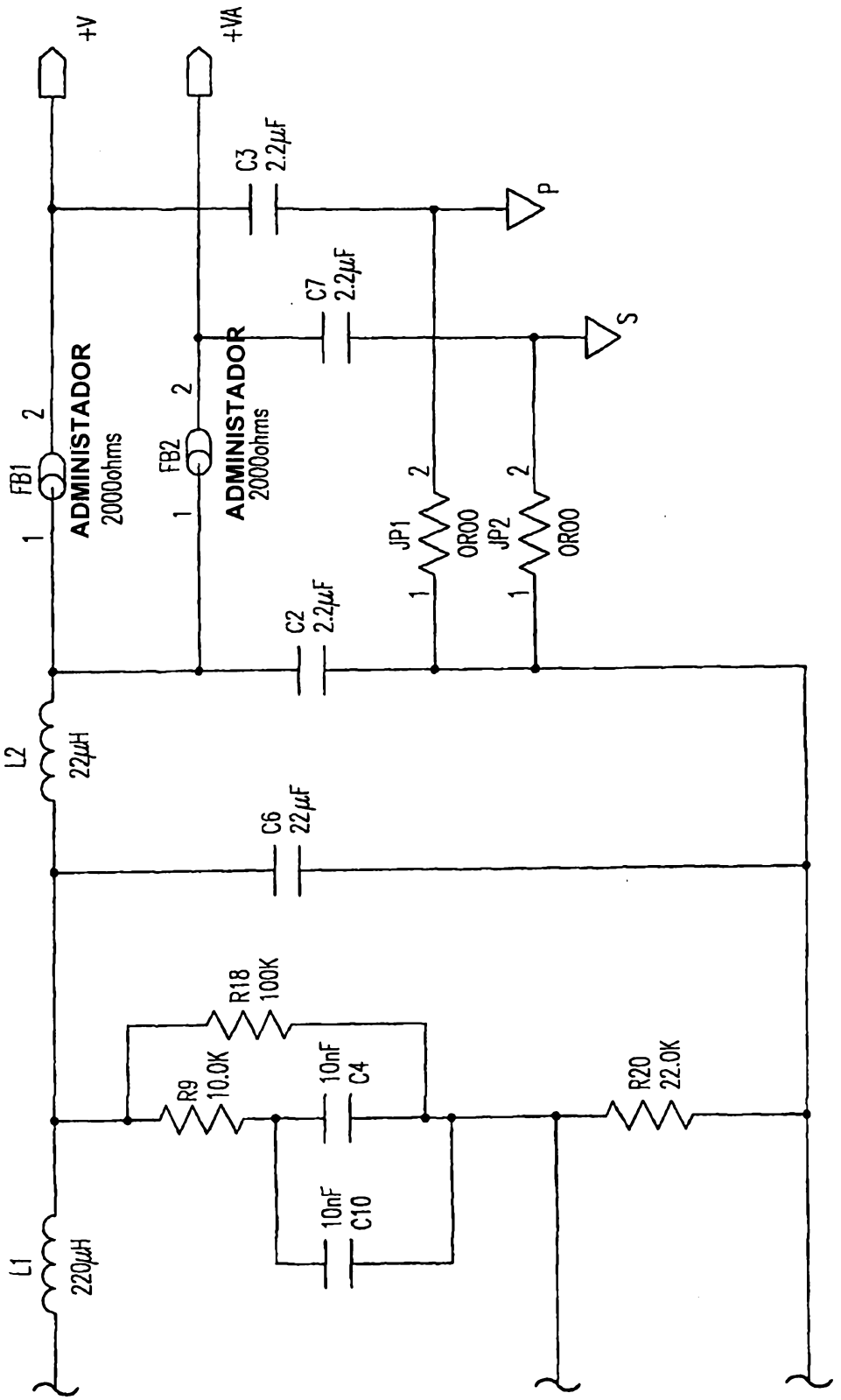

+V
+VA
C3
2.2µF
P
C7
2.2µF
S
FB1
ADMINISTADOR
2000ohms
FB2
ADMINISTADOR
2000ohms
JP1
0R00
JP2
0R00
C2
2.2µF
L2
22µH
C6
22µF
R18
100K
R9
10.0K
10nF
C4
10nF
C10
R20
22.0K
L1
220µH

FIG.29A-3

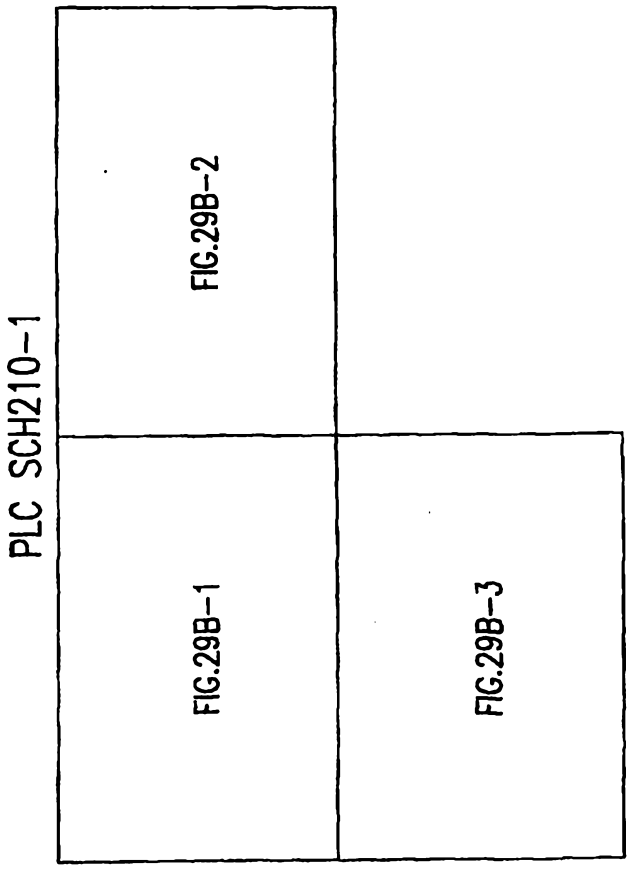


FIG.29B

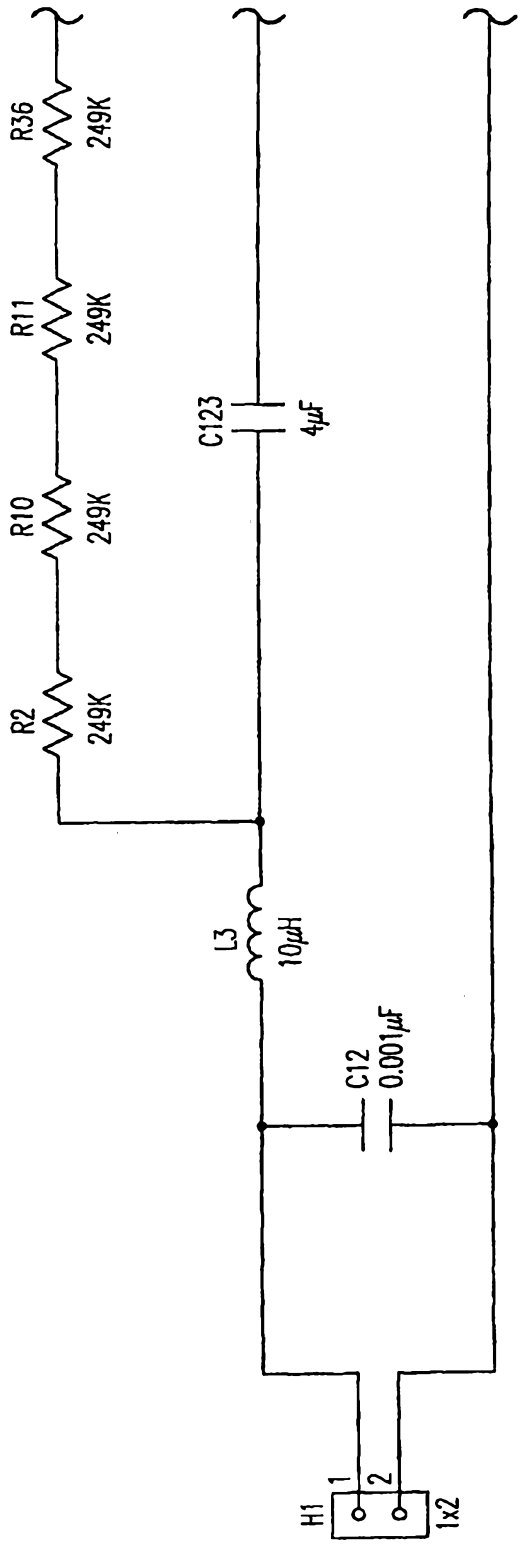

R36
249K
R11
249K
C123
4µF
R10
249K
R2
249K
L3
10µH
C12
0.001µF
H1
1
2
1x2

FIG.29B-1

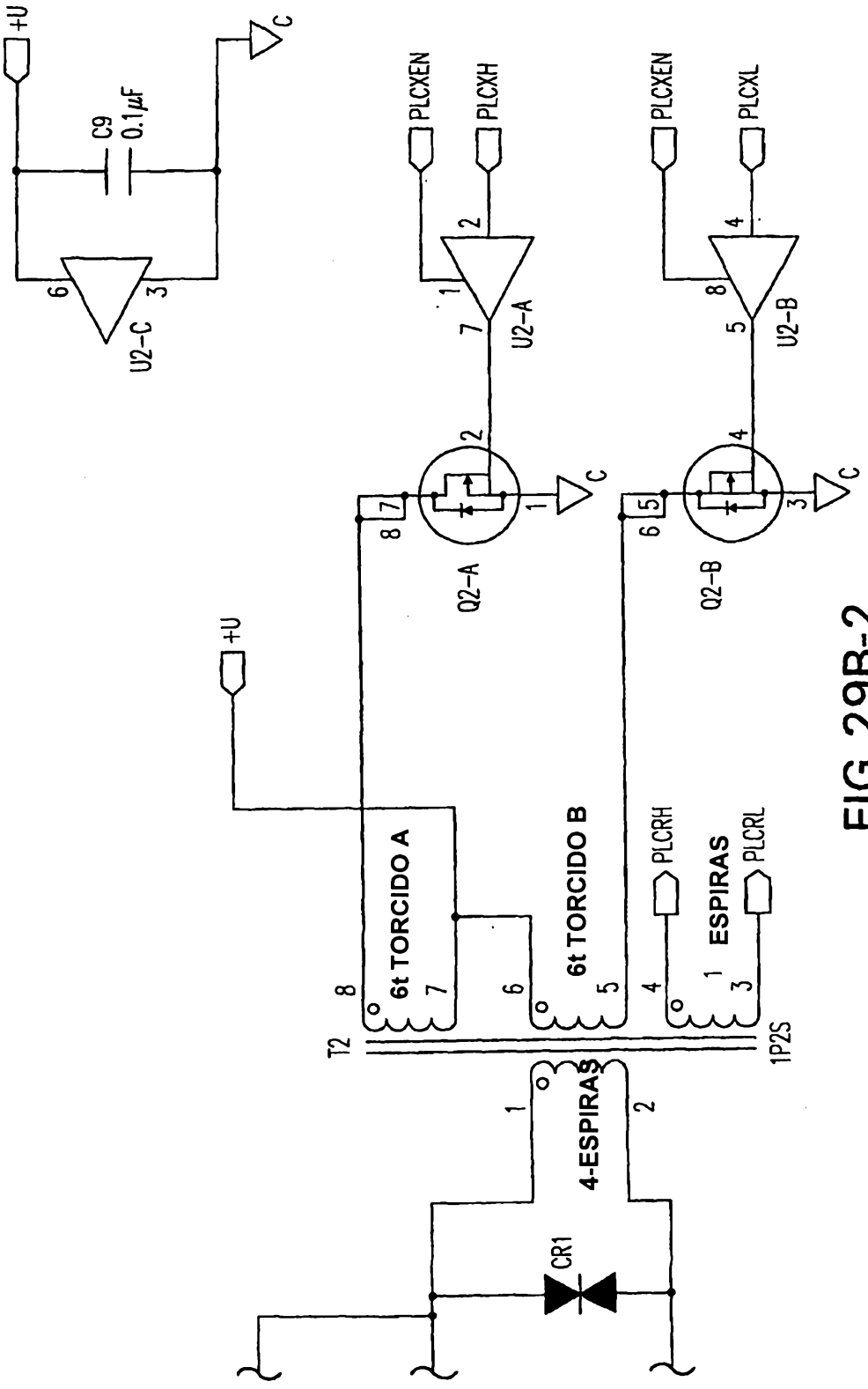

+U
C9
0.1µF
C
6
3
U2-C
PLCXEN
PLCXH
2
1
7
U2-A
PLCXEN
PLCXL
4
8
5
U2-B
2
4
8
7
1
C
Q2-A
6
5
3
C
Q2-B
+U
8
7
6t TORCIDO A
6
5
6t TORCIDO B
4
1
3
PLCRH
ESPIRAS
PLCRL
T2
1P2S
1
4-ESPIRAS
2
CR1

FIG.29B-2

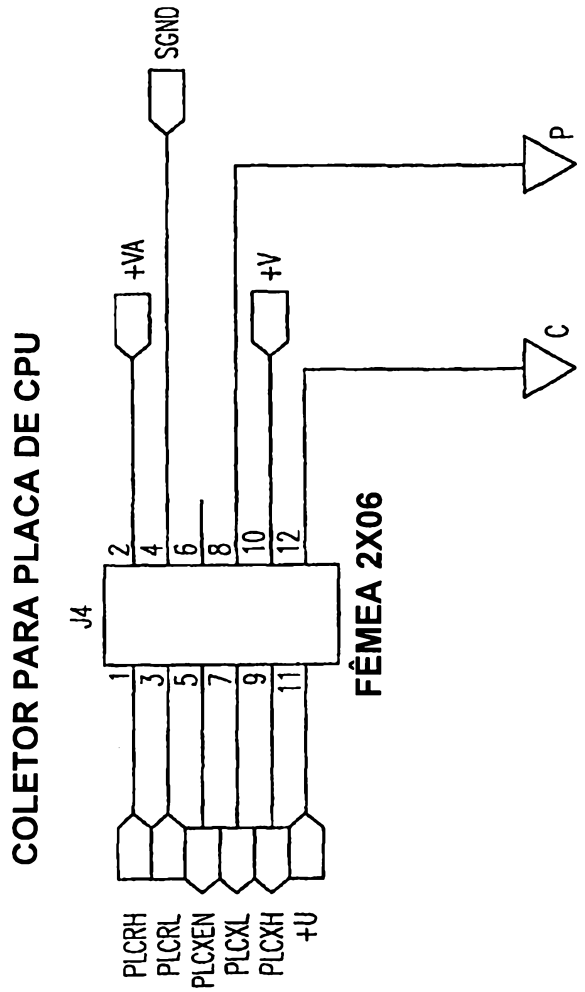

COLETOR PARA PLACA DE CPU
J4
FÊMEA 2X06
SGND
+VA
+V
P
C
PLCRH
PLCRL
PLCXEN
PLCXL
PLCXH
+U
1
3
5
7
9
11
2
4
6
8
10
12

FIG.29B-3

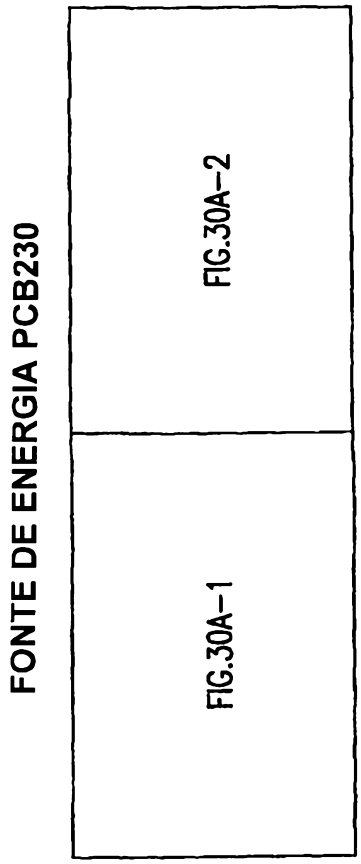


FIG.30A

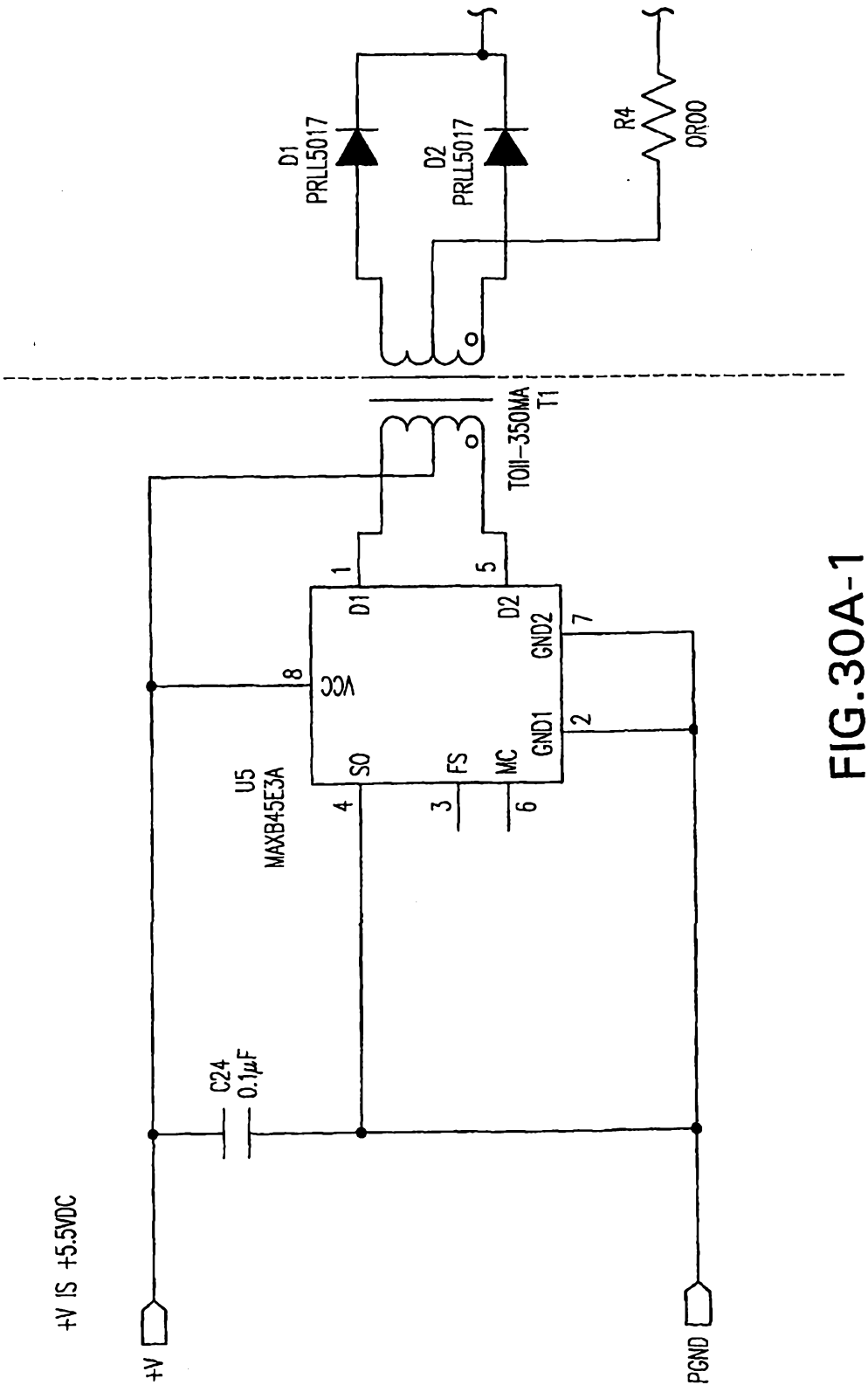

D1
PRLL5017
D2
PRLL5017
R4
ORDO
T1
TOII-350MA
1
5
D1
D2
GND2
7
8
VCC
2
GND1
U5
MAXB45E3A
SO
FS
MC
4
3
6
C24
0.1µF
+V IS +5.5VDC
+V
PGND

FIG.30A-1

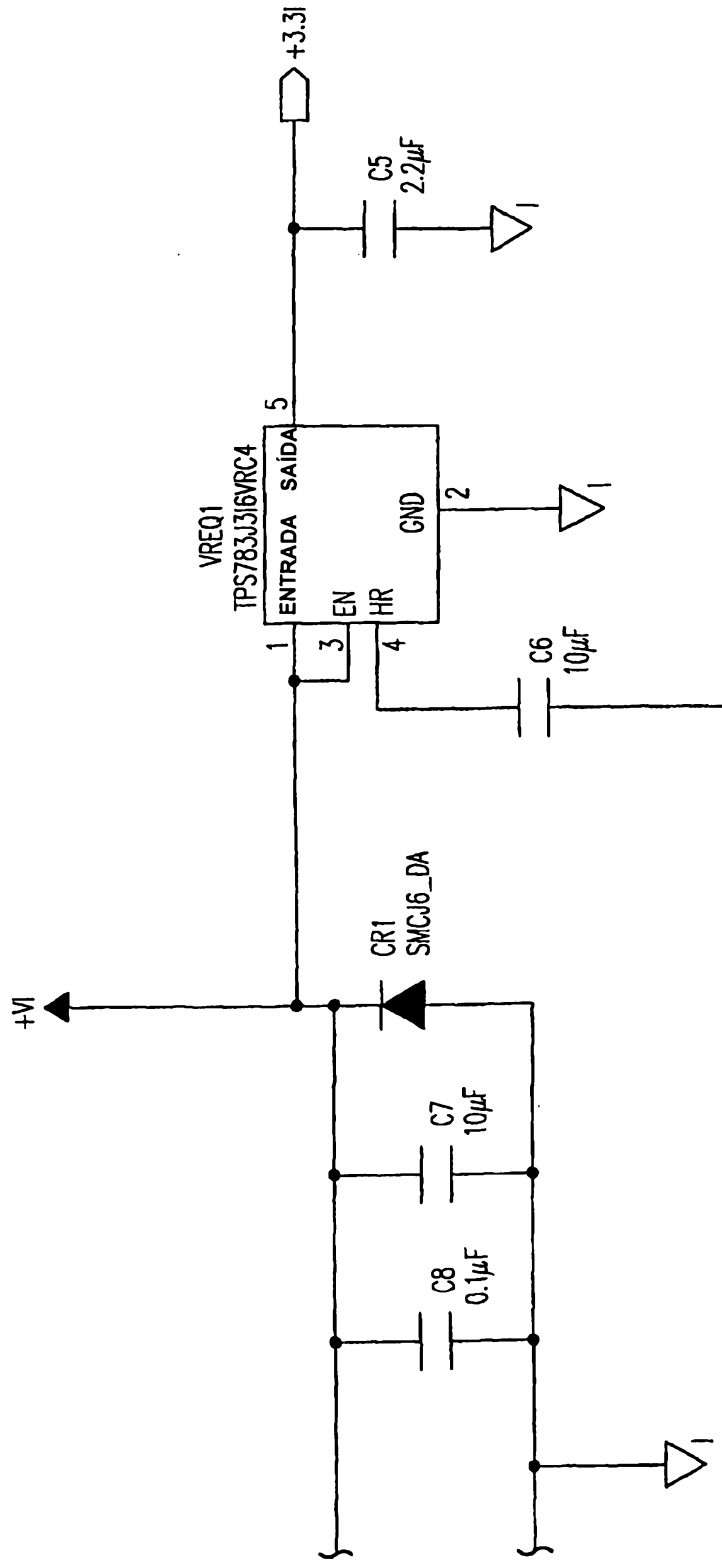

+3.3I
C5
2.2µF
VREQ1
TPS783J3I6VRC4
ENTRADA
SAÍDA
GND
EN
HR
1
2
3
4
5
C6
10µF
CR1
SMCJ6_DA
+VI
C7
10µF
C8
0.1µF

FIG.30A-2

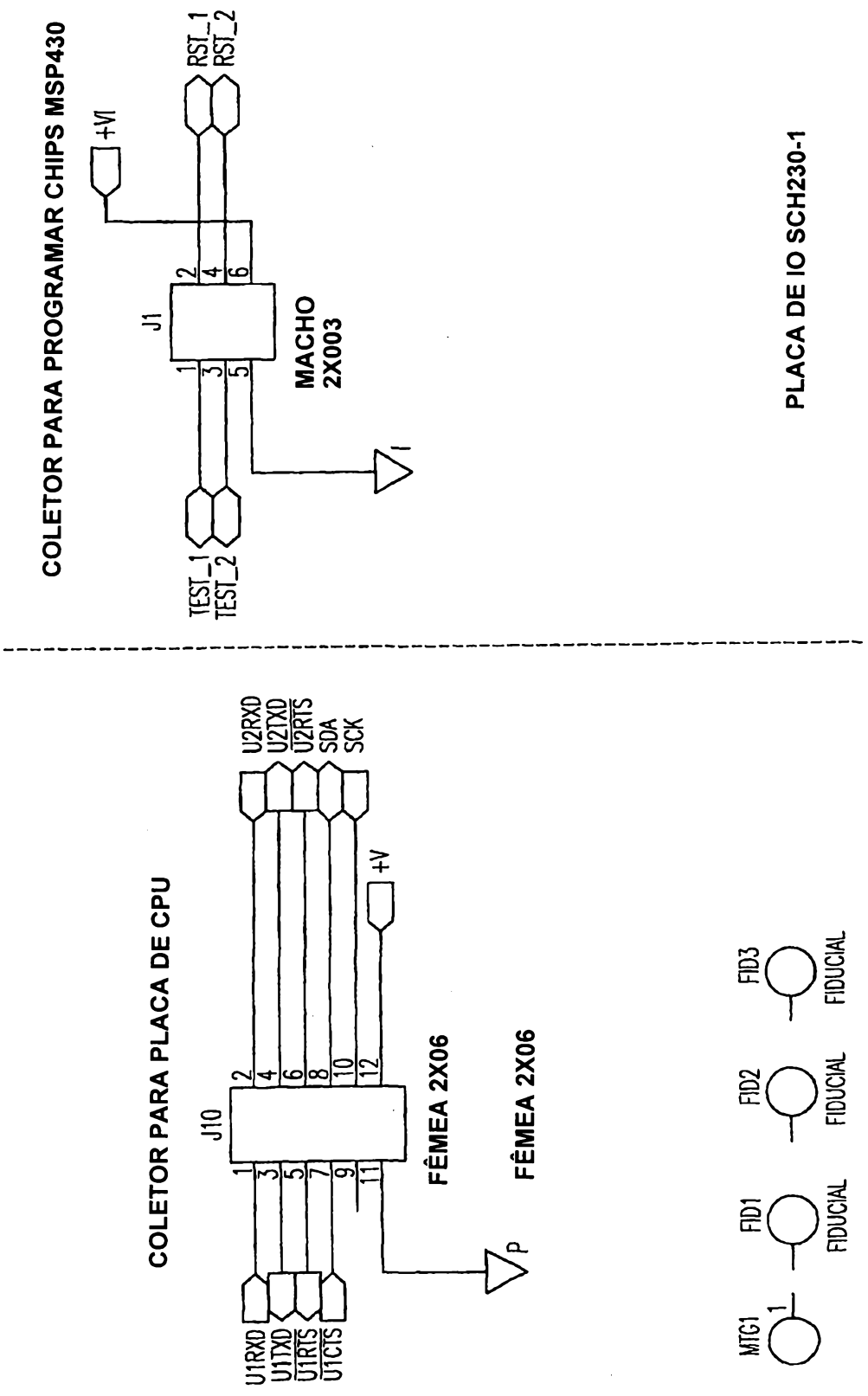

COLETOR PARA PROGRAMAR CHIPS MSP430
+VI
RST_1
RST_2
J1
MACHO
2X003
TEST_1
TEST_2
PLACA DE IO SCH230-1
COLETOR PARA PLACA DE CPU
U2RXD
U2TXD
U2RTS
SDA
SCK
+V
J10
FÊMEA 2X06
FÊMEA 2X06
U1RXD
U1TXD
U1RTS
U1CTS
MTG1
FID1
FIDUCIAL
FID2
FIDUCIAL
FID3
FIDUCIAL

FIG.30B

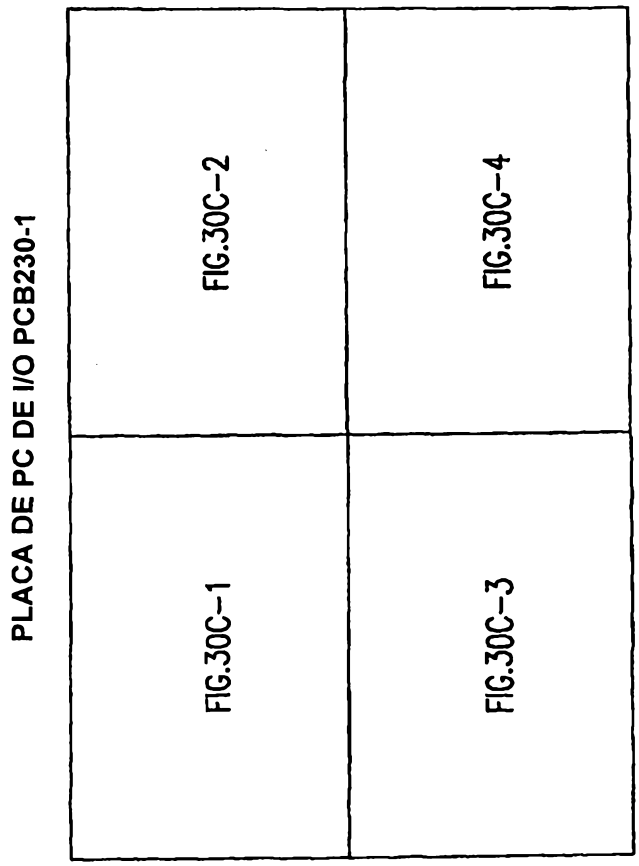


FIG.30C

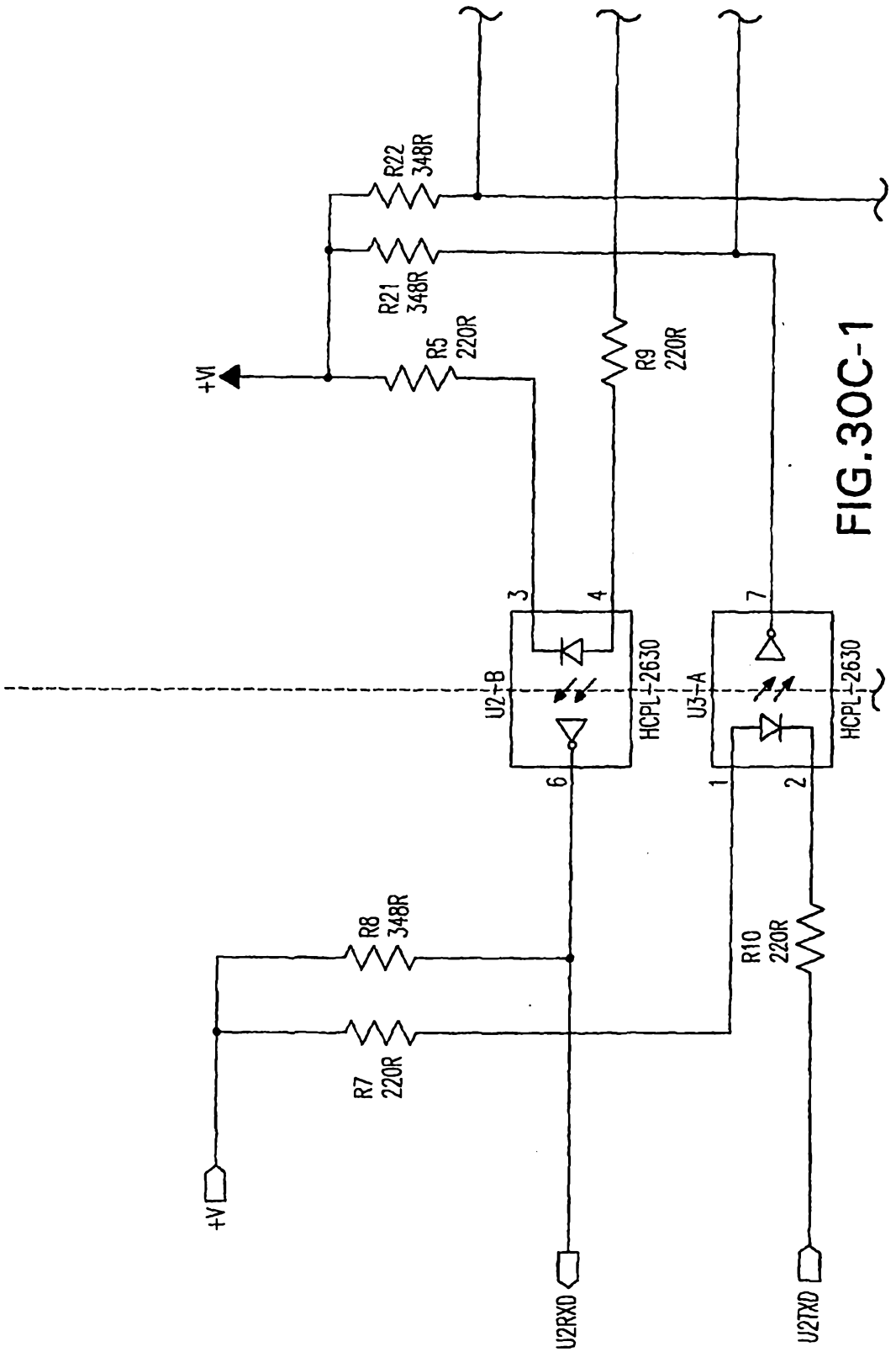

R22
348R
R21
348R
R5
220R
R9
220R
+VI
3
4
7
U2:B
HCPL-2630
U3:A
HCPL-2630
6
1
2
R8
348R
R7
220R
R10
220R
+V
U2RXD
U2TXD

FIG.30C-1

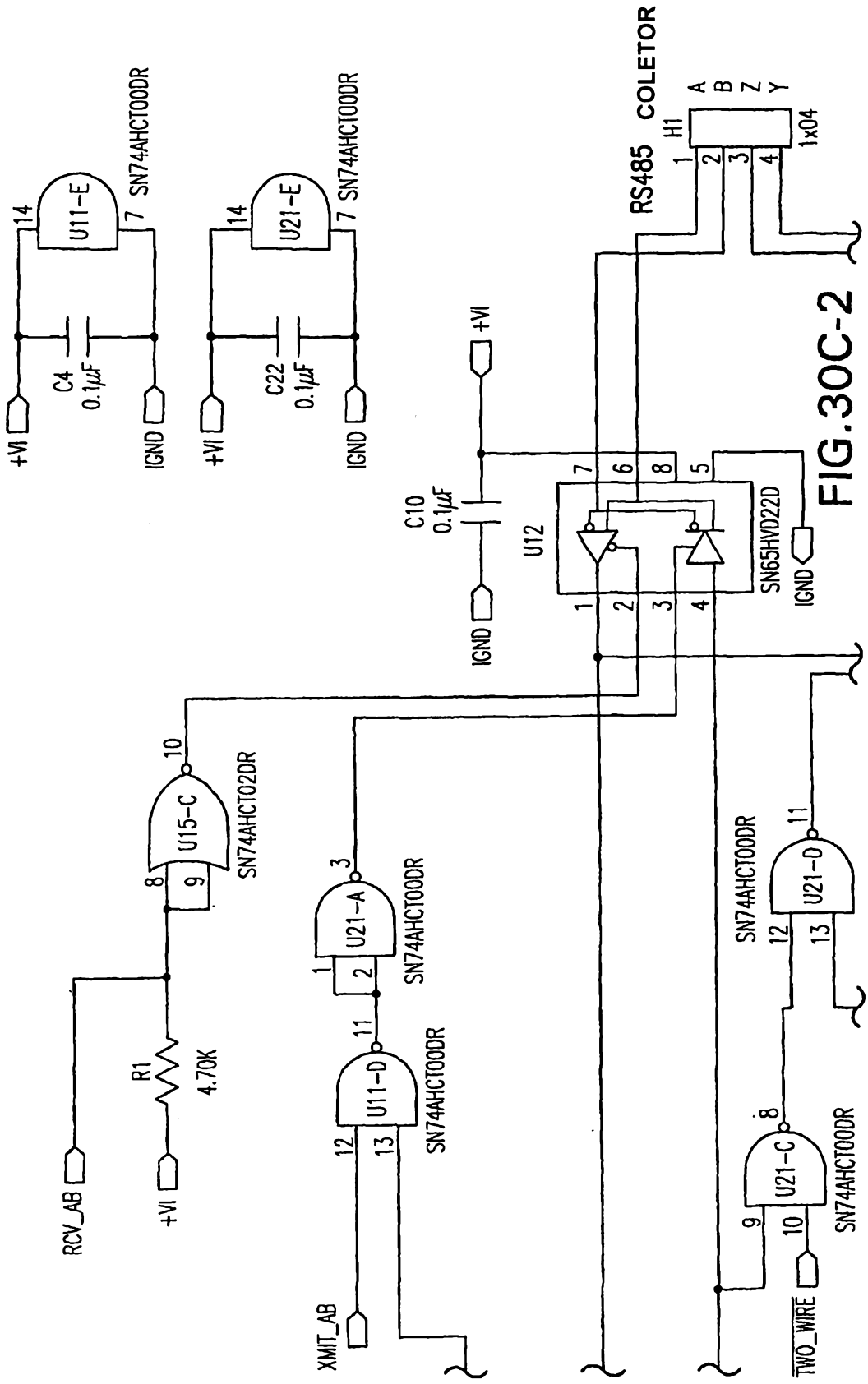

RS485 COLETOR
H1
1x04
A
B
Z
Y
U11-E
SN74AHCT00DR
U21-E
SN74AHCT00DR
C4
0.1µF
C22
0.1µF
+VI
IGND
C10
0.1µF
U12
SN65HVD22D
U15-C
SN74AHCT02DR
U21-A
SN74AHCT00DR
U11-D
SN74AHCT00DR
U21-D
SN74AHCT00DR
U21-C
SN74AHCT00DR
R1
4.70K
RCV_AB
XMIT_AB
TWO_WIRE

FIG.30C-2

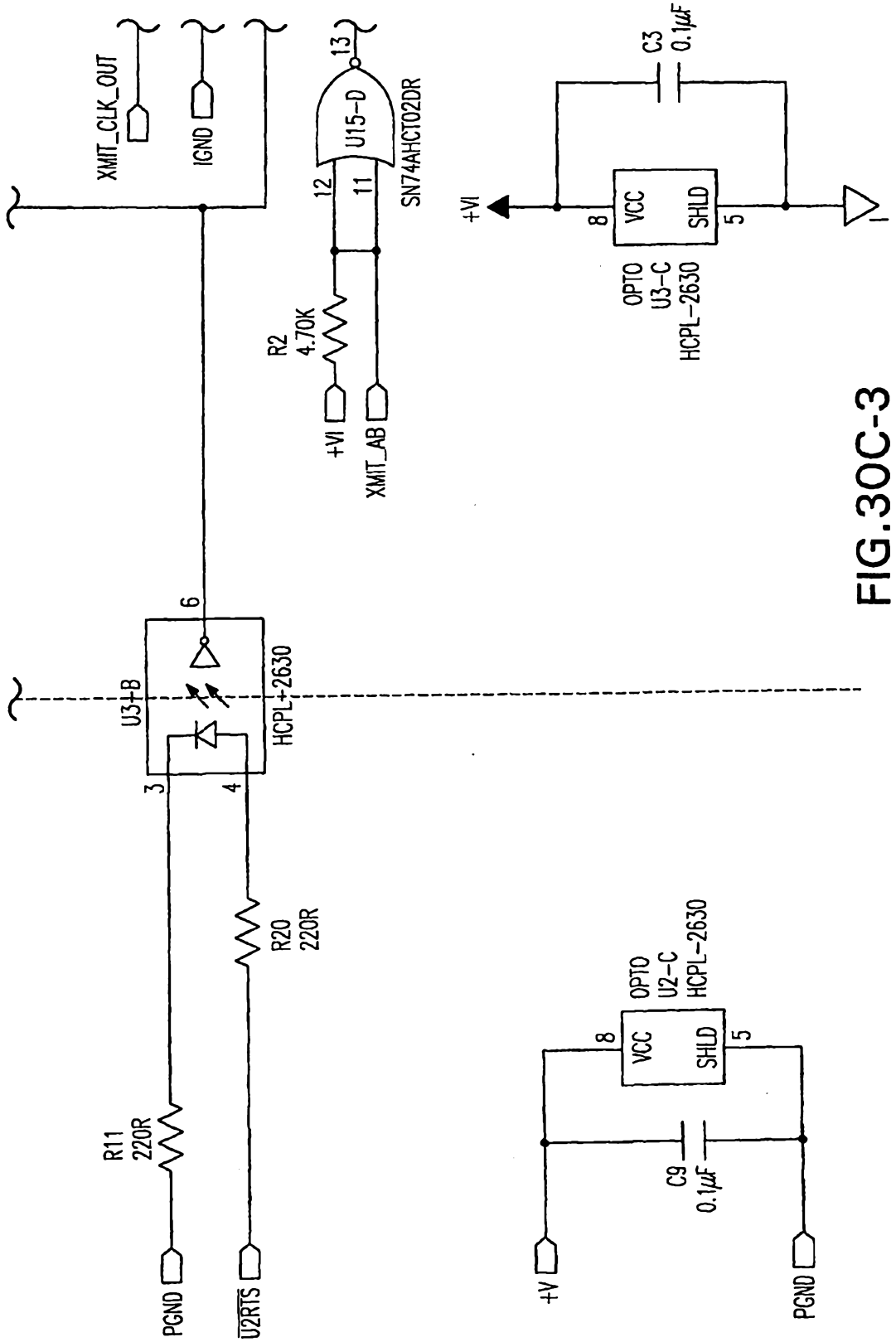

XMIT_CLK_OUT
IGND
U15-D
SN74AHCT02DR
13
12
11
R2
4.70K
+VI
XMIT_AB
U3-B
HCPL-2630
6
3
4
R11
220R
R20
220R
PGND
U2RTS
C3
0.1µF
+VI
8
VCC
SHLD
5
OPTO
U3-C
HCPL-2630
OPTO
U2-C
HCPL-2630
C9
0.1µF
+V
PGND

FIG.30C-3

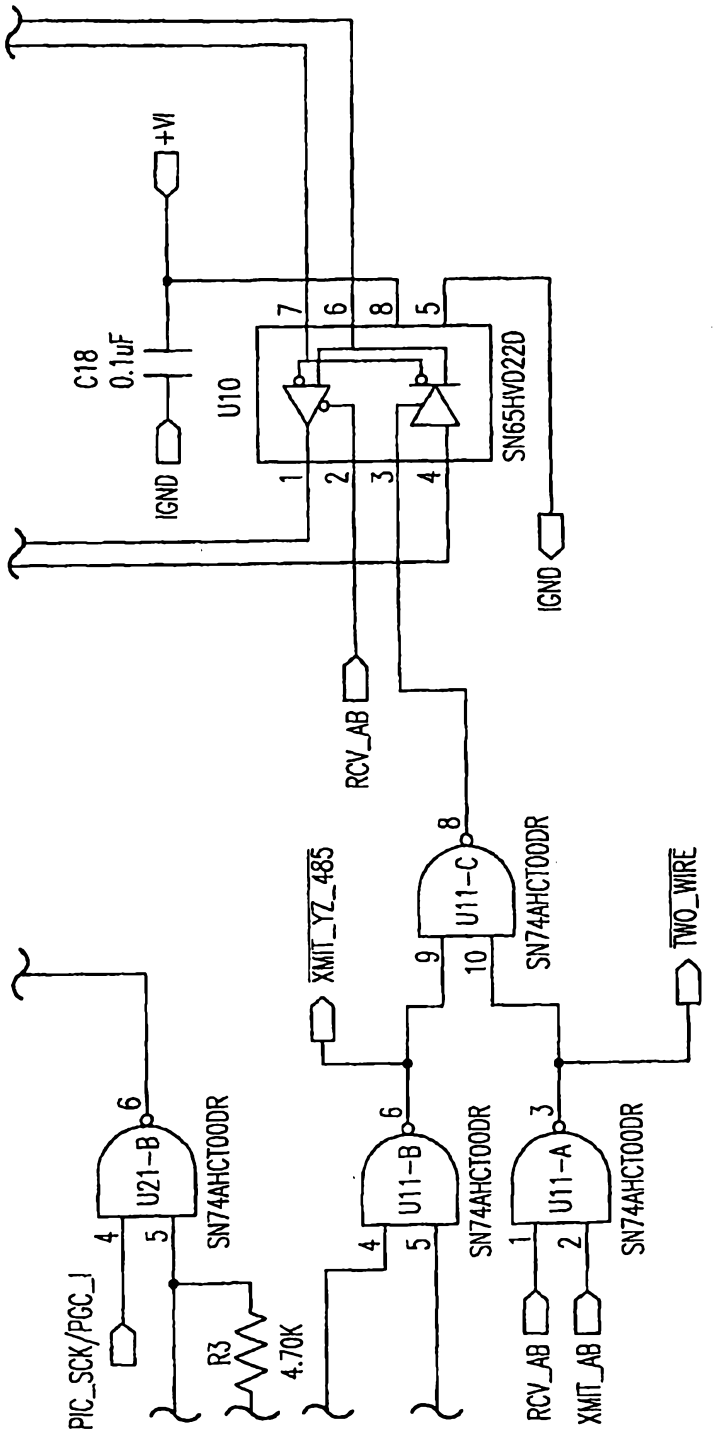

+VI
C18
0.1uF
IGND
U10
SN65HVD22D
IGND
RCV_AB
XMIT_YZ_485
U11-C
SN74AHCT00DR
TWO_WIRE
U21-B
SN74AHCT00DR
U11-B
SN74AHCT00DR
U11-A
SN74AHCT00DR
PIC_SCK/PGC_I
R3
4.70K
RCV_AB
XMIT_AB

FIG.30C-4

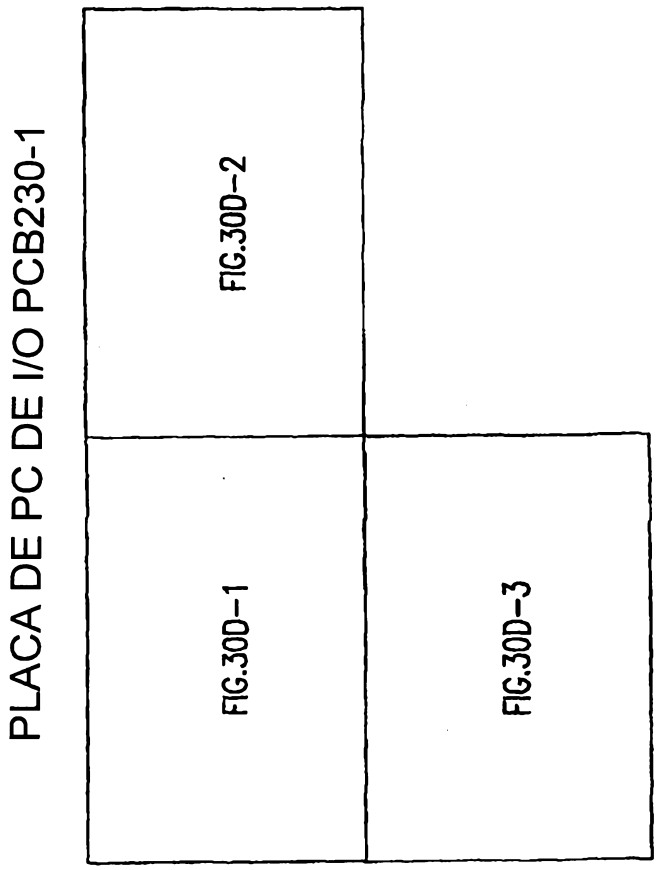


FIG.30D

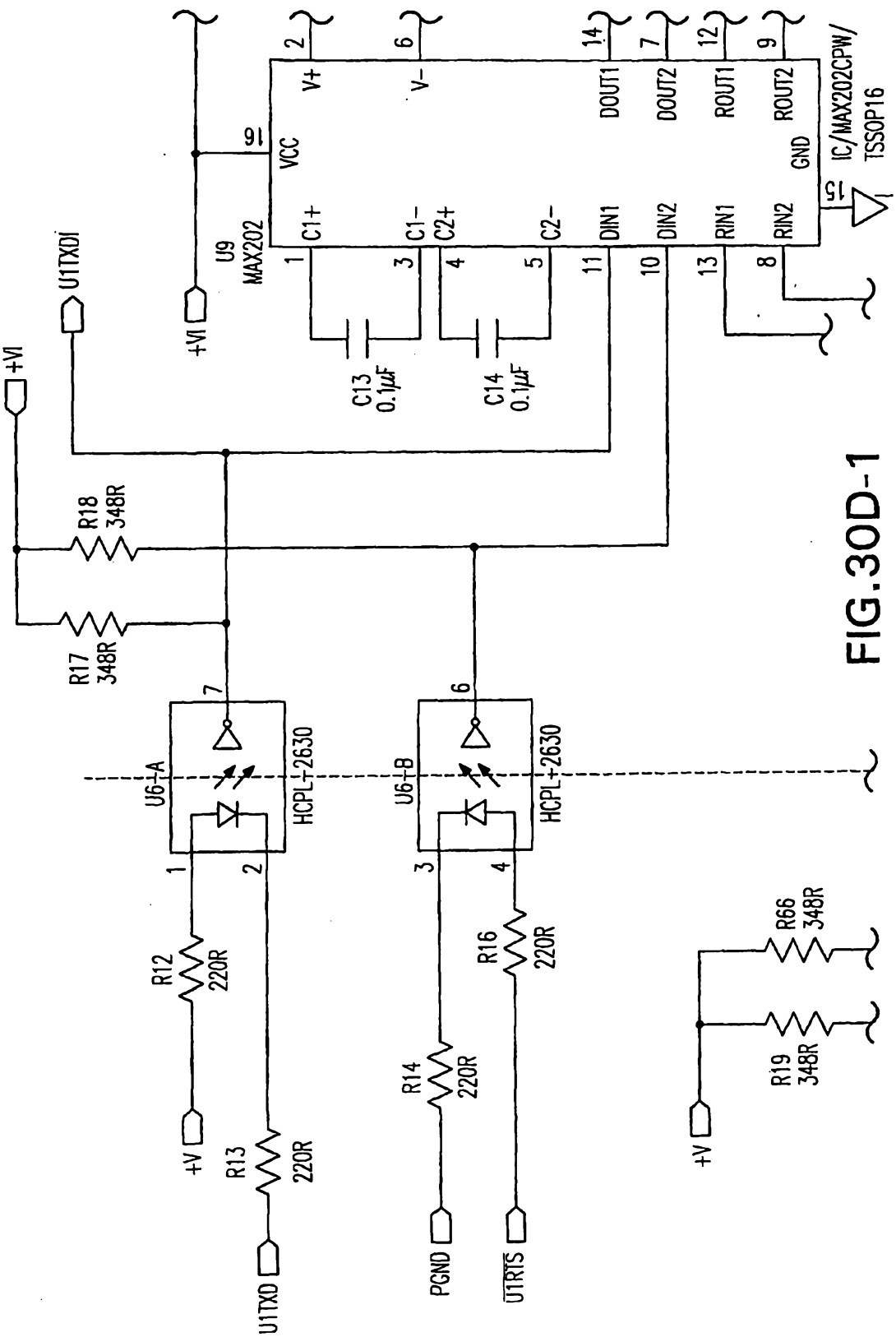

+VI
U1TXDI
R18
348R
R17
348R
U6-A
HCPL-2630
U6-B
HCPL-2630
R12
220R
R13
220R
R14
220R
R16
220R
+V
U1TXD
PGND
U1RTS
U9
MAX202
VCC
V+
V-
C1+
C1-
C2+
C2-
DIN1
DIN2
RIN1
RIN2
DOUT1
DOUT2
ROUT1
ROUT2
GND
C13
0.1μF
C14
0.1μF
IC/MAX202CPW/
TSSOP16
R66
348R
R19
348R

FIG.30D-1

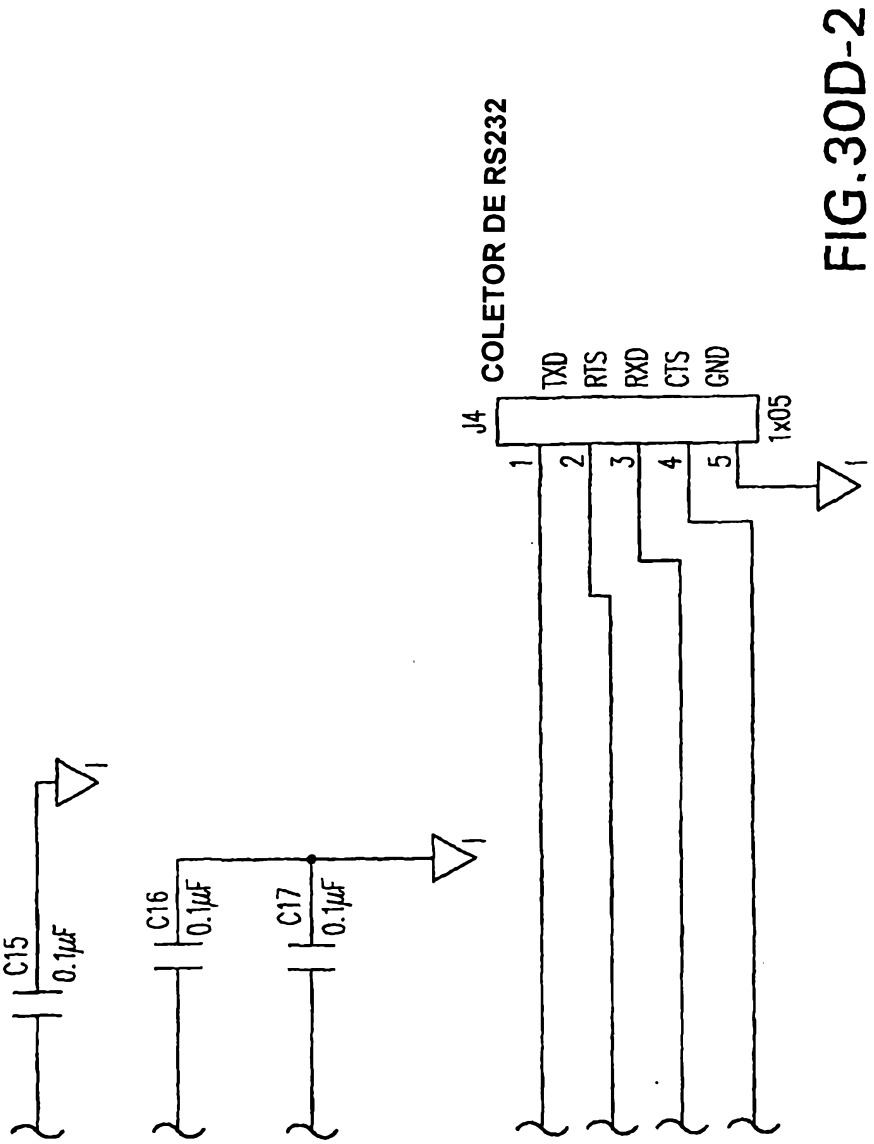

C15
0.1µF
C16
0.1µF
C17
0.1µF
J4
COLETOR DE RS232
TXD
RTS
RXD
CTS
GND
1x05
1
2
3
4
5

FIG.30D-2

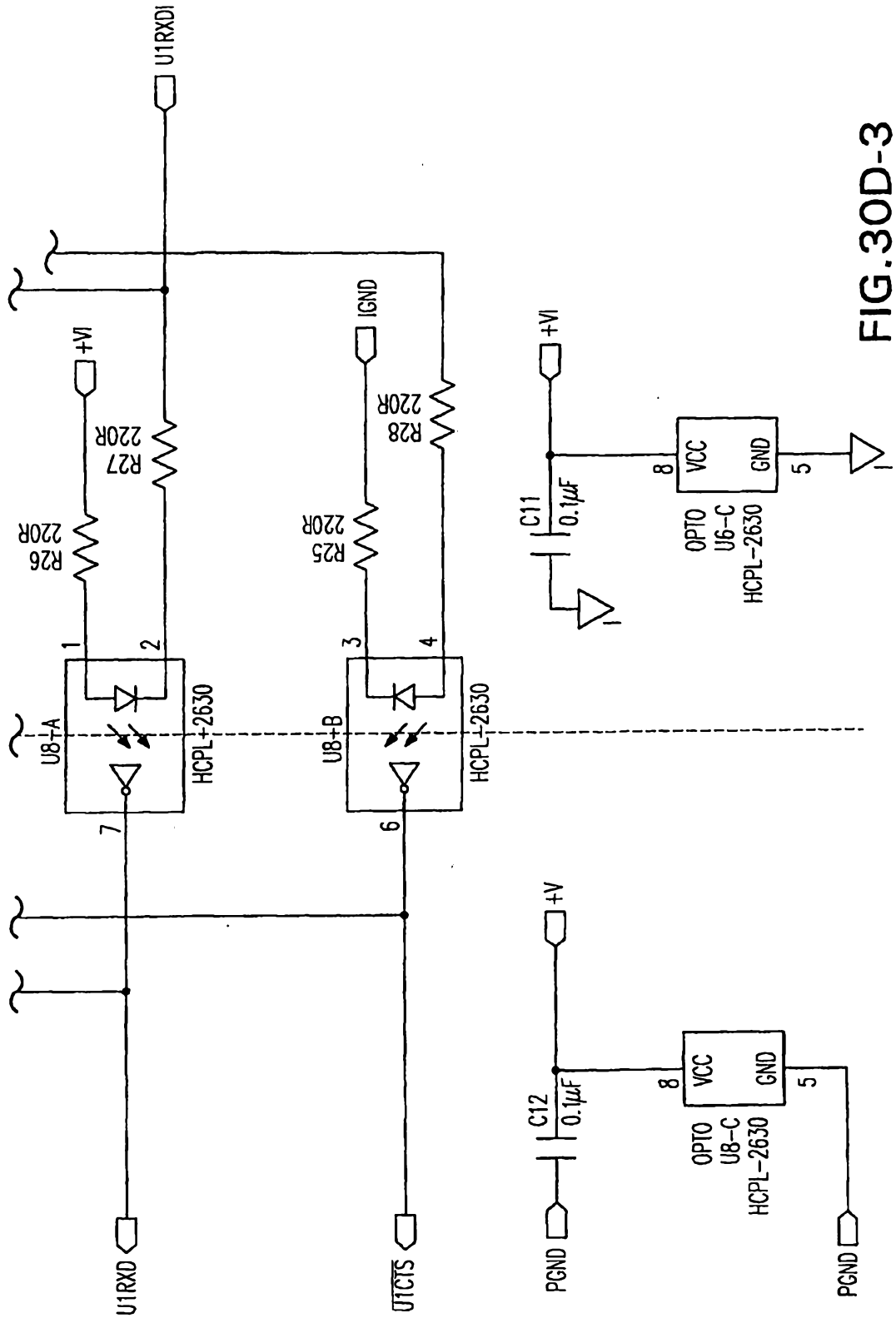

U1RXDI
+VI
R27
220R
R26
220R
IGND
R28
220R
R25
220R
U8-A
HCPL-2630
U8-B
HCPL-2630
U1RXD
U1CTS
+VI
C11
0.1µF
OPTO
U6-C
HCPL-2630
VCC
GND
+V
C12
0.1µF
OPTO
U8-C
HCPL-2630
PGND
PGND

FIG.30D-3

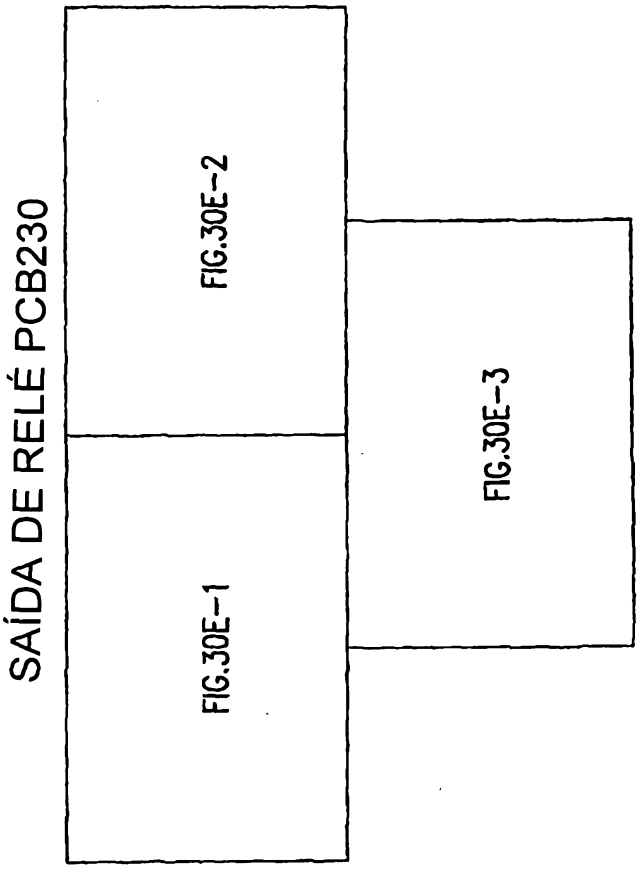


FIG.30E

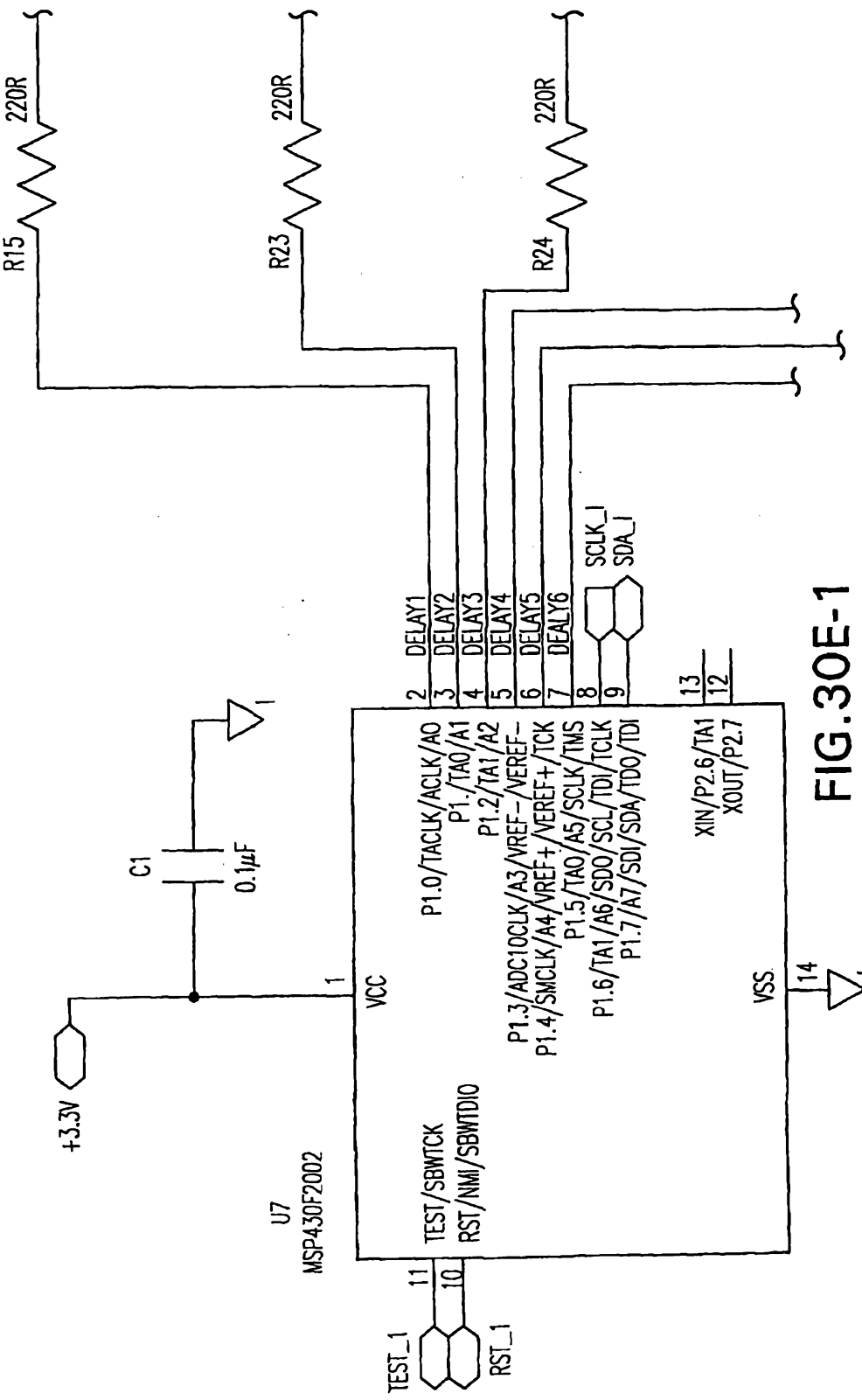

R15
220R
R23
220R
R24
220R
DELAY1
DELAY2
DELAY3
DELAY4
DELAY5
DELAY6
SCLK_1
SDA_1
C1
0.1µF
+3.3V
U7
MSP430F2002
VCC
VSS
TEST/SBWTCK
RST/NMI/SBWTDIO
P1.0/TACLK/ACLK/A0
P1.1/TA0/A1
P1.2/TA1/A2
P1.3/ADC10CLK/A3/VREF-/VEREF-
P1.4/SMCLK/A4/VREF+/VEREF+/TCK
P1.5/TA0/A5/SCLK/TMS
P1.6/TA1/A6/SDO/SCL/TDI/TCLK
P1.7/A7/SDI/SDA/TDO/TDI
XIN/P2.6/TA1
XOUT/P2.7
TEST_1
RST_1

FIG.30E-1

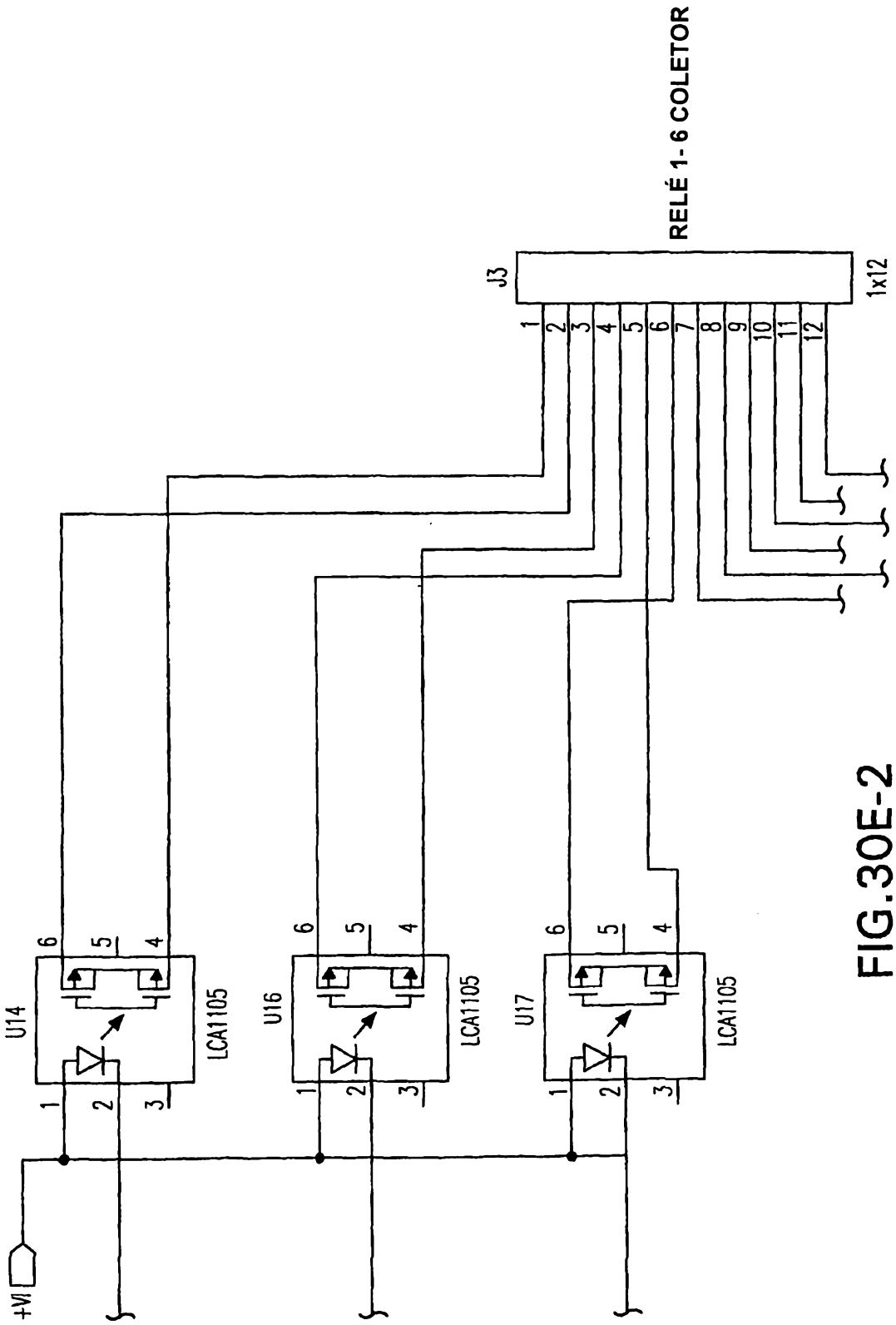

RELÉ 1- 6 COLETOR
J3
1x12
1
2
3
4
5
6
7
8
9
10
11
12
U14
U16
U17
LCA1105
+VI

FIG.30E-2

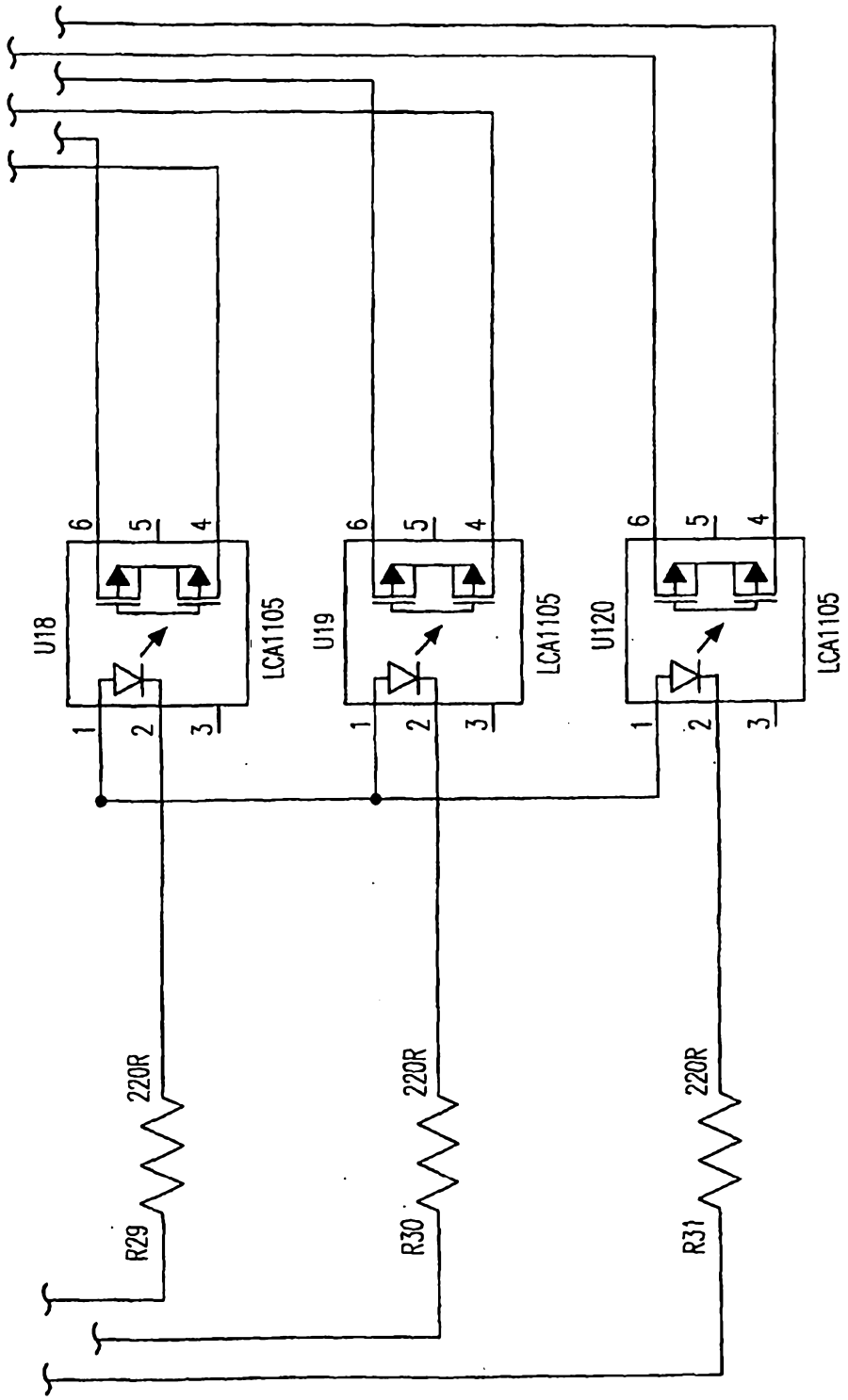

U18
U19
U120
LCA1105
LCA1105
LCA1105
1
2
3
4
5
6
220R
220R
220R
R29
R30
R31

FIG.30E-3

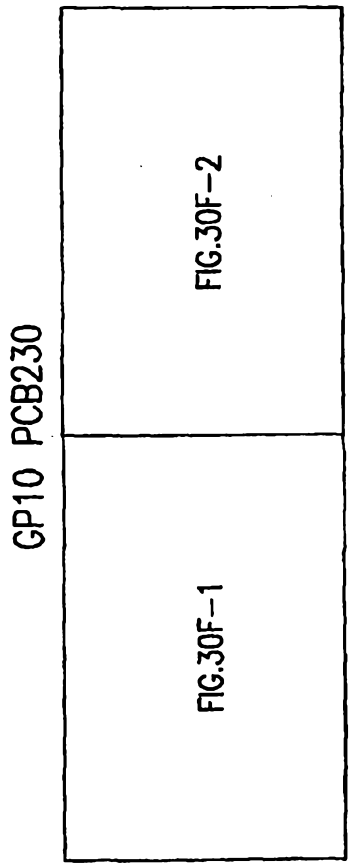


FIG.30F

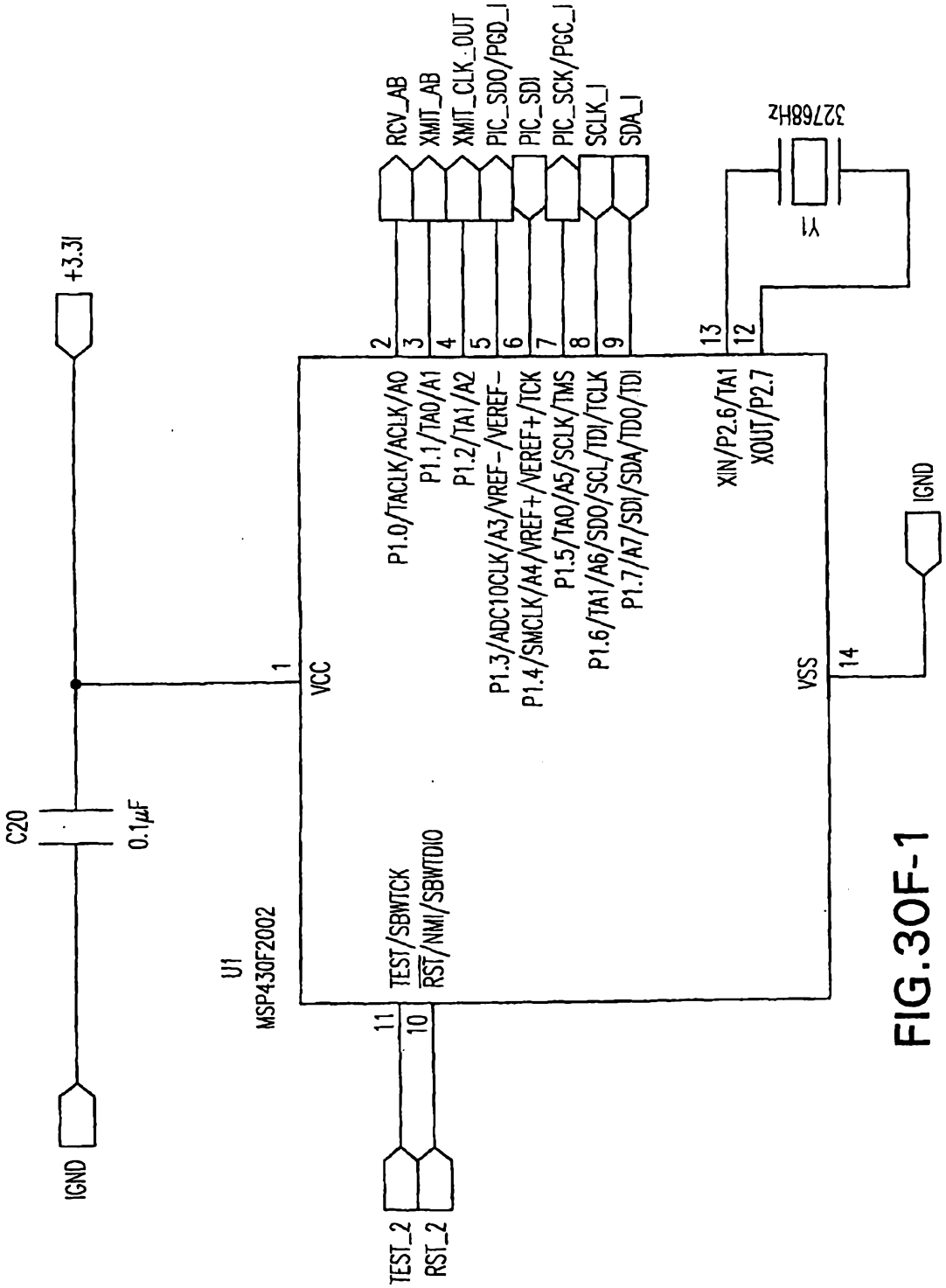

+3.3I
C20
0.1µF
IGND
U1
MSP430F2002
VCC
1
TEST/SBWTCK
RST/NMI/SBWTDIO
11
10
TEST_2
RST_2
P1.0/TACLK/ACLK/A0
P1.1/TA0/A1
P1.2/TA1/A2
P1.3/ADC10CLK/A3/VREF-/VEREF-
P1.4/SMCLK/A4/VREF+/VEREF+/TCK
P1.5/TA0/A5/SCLK/TMS
P1.6/TA1/A6/SDO/SCL/TDI/TCLK
P1.7/A7/SDI/SDA/TDO/TDI
2
3
4
5
6
7
8
9
RCV_AB
XMIT_AB
XMIT_CLK_OUT
PIC_SDO/PGD_I
PIC_SDI
PIC_SCK/PGC_I
SCLK_I
SDA_I
XIN/P2.6/TA1
XOUT/P2.7
13
12
32768HZ
Y1
VSS
14
IGND

FIG.30F-1

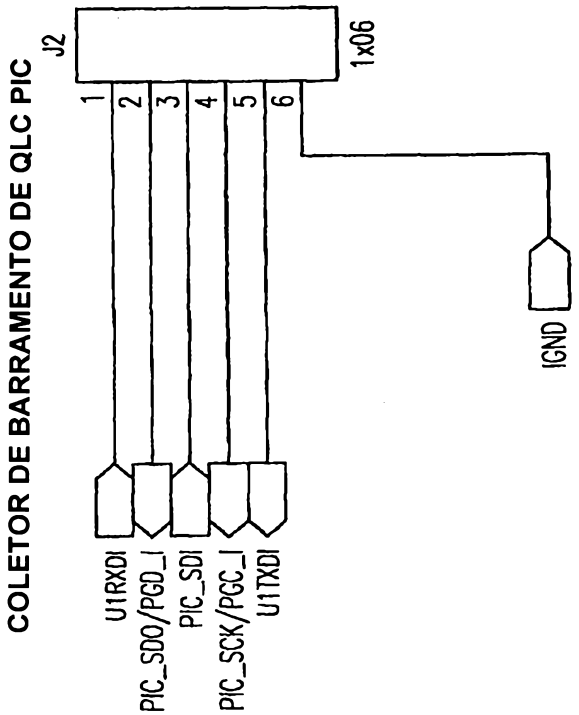

COLETOR DE BARRAMENTO DE QLC PIC
J2
1x06
1
2
3
4
5
6
U1RXDI
PIC_SDO/PGD_I
PIC_SDI
PIC_SCK/PGC_I
U1TXDI
IGND

FIG.30F-2

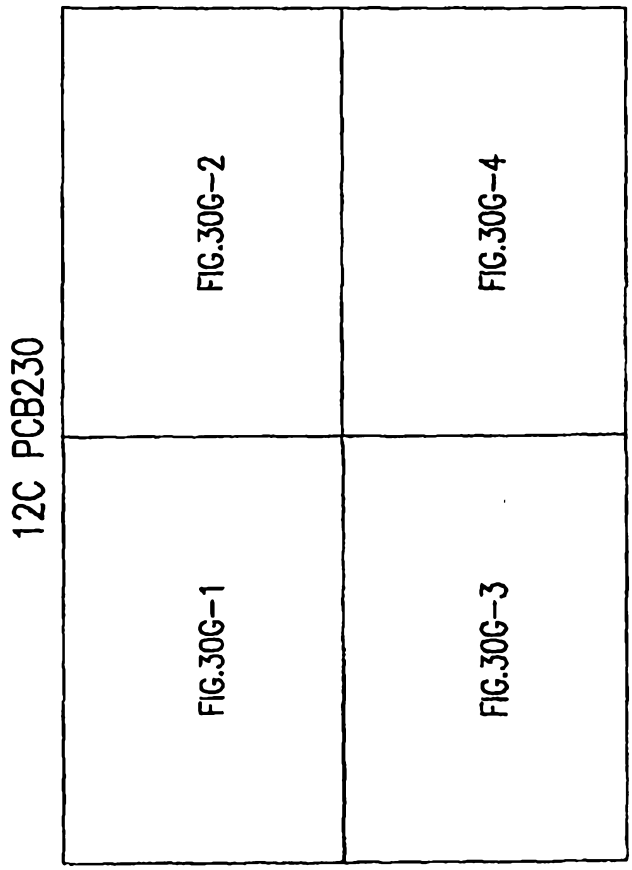


FIG.30G

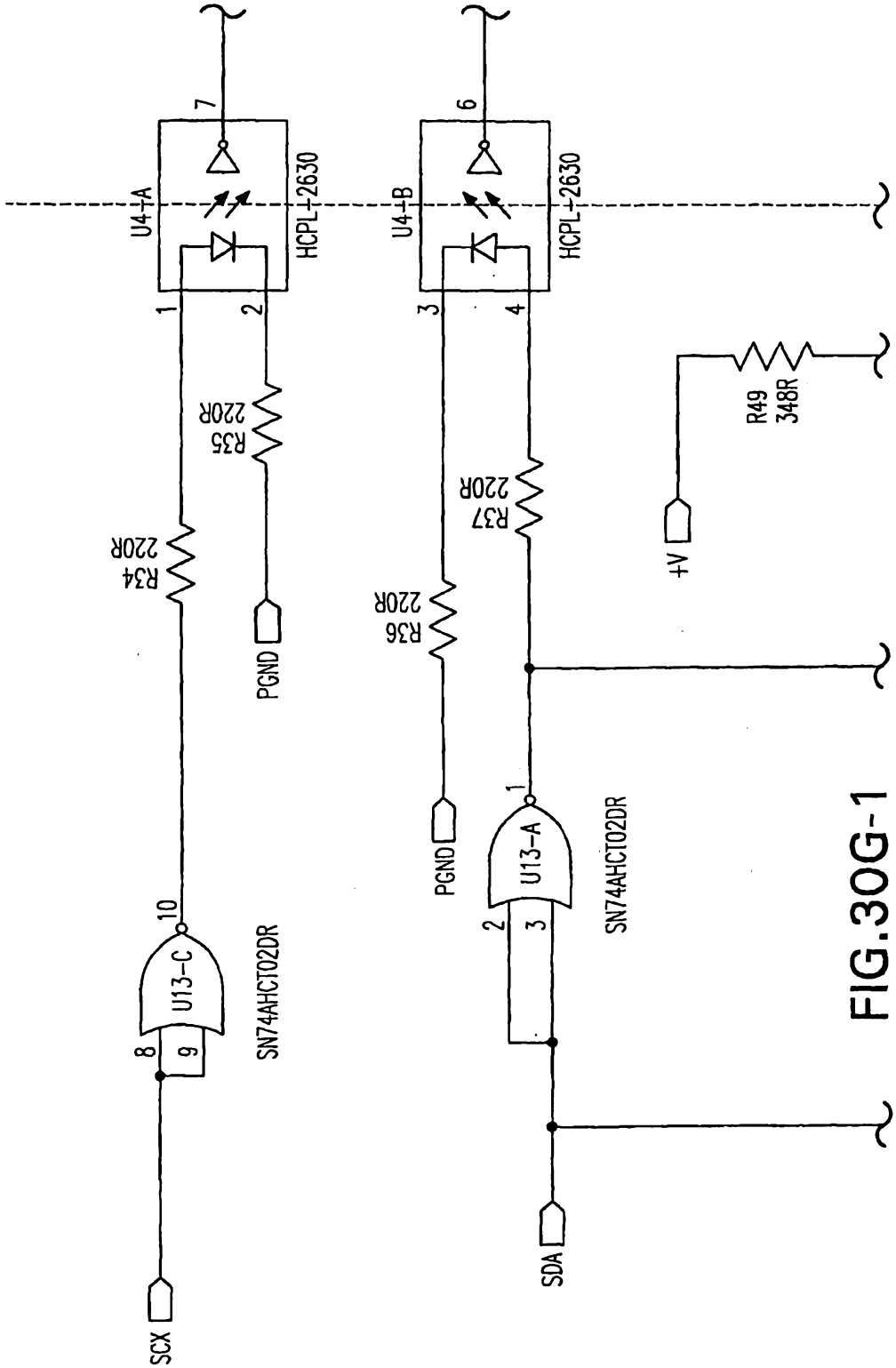

U4-A
HCPL-2630
7
1
2
R35
220R
R34
220R
PGND
SCX
U13-C
SN74AHCT02DR
10
8
9
U4-B
HCPL-2630
6
3
4
R37
220R
R36
220R
R49
348R
+V
PGND
SDA
U13-A
SN74AHCT02DR
1
2
3

FIG.30G-1

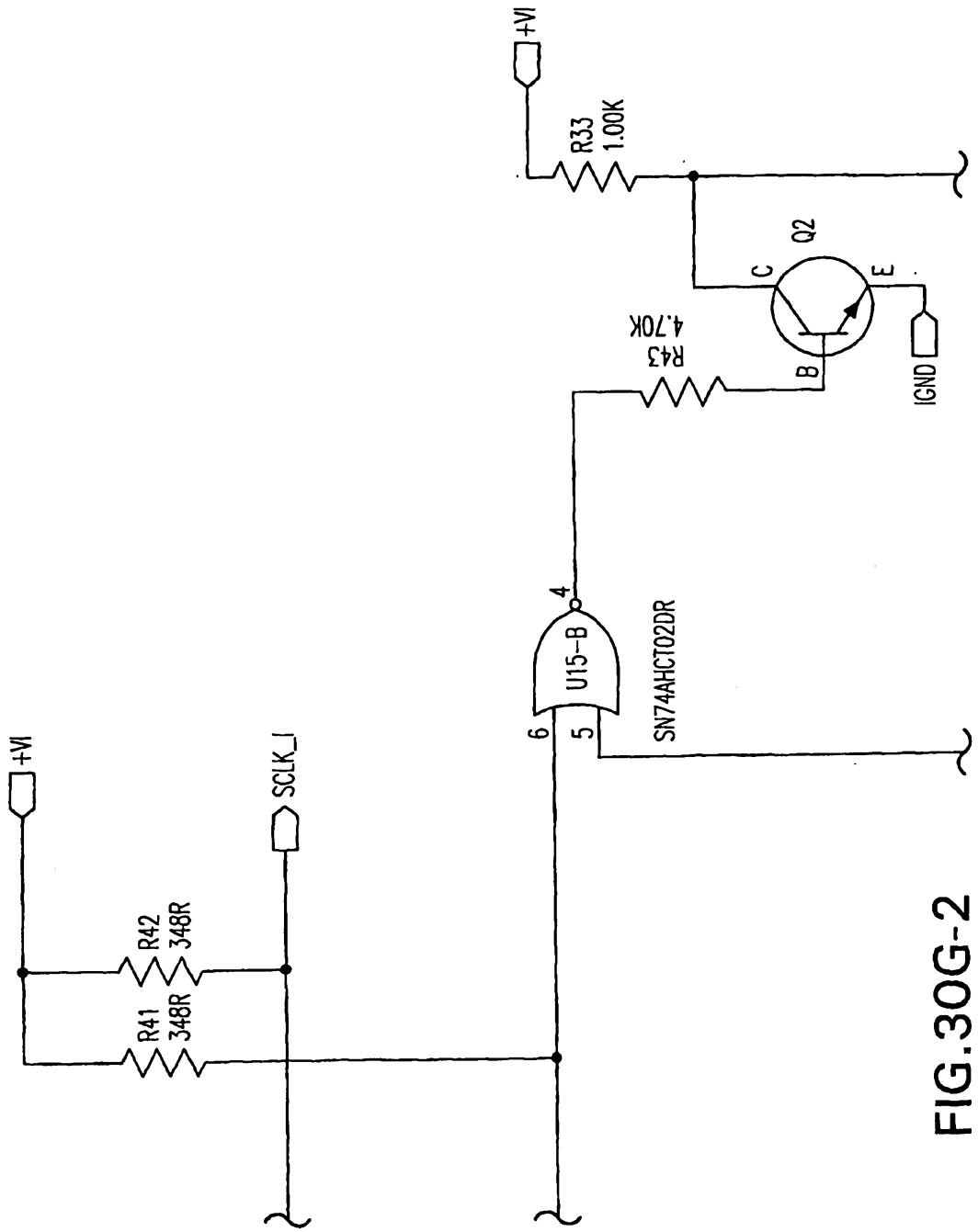

+VI
R33
1.00K
Q2
C
E
B
R43
4.70K
IGND
4
U15-B
SN74AHCT02DR
6
5
+VI
SCLK_I
R42
348R
R41
348R

FIG.30G-2

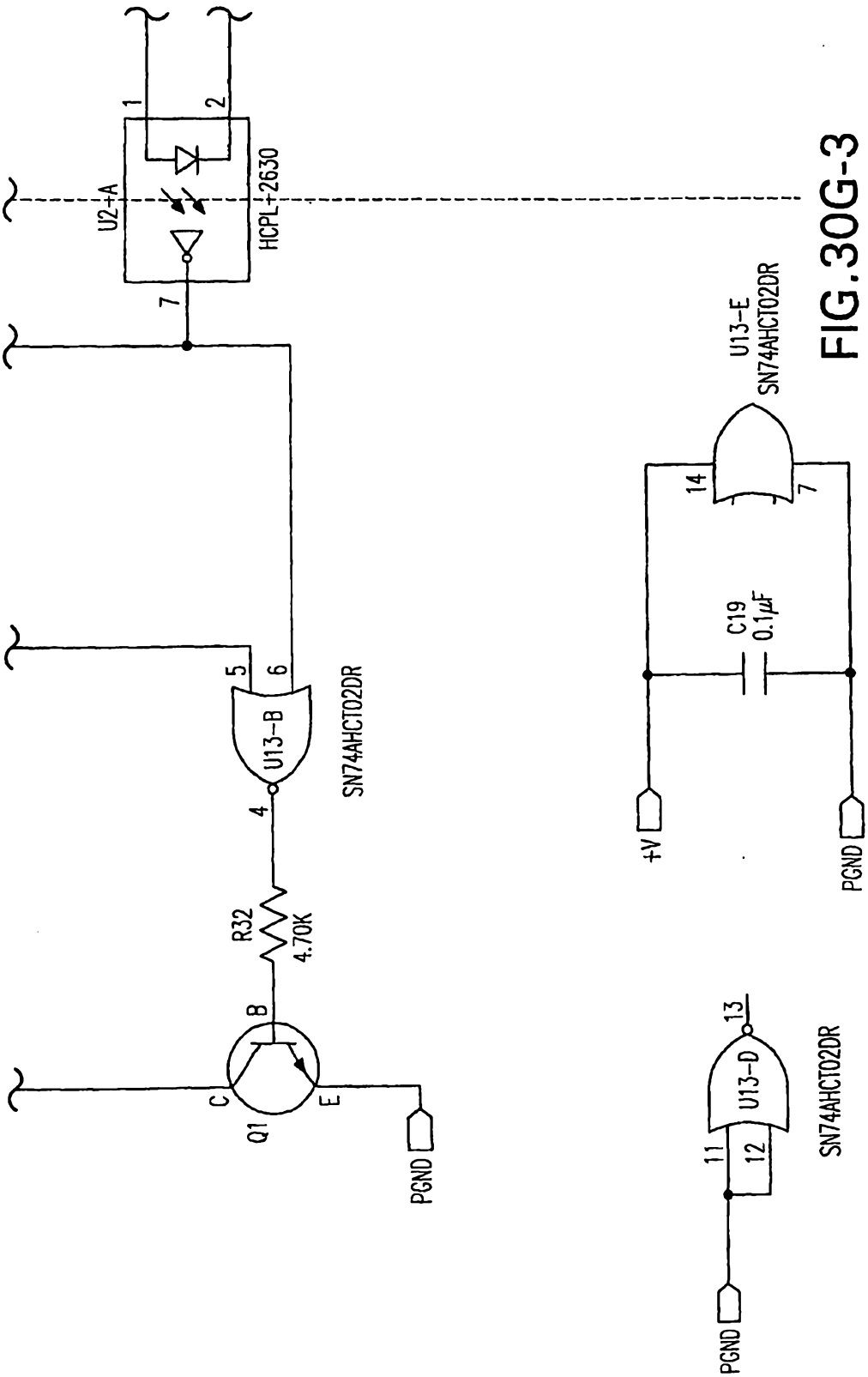

U2-A
HCPL-2630
1
2
7
U13-B
SN74AHCT02DR
5
6
4
R32
4.70K
B
C
E
Q1
PGND
U13-E
SN74AHCT02DR
14
7
C19
0.1µF
+V
PGND
U13-D
SN74AHCT02DR
13
11
12
PGND

FIG.30G-3

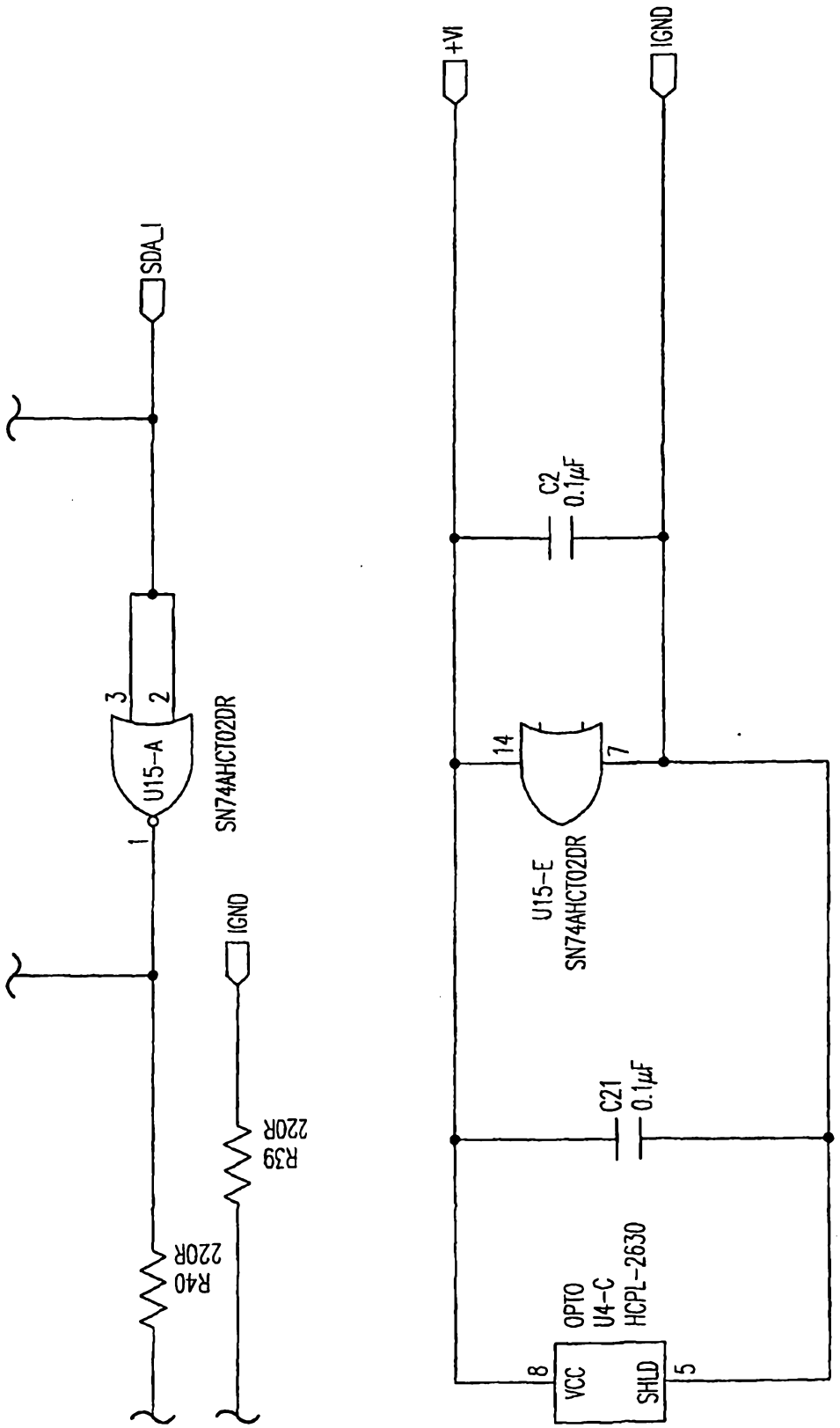

SDA_I
U15-A
SN74AHCT02DR
IGND
R40
220R
R39
220R
+VI
IGND
C2
0.1µF
U15-E
SN74AHCT02DR
C21
0.1µF
OPTO
U4-C
HCPL-2630
VCC
SHLD

FIG.30G-4

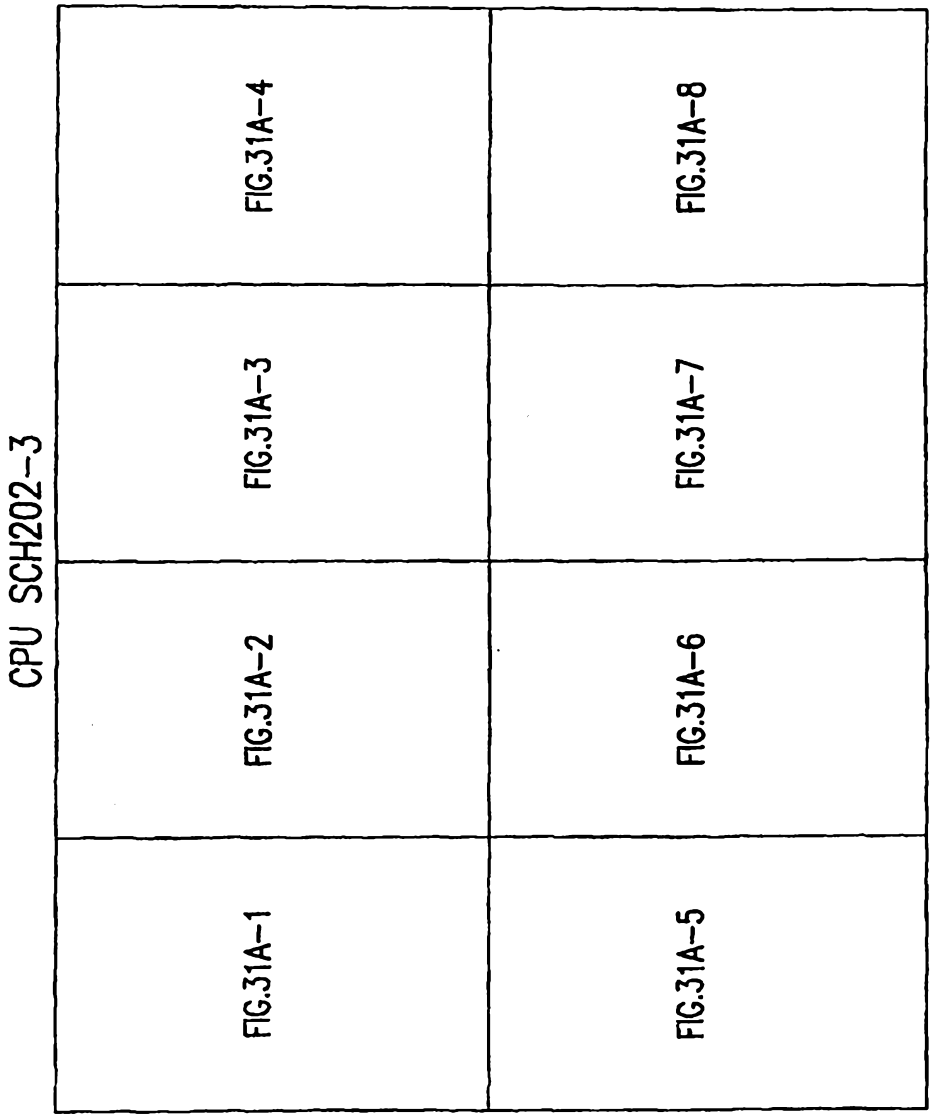

CPU SCH202-3
FIG.31A-1
FIG.31A-2
FIG.31A-3
FIG.31A-4
FIG.31A-5
FIG.31A-6
FIG.31A-7
FIG.31A-8

FIG.31A

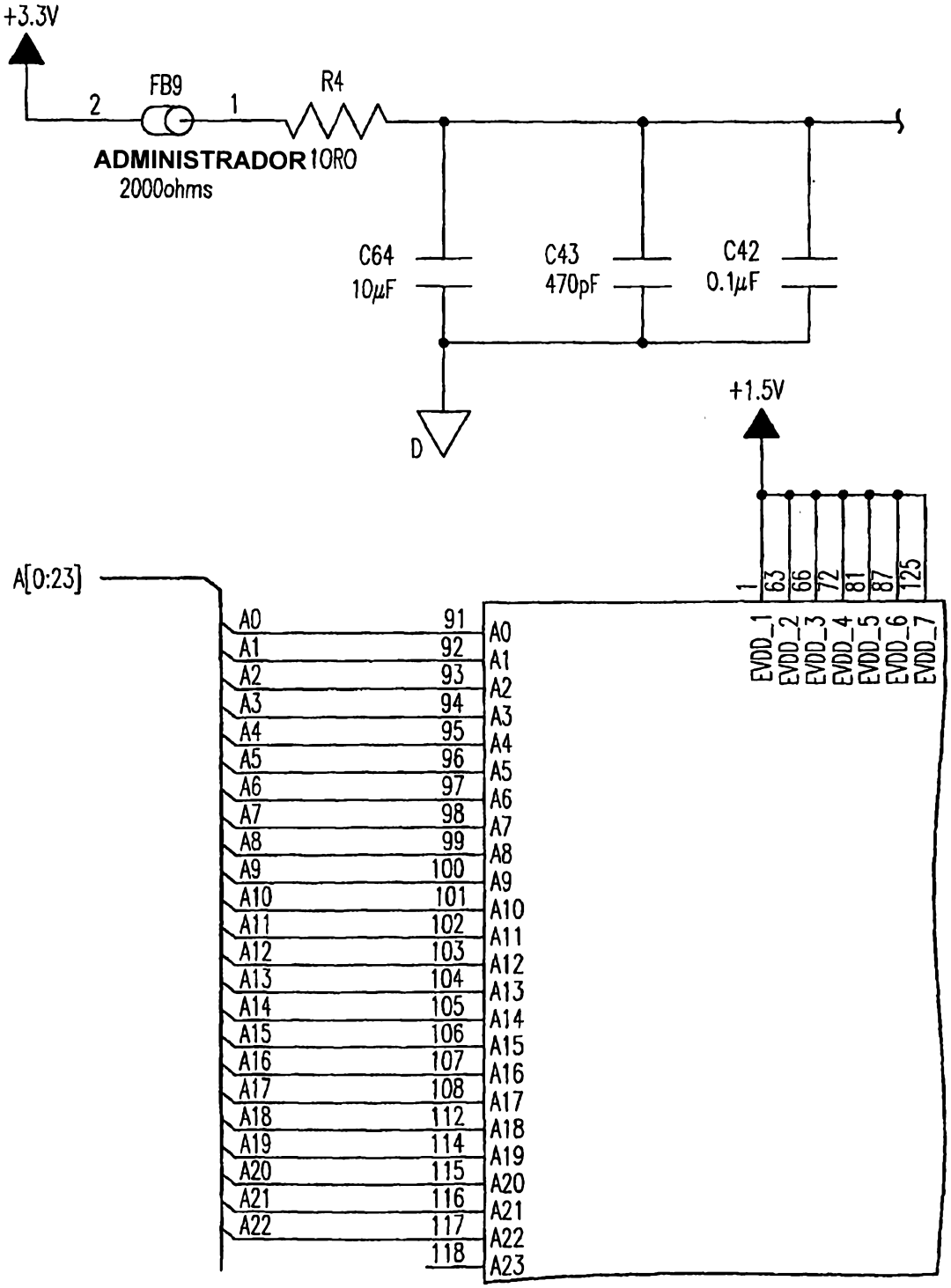

+3.3V
FB9
2
1
R4
ADMINISTRADOR 1ORO
2000ohms
C64
10µF
C43
470pF
C42
0.1µF
D
+1.5V
A[0:23]
A0 91
A1 92
A2 93
A3 94
A4 95
A5 96
A6 97
A7 98
A8 99
A9 100
A10 101
A11 102
A12 103
A13 104
A14 105
A15 106
A16 107
A17 108
A18 112
A19 114
A20 115
A21 116
A22 117
118
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
A17
A18
A19
A20
A21
A22
A23
1
63
66
72
81
87
125
EVDD_1
EVDD_2
EVDD_3
EVDD_4
EVDD_5
EVDD_6
EVDD_7

FIG.31A-1

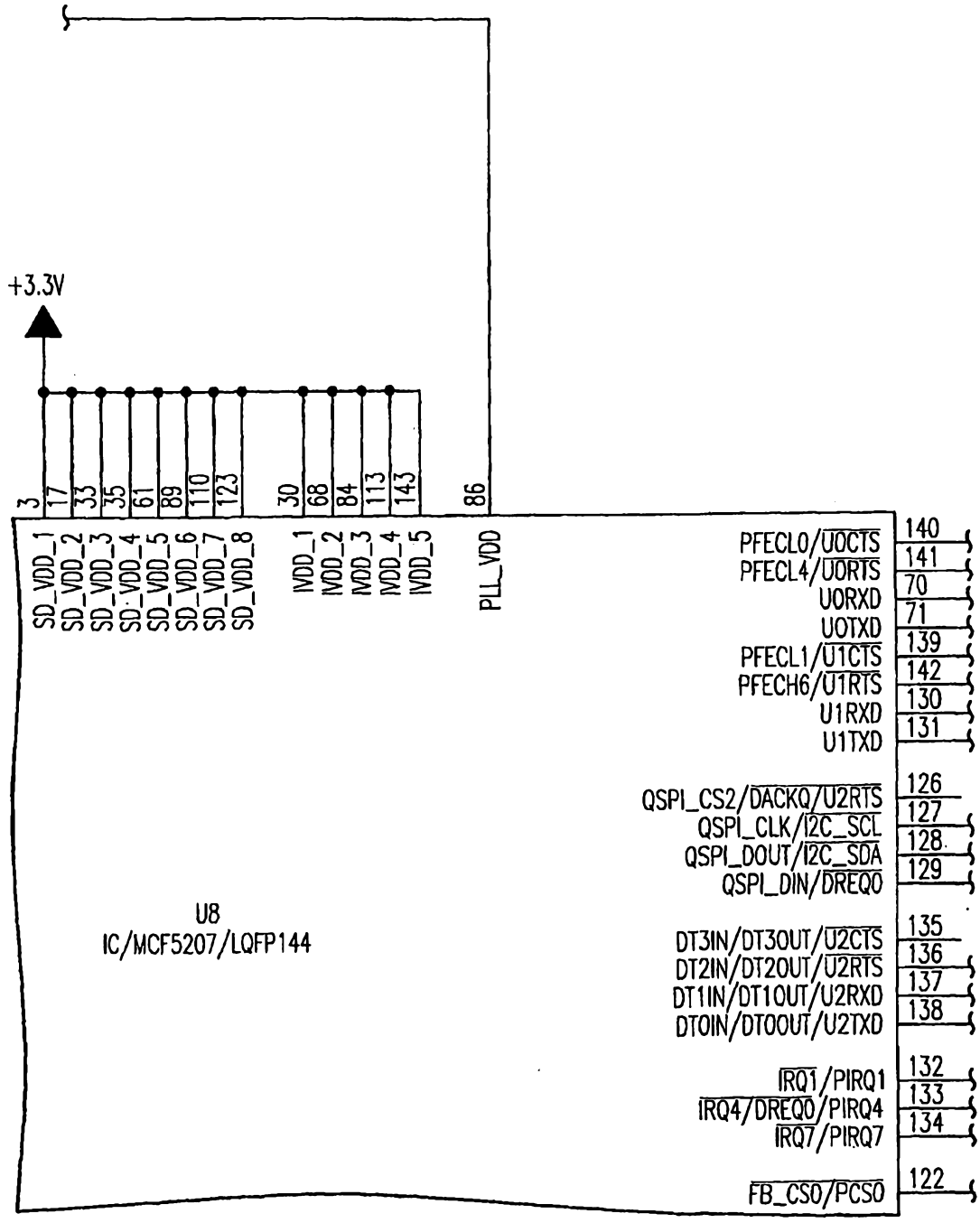

+3.3V
3
17
33
35
61
89
110
123
30
68
84
113
143
86
SD_VDD_1
SD_VDD_2
SD_VDD_3
SD_VDD_4
SD_VDD_5
SD_VDD_6
SD_VDD_7
SD_VDD_8
IVDD_1
IVDD_2
IVDD_3
IVDD_4
IVDD_5
PLL_VDD
PFECL0/U0CTS 140
PFECL4/U0RTS 141
U0RXD 70
U0TXD 71
PFECL1/U1CTS 139
PFECH6/U1RTS 142
U1RXD 130
U1TXD 131
QSPI_CS2/DACK0/U2RTS 126
QSPI_CLK/I2C_SCL 127
QSPI_DOUT/I2C_SDA 128
QSPI_DIN/DREQ0 129
U8
IC/MCF5207/LQFP144
DT3IN/DT3OUT/U2CTS 135
DT2IN/DT2OUT/U2RTS 136
DT1IN/DT1OUT/U2RXD 137
DT0IN/DT0OUT/U2TXD 138
IRQ1/PIRQ1 132
IRQ4/DREQ0/PIRQ4 133
IRQ7/PIRQ7 134
FB_CS0/PCS0 122

FIG.31A-2

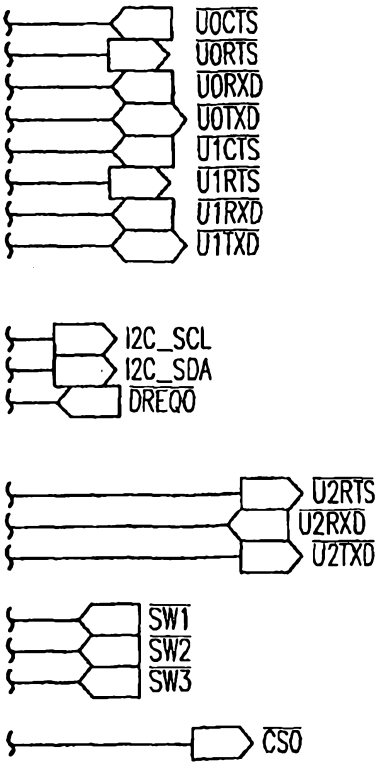

U0CTS
U0RTS
U0RXD
U0TXD
U1CTS
U1RTS
U1RXD
U1TXD
I2C_SCL
I2C_SDA
DREQ0
U2RTS
U2RXD
U2TXD
SW1
SW2
SW3
CS0

FIG.31A-3

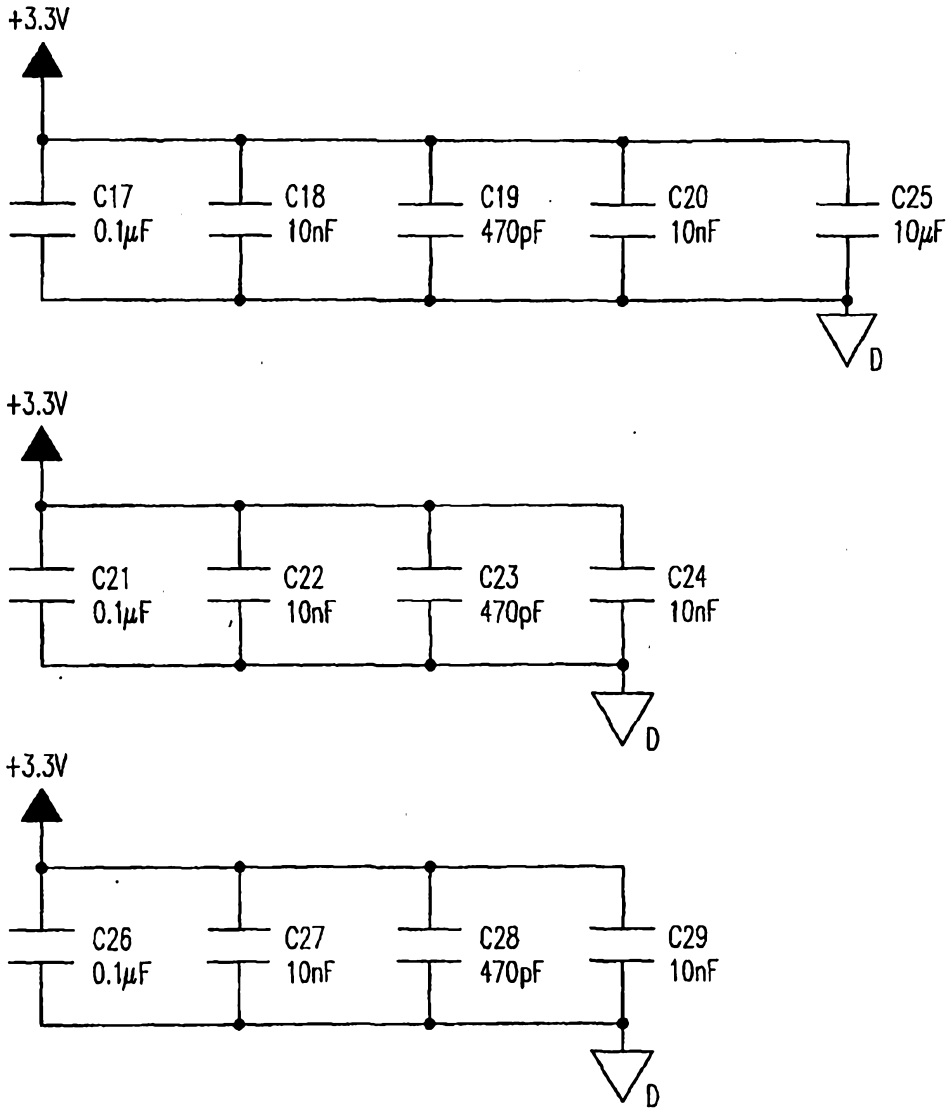

+3.3V
C17
0.1μF
C18
10nF
C19
470pF
C20
10nF
C25
10μF
D
+3.3V
C21
0.1μF
C22
10nF
C23
470pF
C24
10nF
D
+3.3V
C26
0.1μF
C27
10nF
C28
470pF
C29
10nF
D

FIG.31A-4

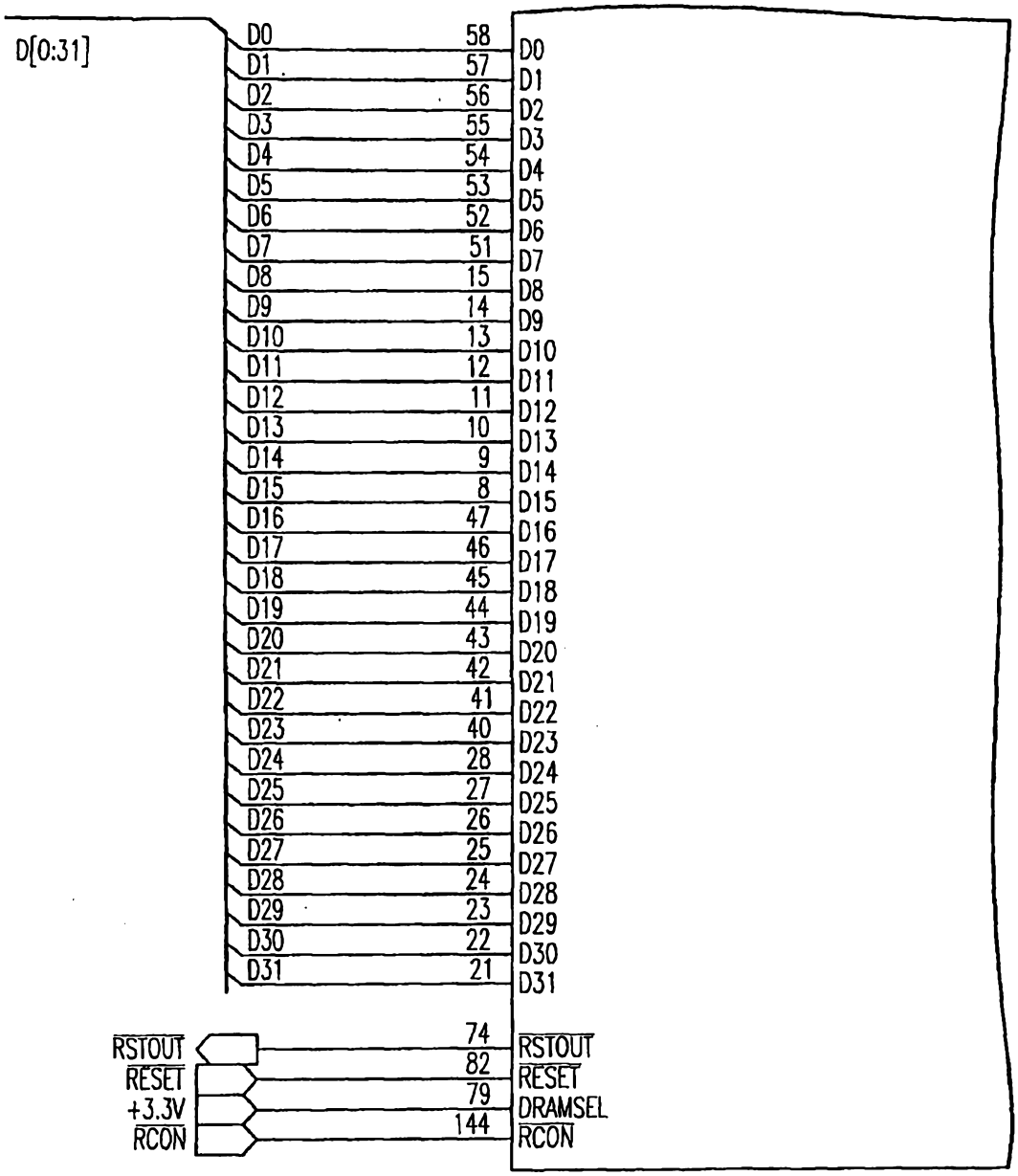

D[0:31]
D0 58 D0
D1 57 D1
D2 56 D2
D3 55 D3
D4 54 D4
D5 53 D5
D6 52 D6
D7 51 D7
D8 15 D8
D9 14 D9
D10 13 D10
D11 12 D11
D12 11 D12
D13 10 D13
D14 9 D14
D15 8 D15
D16 47 D16
D17 46 D17
D18 45 D18
D19 44 D19
D20 43 D20
D21 42 D21
D22 41 D22
D23 40 D23
D24 28 D24
D25 27 D25
D26 26 D26
D27 25 D27
D28 24 D28
D29 23 D29
D30 22 D30
D31 21 D31
RSTOUT 74 RSTOUT
RESET 82 RESET
+3.3V 79 DRAMSEL
RCON 144 RCON

FIG.31A-5

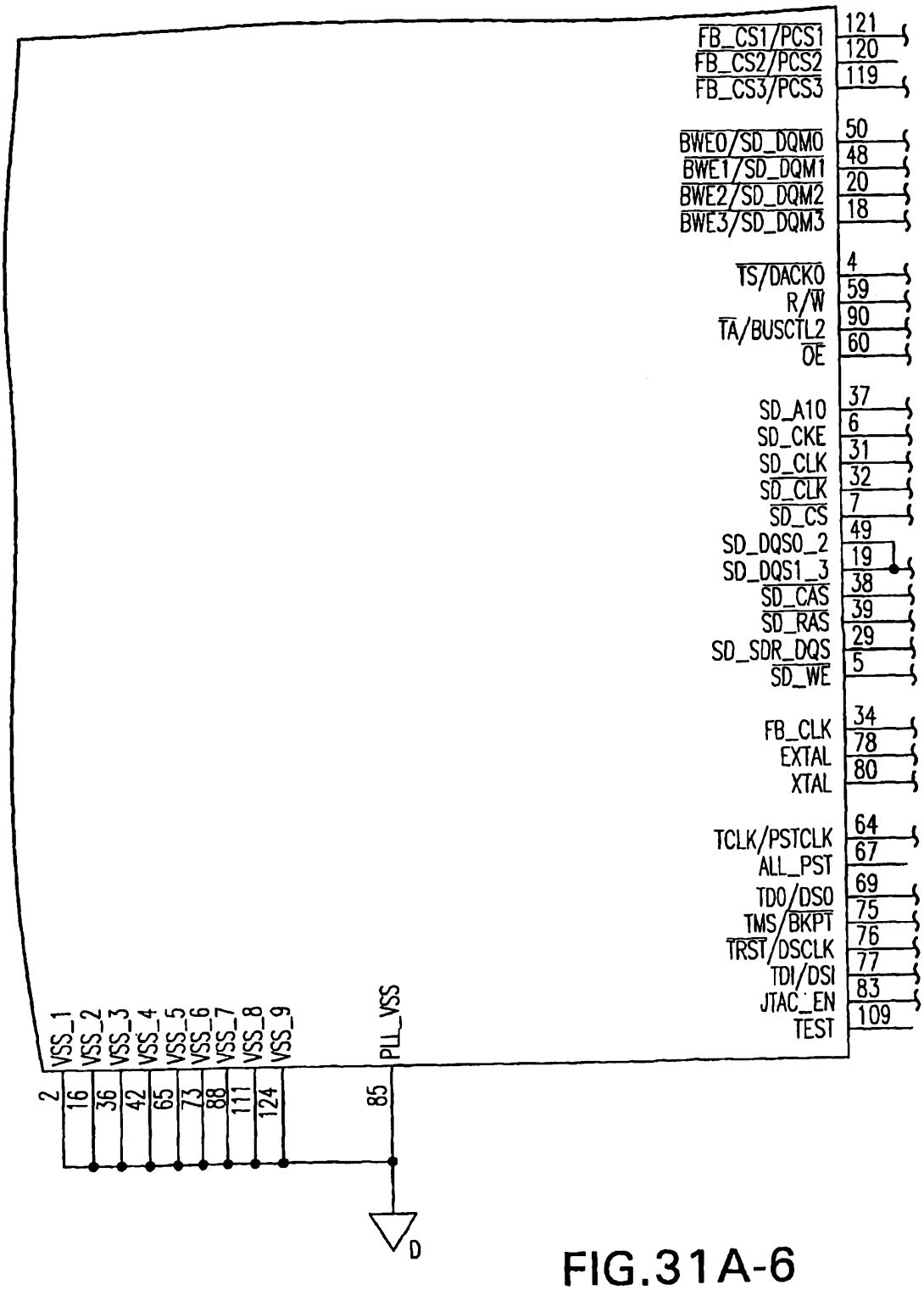

FB_CS1/PCS1 121
FB_CS2/PCS2 120
FB_CS3/PCS3 119
BWE0/SD_DQM0 50
BWE1/SD_DQM1 48
BWE2/SD_DQM2 20
BWE3/SD_DQM3 18
TS/DACK0 4
R/W 59
TA/BUSCTL2 90
OE 60
SD_A10 37
SD_CKE 6
SD_CLK 31
SD_CLK 32
SD_CS 7
SD_DQS0_2 49
SD_DQS1_3 19
SD_CAS 38
SD_RAS 39
SD_SDR_DQS 29
SD_WE 5
FB_CLK 34
EXTAL 78
XTAL 80
TCLK/PSTCLK 64
ALL_PST 67
TDO/DSO 69
TMS/BKPT 75
TRST/DSCLK 76
TDI/DSI 77
JTAC_EN 83
TEST 109
VSS_1 2
VSS_2 16
VSS_3 36
VSS_4 42
VSS_5 65
VSS_6 73
VSS_7 88
VSS_8 111
VSS_9 124
PLL_VSS 85
D

FIG.31A-6

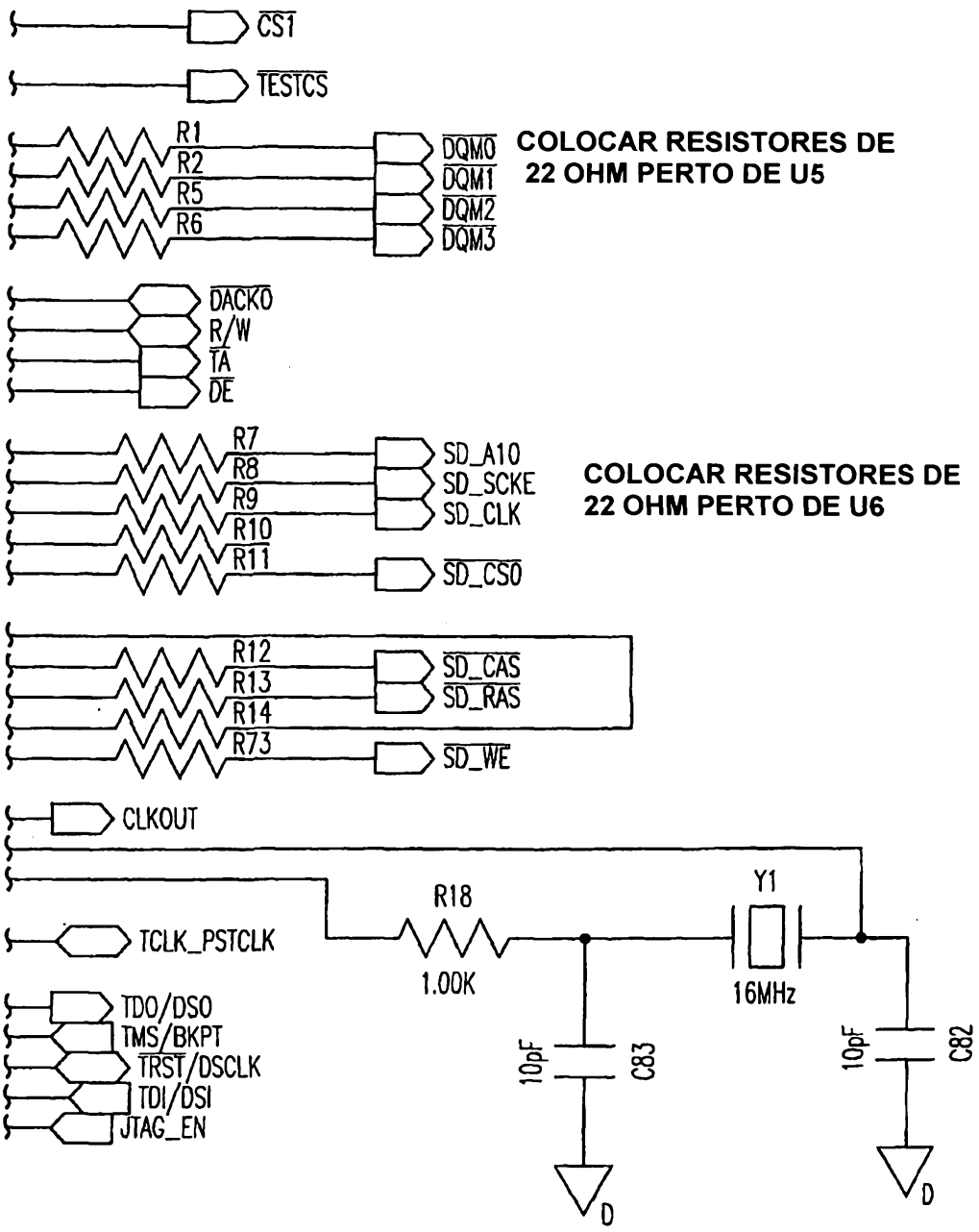

CS1
TESTCS
R1
R2
R5
R6
DQM0
DQM1
DQM2
DQM3
COLOCAR RESISTORES DE 22 OHM PERTO DE U5
DACK0
R/W
TA
DE
R7
R8
R9
R10
R11
SD_A10
SD_SCKE
SD_CLK
SD_CS0
COLOCAR RESISTORES DE 22 OHM PERTO DE U6
R12
R13
R14
R73
SD_CAS
SD_RAS
SD_WE
CLKOUT
TCLK_PSTCLK
TDO/DSO
TMS/BKPT
TRST/DSCLK
TDI/DSI
JTAG_EN
R18
1.00K
Y1
16MHz
10pF
C83
10pF
C82
D
D

FIG.31A-7

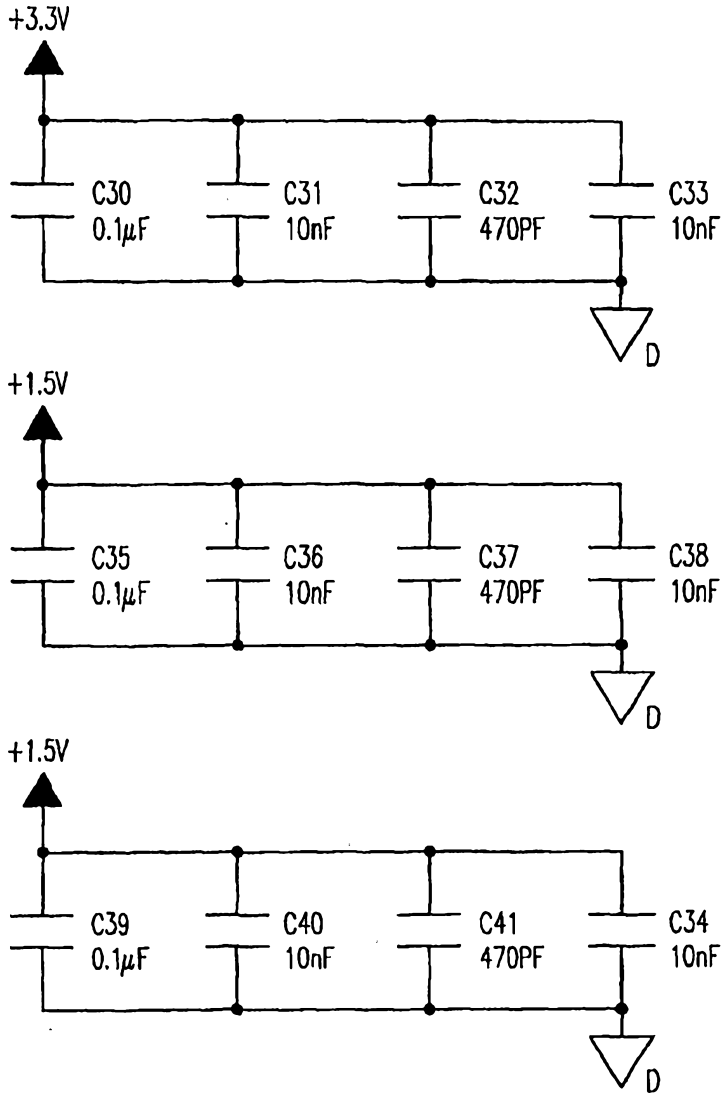

+3.3V
C30
0.1μF
C31
10nF
C32
470PF
C33
10nF
D
+1.5V
C35
0.1μF
C36
10nF
C37
470PF
C38
10nF
D
+1.5V
C39
0.1μF
C40
10nF
C41
470PF
C34
10nF
D

FIG.31A-8

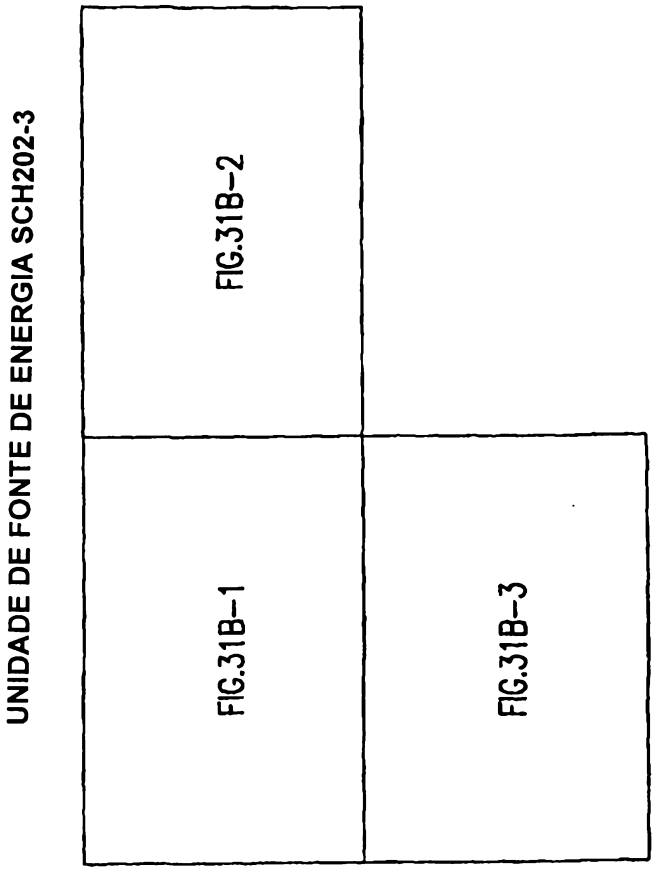

UNIDADE DE FONTE DE ENERGIA SCH202-3
FIG.31B-2
FIG.31B-1
FIG.31B-3

FIG.31B

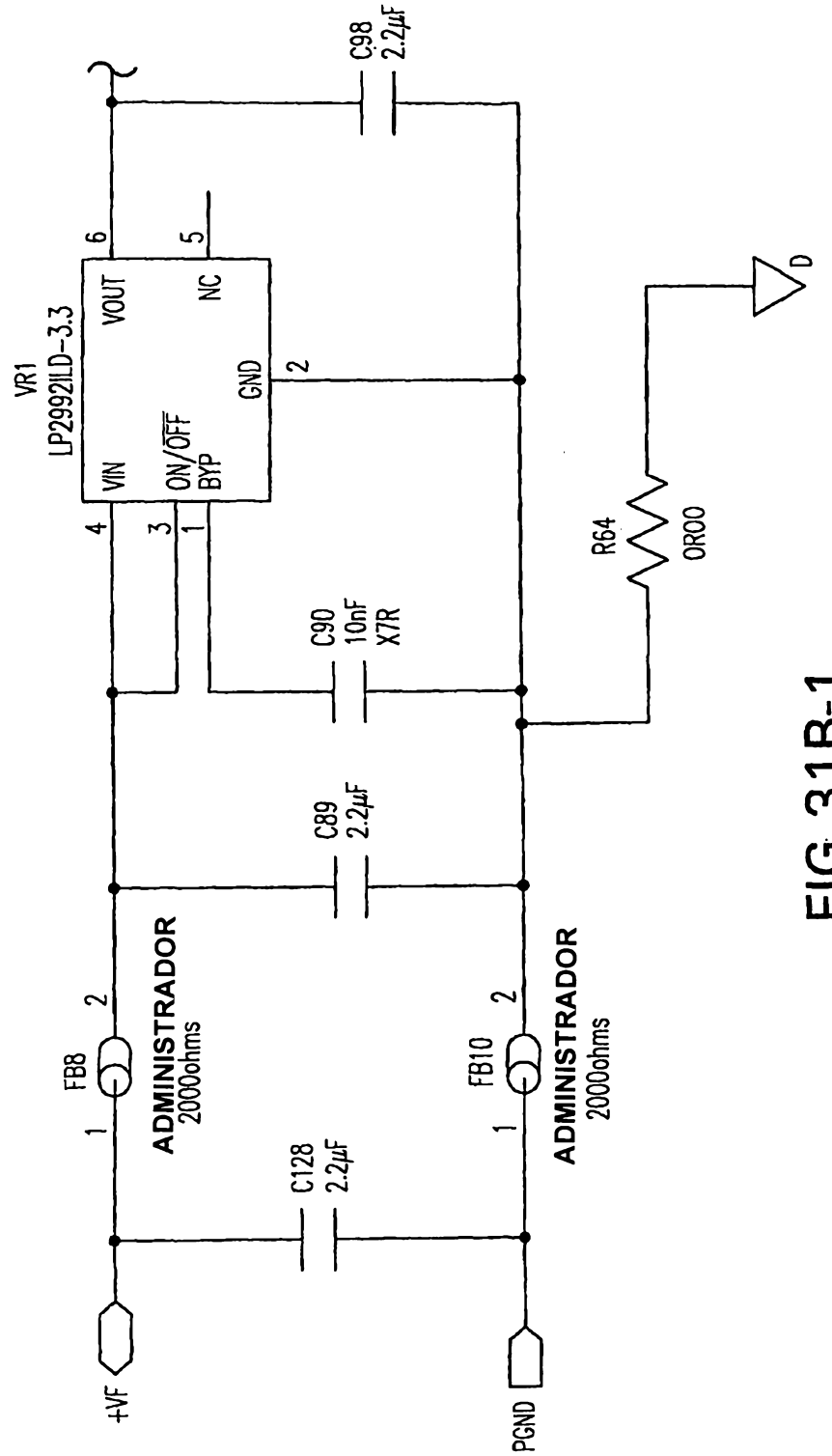

VR1
LP2992ILD-3.3
VIN
ON/OFF
BYP
GND
NC
VOUT
C98
2.2µF
C90
10nF
X7R
C89
2.2µF
C128
2.2µF
R64
0R00
D
FB8
FB10
ADMINISTRADOR
2000ohms
ADMINISTRADOR
2000ohms
+VF
PGND

FIG.31B-1

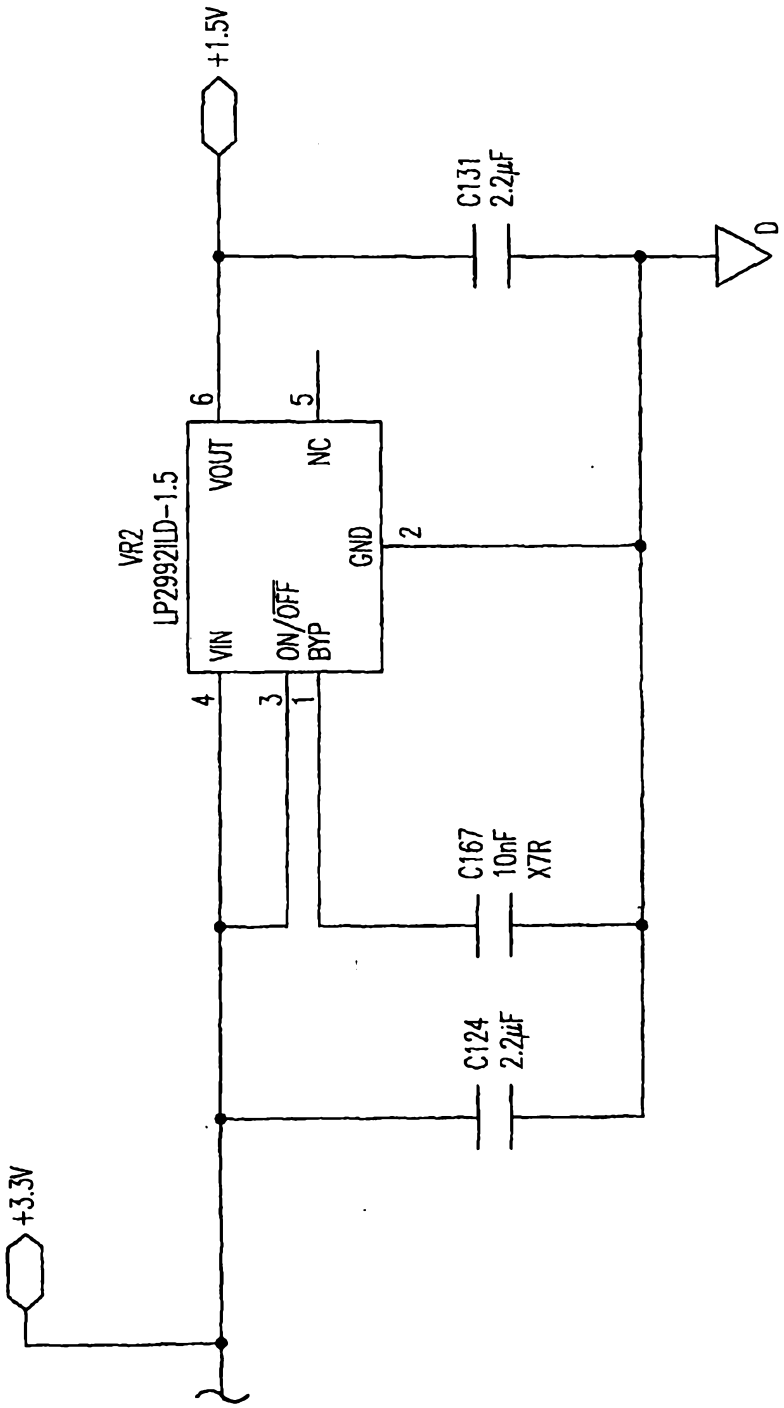

+1.5V
C131
2.2µF
D
VR2
LP2992ILD-1.5
VOUT
NC
GND
VIN
ON/OFF
BYP
6
5
2
4
3
1
C167
10nF
X7R
C124
2.2µF
+3.3V

FIG.31B-2

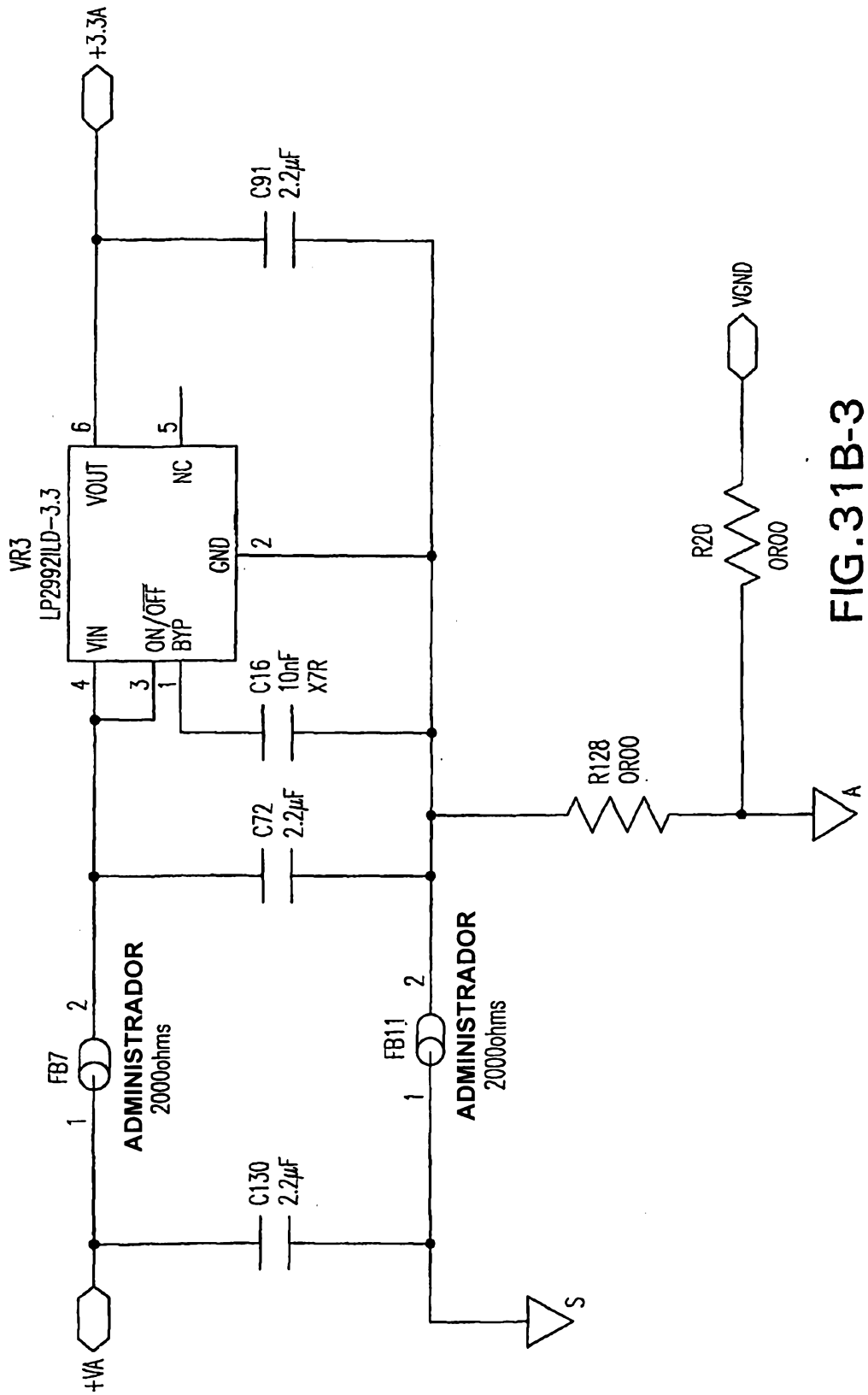

+3.3A
C91
2.2µF
VR3
LP2992ILD-3.3
VOUT
NC
GND
VIN
ON/OFF
BYP
6
5
2
4
3
1
C16
10nF
X7R
C72
2.2µF
R128
0R00
VGND
R20
0R00
A
FB7
2
1
ADMINISTRADOR
2000ohms
FB11
2
1
ADMINISTRADOR
2000ohms
C130
2.2µF
+VA
S

FIG.31B-3

RTC/FRAUDE TEMP/REAJUSTE PCB202-3

FIG.31C-1	FIG.31C-2	FIG.31C-3
FIG.31C-4	FIG.31C-5	FIG.31C-6

FIG.31C

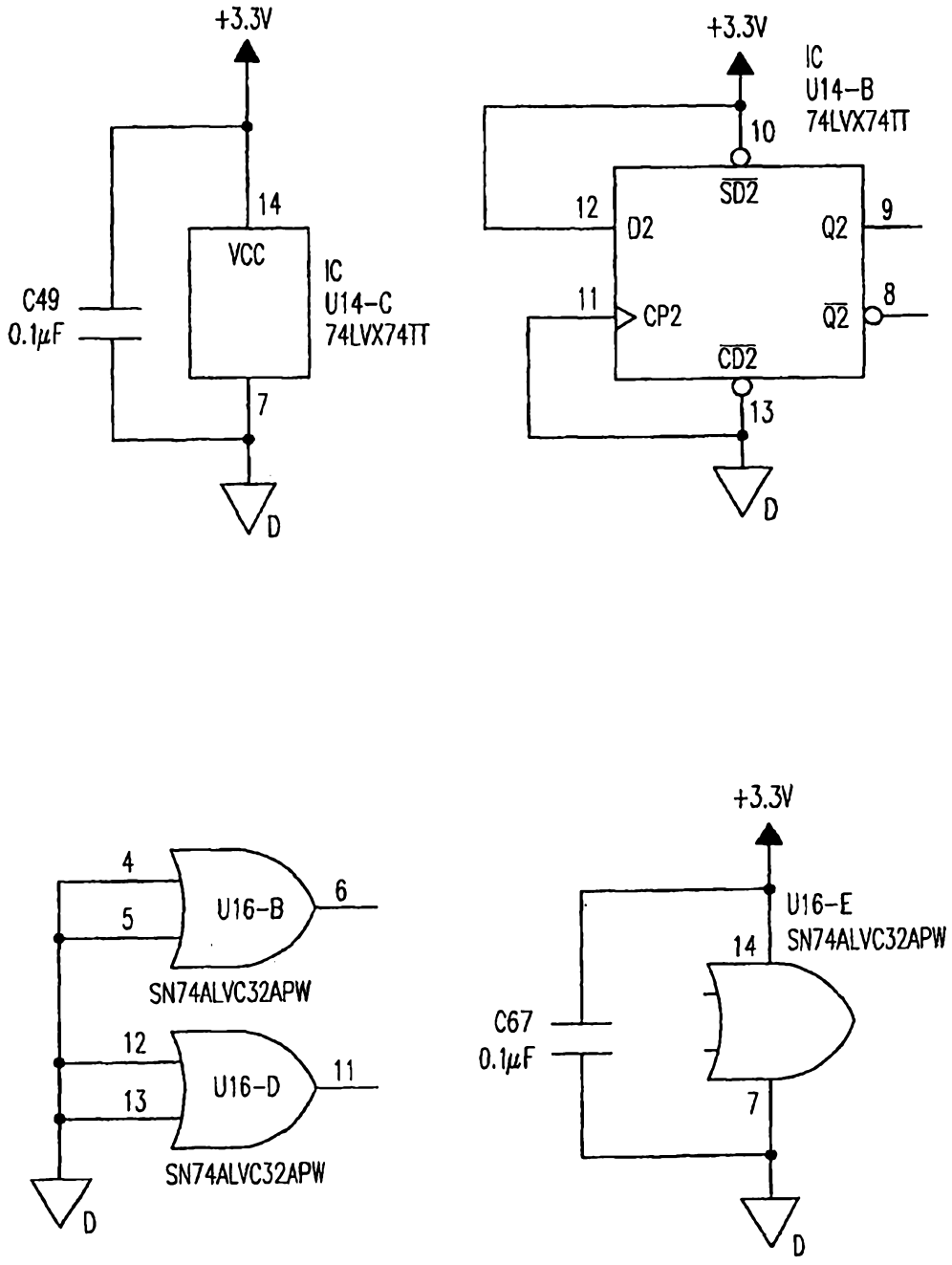

+3.3V
14
VCC
IC
U14-C
74LVX74TT
C49
0.1μF
7
D
+3.3V
IC
U14-B
74LVX74TT
10
SD2
12
D2
Q2
9
11
CP2
Q2
8
CD2
13
D
4
5
U16-B
6
SN74ALVC32APW
12
13
U16-D
11
SN74ALVC32APW
D
+3.3V
U16-E
SN74ALVC32APW
14
C67
0.1μF
7
D

FIG.31C-1

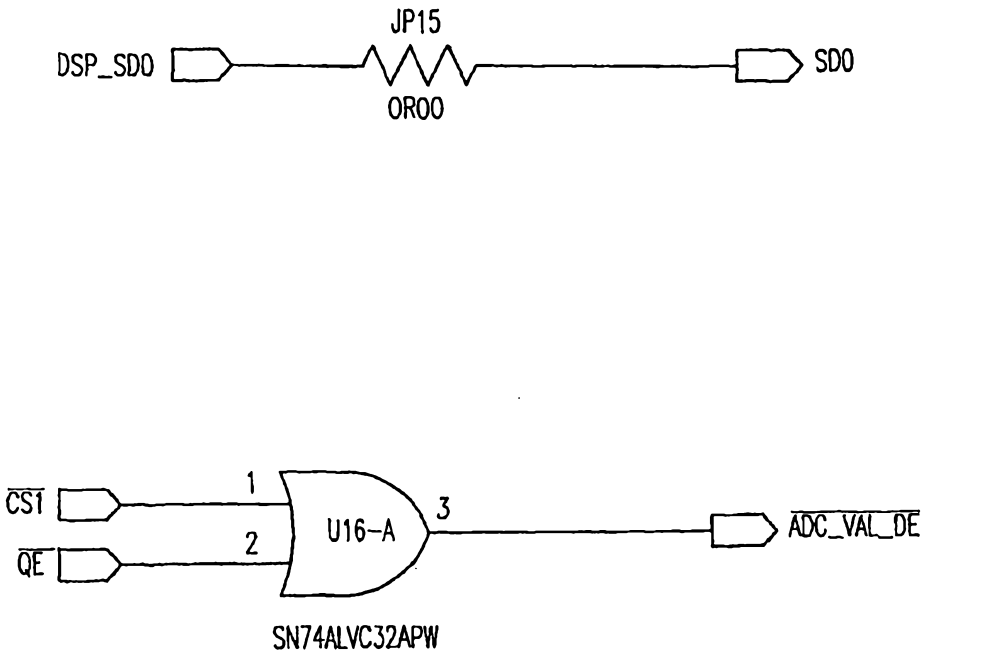

JP15
DSP_SDO
SDO
0R00
CS1
QE
1
2
U16-A
3
ADC_VAL_DE
SN74ALVC32APW
DACKO
IC
U14-A
74LVX74TT
4
SD1
2
D1
Q1
5
DREQ0
DSP_SPI_CS1
3
CP1
Q1
6
CD1
1
D
+3.3V

FIG.31C-2

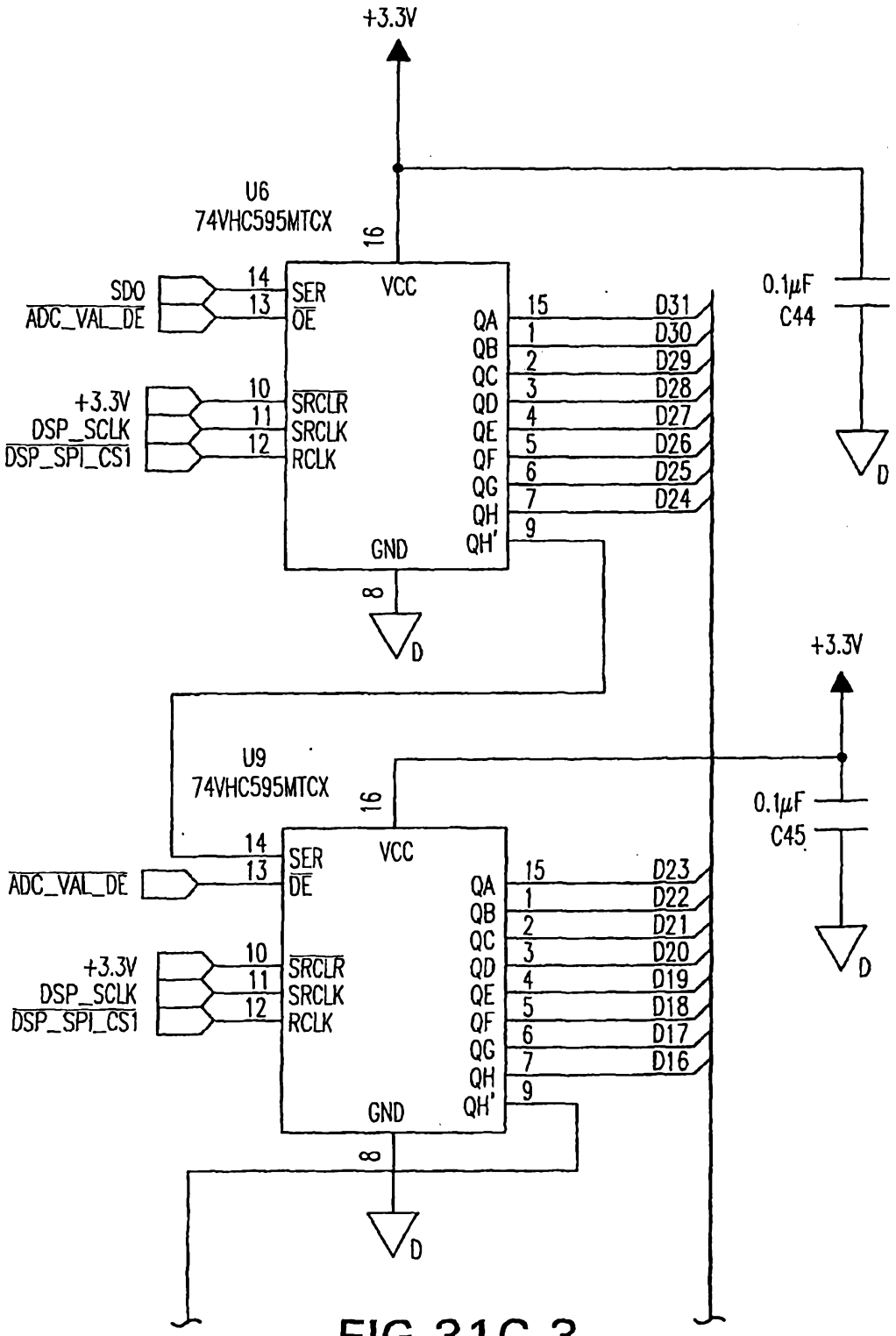

+3.3V
U6
74VHC595MTCX
16
VCC
SDO
ADC_VAL_DE
14
13
SER
OE
+3.3V
DSP_SCLK
DSP_SPI_CS1
10
11
12
SRCLR
SRCLK
RCLK
QA
QB
QC
QD
QE
QF
QG
QH
QH'
15
1
2
3
4
5
6
7
9
D31
D30
D29
D28
D27
D26
D25
D24
GND
8
D
0.1µF
C44
+3.3V
U9
74VHC595MTCX
16
VCC
14
13
SER
DE
ADC_VAL_DE
+3.3V
DSP_SCLK
DSP_SPI_CS1
10
11
12
SRCLR
SRCLK
RCLK
QA
QB
QC
QD
QE
QF
QG
QH
QH'
15
1
2
3
4
5
6
7
9
D23
D22
D21
D20
D19
D18
D17
D16
GND
8
D
0.1µF
C45

FIG.31C-3

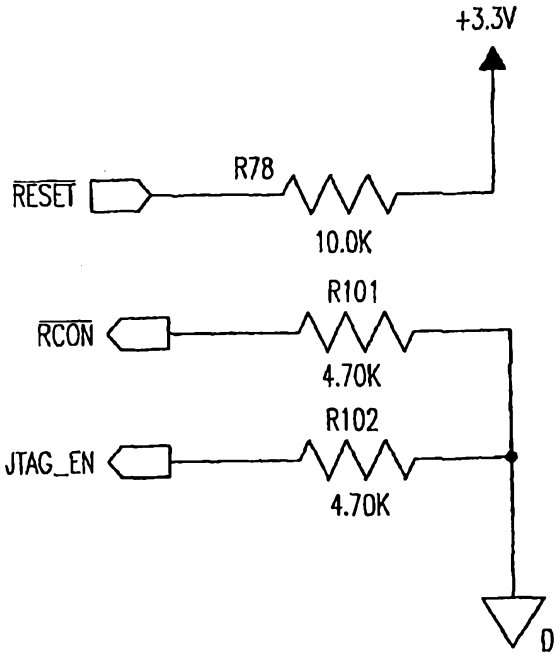


FIG.31C-4

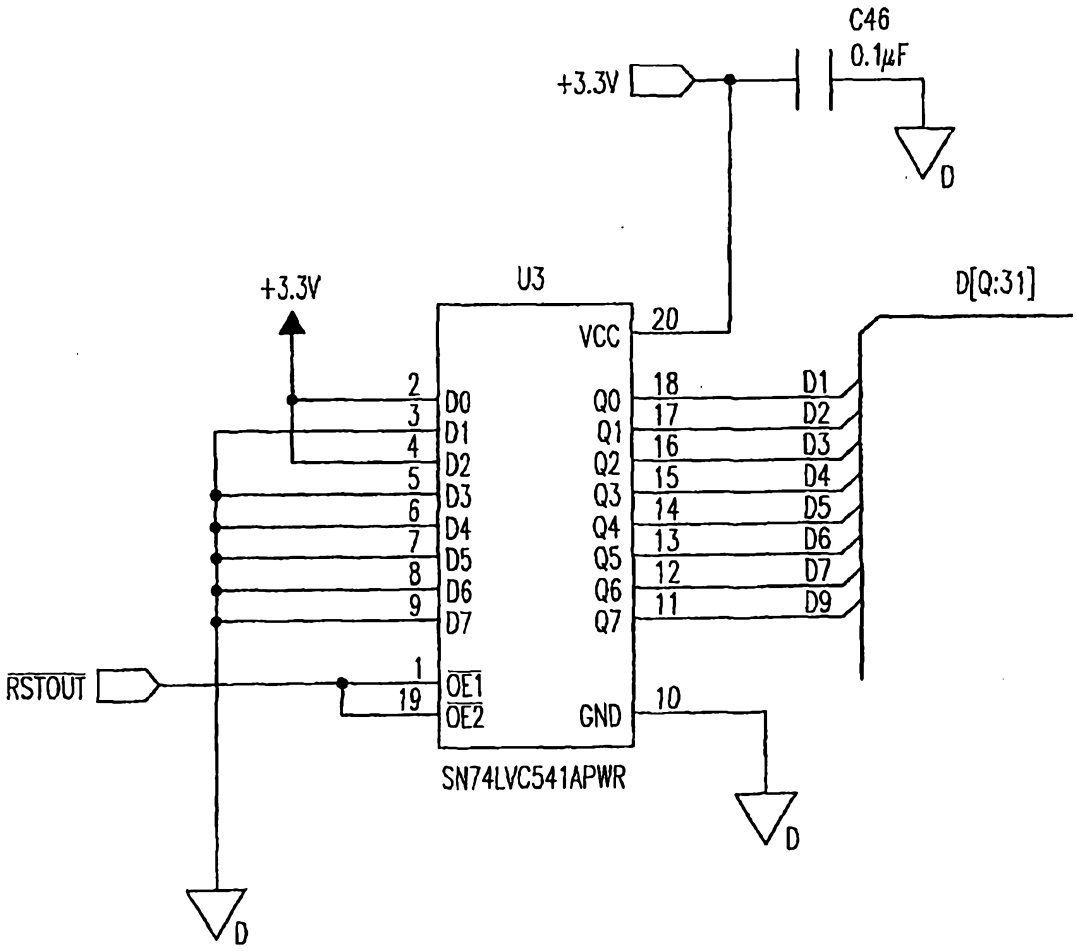

C46
0.1μF
+3.3V
D
+3.3V
U3
D[Q:31]
VCC
20
2
3
4
5
6
7
8
9
D0
D1
D2
D3
D4
D5
D6
D7
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
18
17
16
15
14
13
12
11
D1
D2
D3
D4
D5
D6
D7
D9
RSTOUT
1
19
OE1
OE2
GND
10
SN74LVC541APWR
D
D

FIG.31C-5

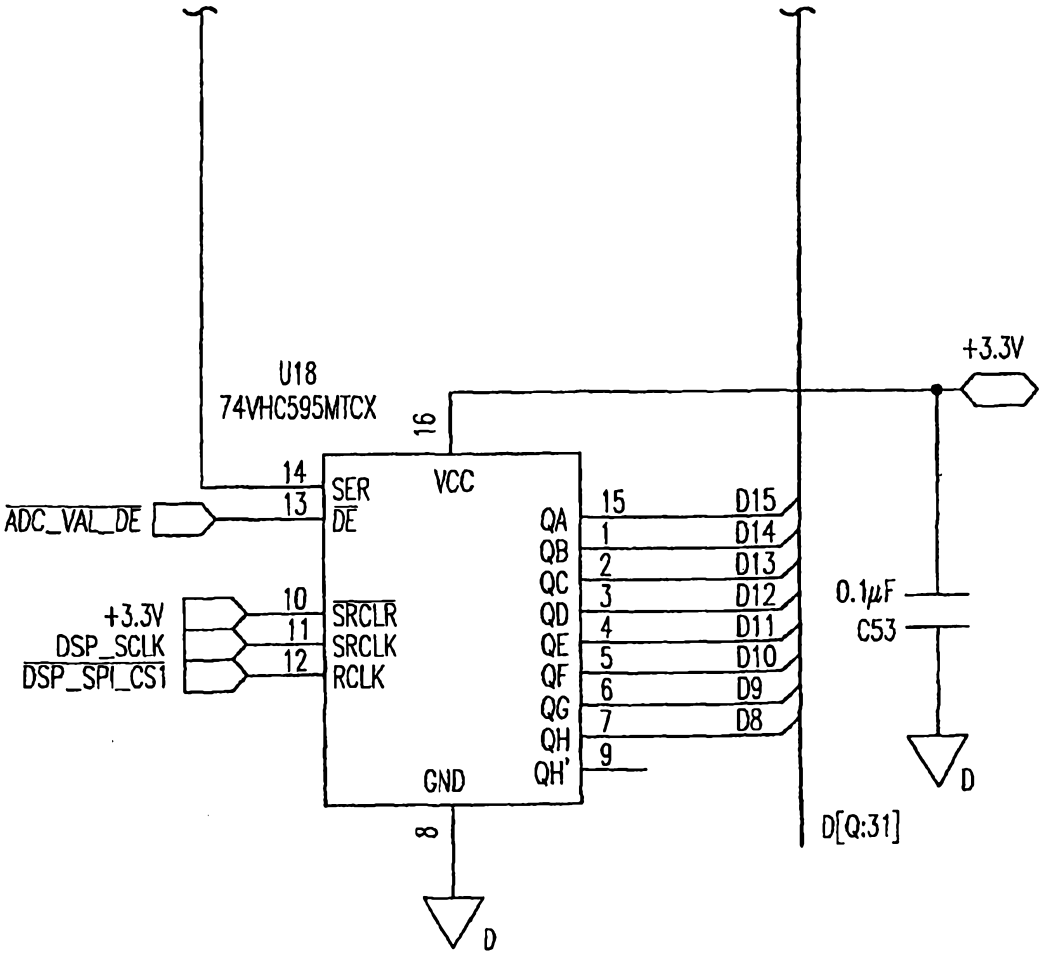

U18
74VHC595MTCX
16
VCC
14
SER
13
DE
ADC_VAL_DE
10
SRCLR
11
SRCLK
12
RCLK
+3.3V
DSP_SCLK
DSP_SPI_CS1
QA
QB
QC
QD
QE
QF
QG
QH
QH'
15
1
2
3
4
5
6
7
9
D15
D14
D13
D12
D11
D10
D9
D8
GND
8
D
+3.3V
0.1µF
C53
D
D[Q:31]

FIG.31C-6

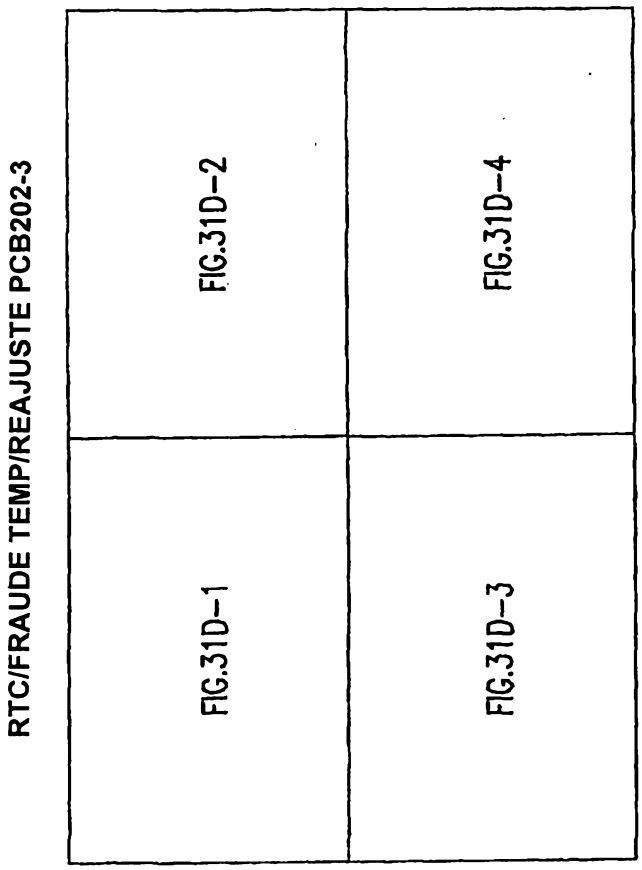

RTC/FRAUDE TEMP/REAJUSTE PCB202-3
FIG.31D-1
FIG.31D-2
FIG.31D-3
FIG.31D-4

FIG.31D

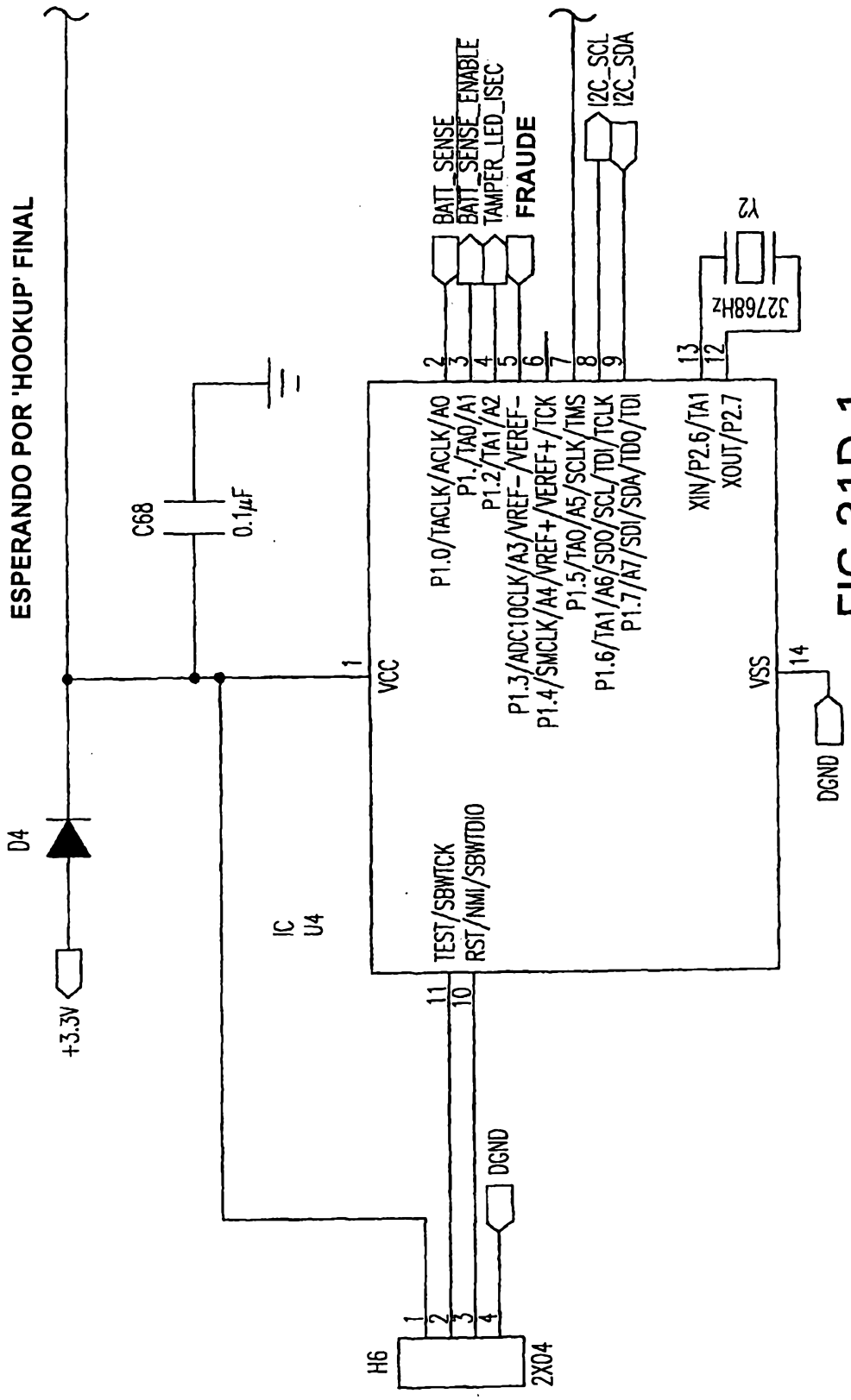

ESPERANDO POR 'HOOKUP' FINAL
+3.3V
D4
C68
0.1µF
IC
U4
VCC
1
TEST/SBWTCK
RST/NMI/SBWTDIO
11
10
P1.0/TACLK/ACLK/A0
P1./TA0/A1
P1.2/TA1/A2
P1.3/ADC10CLK/A3/VREF-/VEREF-
P1.4/SMCLK/A4/VREF+/VEREF+/TCK
P1.5/TA0/A5/SCLK/TMS
P1.6/TA1/A6/SDO/SCL/TDI/TCLK
P1.7/A7/SDI/SDA/TDO/TDI
2
3
4
5
6
7
8
9
BATT_SENSE
BATT_SENSE_ENABLE
TAMPER_LED_ISEC
FRAUDE
I2C_SCL
I2C_SDA
XIN/P2.6/TA1
XOUT/P2.7
13
12
Y2
32768Hz
VSS
14
DGND
DGND
H6
2X04
1
2
3
4

FIG.31D-1

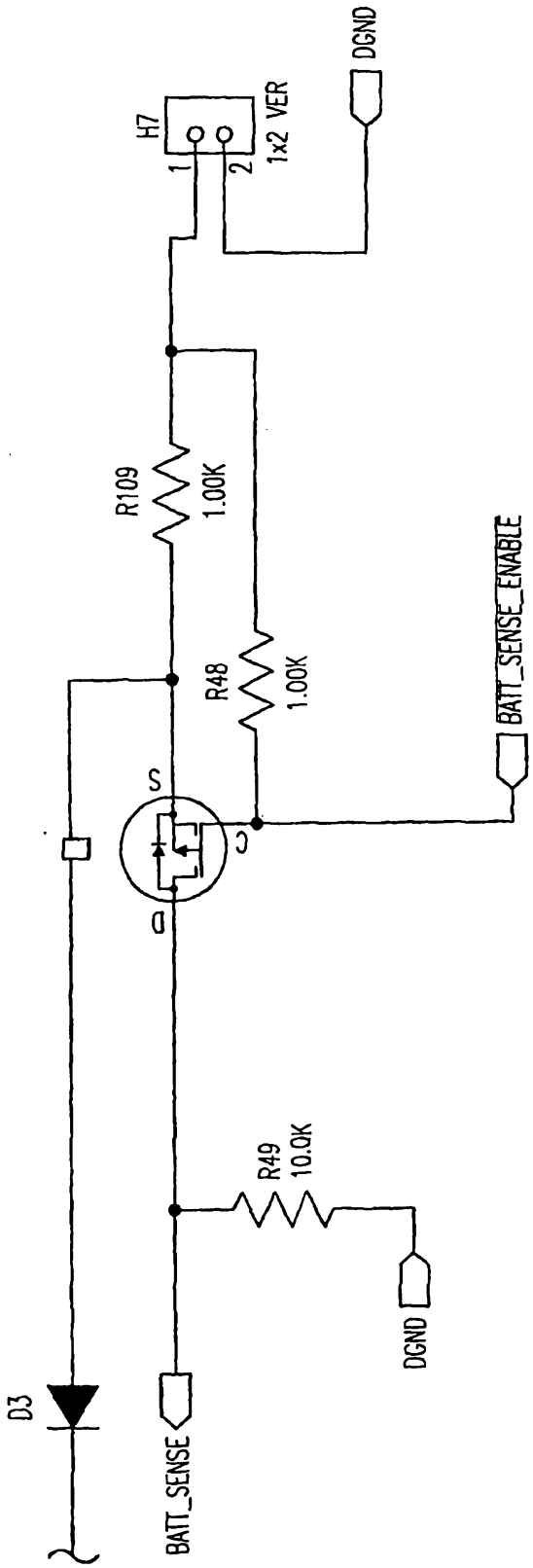

H7
1
2
1x2 VER
DGND
R109
1.00K
R48
1.00K
BATT_SENSE_ENABLE
S
G
D
R49
10.0K
DGND
D3
BATT_SENSE
RESET

FIG.31D-2

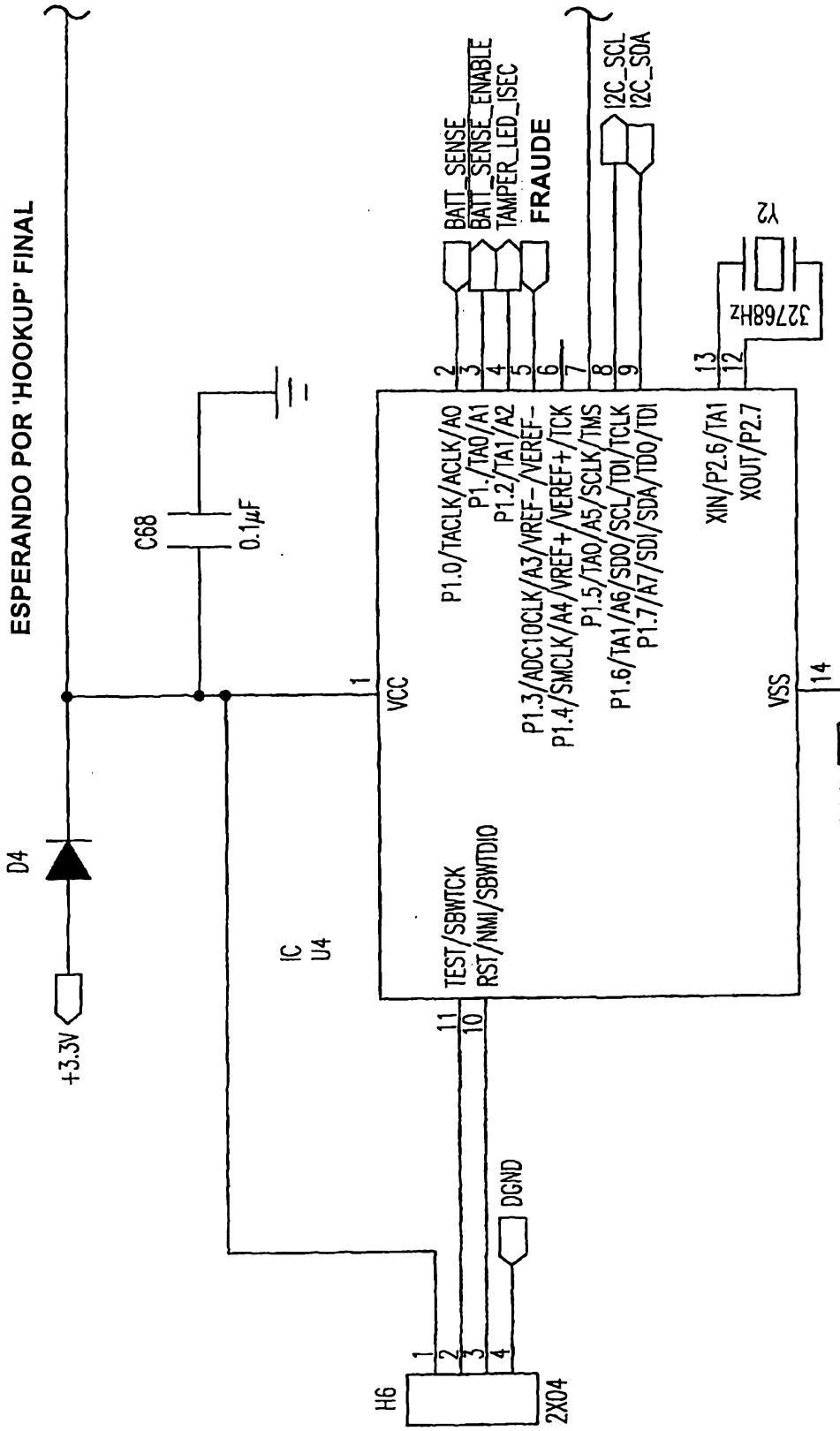

ESPERANDO POR 'HOOKUP' FINAL
D4
+3.3V
C68
0.1µF
IC
U4
VCC
1
TEST/SBWTCK
RST/NMI/SBWTDIO
11
10
P1.0/TACLK/ACLK/A0
P1./TA0/A1
P1.2/TA1/A2
P1.3/ADC10CLK/A3/VREF-/VEREF-
P1.4/SMCLK/A4/VREF+/VEREF+/TCK
P1.5/TA0/A5/SCLK/TMS
P1.6/TA1/A6/SDO/SCL/TDI/TCLK
P1.7/A7/SDI/SDA/TDO/TDI
2
3
4
5
6
7
8
9
BATT_SENSE
BATT_SENSE_ENABLE
TAMPER_LED_1SEC
FRAUDE
I2C_SCL
I2C_SDA
XIN/P2.6/TA1
XOUT/P2.7
13
12
Y2
32768Hz
VSS
14
DGND
DGND
H6
2X04
1
2
3
4

FIG.31D-1

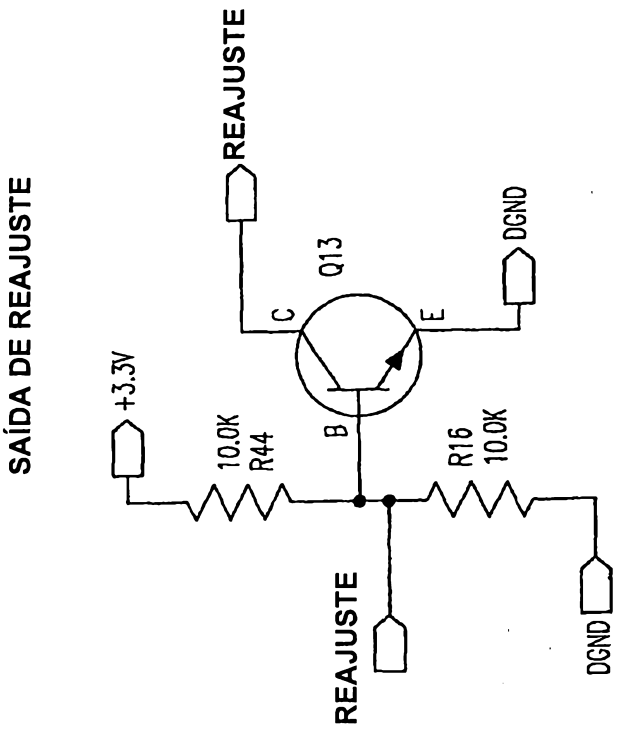

SAÍDA DE REAJUSTE
+3.3V
10.0K
R44
REAJUSTE
C
Q13
B
E
DGND
R16
10.0K
REAJUSTE
DGND

FIG.31D-4

MEMÓRIA FLASH SCH202-3

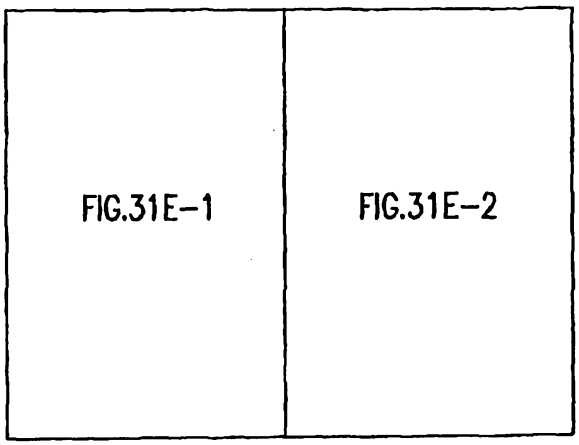


FIG.31E

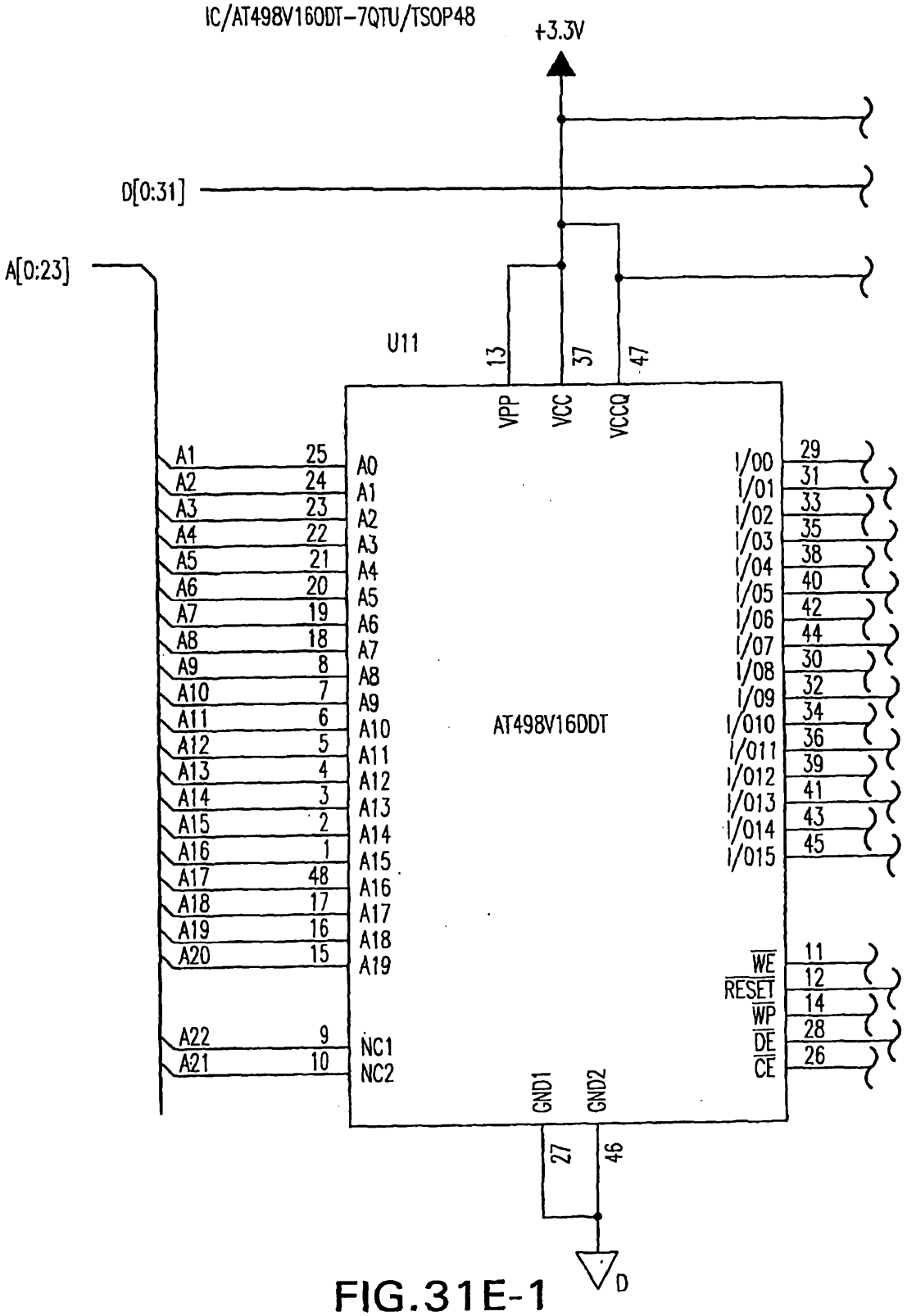

IC/AT498V160DT-7QTU/TSOP48
+3.3V
D[0:31]
A[0:23]
U11
13
37
47
VPP
VCC
VCCQ
A1 25
A2 24
A3 23
A4 22
A5 21
A6 20
A7 19
A8 18
A9 8
A10 7
A11 6
A12 5
A13 4
A14 3
A15 2
A16 1
A17 48
A18 17
A19 16
A20 15
A22 9
A21 10
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
A17
A18
A19
NC1
NC2
AT498V16DDT
I/O0 29
I/O1 31
I/O2 33
I/O3 35
I/O4 38
I/O5 40
I/O6 42
I/O7 44
I/O8 30
I/O9 32
I/O10 34
I/O11 36
I/O12 39
I/O13 41
I/O14 43
I/O15 45
WE 11
RESET 12
WP 14
DE 28
CE 26
GND1
GND2
27
46
D

FIG.31E-1

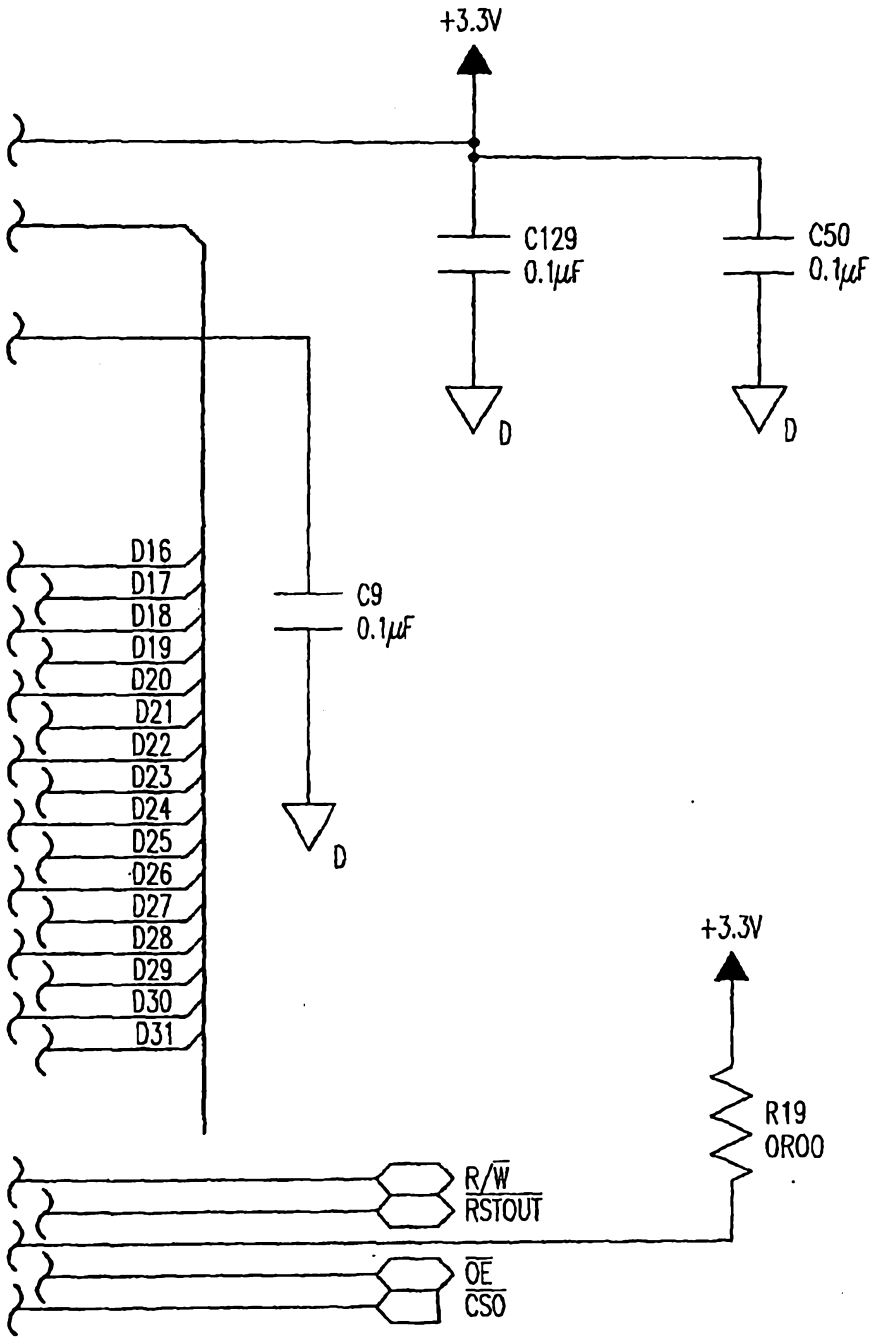

+3.3V
C129
0.1μF
C50
0.1μF
D
D
D16
D17
D18
D19
D20
D21
D22
D23
D24
D25
D26
D27
D28
D29
D30
D31
C9
0.1μF
D
+3.3V
R19
0R00
R/W
RSTOUT
OE
CS0

FIG.31E-2

MEMÓRIA FLASH SCH202-3

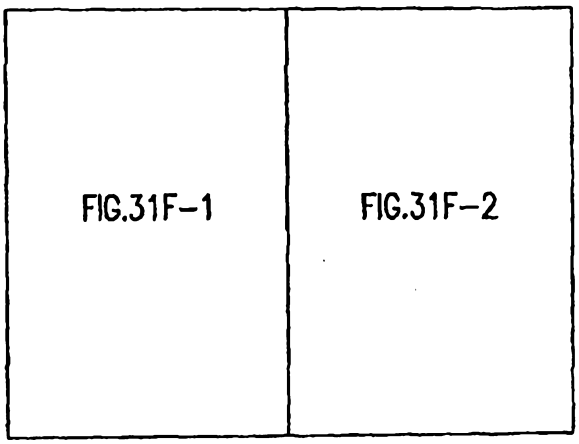


FIG.31F

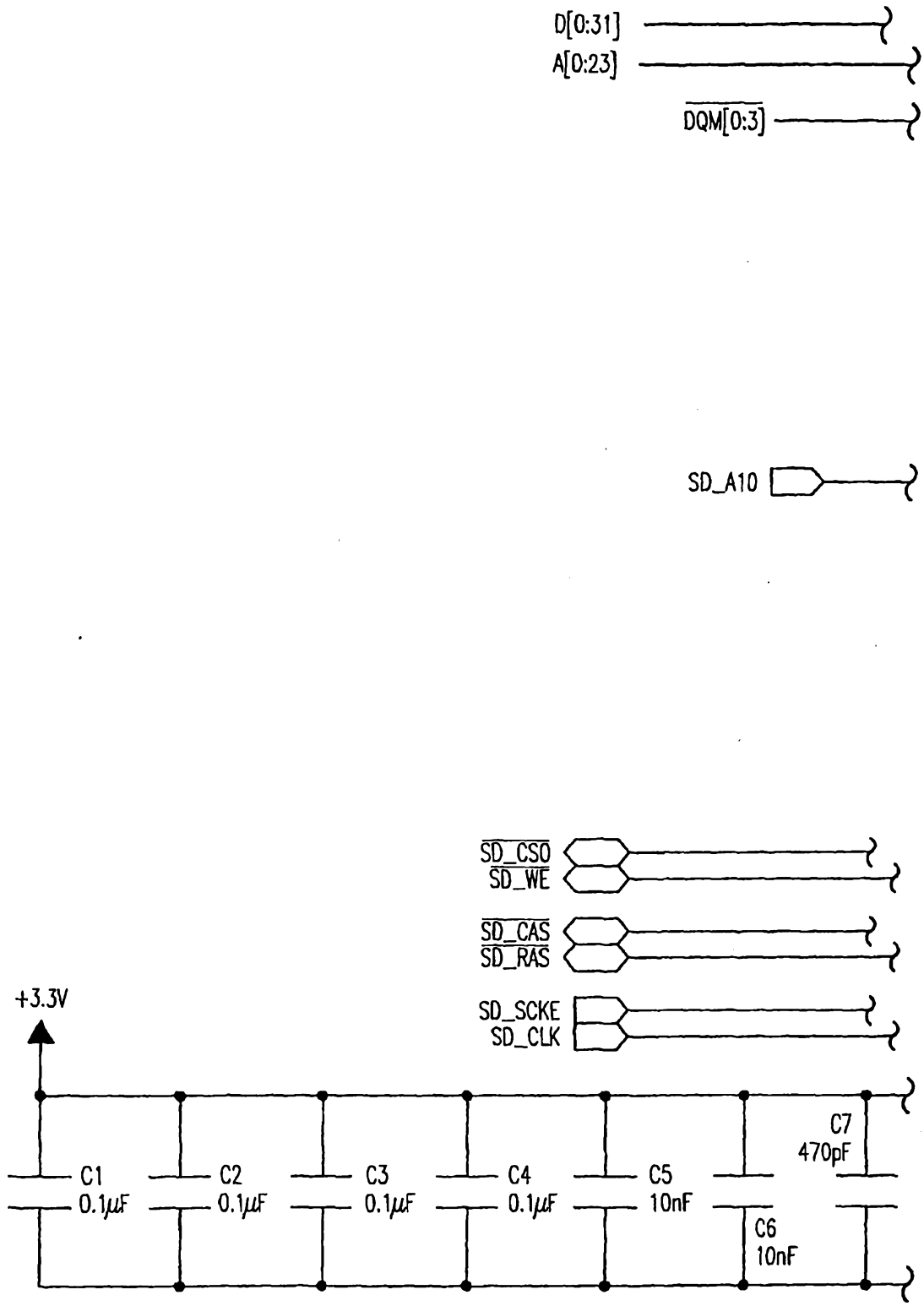

D[0:31]
A[0:23]
DQM[0:3]
SD_A10
SD_CS0
SD_WE
SD_CAS
SD_RAS
SD_SCKE
SD_CLK
+3.3V
C1
0.1μF
C2
0.1μF
C3
0.1μF
C4
0.1μF
C5
10nF
C6
10nF
C7
470pF

FIG.31F-1

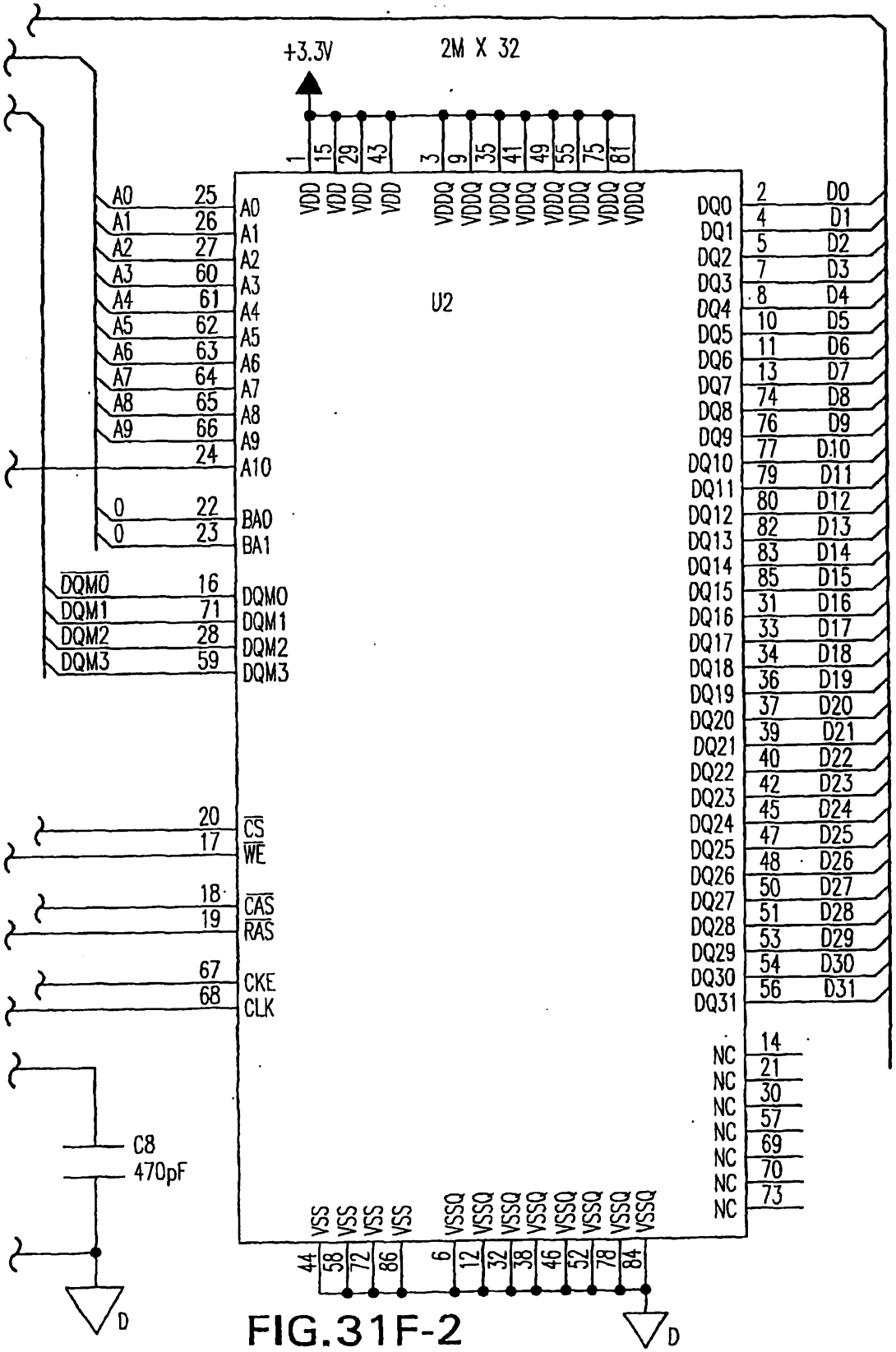

+3.3V
2M X 32
U2
VDD
VDDQ
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
BA0
BA1
DQM0
DQM1
DQM2
DQM3
CS
WE
CAS
RAS
CKE
CLK
DQ0
DQ1
DQ2
DQ3
DQ4
DQ5
DQ6
DQ7
DQ8
DQ9
DQ10
DQ11
DQ12
DQ13
DQ14
DQ15
DQ16
DQ17
DQ18
DQ19
DQ20
DQ21
DQ22
DQ23
DQ24
DQ25
DQ26
DQ27
DQ28
DQ29
DQ30
DQ31
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
D11
D12
D13
D14
D15
D16
D17
D18
D19
D20
D21
D22
D23
D24
D25
D26
D27
D28
D29
D30
D31
NC
VSS
VSSQ
C8
470pF
D

FIG.31F-2

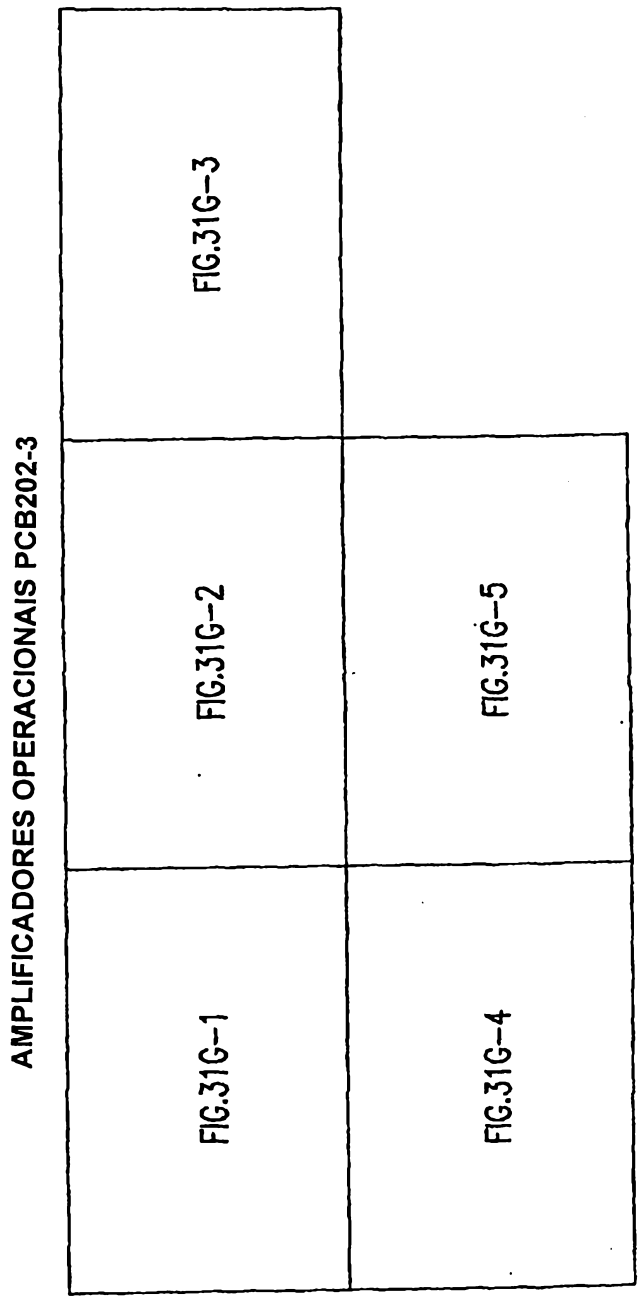

AMPLIFICADORES OPERACIONAIS PCB202-3
FIG.31G-1
FIG.31G-2
FIG.31G-3
FIG.31G-4
FIG.31G-5

FIG.31G

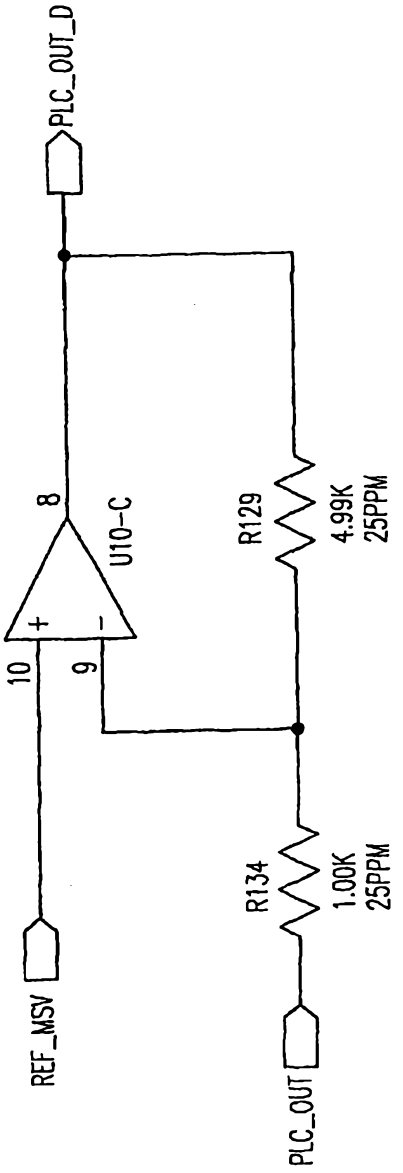

PLC_OUT_D
8
U10-C
10
9
+
-
R129
4.99K
25PPM
R134
1.00K
25PPM
REF_MSV
PLC_OUT
VA_OUT_D
7
U10-B
5
6
R135
4.99K
25PPM
R136
1.00K
25PPM
REF_MSV
VA_OUT

FIG.31G-1

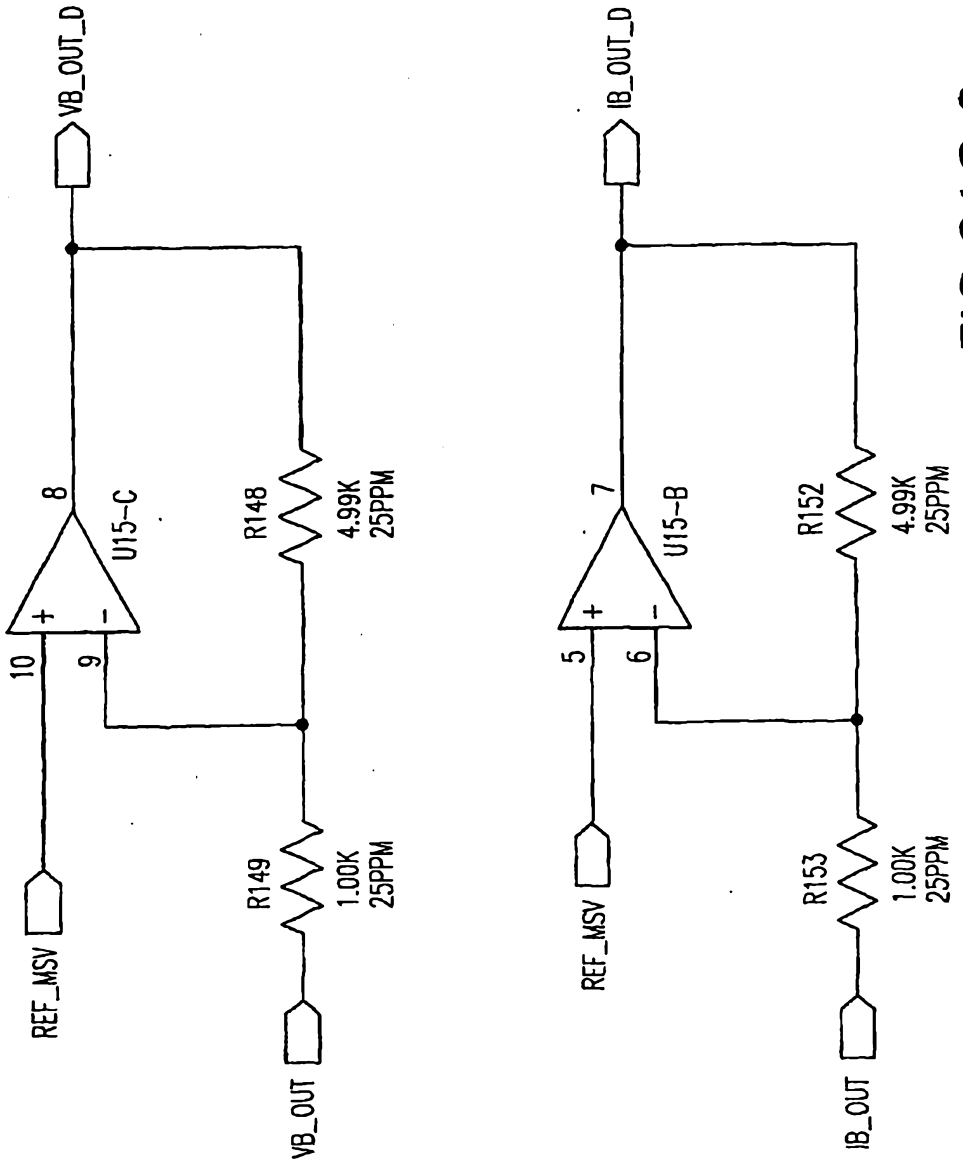

VB_OUT_D
8
U15-C
10
9
REF_MSV
R148
4.99K
25PPM
R149
1.00K
25PPM
VB_OUT
IB_OUT_D
7
U15-B
5
6
REF_MSV
R152
4.99K
25PPM
R153
1.00K
25PPM
IB_OUT

FIG.31G-2

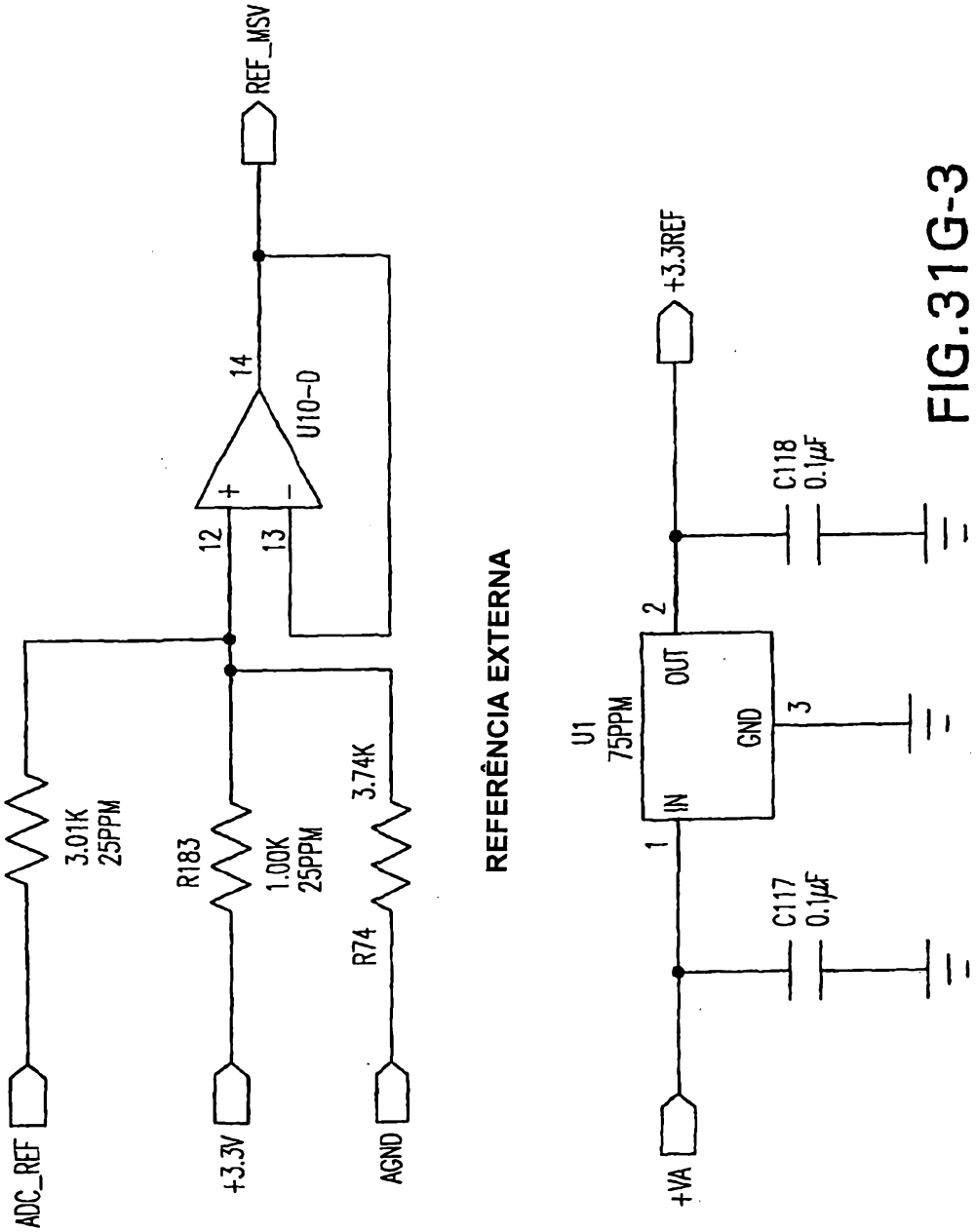

REF_MSV
14
U10-D
12
13
+
-
R184
3.01K
25PPM
R183
1.00K
25PPM
R74
3.74K
ADC_REF
+3.3V
AGND
REFERÊNCIA EXTERNA
U1
75PPM
IN
OUT
GND
1
2
3
+3.3REF
C118
0.1µF
C117
0.1µF
+VA

FIG.31G-3

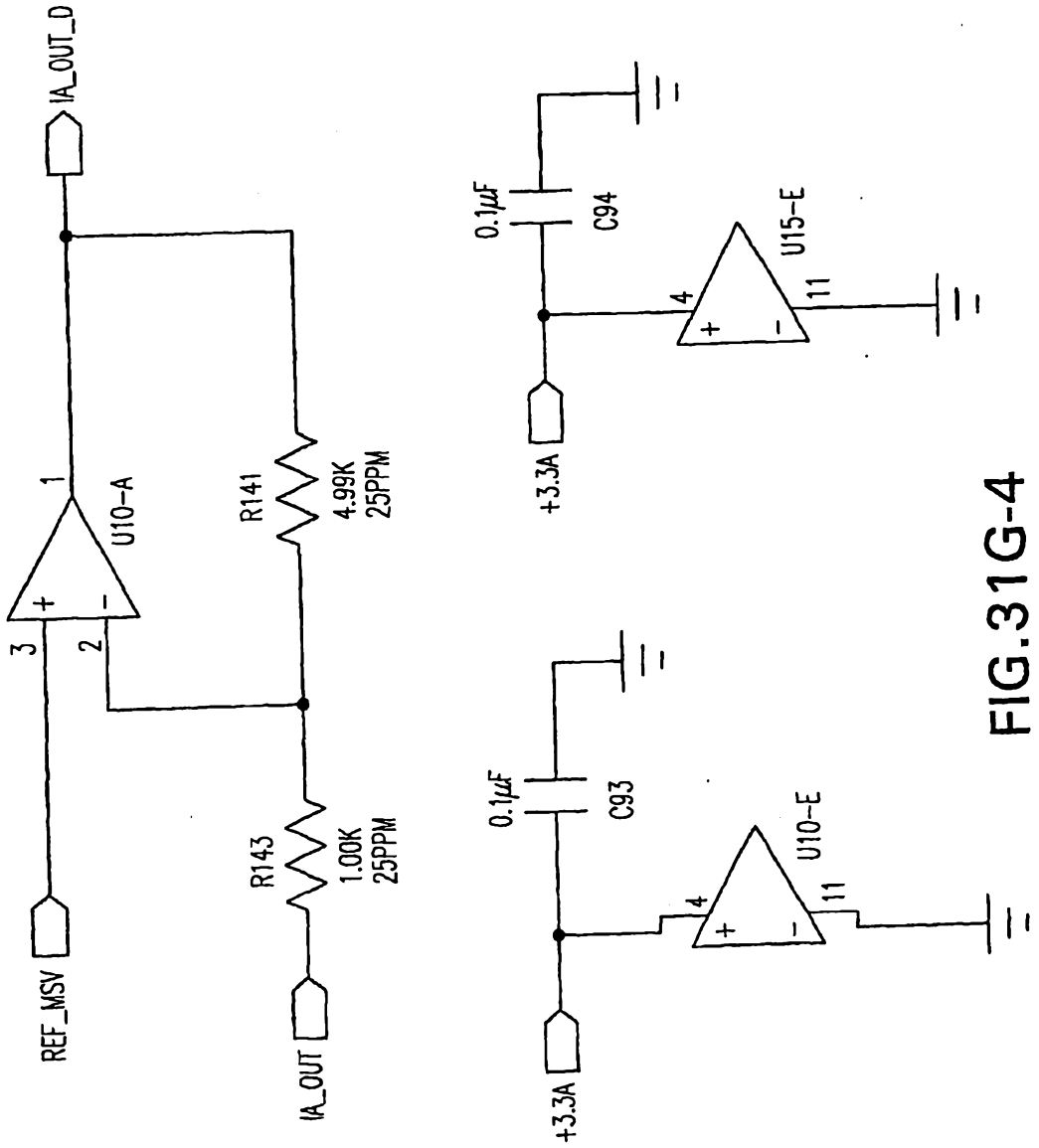

IA_OUT_D
U10-A
1
2
3
+
-
R141
4.99K
25PPM
R143
1.00K
25PPM
REF_MSV
IA_OUT
0.1µF
C94
U15-E
+3.3A
0.1µF
C93
U10-E
+3.3A
4
11

FIG.31G-4

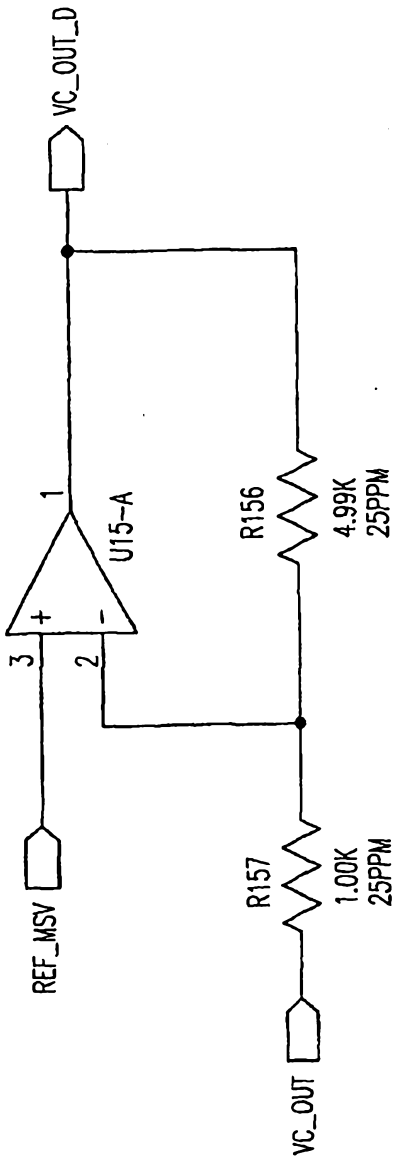

VC_OUT_D
1
U15-A
3
2
+
-
REF_MSV
R156
4.99K
25PPM
R157
1.00K
25PPM
VC_OUT
IC_OUT_D
14
U15-D
12
13
+
-
REF_MSV
R163
4.99K
25PPM
R164
1.00K
25PPM
IC_OUT

FIG.31G-5

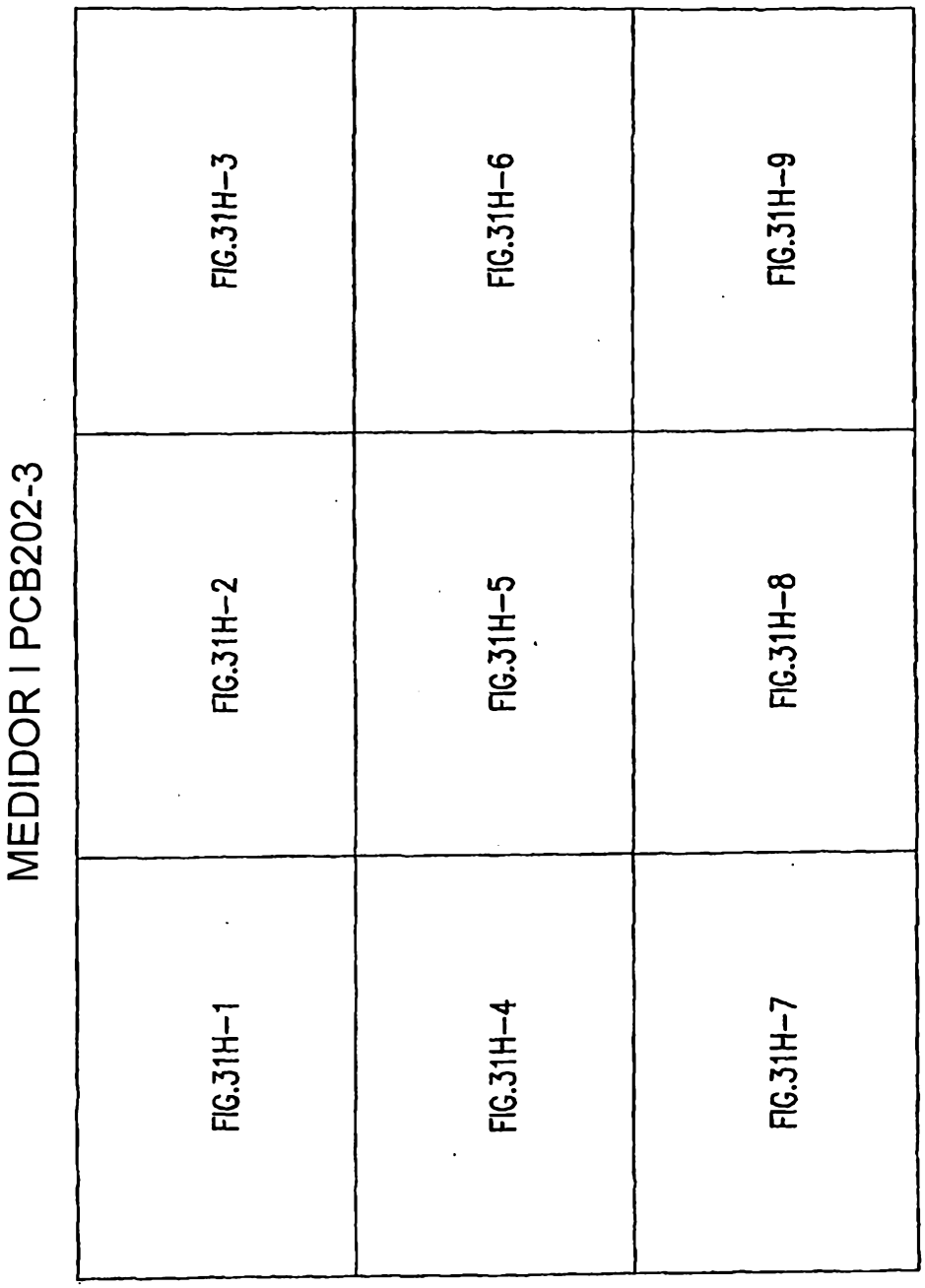

MEDIDOR I PCB202-3
FIG.31H-1
FIG.31H-2
FIG.31H-3
FIG.31H-4
FIG.31H-5
FIG.31H-6
FIG.31H-7
FIG.31H-8
FIG.31H-9

FIG.31H

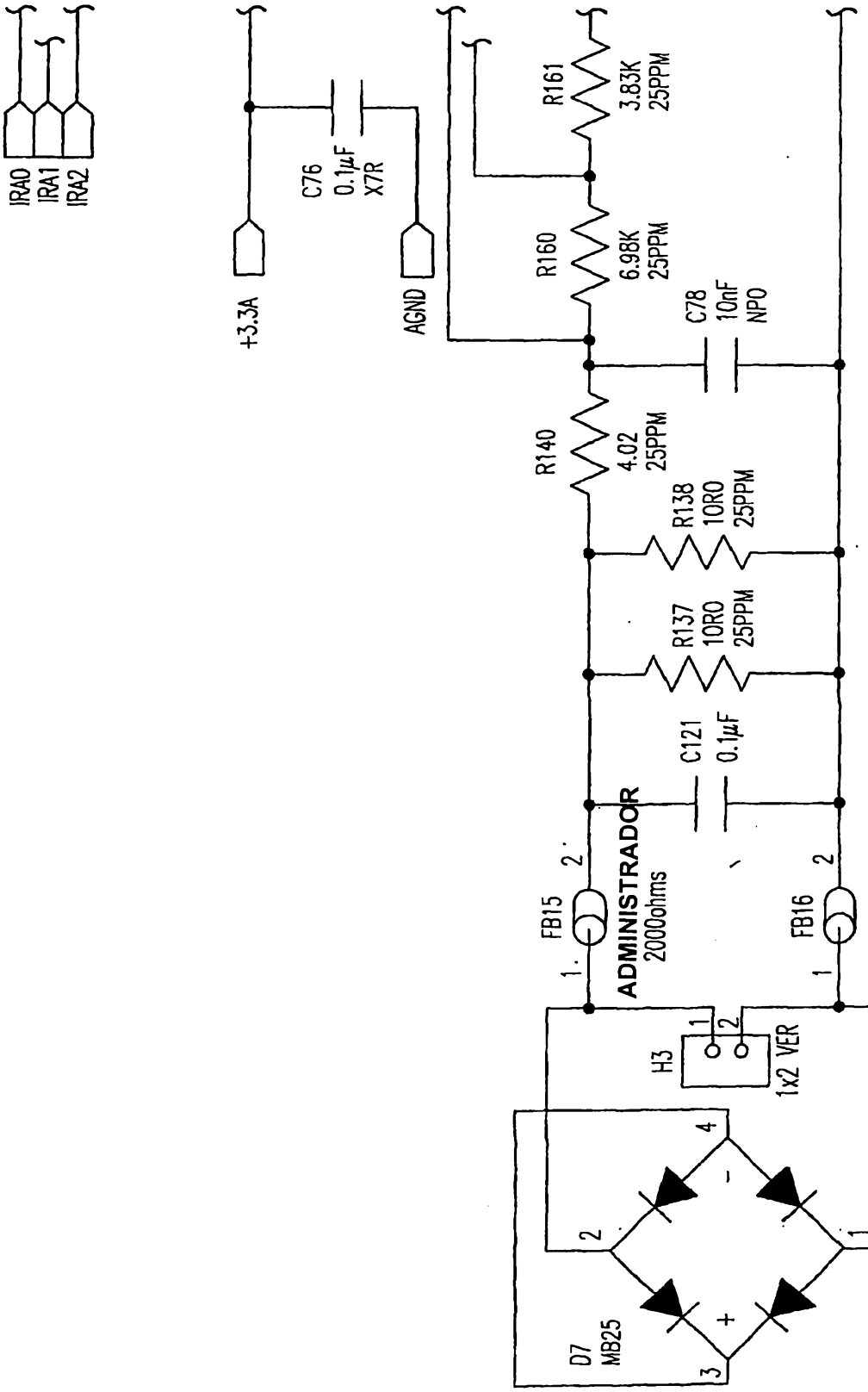

IRA0
IRA1
IRA2
+3.3A
C76
0.1µF
X7R
AGND
R161
3.83K
25PPM
R160
6.98K
25PPM
C78
10nF
NPO
R140
4.02
25PPM
R138
10RO
25PPM
R137
10RO
25PPM
C121
0.1µF
FB15
ADMINISTRADOR
2000ohms
FB16
ADMINISTRADOR
2000ohms
H3
1x2 VER
D7
MB25

FIG.31H-1

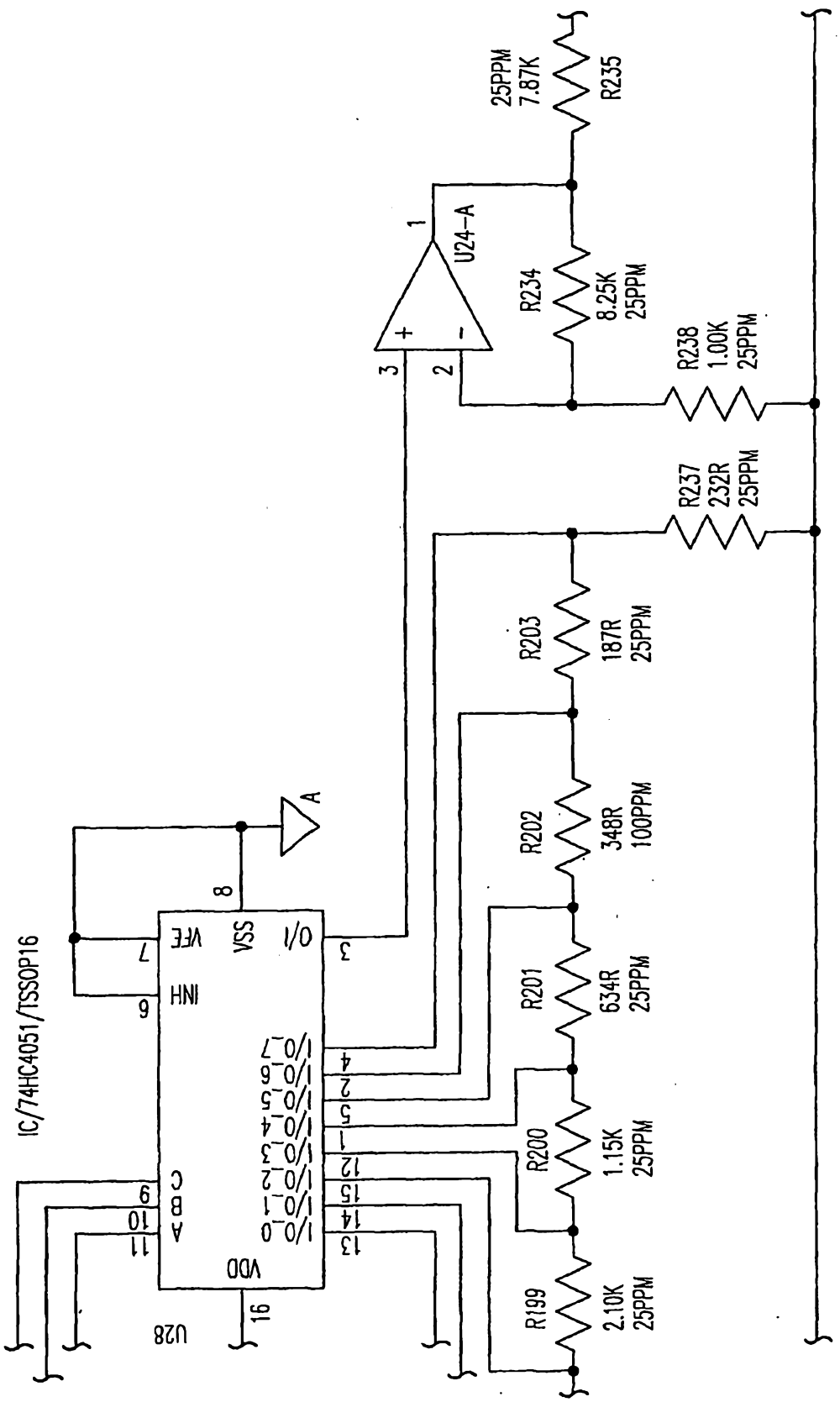

IC/74HC4051/TSSOP16
U28
VDD
VSS
VEE
INH
A
B
C
I/O
I/O_0
I/O_1
I/O_2
I/O_3
I/O_4
I/O_5
I/O_6
I/O_7
U24-A
R199 2.10K 25PPM
R200 1.15K 25PPM
R201 634R 25PPM
R202 348R 100PPM
R203 187R 25PPM
R237 232R 25PPM
R238 1.00K 25PPM
R234 8.25K 25PPM
R235 7.87K 25PPM

FIG.31H-2

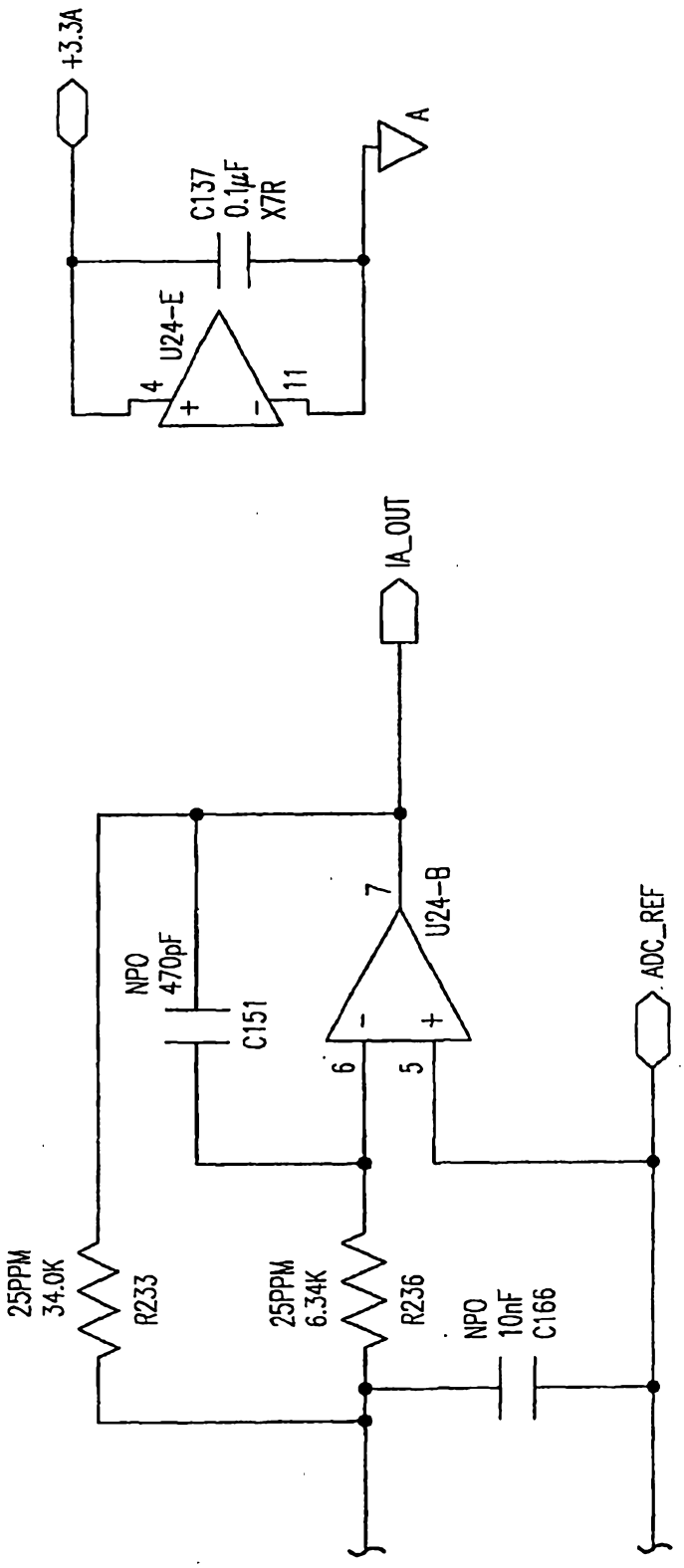

+3.3A
C137
0.1µF
X7R
A
U24-E
4
11
IA_OUT
NPO
470pF
C151
7
U24-B
6
5
ADC_REF
25PPM
34.0K
R233
25PPM
6.34K
R236
NPO
10nF
C166

FIG.31H-3

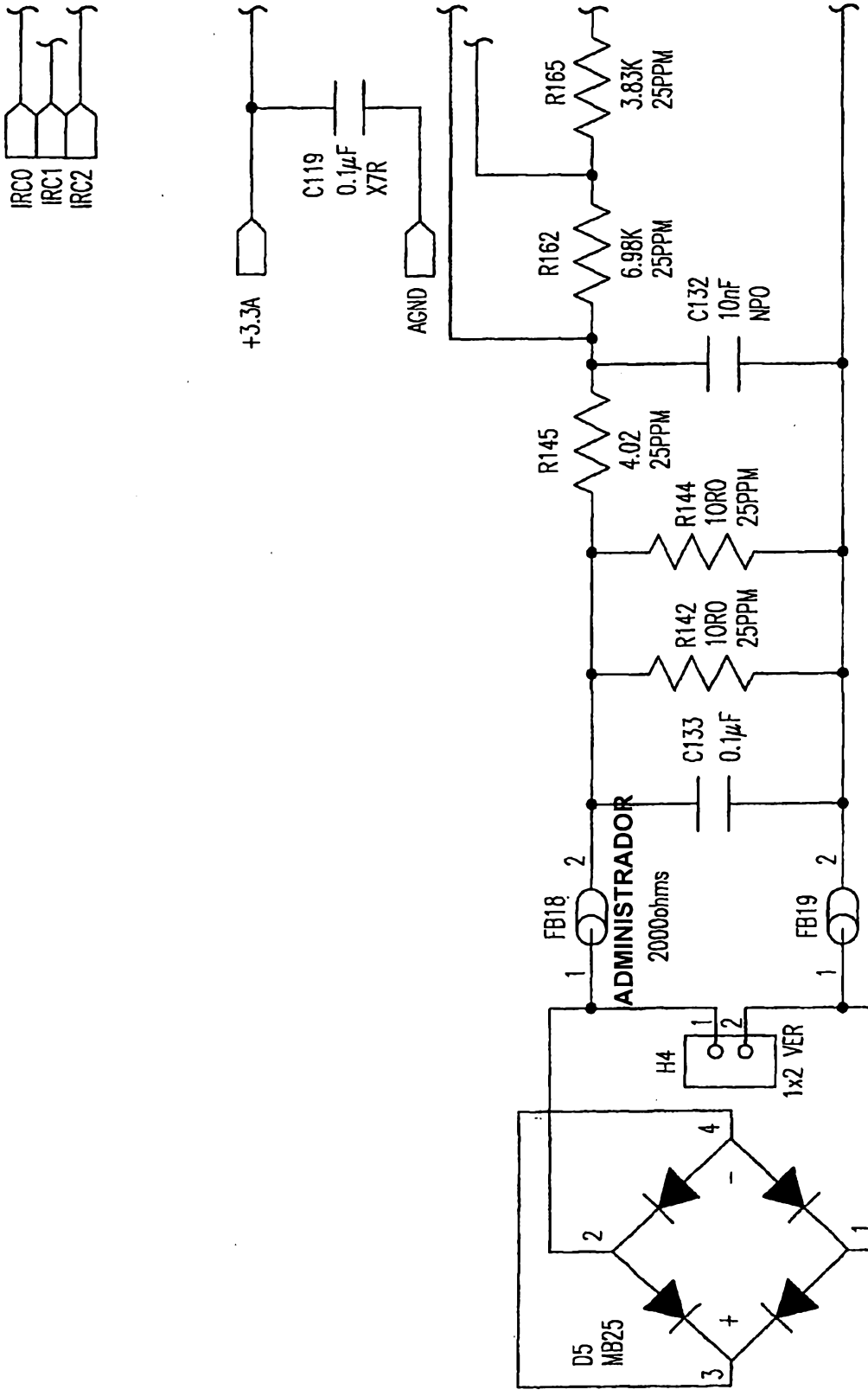

IRC0
IRC1
IRC2
+3.3A
C119
0.1µF
X7R
AGND
R165
3.83K
25PPM
R162
6.98K
25PPM
C132
10nF
NPO
R145
4.02
25PPM
R144
10RO
25PPM
R142
10RO
25PPM
C133
0.1µF
FB18
ADMINISTRADOR
2000ohms
FB19
ADMINISTRADOR
2000ohms
H4
1x2 VER
D5
MB25

FIG.31H-4

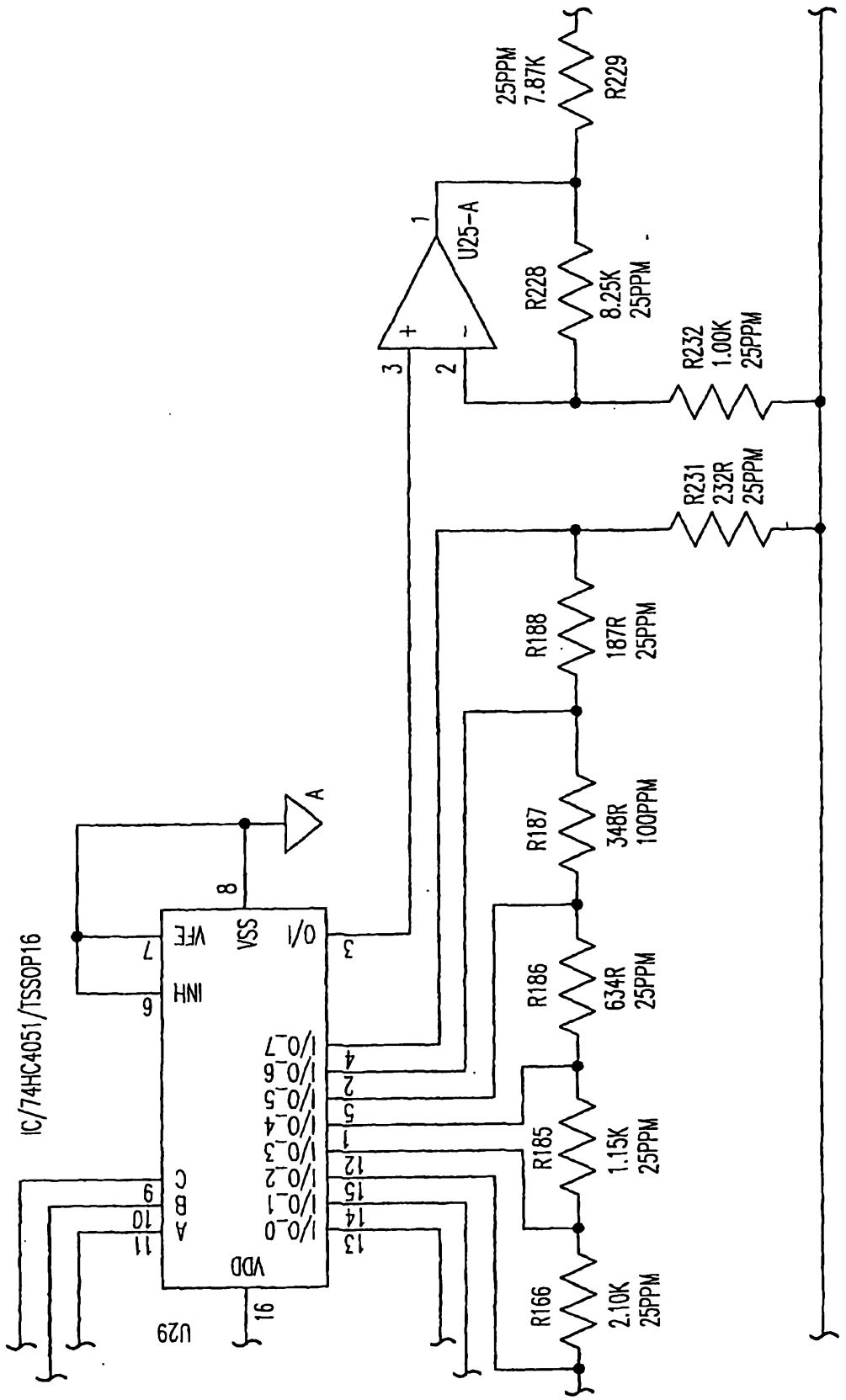

IC/74HC4051/TSSOP16
U29
VDD
VEE
VSS
INH
A
B
C
I/O
I/O_0
I/O_1
I/O_2
I/O_3
I/O_4
I/O_5
I/O_6
I/O_7
U25-A
R166 2.10K 25PPM
R185 1.15K 25PPM
R186 634R 25PPM
R187 348R 100PPM
R188 187R 25PPM
R231 232R 25PPM
R232 1.00K 25PPM
R228 8.25K 25PPM
R229 7.87K 25PPM

FIG.31H-5

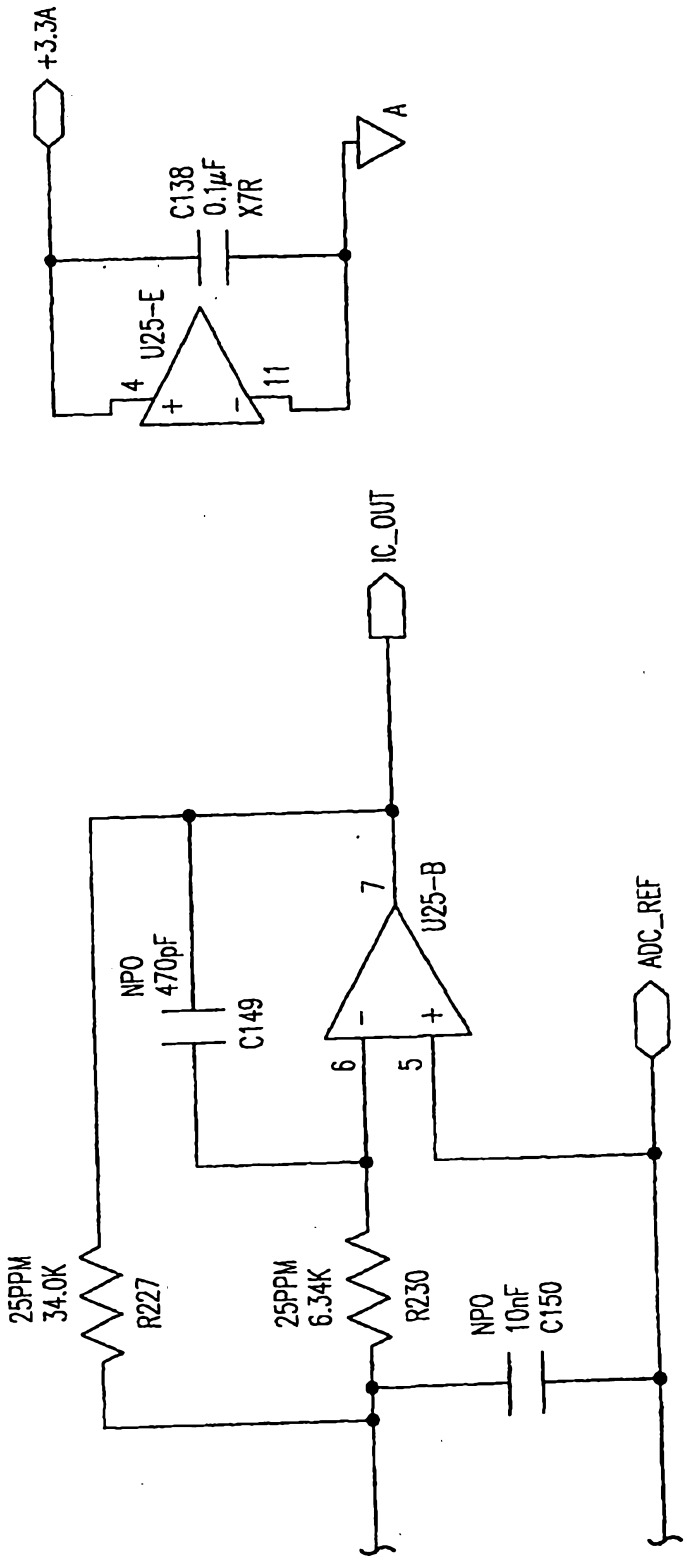

+3.3A
A
C138
0.1µF
X7R
U25-E
4
11
IC_OUT
NPO
470pF
C149
7
U25-B
6
5
ADC_REF
25PPM
34.0K
R227
25PPM
6.34K
R230
NPO
10nF
C150

FIG.31H-6

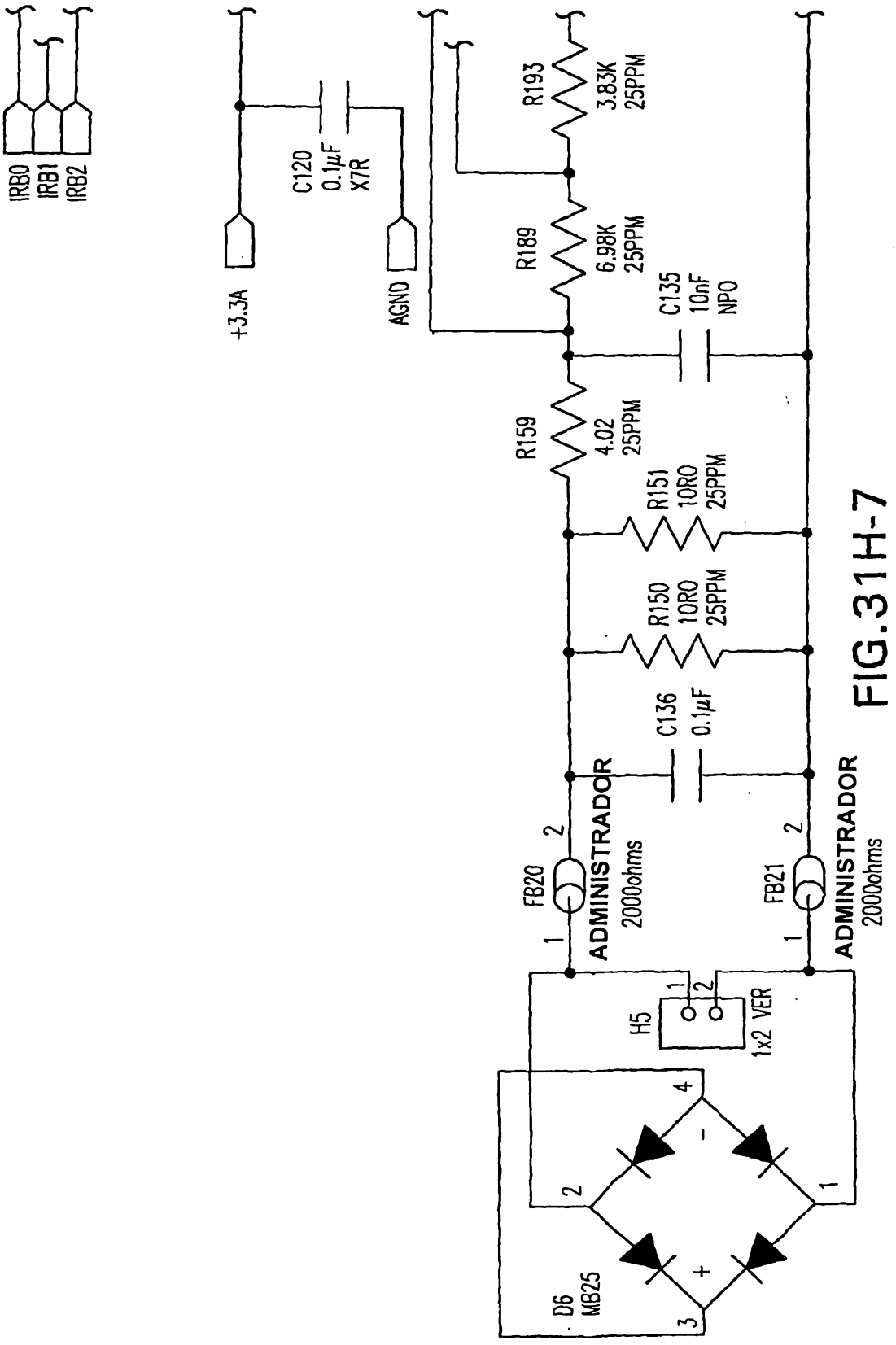

IRB0
IRB1
IRB2
+3.3A
C120
0.1µF
X7R
AGND
R193
3.83K
25PPM
R189
6.98K
25PPM
C135
10nF
NPO
R159
4.02
25PPM
R151
10RO
25PPM
R150
10RO
25PPM
C136
0.1µF
FB20
ADMINISTRADOR
2000ohms
FB21
ADMINISTRADOR
2000ohms
H5
1x2 VER
D6
MB25

FIG.31H-7

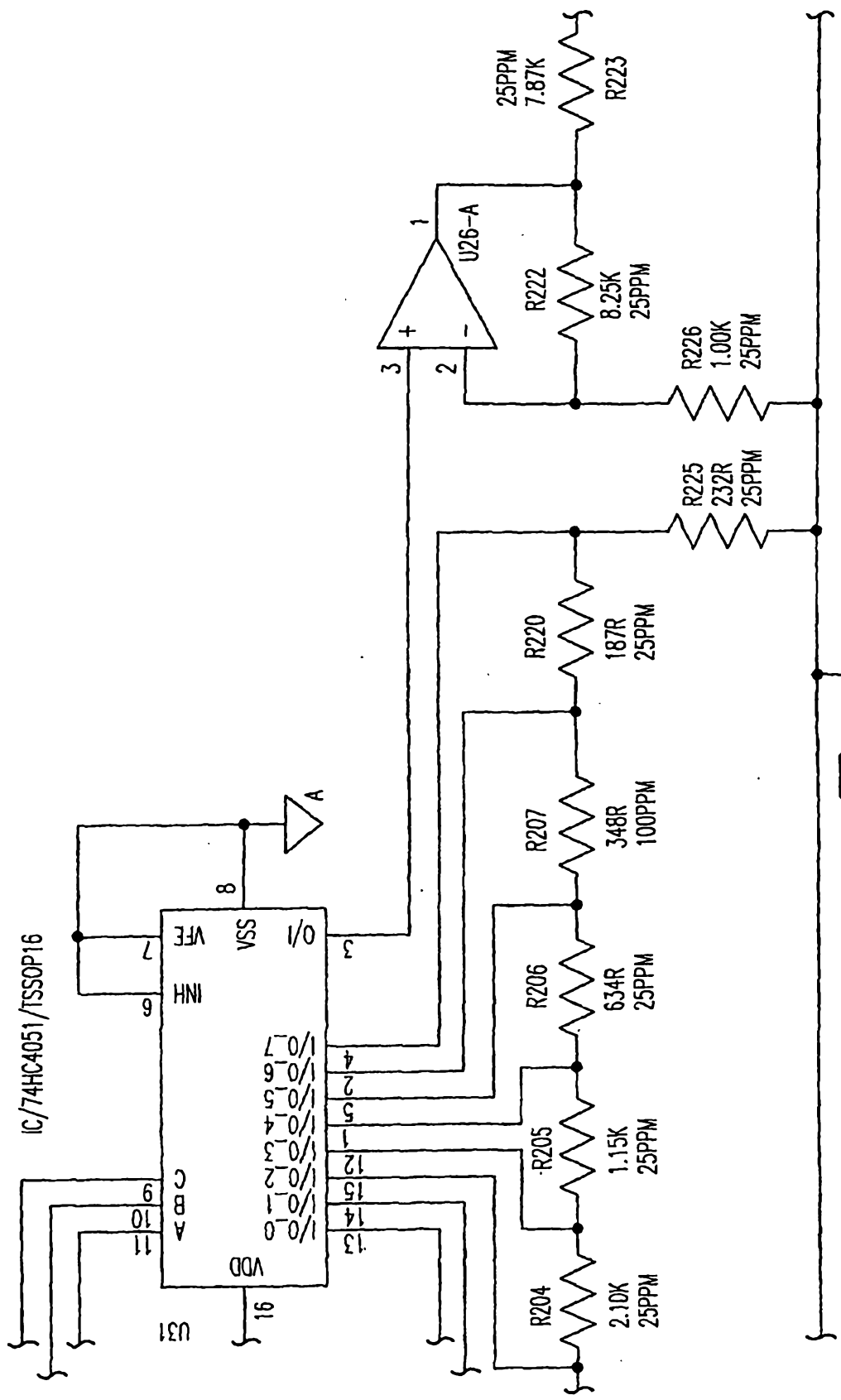

IC/74HC4051/TSSOP16
U31
VDD
VSS
VEE
INH
A
B
C
I/O
I/O_0
I/O_1
I/O_2
I/O_3
I/O_4
I/O_5
I/O_6
I/O_7
U26-A
R204
2.10K
25PPM
R205
1.15K
25PPM
R206
634R
25PPM
R207
348R
100PPM
R220
187R
25PPM
R225
232R
25PPM
R226
1.00K
25PPM
R222
8.25K
25PPM
R223
7.87K
25PPM
ADC_REF

FIG.31H-8

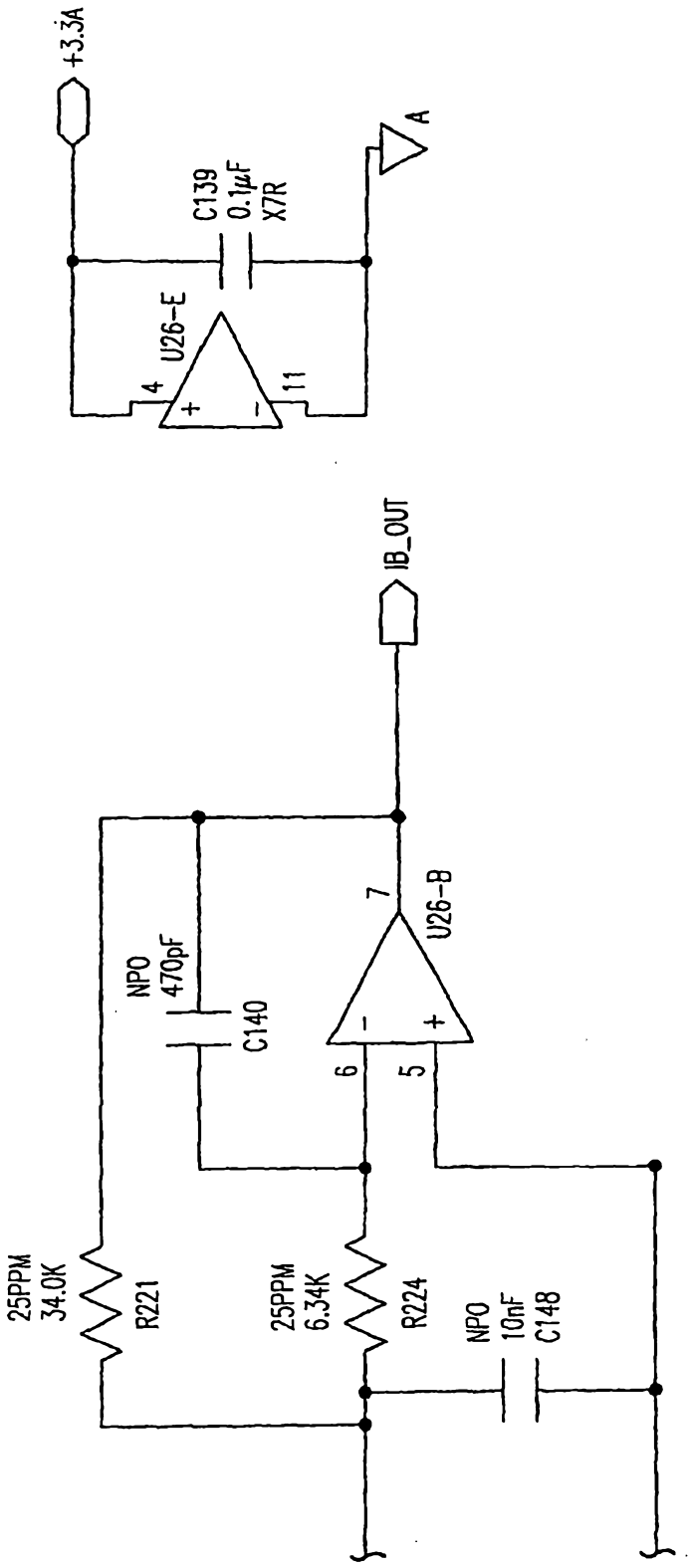

+3.3A
C139
0.1µF
X7R
A
U26-E
4
11
IB_OUT
NPO
470pF
C140
7
U26-B
6
5
25PPM
34.0K
R221
25PPM
6.34K
R224
NPO
10nF
C148

FIG.31H-9

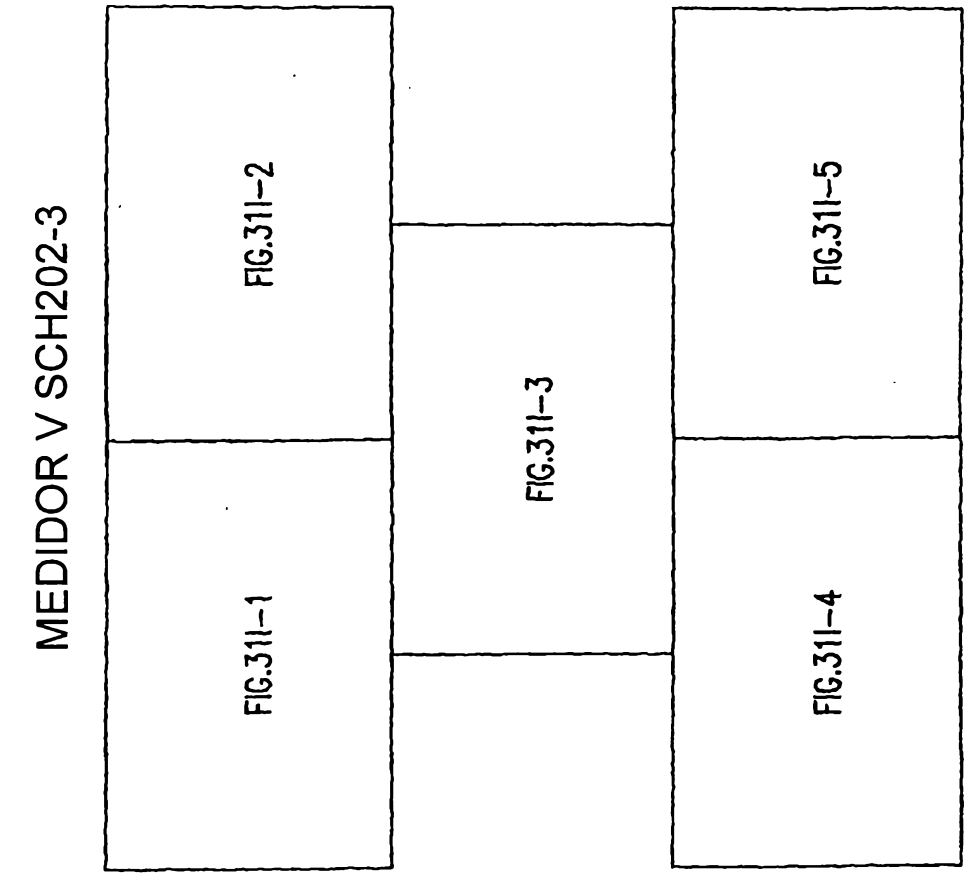

MEDIDOR V SCH202-3
FIG.311-1
FIG.311-2
FIG.311-3
FIG.311-4
FIG.311-5

FIG.311

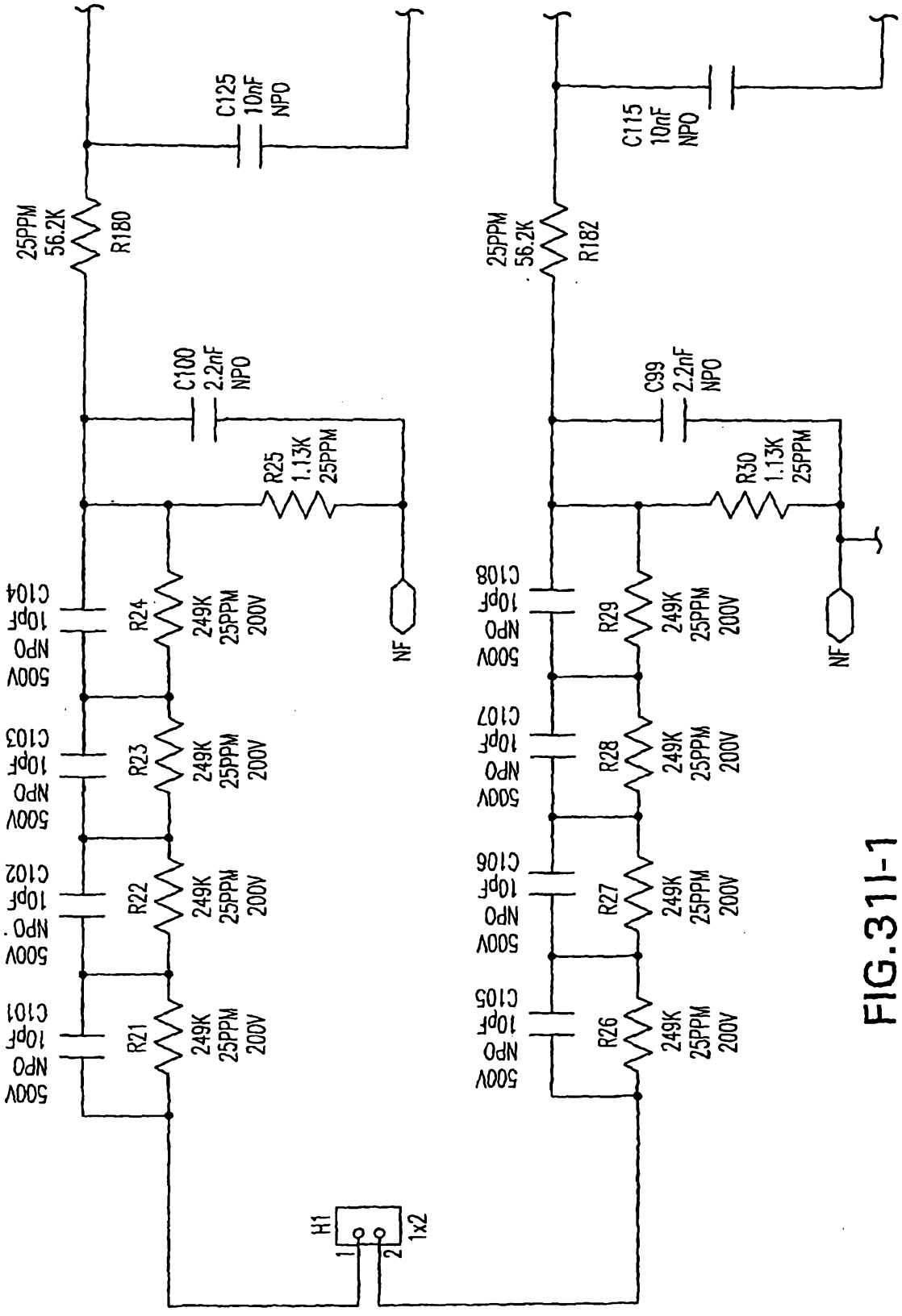

C125
10nF
NPO
25PPM
56.2K
R180
C100
2.2nF
NPO
R25
1.13K
25PPM
NF
C104
10pF
NPO
500V
R24
249K
25PPM
200V
C103
10pF
NPO
500V
R23
249K
25PPM
200V
C102
10pF
NPO
500V
R22
249K
25PPM
200V
C101
10pF
NPO
500V
R21
249K
25PPM
200V
H1
1
2
1x2
C115
10nF
NPO
25PPM
56.2K
R182
C99
2.2nF
NPO
R30
1.13K
25PPM
NF
C108
10pF
NPO
500V
R29
249K
25PPM
200V
C107
10pF
NPO
500V
R28
249K
25PPM
200V
C106
10pF
NPO
500V
R27
249K
25PPM
200V
C105
10pF
NPO
500V
R26
249K
25PPM
200V

FIG.311-1

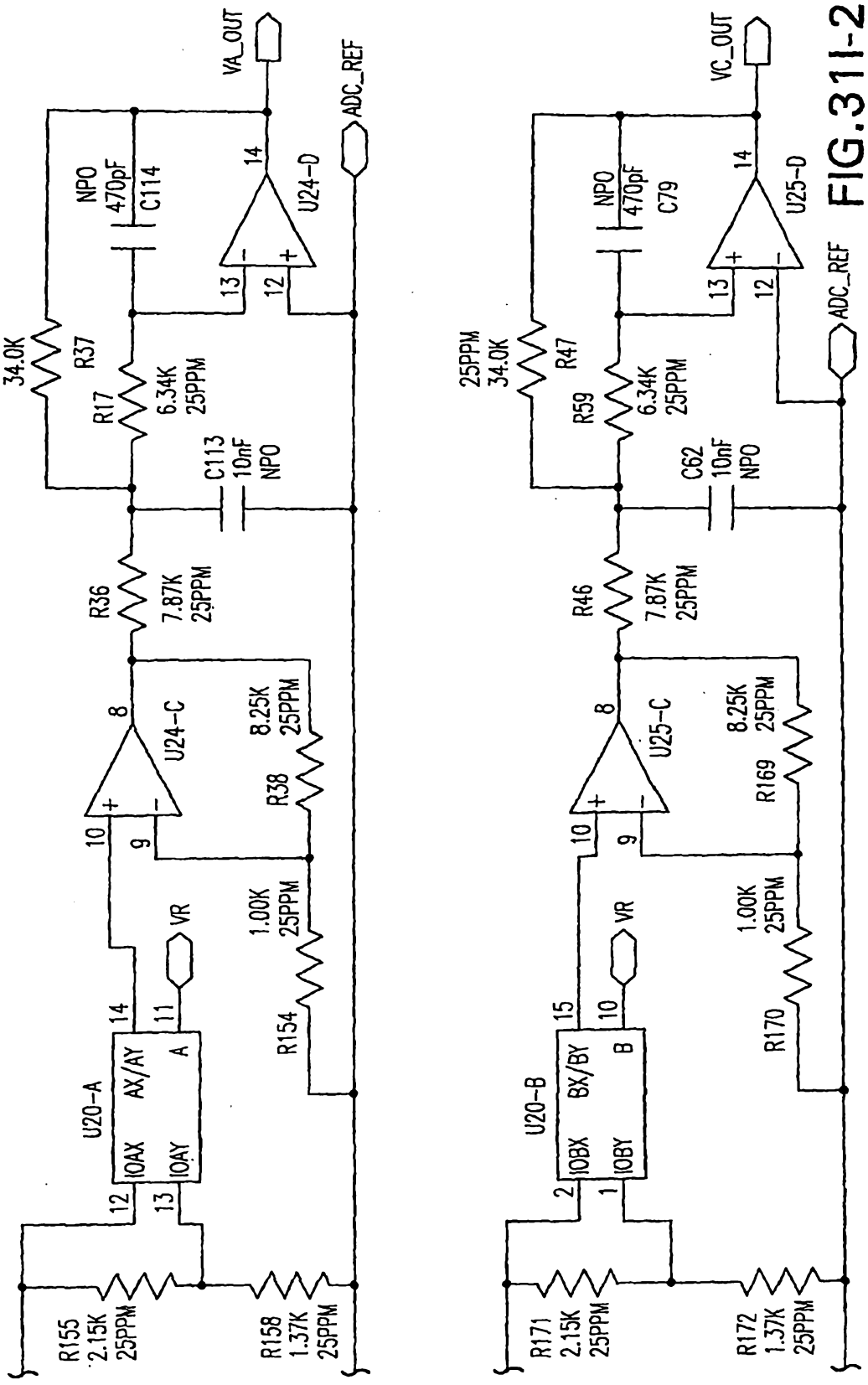

VA_OUT
ADC_REF
NPO
470pF
C114
U24-D
14
13
12
25PPM
34.0K
R37
R17
6.34K
25PPM
C113
10nF
NPO
R36
7.87K
25PPM
8
10
9
U24-C
8.25K
25PPM
R38
1.00K
25PPM
R154
VR
14
11
U20-A
AX/AY
A
IOAX
IOAY
12
13
R155
2.15K
25PPM
R158
1.37K
25PPM
VC_OUT
ADC_REF
NPO
470pF
C79
U25-D
14
13
12
25PPM
34.0K
R47
R59
6.34K
25PPM
C62
10nF
NPO
R46
7.87K
25PPM
8
10
9
U25-C
8.25K
25PPM
R169
1.00K
25PPM
R170
VR
15
10
U20-B
BX/BY
B
IOBX
IOBY
2
1
R171
2.15K
25PPM
R172
1.37K
25PPM

FIG.311-2

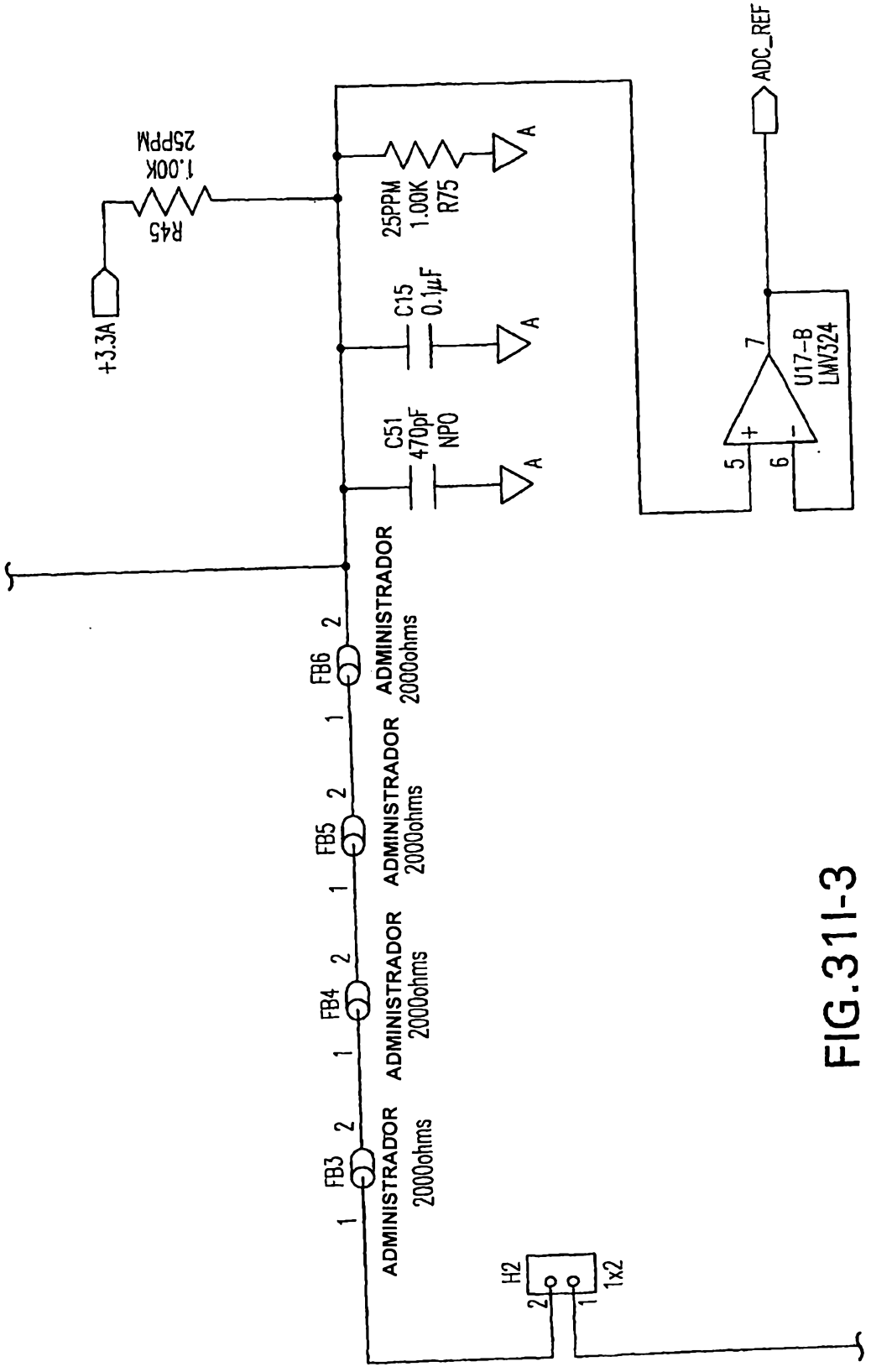

ADC_REF
U17-B
LMV324
7
5
6
+
-
25PPM
1.00K
R75
C15
0.1µF
C51
470pF
NPO
A
R45
1.00K
25PPM
+3.3A
FB6
FB5
FB4
FB3
ADMINISTRADOR
2000ohms
H2
1x2
1
2

FIG.311-3

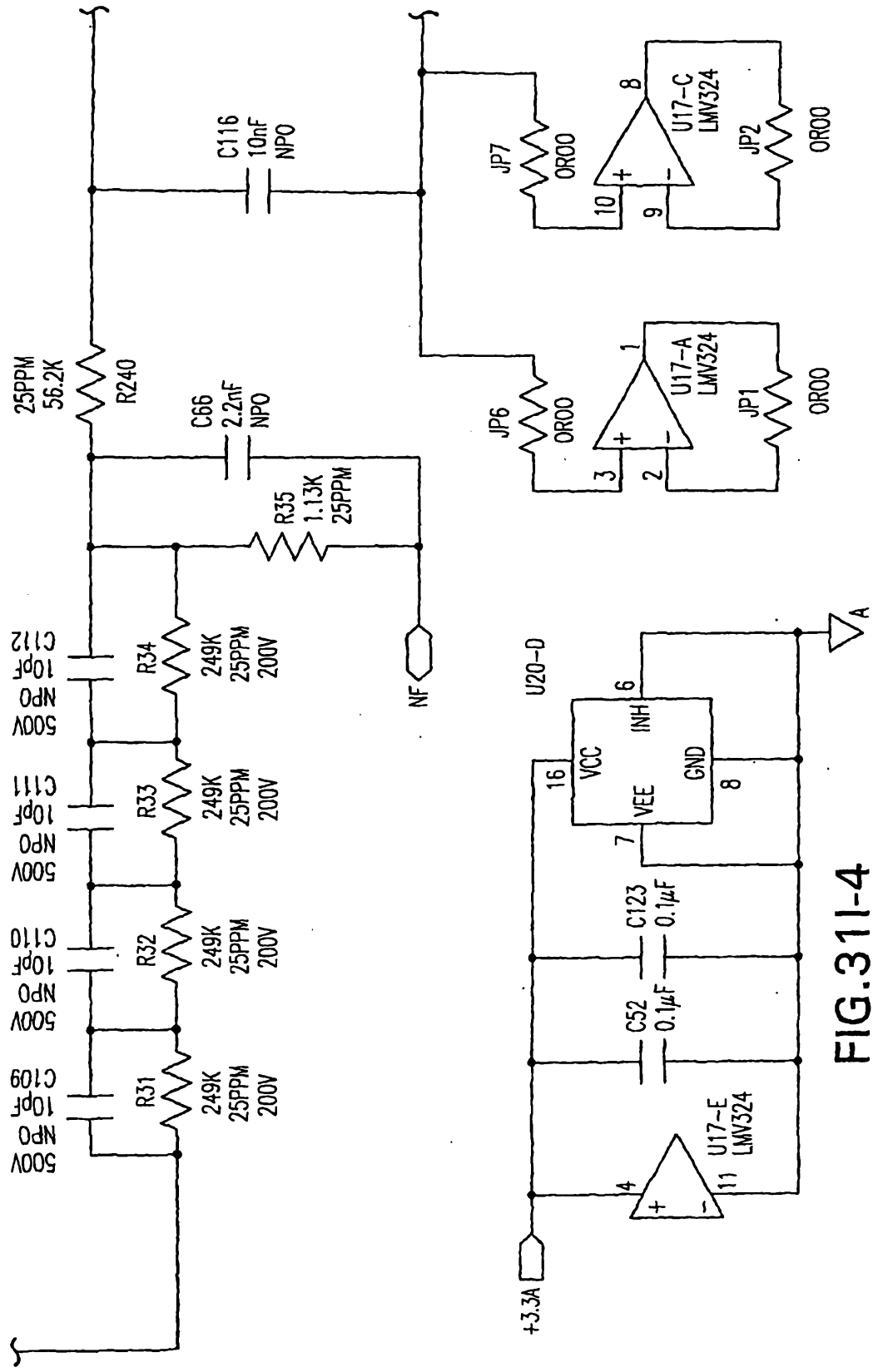

C116
10nF
NPO
JP7
0R00
U17-C
LMV324
JP2
0R00
25PPM
56.2K
R240
C66
2.2nF
NPO
JP6
0R00
U17-A
LMV324
JP1
0R00
R35
1.13K
25PPM
NF
C112
10pF
NPO
500V
R34
249K
25PPM
200V
C111
10pF
NPO
500V
R33
249K
25PPM
200V
C110
10pF
NPO
500V
R32
249K
25PPM
200V
C109
10pF
NPO
500V
R31
249K
25PPM
200V
U20-D
VCC
INH
GND
VEE
C123
0.1µF
C52
0.1µF
U17-E
LMV324
+3.3A
A

FIG.311-4

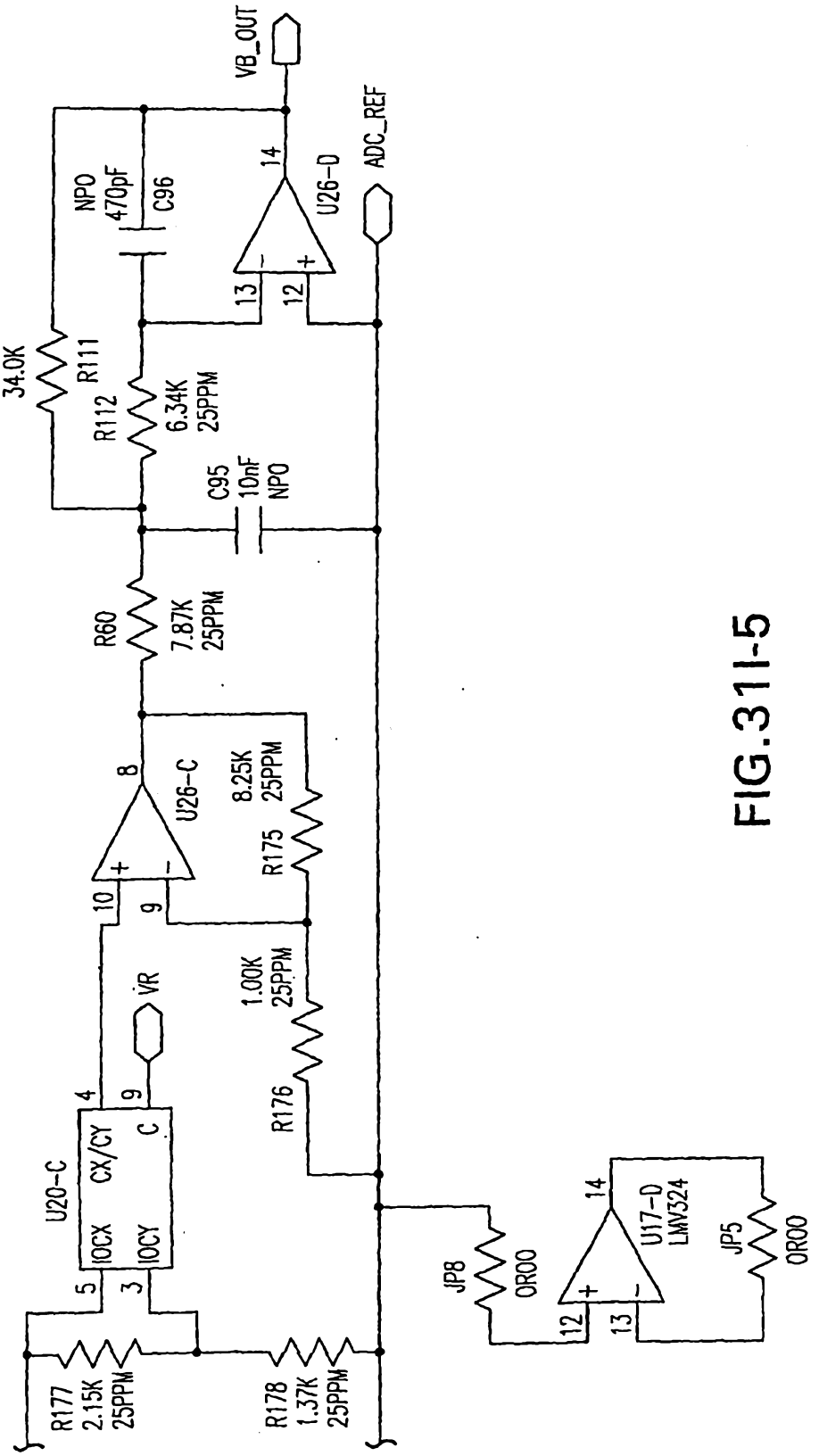

VB_OUT
ADC_REF
NPO
470pF
C96
U26-D
14
13
12
25PPM
34.0K
R111
R112
6.34K
25PPM
C95
10nF
NPO
R60
7.87K
25PPM
8.25K
25PPM
R175
U26-C
8
10
9
VR
1.00K
25PPM
R176
U20-C
CX/CY
IOCX
IOCY
C
4
9
5
3
R177
2.15K
25PPM
R178
1.37K
25PPM
JP8
0R00
U17-D
LMV324
14
12
13
JP5
0R00

FIG.311-5

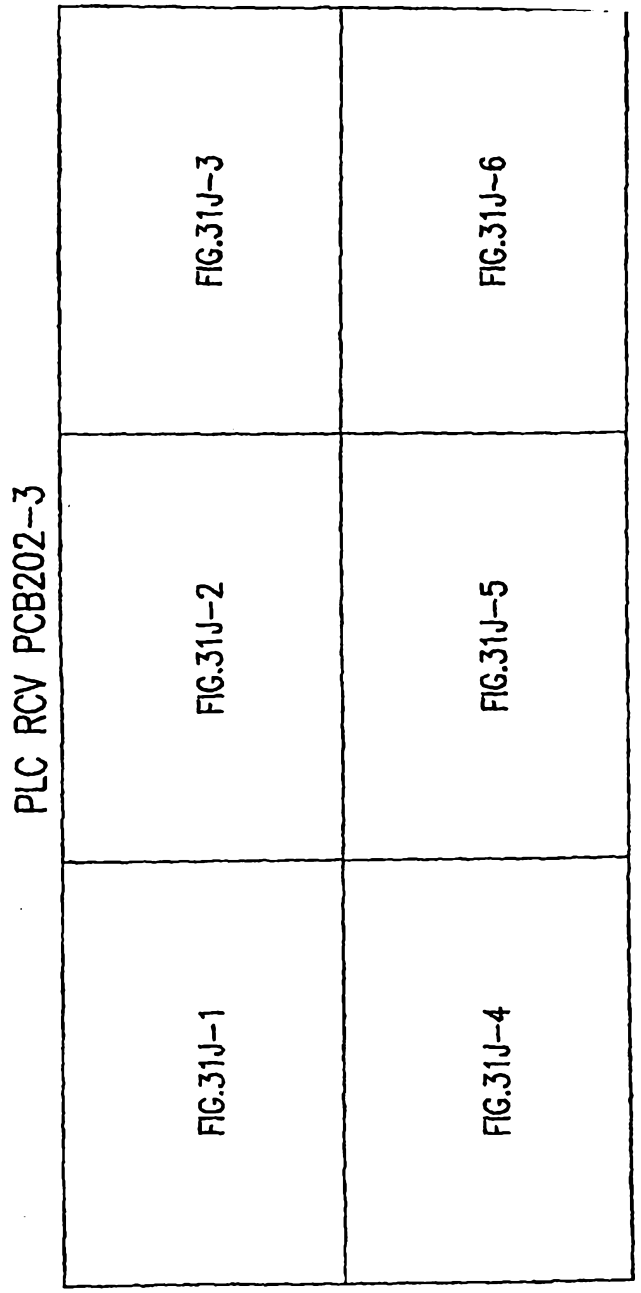

PLC RCV PCB202-3
FIG.31J-1
FIG.31J-2
FIG.31J-3
FIG.31J-4
FIG.31J-5
FIG.31J-6

FIG.31J

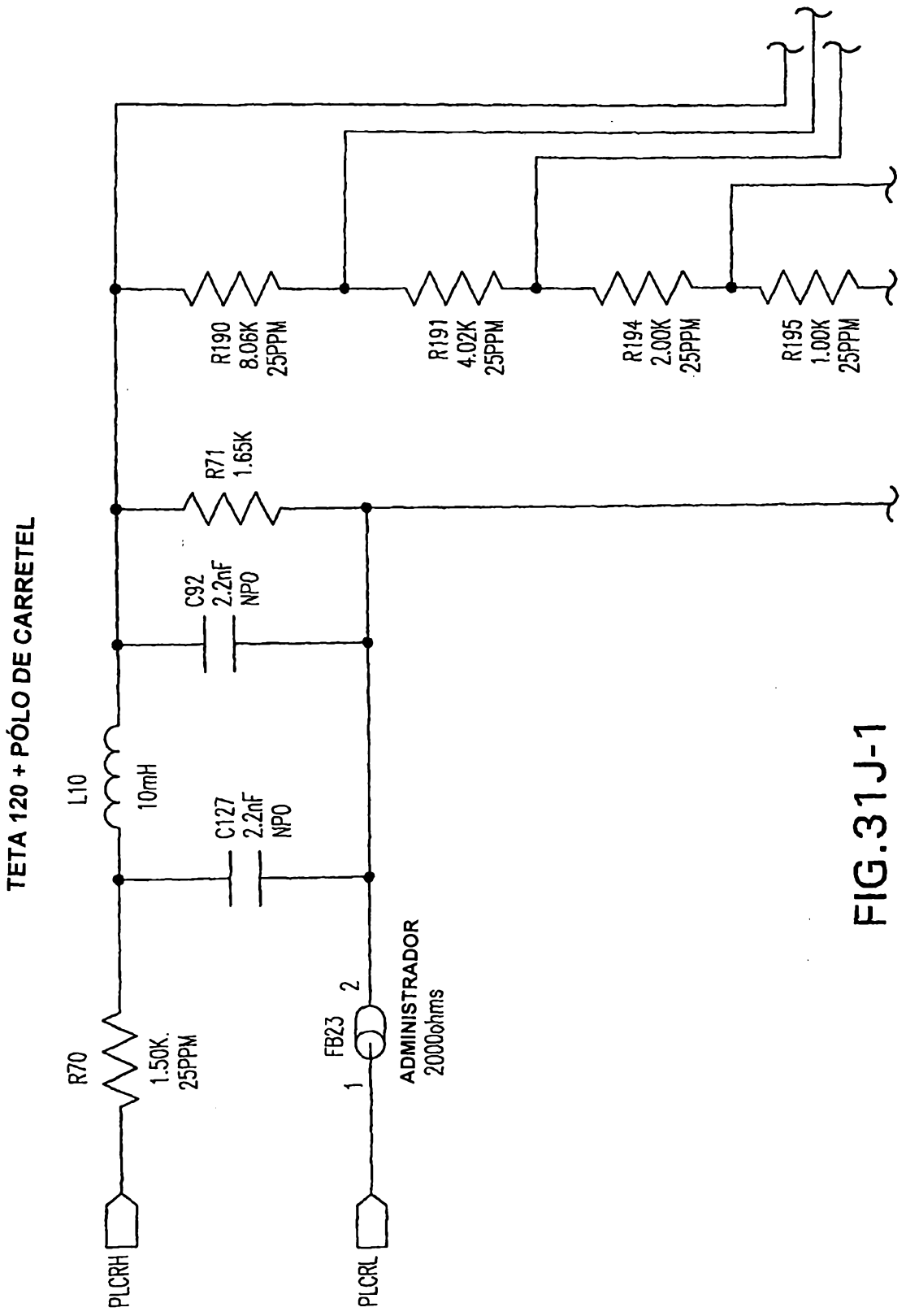

TETA 120 + PÓLO DE CARRETEL
R190
8.06K
25PPM
R191
4.02K
25PPM
R194
2.00K
25PPM
R195
1.00K
25PPM
R71
1.65K
C92
2.2nF
NPO
L10
10mH
C127
2.2nF
NPO
R70
1.50K.
25PPM
FB23
2
1
ADMINISTRADOR
2000ohms
PLCRH
PLCRL

FIG.31J-1

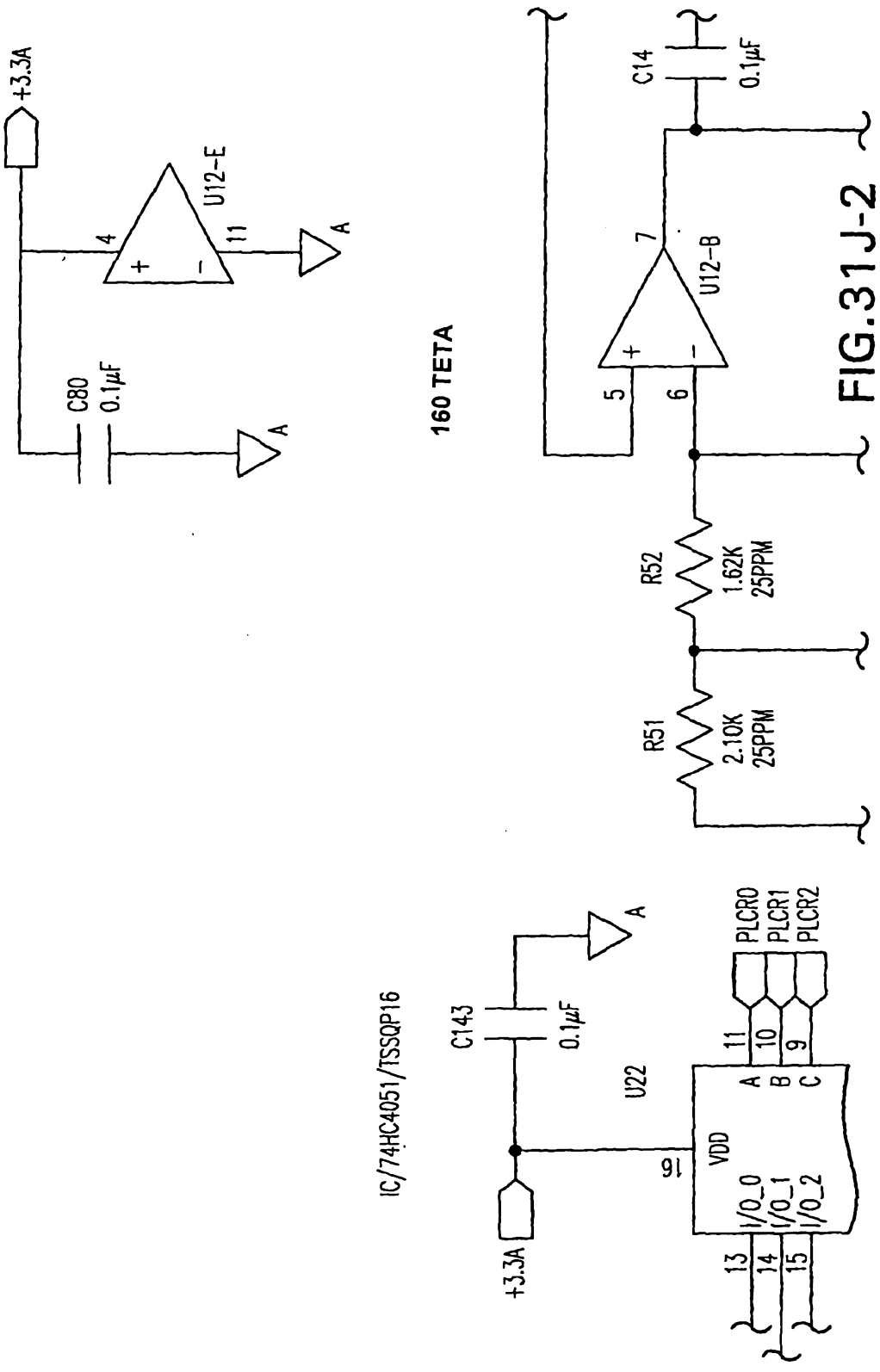

+3.3A
U12-E
4
11
C80
0.1µF
A
160 TETA
C14
0.1µF
7
U12-B
5
6
R52
1.62K
25PPM
R51
2.10K
25PPM
IC/74HC4051/TSSOP16
C143
0.1µF
U22
PLCR0
PLCR1
PLCR2
11
10
9
A
B
C
VDD
I/O_0
I/O_1
I/O_2
13
14
15
+3.3A

FIG.31J-2

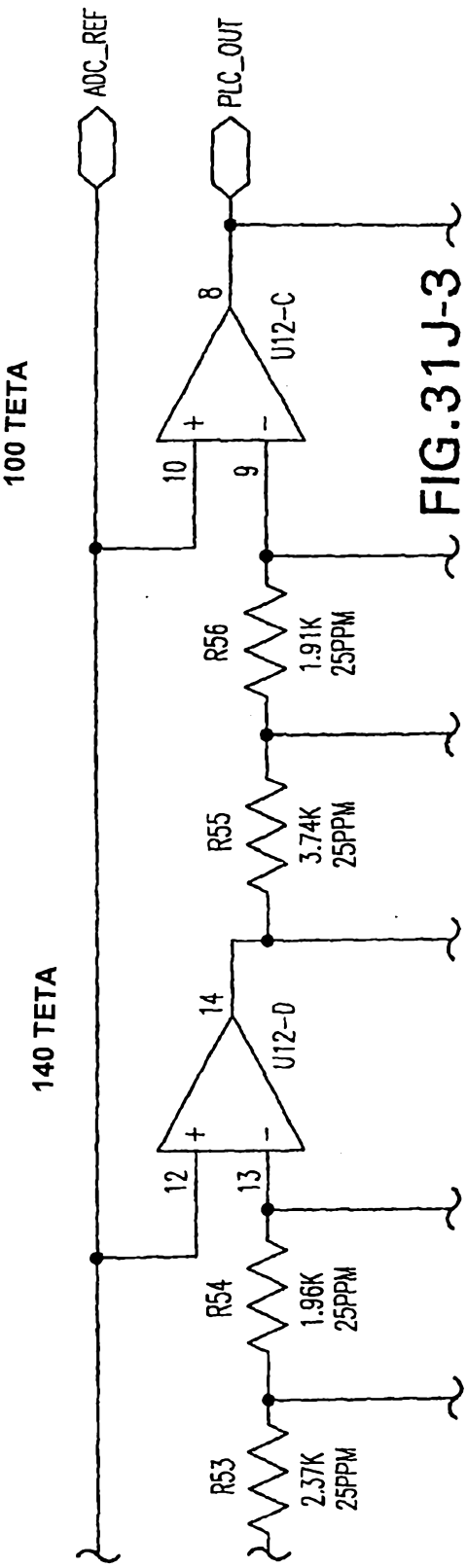

ADC_REF
PLC_OUT
100 TETA
140 TETA
U12-C
U12-D
8
9
10
12
13
14
R53
2.37K
25PPM
R54
1.96K
25PPM
R55
3.74K
25PPM
R56
1.91K
25PPM

FIG.31J-3

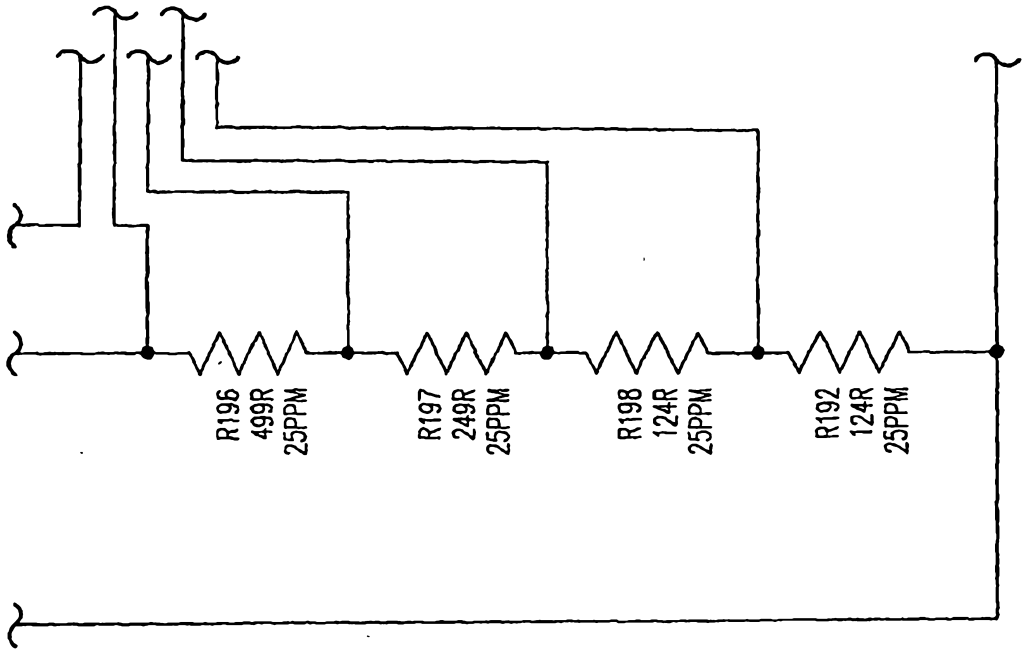

R196
499R
25PPM
R197
249R
25PPM
R198
124R
25PPM
R192
124R
25PPM

FIG.31J-4

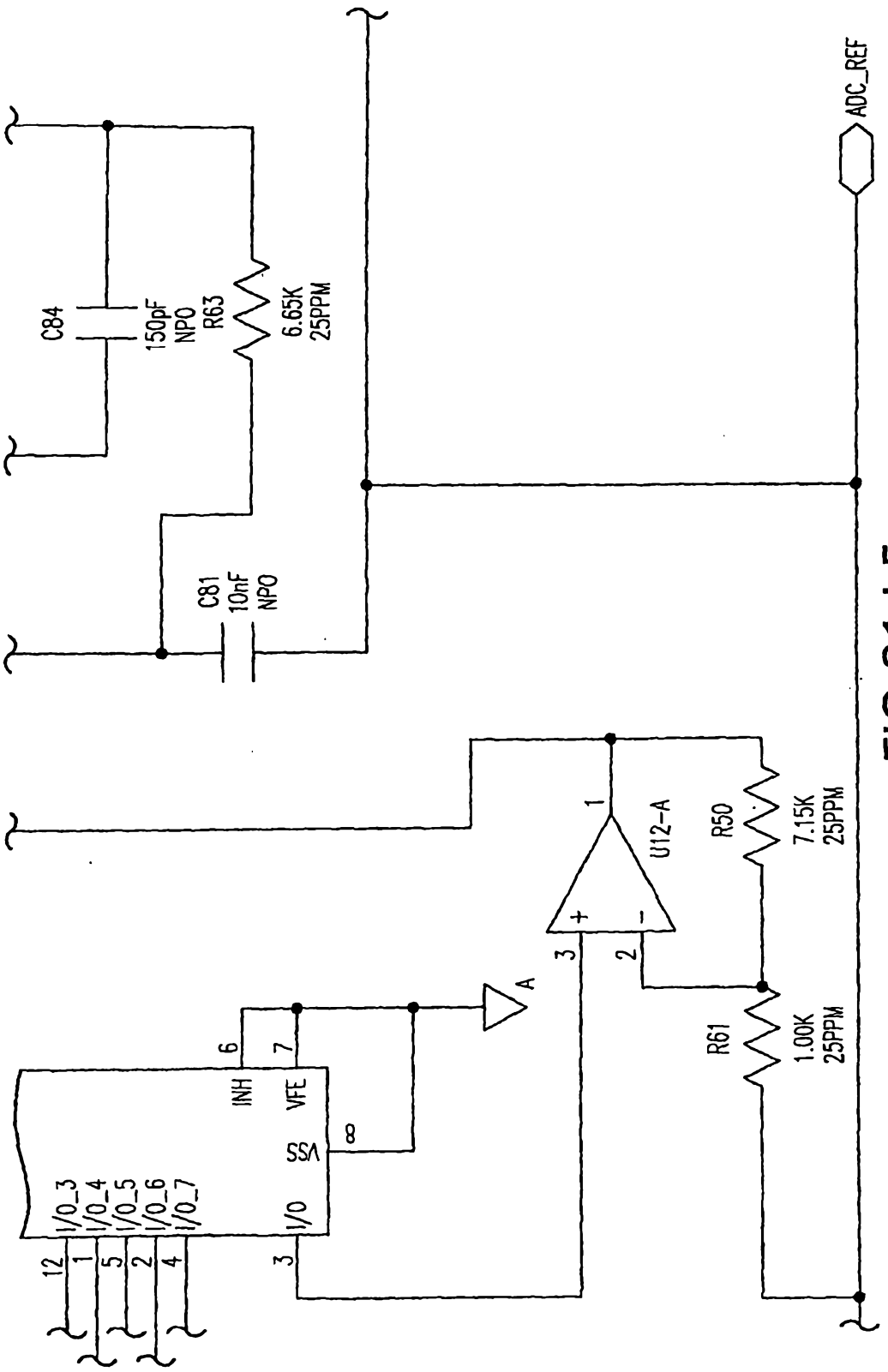

ADC_REF
C84
150pF
NPO
R63
6.65K
25PPM
C81
10nF
NPO
U12-A
R50
7.15K
25PPM
R61
1.00K
25PPM
A
INH
VFE
SSA
I/O
I/O_3
I/O_4
I/O_5
I/O_6
I/O_7

FIG.31J-5

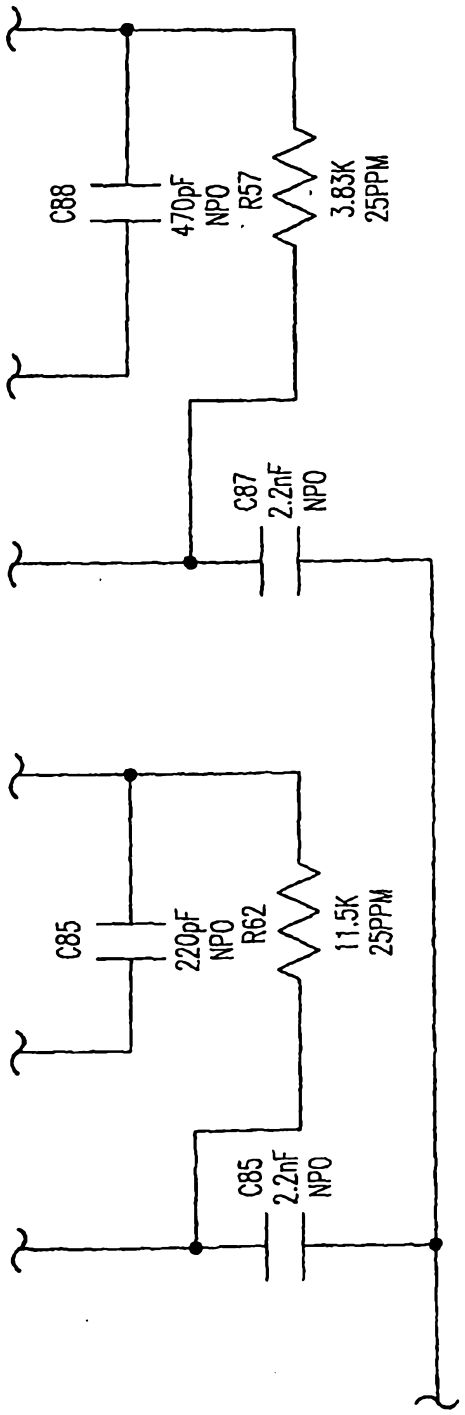

C88
470pF
NPO
R57
3.83K
25PPM
C87
2.2nF
NPO
C85
220pF
NPO
R62
11.5K
25PPM
C85
2.2nF
NPO

FIG.31J-6

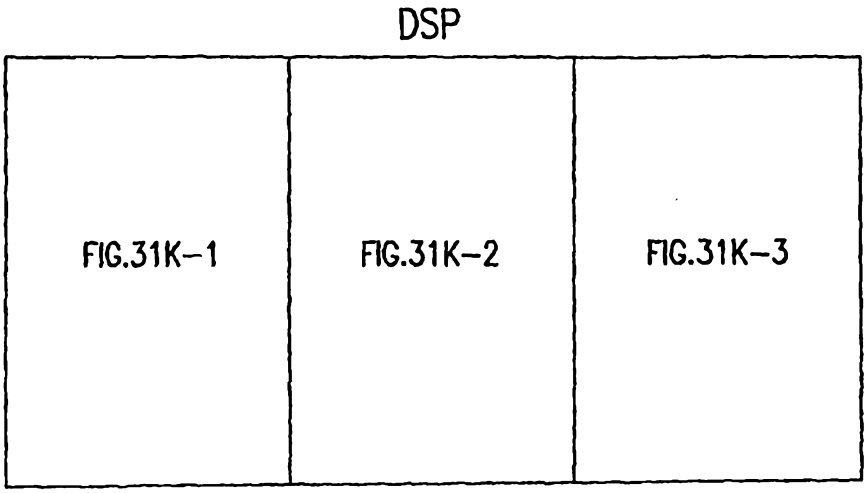
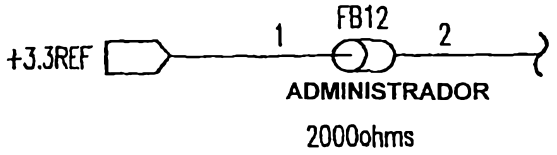


FIG.31K



PORTA JTAG

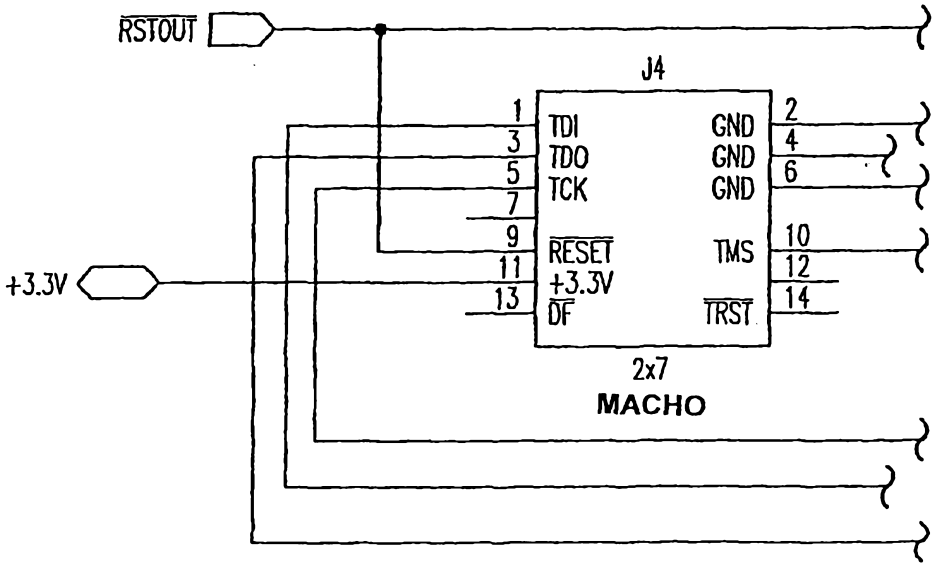


FIG.31K-1

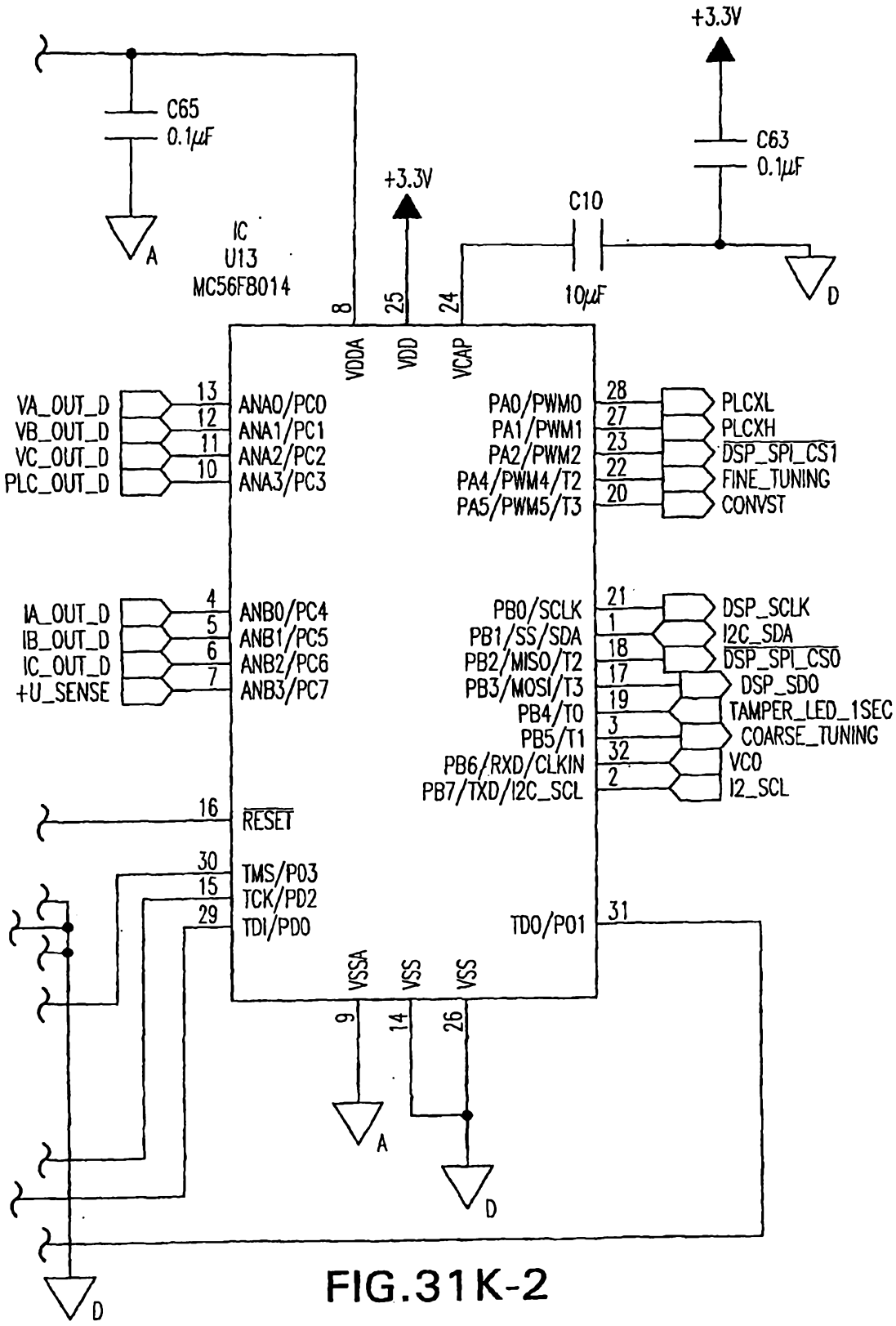

+3.3V
C65
0.1μF
C63
0.1μF
+3.3V
C10
IC
U13
MC56F8014
10μF
A
D
8
25
24
VDDA
VDD
VCAP
VA_OUT_D
VB_OUT_D
VC_OUT_D
PLC_OUT_D
13
12
11
10
ANA0/PC0
ANA1/PC1
ANA2/PC2
ANA3/PC3
PA0/PWM0
PA1/PWM1
PA2/PWM2
PA4/PWM4/T2
PA5/PWM5/T3
28
27
23
22
20
PLCXL
PLCXH
DSP_SPI_CS1
FINE_TUNING
CONVST
IA_OUT_D
IB_OUT_D
IC_OUT_D
+U_SENSE
4
5
6
7
ANB0/PC4
ANB1/PC5
ANB2/PC6
ANB3/PC7
PB0/SCLK
PB1/SS/SDA
PB2/MISO/T2
PB3/MOSI/T3
PB4/T0
PB5/T1
PB6/RXD/CLKIN
PB7/TXD/I2C_SCL
21
1
18
17
19
3
32
2
DSP_SCLK
I2C_SDA
DSP_SPI_CS0
DSP_SDO
TAMPER_LED_1SEC
COARSE_TUNING
VCO
I2_SCL
16
RESET
30
15
29
TMS/P03
TCK/PD2
TDI/PD0
TDO/P01
31
VSSA
VSS
VSS
9
14
26
A
D
D

FIG.31K-2

CONECTADO A ADC DE DSP PARA SENTIR NÍVEIS DE FONTE DE ENERGIA DE ENTRADA

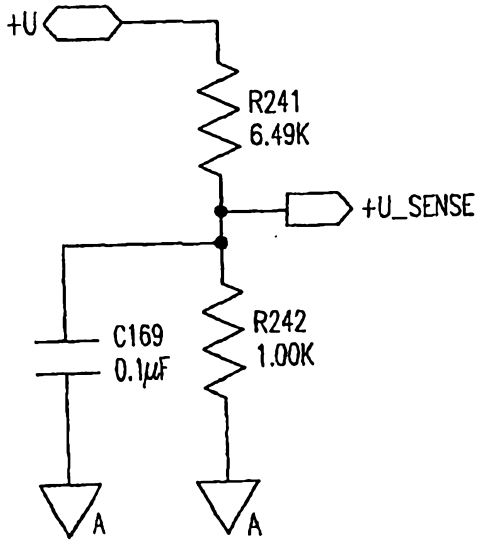


FIG.31K-3

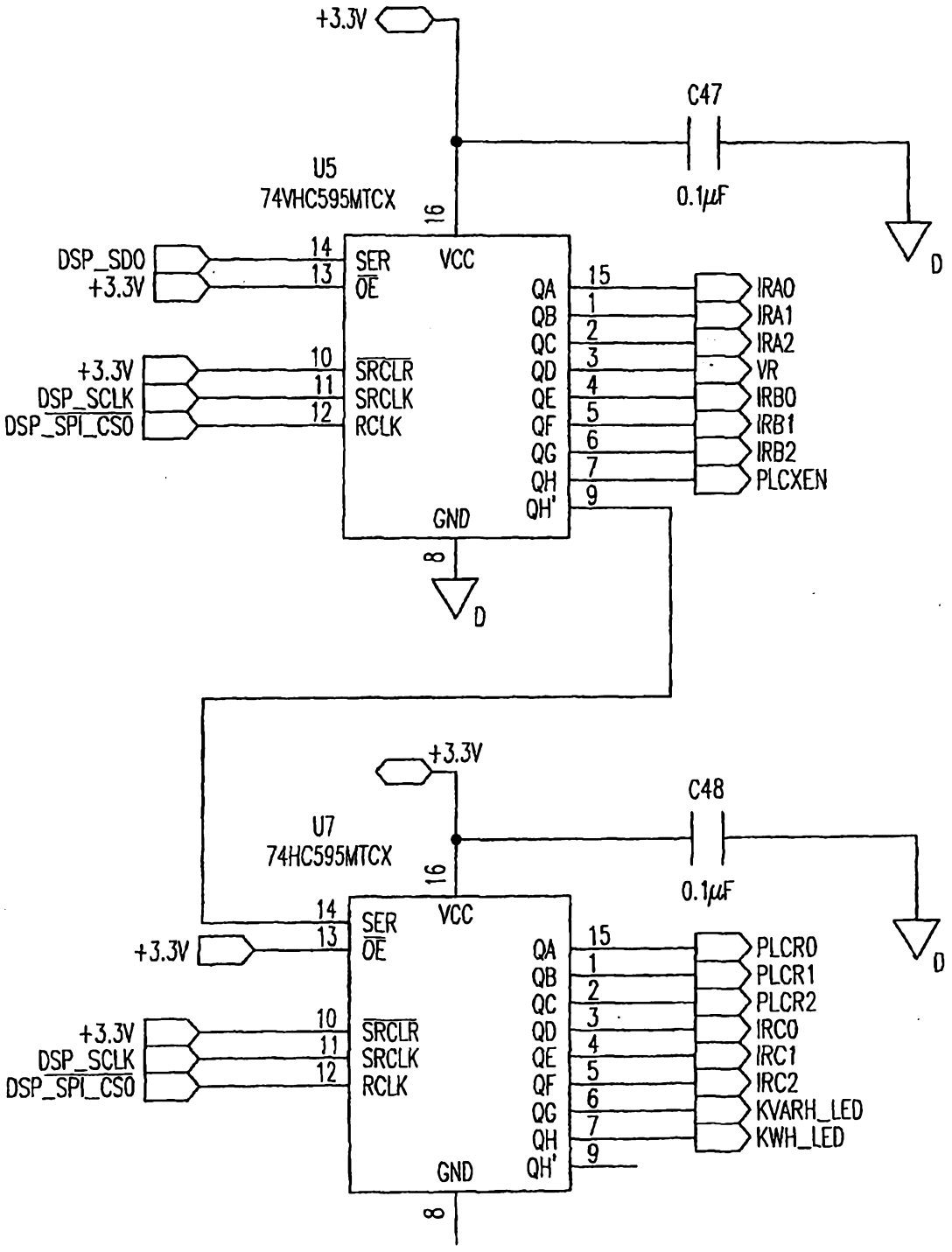

+3.3V
C47
0.1µF
U5
74VHC595MTCX
16
VCC
14
SER
13
OE
DSP_SDO
+3.3V
10
SRCLR
11
SRCLK
12
RCLK
+3.3V
DSP_SCLK
DSP_SPI_CS0
GND
8
QA
QB
QC
QD
QE
QF
QG
QH
QH'
15
1
2
3
4
5
6
7
9
IRA0
IRA1
IRA2
VR
IRB0
IRB1
IRB2
PLCXEN
D
+3.3V
C48
0.1µF
U7
74HC595MTCX
16
VCC
14
SER
13
OE
+3.3V
10
SRCLR
11
SRCLK
12
RCLK
+3.3V
DSP_SCLK
DSP_SPI_CS0
GND
8
QA
QB
QC
QD
QE
QF
QG
QH
QH'
15
1
2
3
4
5
6
7
9
PLCR0
PLCR1
PLCR2
IRC0
IRC1
IRC2
KVARH_LED
KWH_LED
D
IO EXP

FIG.31L

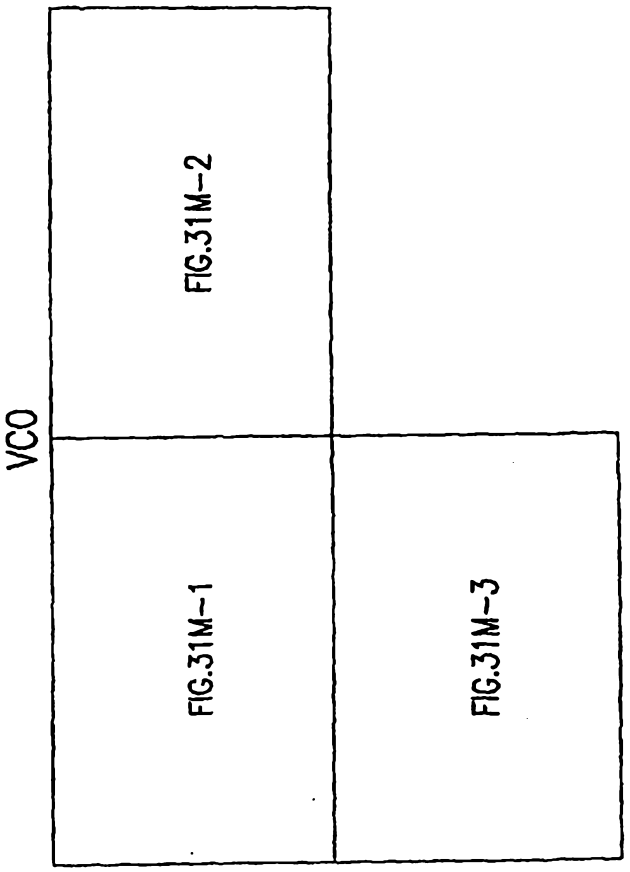


FIG.31M

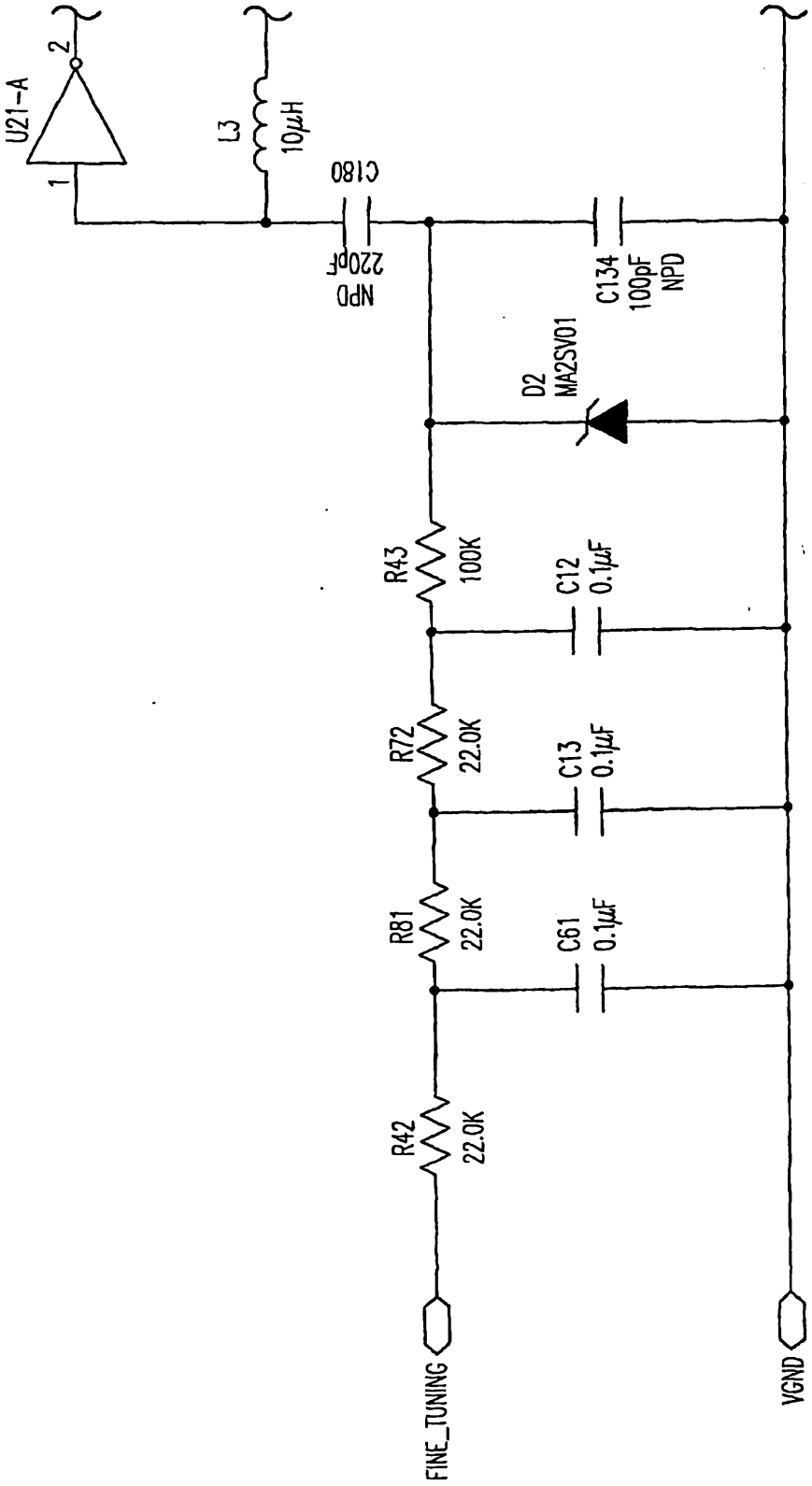

U21-A
2
1
L3
10μH
C180
220pF
NPD
C134
100pF
NPD
D2
MA2SV01
R43
100K
C12
0.1μF
R72
22.0K
C13
0.1μF
R81
22.0K
C61
0.1μF
R42
22.0K
FINE_TUNING
VGND

FIG.31M-1

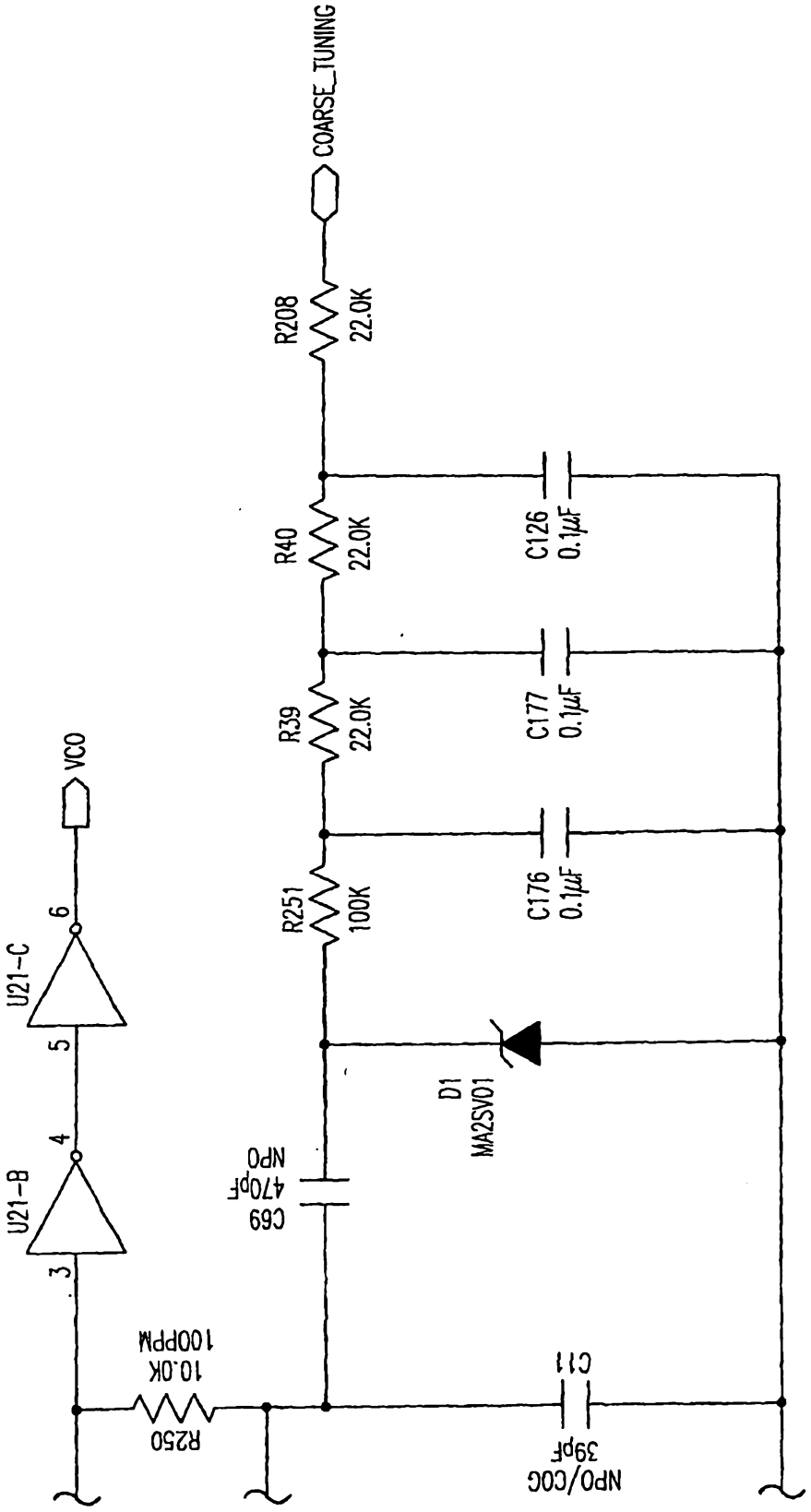

COARSE_TUNING
R208
22.0K
R40
22.0K
R39
22.0K
R251
100K
C126
0.1µF
C177
0.1µF
C176
0.1µF
VCO
U21-C
U21-B
6
5
4
3
D1
MA2SV01
C69
470pF
NPO
R250
10.0K
100PPM
C11
39pF
NP0/C0G

FIG.31M-2

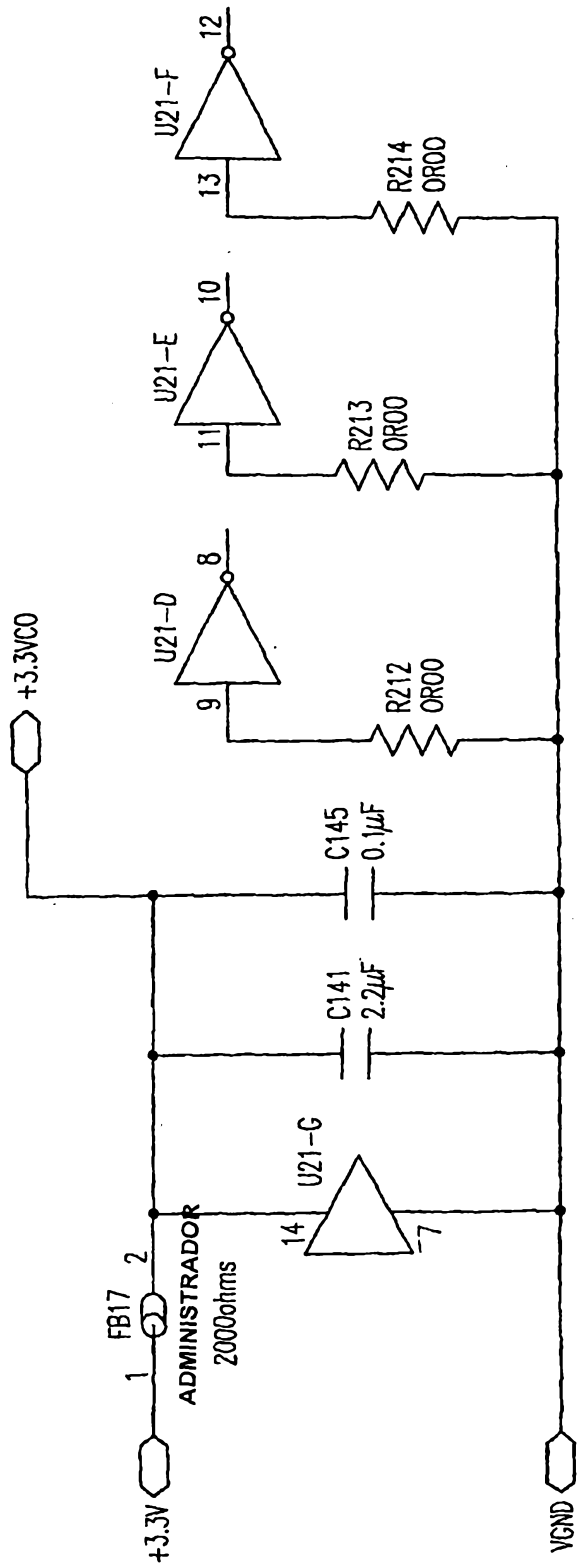

+3.3VCO
U21-D
8
9
R212
OR00
U21-E
10
11
R213
OR00
U21-F
12
13
R214
OR00
C145
0.1μF
C141
2.2μF
U21-G
14
7
FB17
1
2
ADMINISTRADOR
2000ohms
+3.3V
VGND

FIG.31M-3

PLC SCH202-3

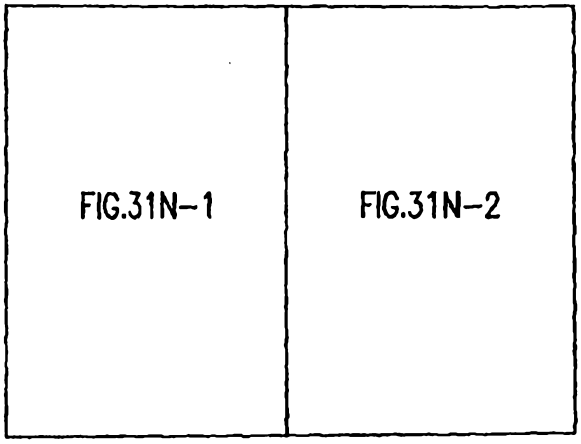
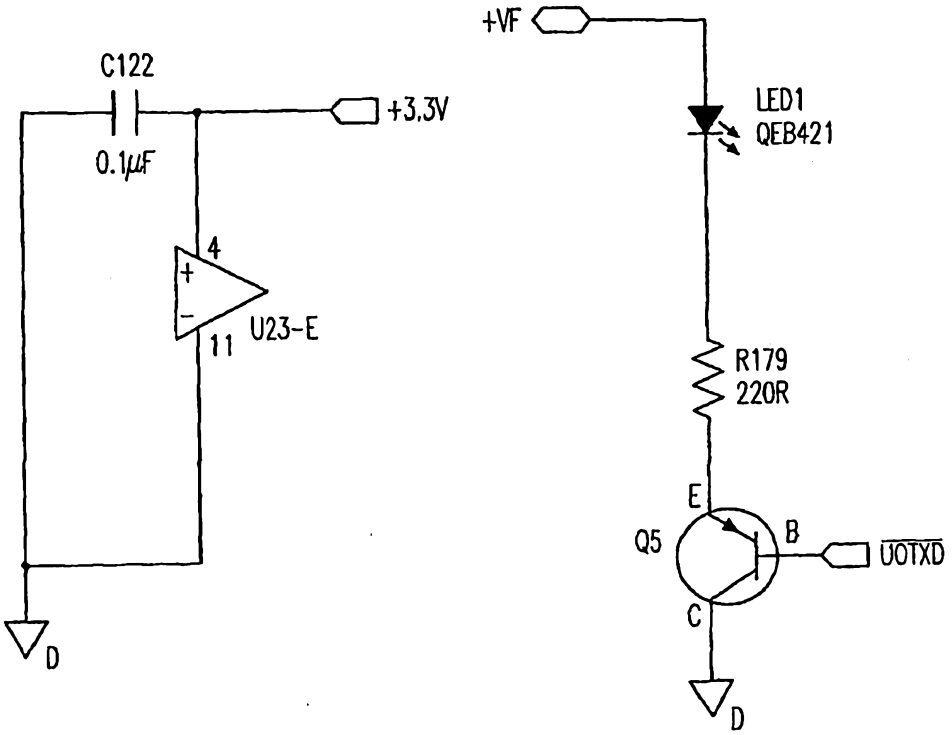


FIG.31N


C122
0.1μF
+3.3V
4
11
U23-E
D
+VF
LED1
QEB421
R179
220R
E
B
C
Q5
U0TXD
D

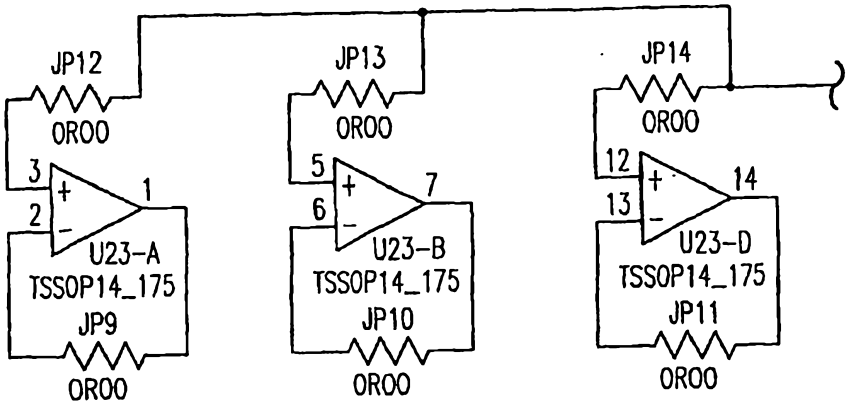

JP12
0R00
3
2
1
U23-A
TSSOP14_175
JP9
0R00
JP13
0R00
5
6
7
U23-B
TSSOP14_175
JP10
0R00
JP14
0R00
12
13
14
U23-D
TSSOP14_175
JP11
0R00

FIG.31N-1

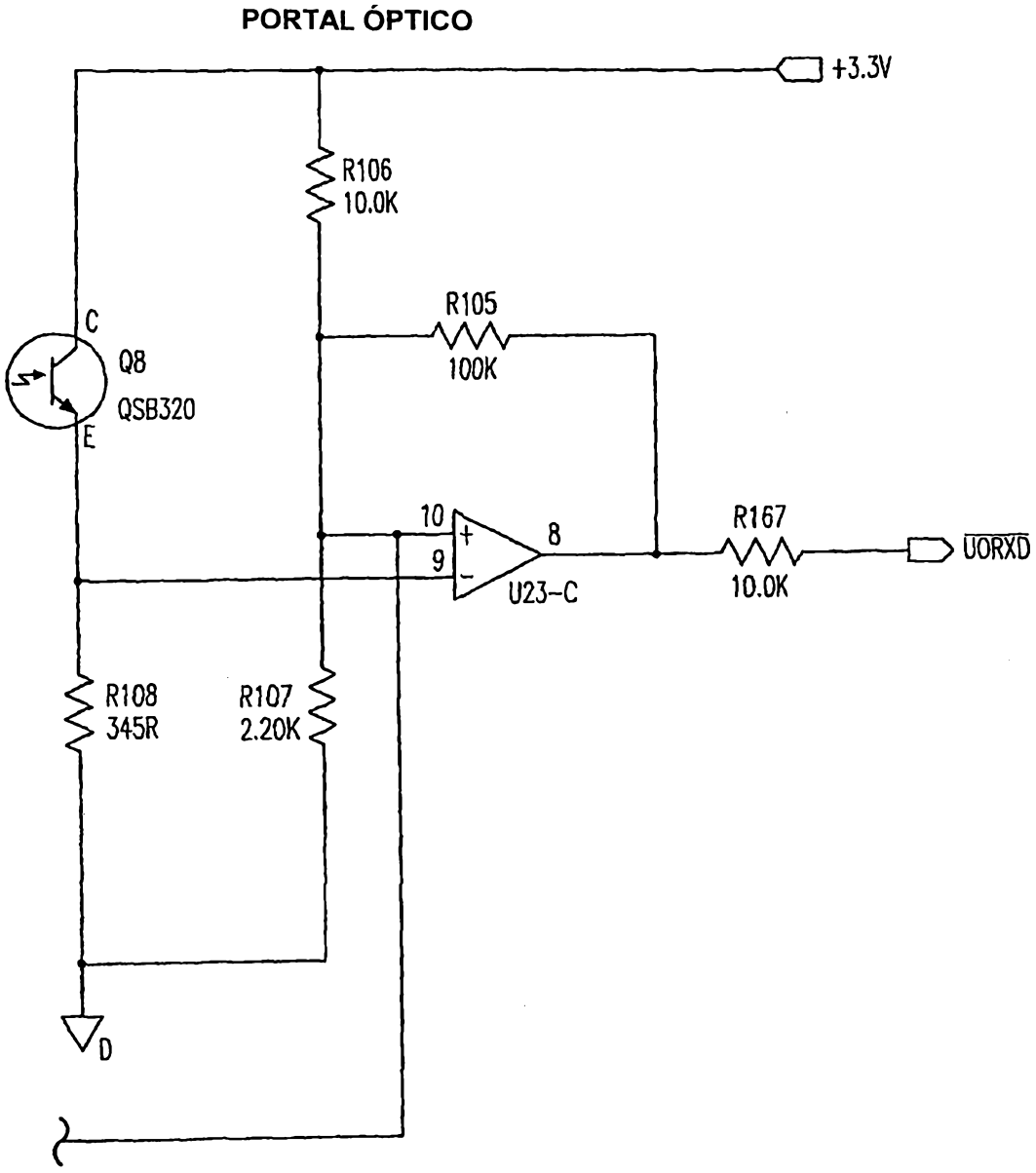

PORTAL ÓPTICO
+3.3V
R106
10.0K
R105
100K
C
Q8
QSB320
E
10
9
8
U23-C
R167
10.0K
UORXD
R108
345R
R107
2.20K
D

FIG.31N-2

I/O SERIAL SCH202-3

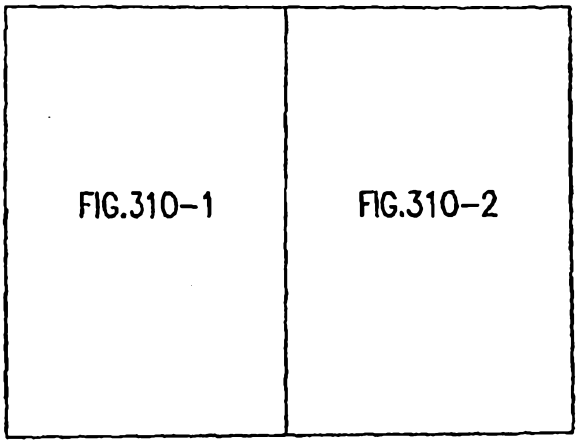
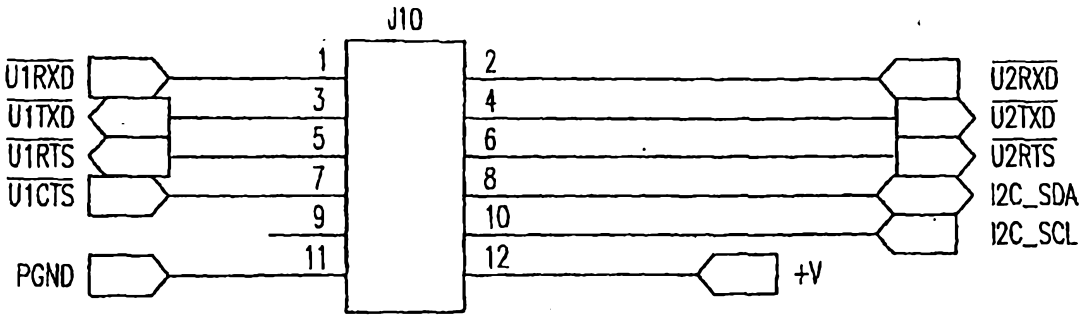


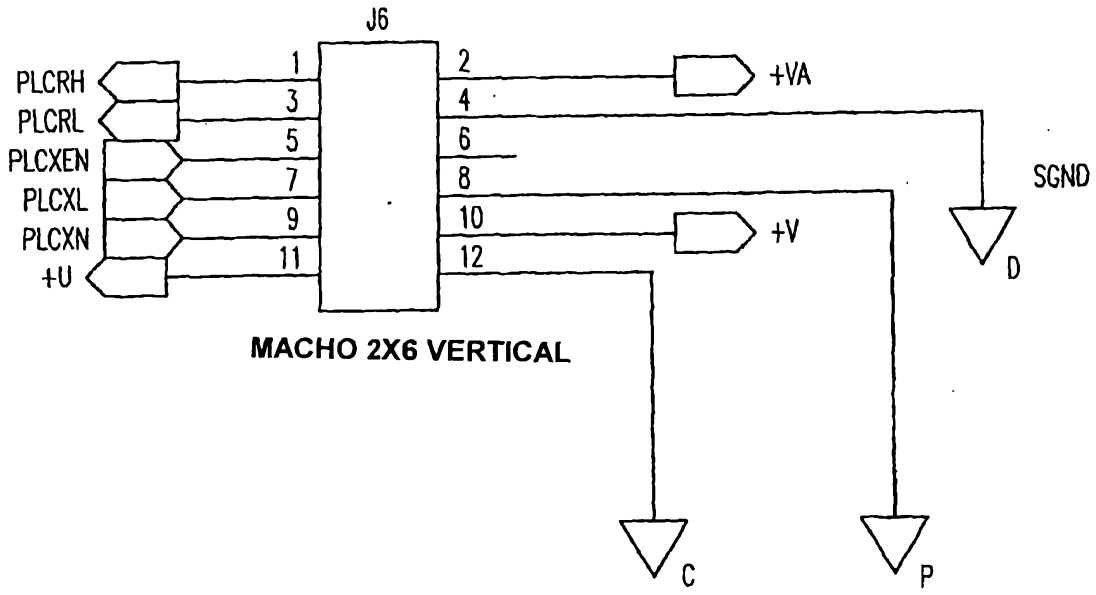
FIG.310

COLETOR PARA PLACA DE IO



MACHO 2X6 VERTICAL

COLETOR PARA PLACA DE POTÊNCIA

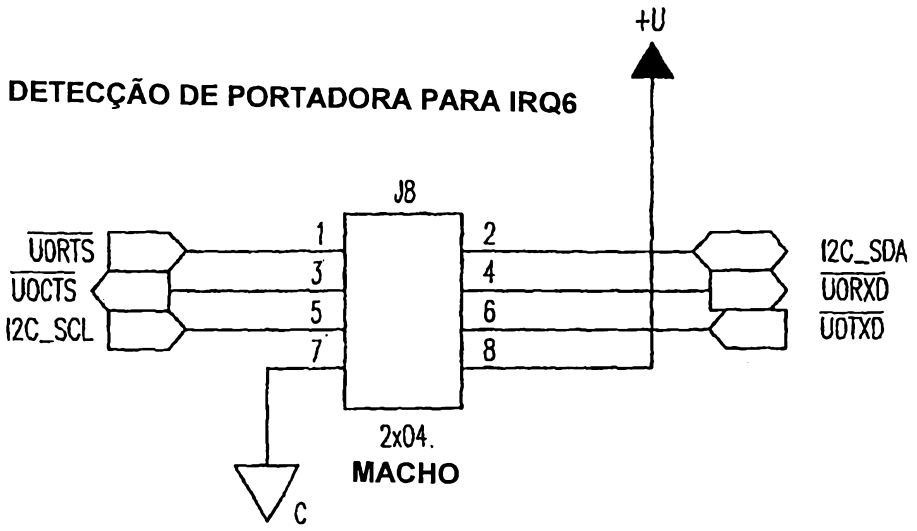


MACHO 2X6 VERTICAL

FIG.310-1

COLETOR PARA PLACA DE MODEM

DETECÇÃO DE PORTADORA PARA IRQ6



COLETOR PARA PLACA DE MOSTRADOR

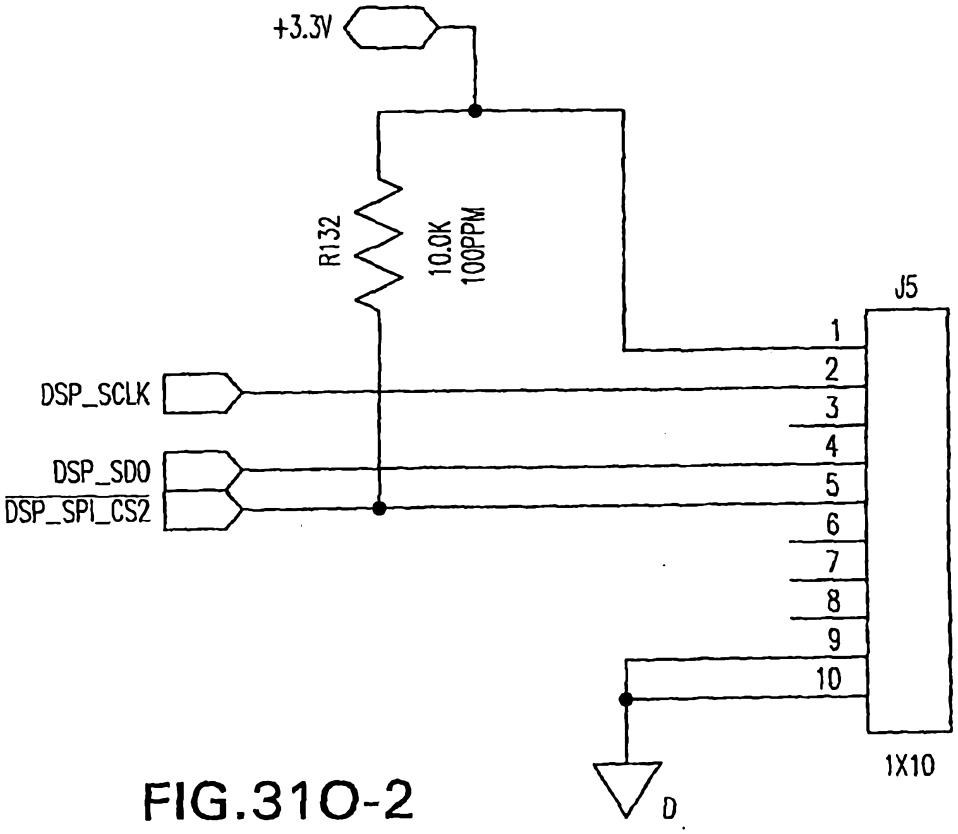


FIG.310-2

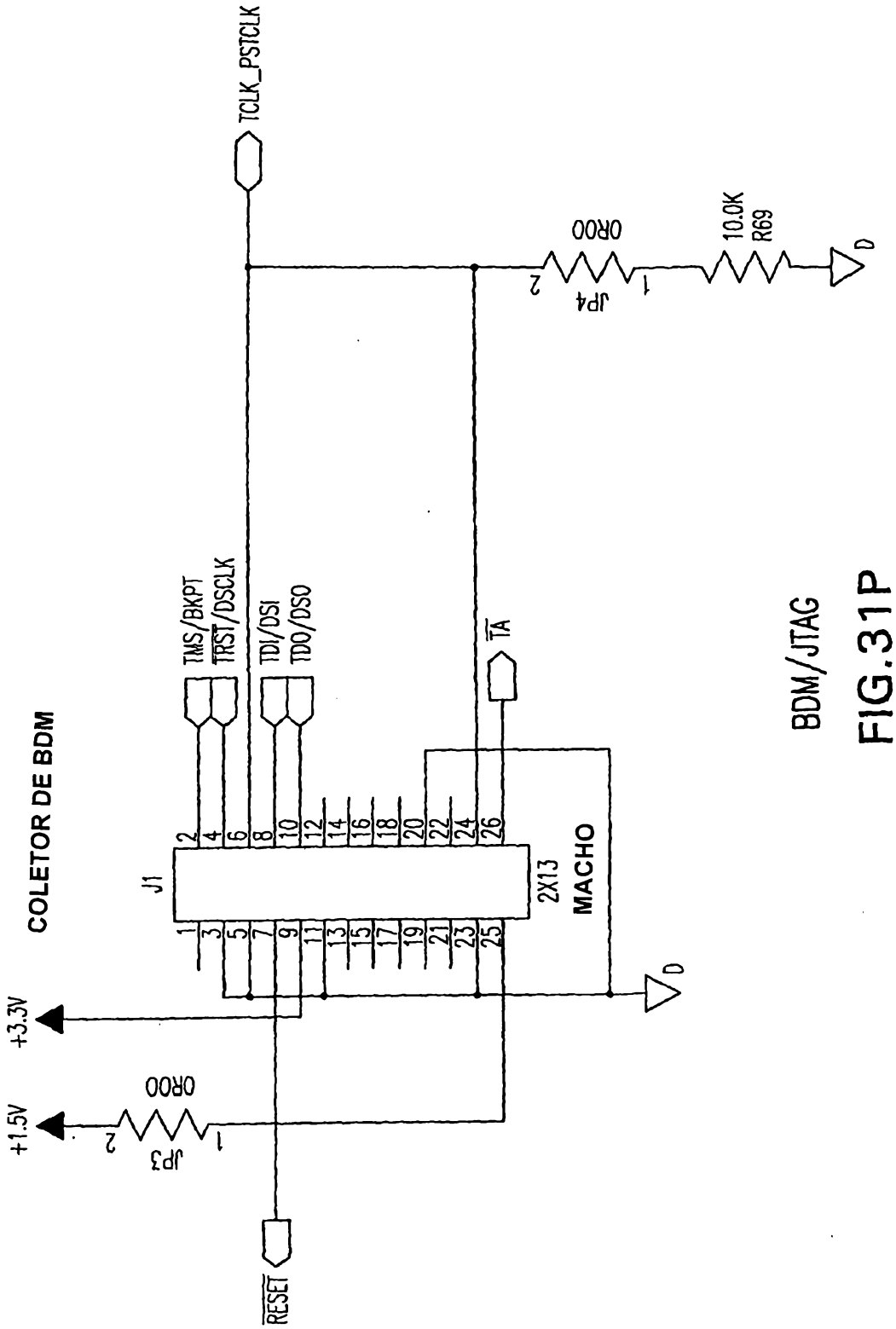

TCLK_PSTCLK
0R00
JP4
10.0K
R69
D
TMS/BKPT
TRST/DSCLK
TDI/DSI
TDO/DSO
TA
COLETOR DE BDM
J1
2X13
MACHO
+3.3V
+1.5V
0R00
JP3
RESET

FIG.31P

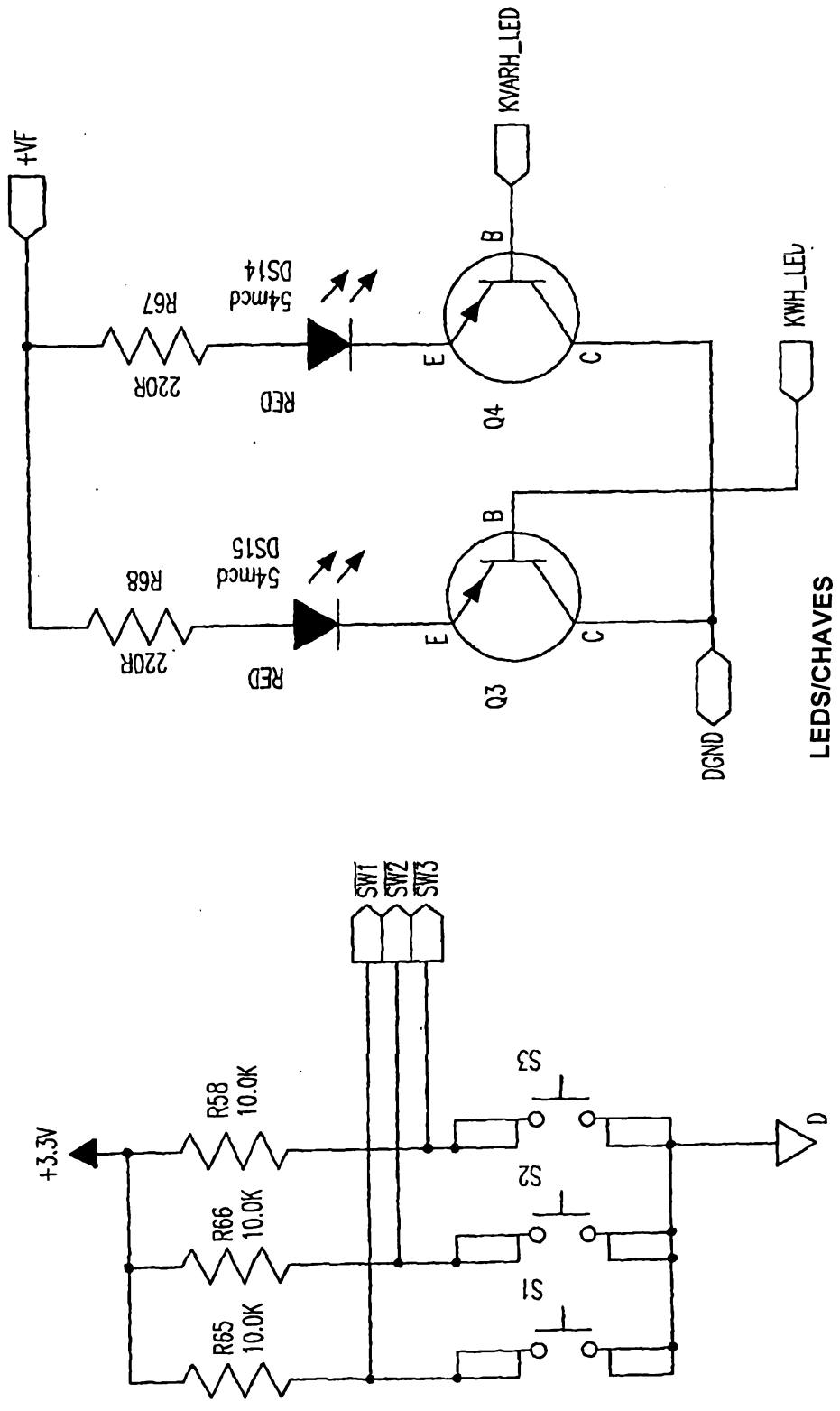

+VF
KVARH_LED
KWH_LED
DS14
54mcd
R67
220R
RED
Q4
DS15
54mcd
R68
220R
RED
Q3
DGND
LEDS/CHAVES
SW1
SW2
SW3
+3.3V
R58
10.0K
R66
10.0K
R65
10.0K
S3
S2
S1
D

FIG.31Q

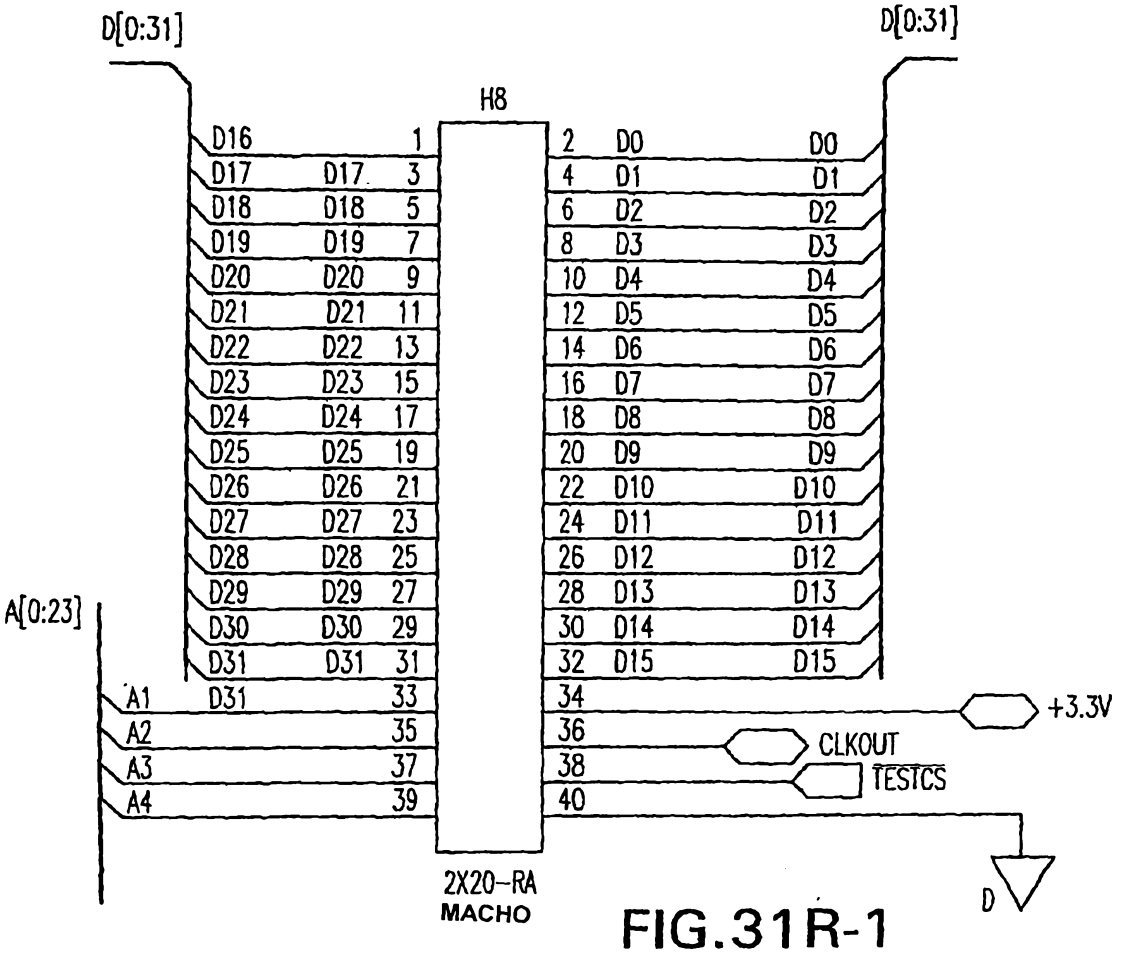


FIG.31R-1

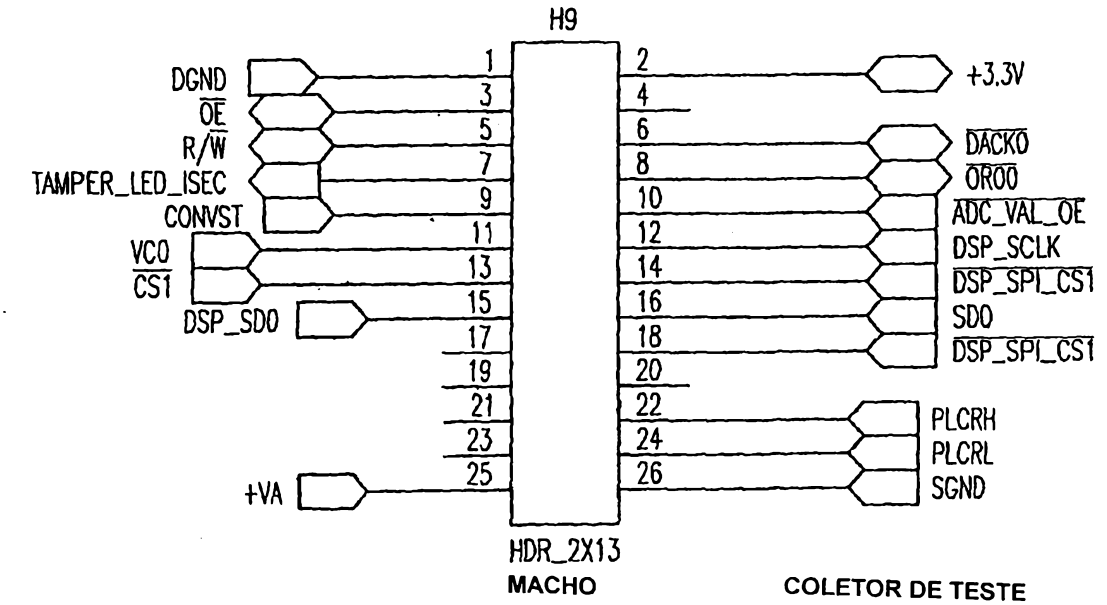


FIG.31R-2

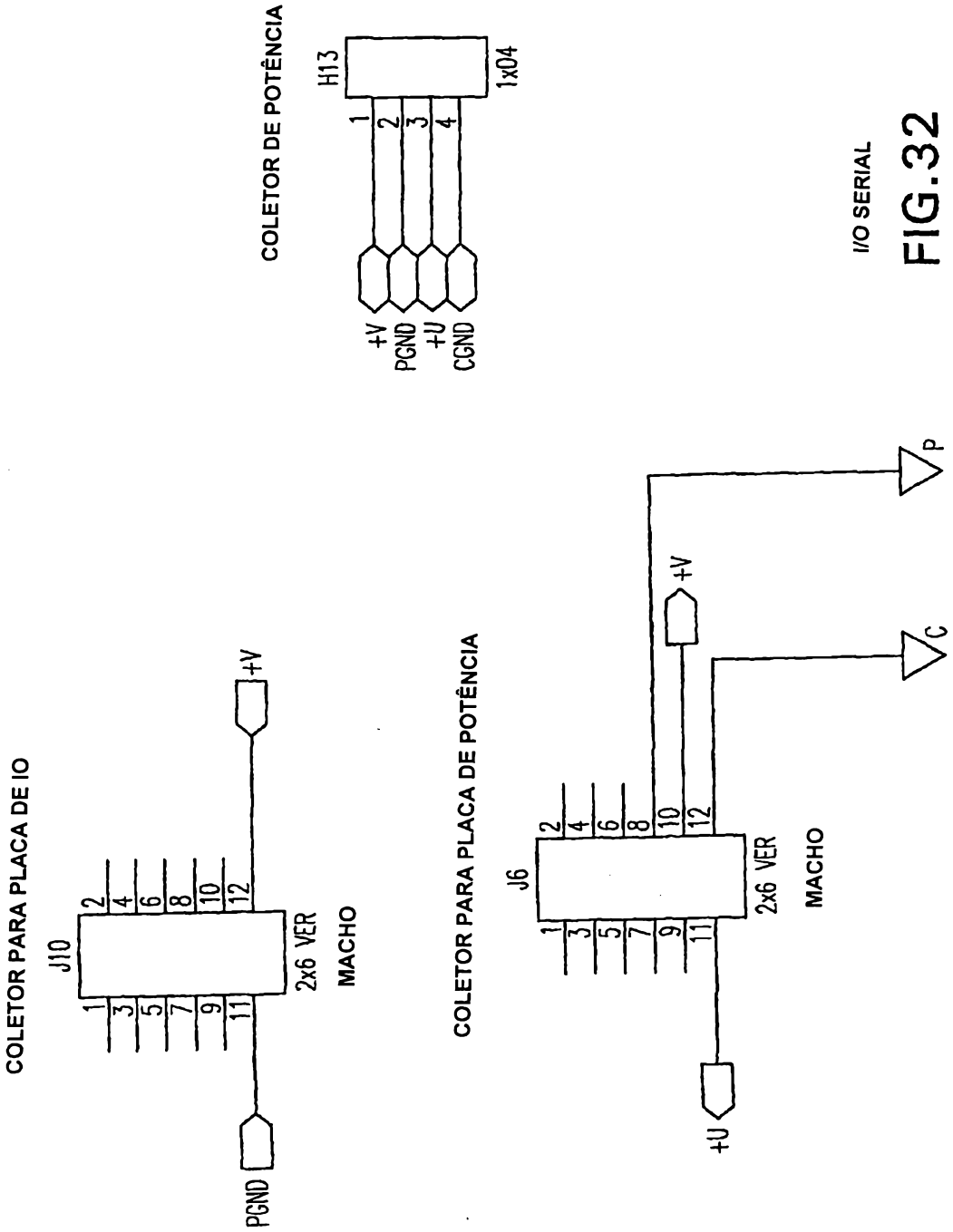

COLETOR DE POTÊNCIA
H13
1x04
1
2
3
4
+V
PGND
+U
CGND
COLETOR PARA PLACA DE IO
J10
2x6 VER
MACHO
1
3
5
7
9
11
2
4
6
8
10
12
+V
PGND
COLETOR PARA PLACA DE POTÊNCIA
J6
2x6 VER
MACHO
1
3
5
7
9
11
2
4
6
8
10
12
+V
+U
P
C
I/O SERIAL

FIG.32

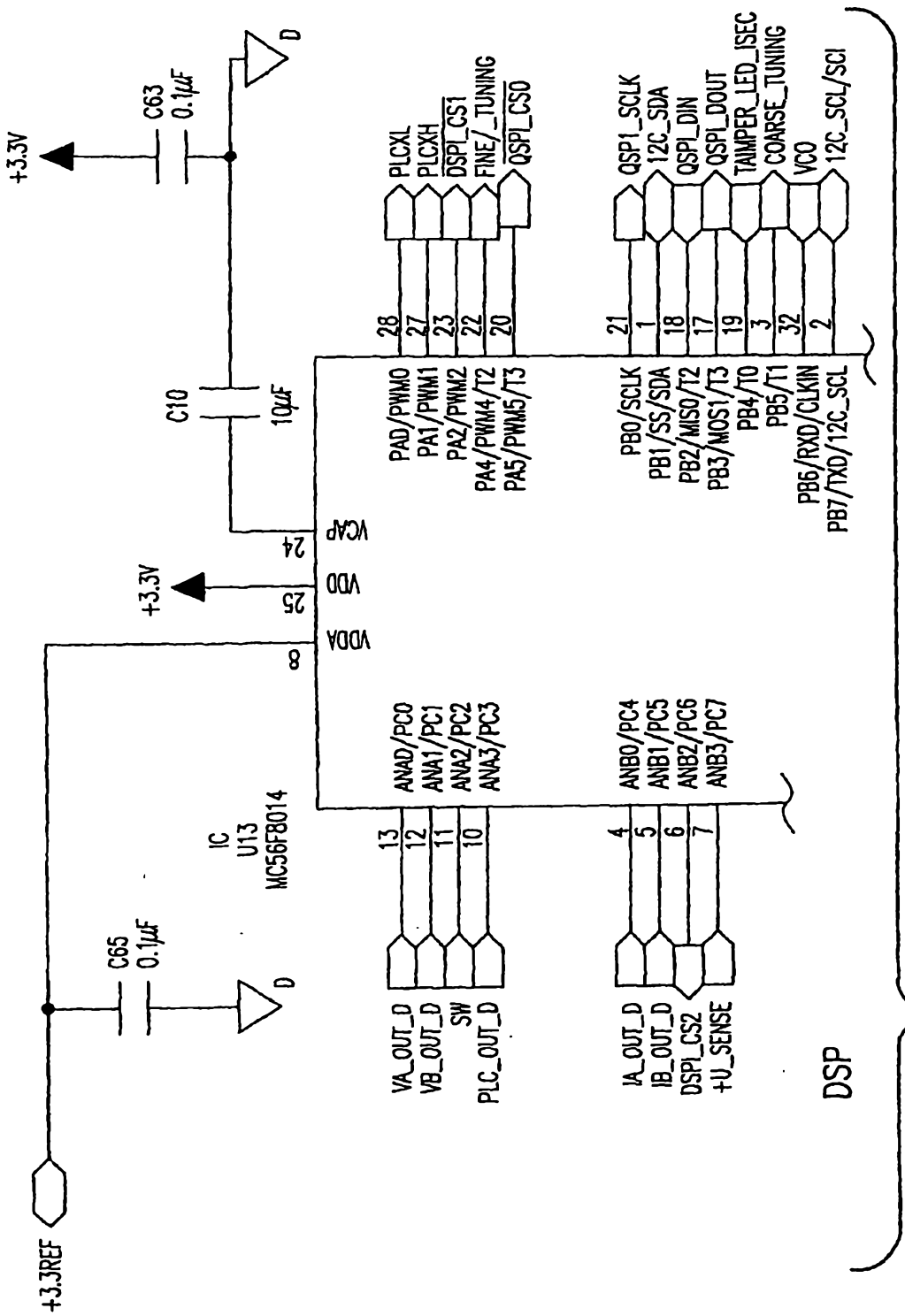

+3.3V
C63
0.1µF
D
C10
10µF
+3.3V
C65
0.1µF
D
IC
U13
MC56F8014
+3.3REF
VCAP
VDD
VDDA
24
25
8
PA0/PWM0
PA1/PWM1
PA2/PWM2
PA4/PWM4/T2
PA5/PWM5/T3
PB0/SCLK
PB1/SS/SDA
PB2/MISO/T2
PB3/MOSI/T3
PB4/T0
PB5/T1
PB6/RXD/CLKIN
PB7/TXD/12C_SCL
ANA0/PC0
ANA1/PC1
ANA2/PC2
ANA3/PC3
ANB0/PC4
ANB1/PC5
ANB2/PC6
ANB3/PC7
28
27
23
22
20
21
1
18
17
19
3
32
2
13
12
11
10
4
5
6
7
PLCXL
PLCXH
DSPI_CS1
FINE/_TUNING
QSPI_CS0
QSPI_SCLK
I2C_SDA
QSPI_DIN
QSPI_DOUT
TAMPER_LED_ISEC
COARSE_TUNING
VCO
I2C_SCL/SCI
VA_OUT_D
VB_OUT_D
SW
PLC_OUT_D
IA_OUT_D
IB_OUT_D
DSPI_CS2
+U_SENSE
DSP
Cont.
Fig.33A-2

FIG.33A-1

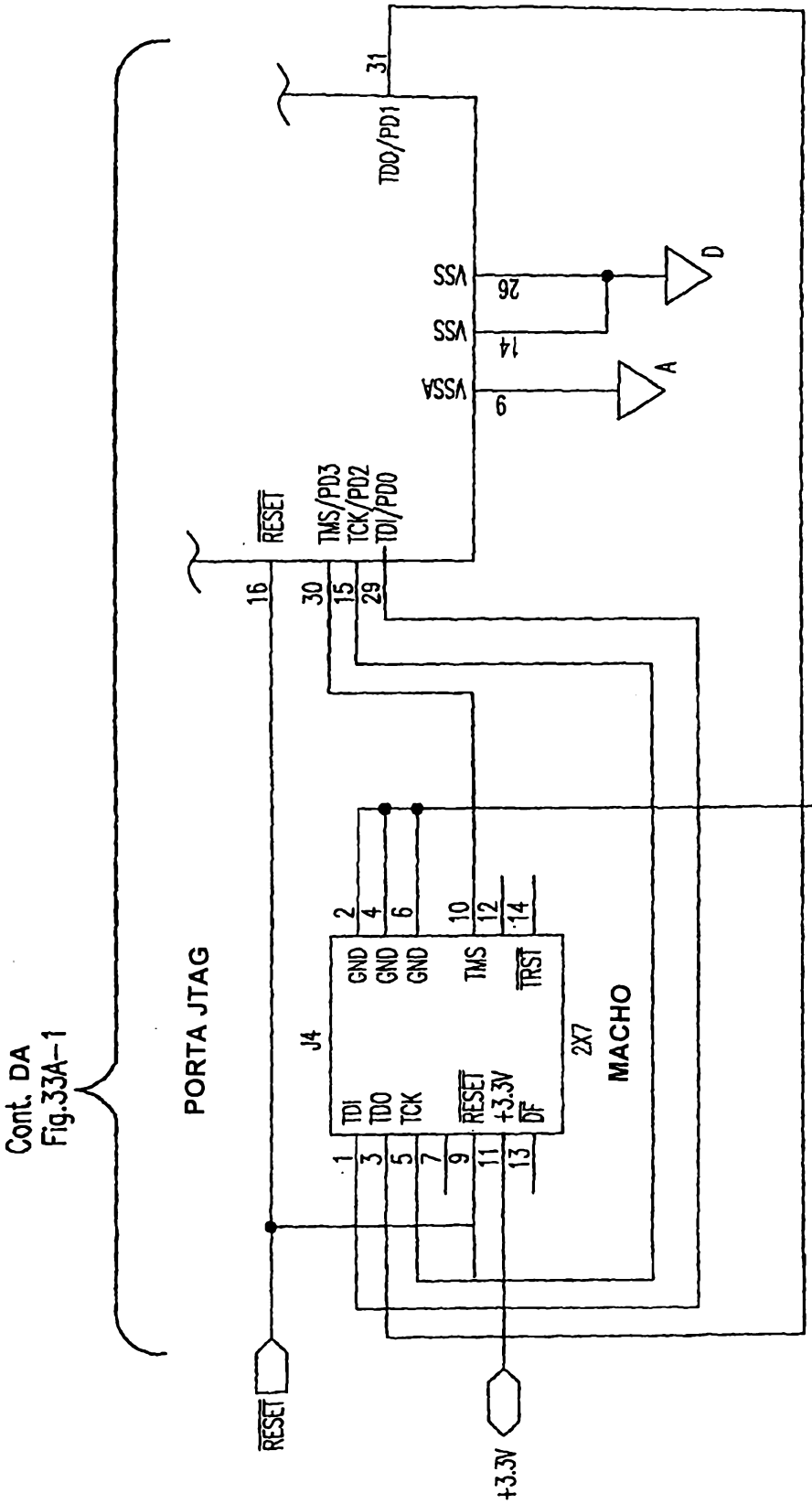

Cont. DA
Fig.33A-1
PORTA JTAG
J4
2X7
MACHO
TDI
TDO
TCK
RESET
+3.3V
DF
GND
GND
GND
TMS
TRST
1
3
5
7
9
11
13
2
4
6
10
12
14
RESET
+3.3V
16
30
15
29
RESET
TMS/PD3
TCK/PD2
TDI/PD0
TDO/PD1
31
VSS
26
VSS
14
VSSA
9
D
A
D
DSP

FIG.33A-2

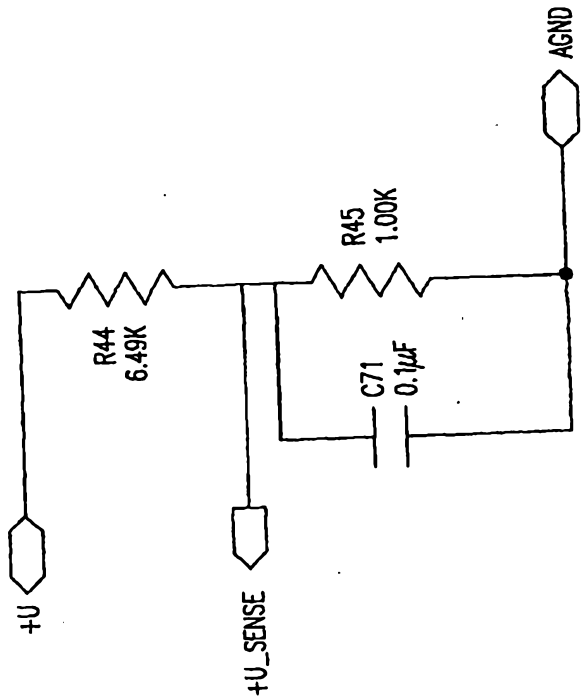


FIG.33A-3

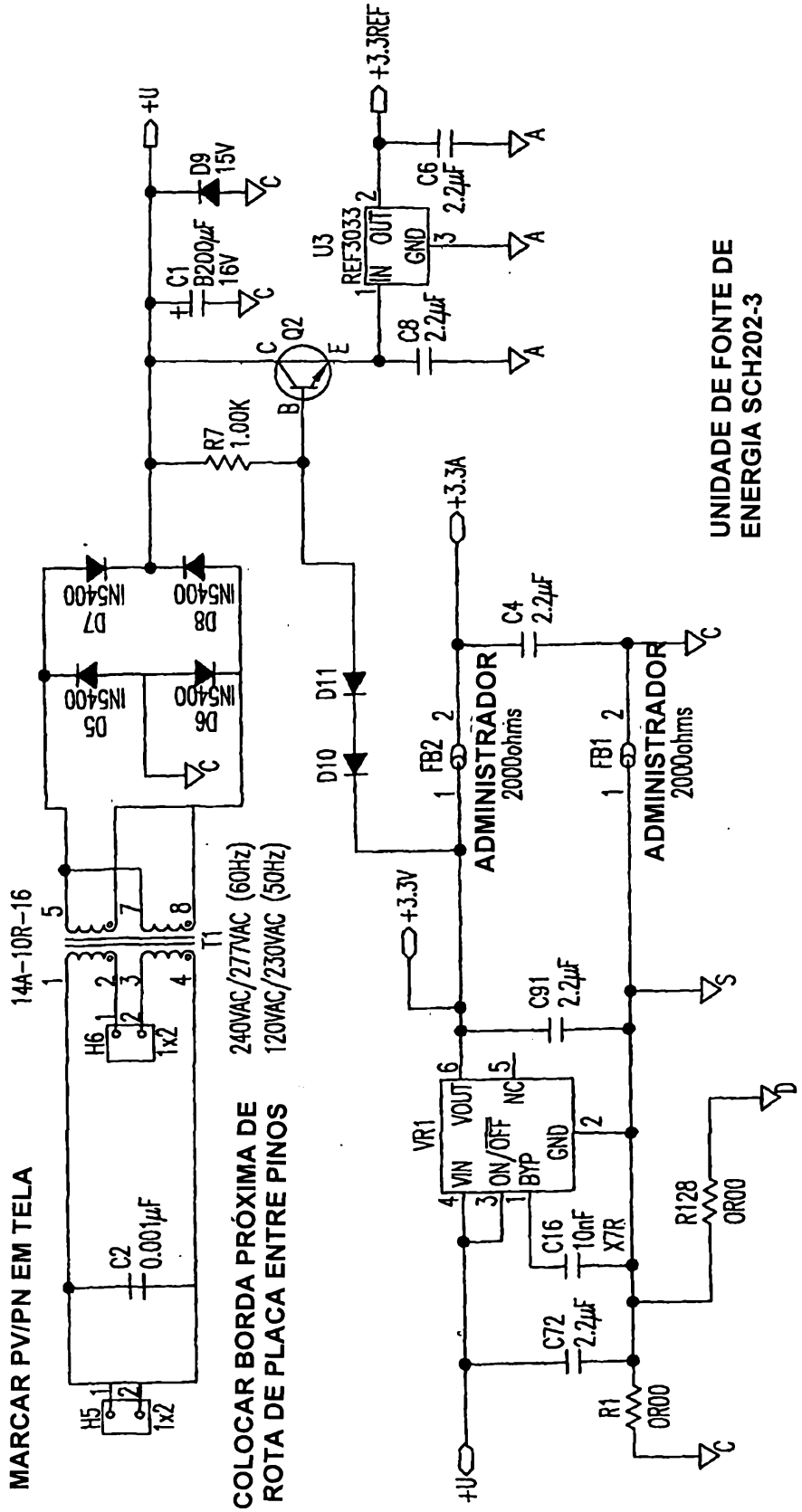
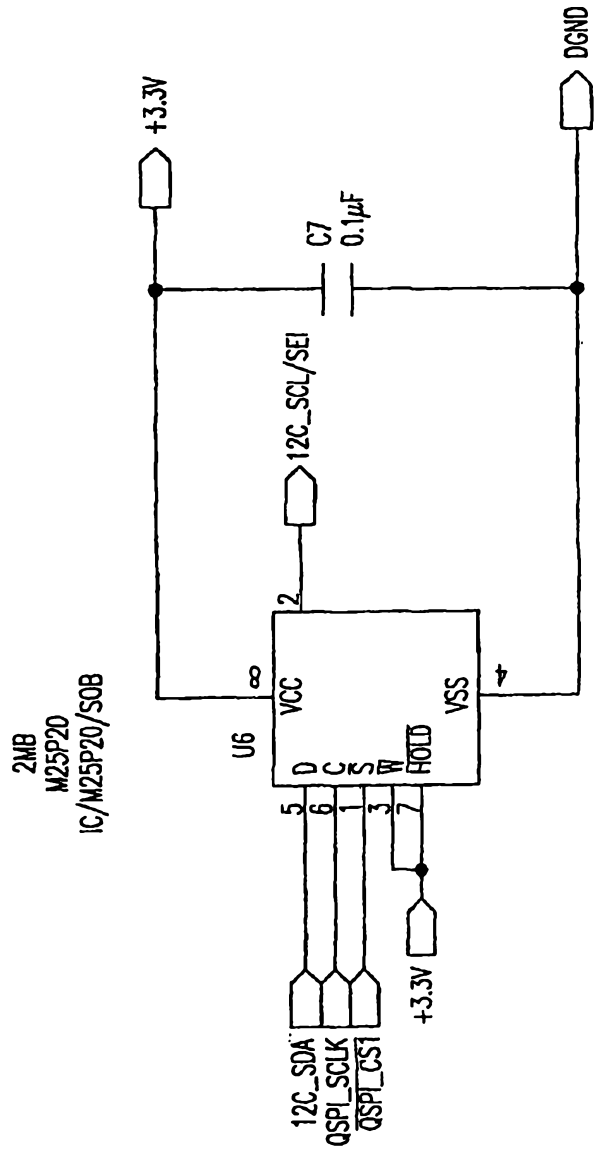

UNIDADE DE FONTE DE ENERGIA SCH202-3
MARCAR PV/PN EM TELA
COLOCAR BORDA PRÓXIMA DE ROTA DE PLACA ENTRE PINOS
14A-10R-16
240VAC/277VAC (60Hz)
120VAC/230VAC (50Hz)
T1
H5
H6
1x2
C2
0.001µF
D5 1N5400
D6 1N5400
D7 1N5400
D8 1N5400
D9 15V
C1 B200µF 16V
+U
R7 1.00K
Q2
U3 REF3033
IN OUT GND
+3.3REF
C6 2.2µF
C8 2.2µF
D10
D11
+3.3A
+3.3V
FB2
FB1
ADMINISTRADOR 2000ohms
C4 2.2µF
C91 2.2µF
VR1
VIN VOUT
ON/OFF BYP
NC GND
C16 10nF X7R
C72 2.2µF
R1 0R00
R128 0R00

FIG.33B



MEMÓRIA FLASH

FIG.33C

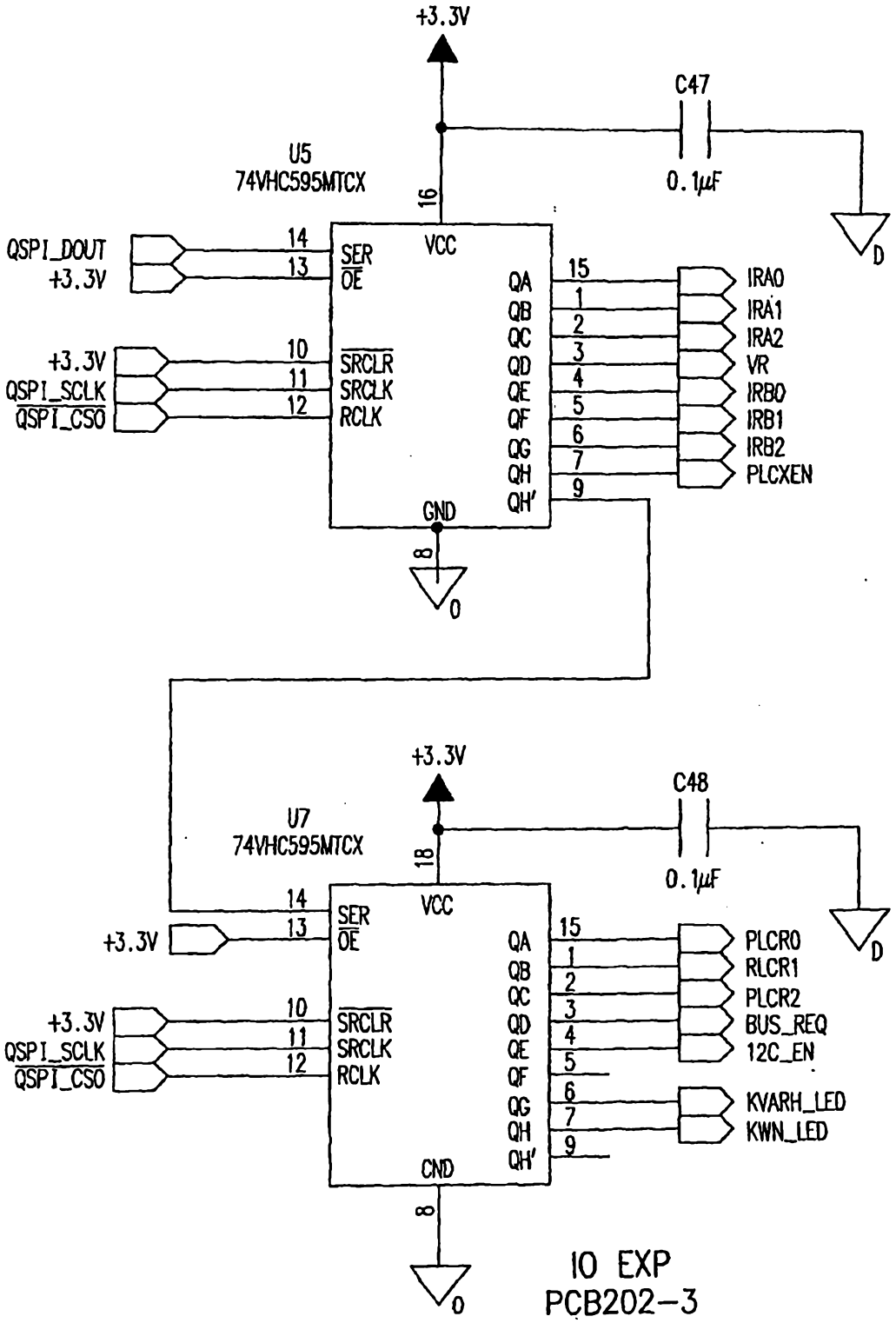

+3.3V
C47
0.1μF
U5
74VHC595MTCX
16
VCC
QSPI_DOUT
+3.3V
14
13
SER
OE
+3.3V
QSPI_SCLK
QSPI_CS0
10
11
12
SRCLR
SRCLK
RCLK
QA
QB
QC
QD
QE
QF
QG
QH
QH'
15
1
2
3
4
5
6
7
9
IRA0
IRA1
IRA2
VR
IRB0
IRB1
IRB2
PLCXEN
GND
8
D
0
+3.3V
C48
0.1μF
U7
74VHC595MTCX
18
VCC
14
13
SER
OE
+3.3V
+3.3V
QSPI_SCLK
QSPI_CS0
10
11
12
SRCLR
SRCLK
RCLK
QA
QB
QC
QD
QE
QF
QG
QH
QH'
15
1
2
3
4
5
6
7
9
PLCR0
RLCR1
PLCR2
BUS_REQ
12C_EN
KVARH_LED
KWN_LED
CND
8
D
0
10 EXP
PCB202-3

FIG.33D

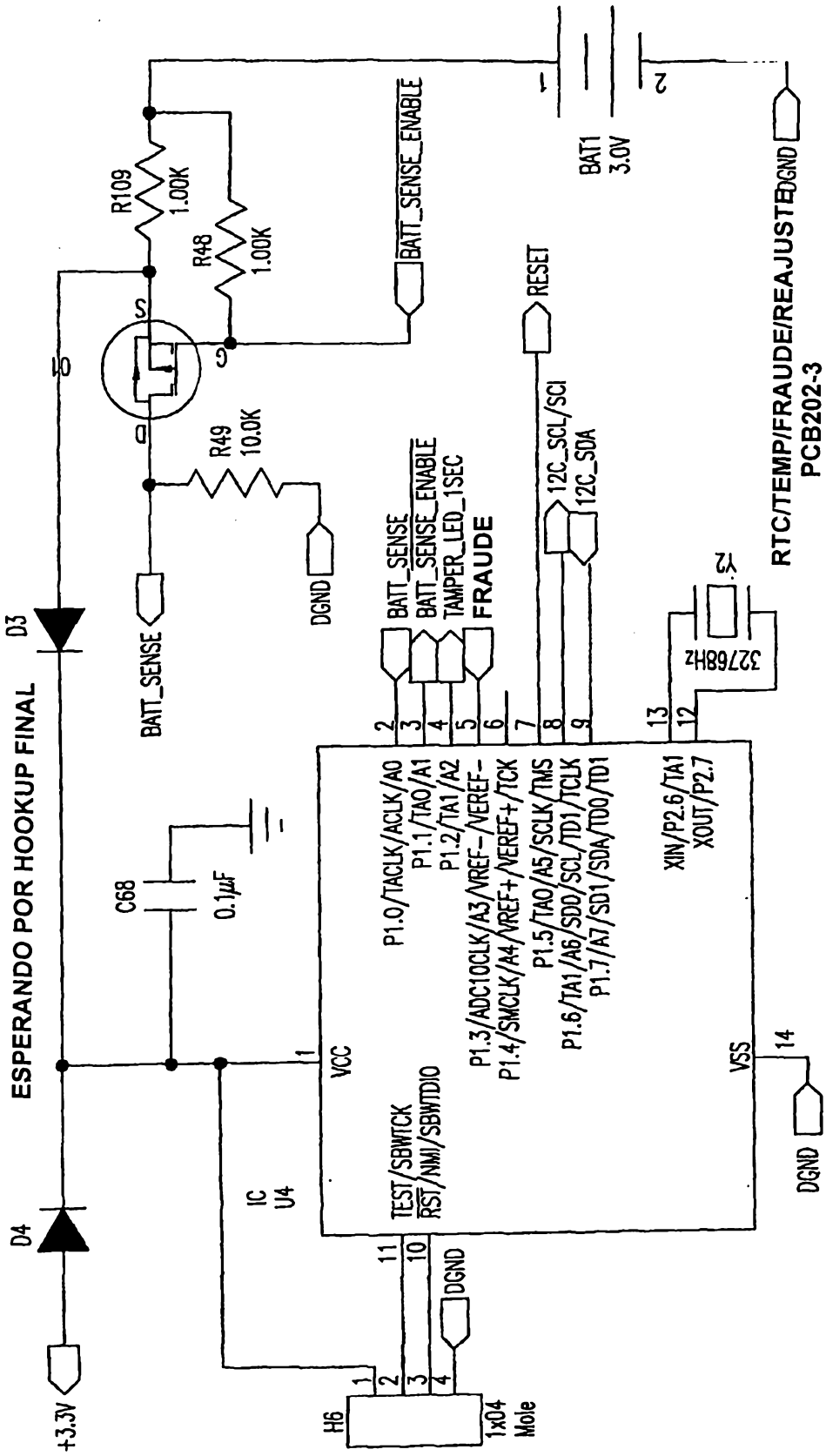

ESPERANDO POR HOOKUP FINAL
D3
D4
+3.3V
BATT_SENSE
R109
1.00K
R48
1.00K
R49
10.0K
DGND
BATT_SENSE_ENABLE
RESET
BAT1
3.0V
RTC/TEMP/FRAUDE/REAJUSTE
PCB202-3
C68
0.1µF
IC
U4
VCC
TEST/SBWTCK
RST/NMI/SBWTDIO
P1.0/TACLK/ACLK/A0
P1.1/TA0/A1
P1.2/TA1/A2
P1.3/ADC10CLK/A3/VREF-/VEREF-
P1.4/SMCLK/A4/VREF+/VEREF+/TCK
P1.5/TA0/A5/SCLK/TMS
P1.6/TA1/A6/SDD/SCL/TD1/TCLK
P1.7/A7/SD1/SDA/TDO/TD1
BATT_SENSE
BATT_SENSE_ENABLE
TAMPER_LED_1SEC
FRAUDE
I2C_SCL/SCI
I2C_SDA
XIN/P2.6/TA1
XOUT/P2.7
VSS
32768HZ
H6
1x04
Mole

FIG.33E-1

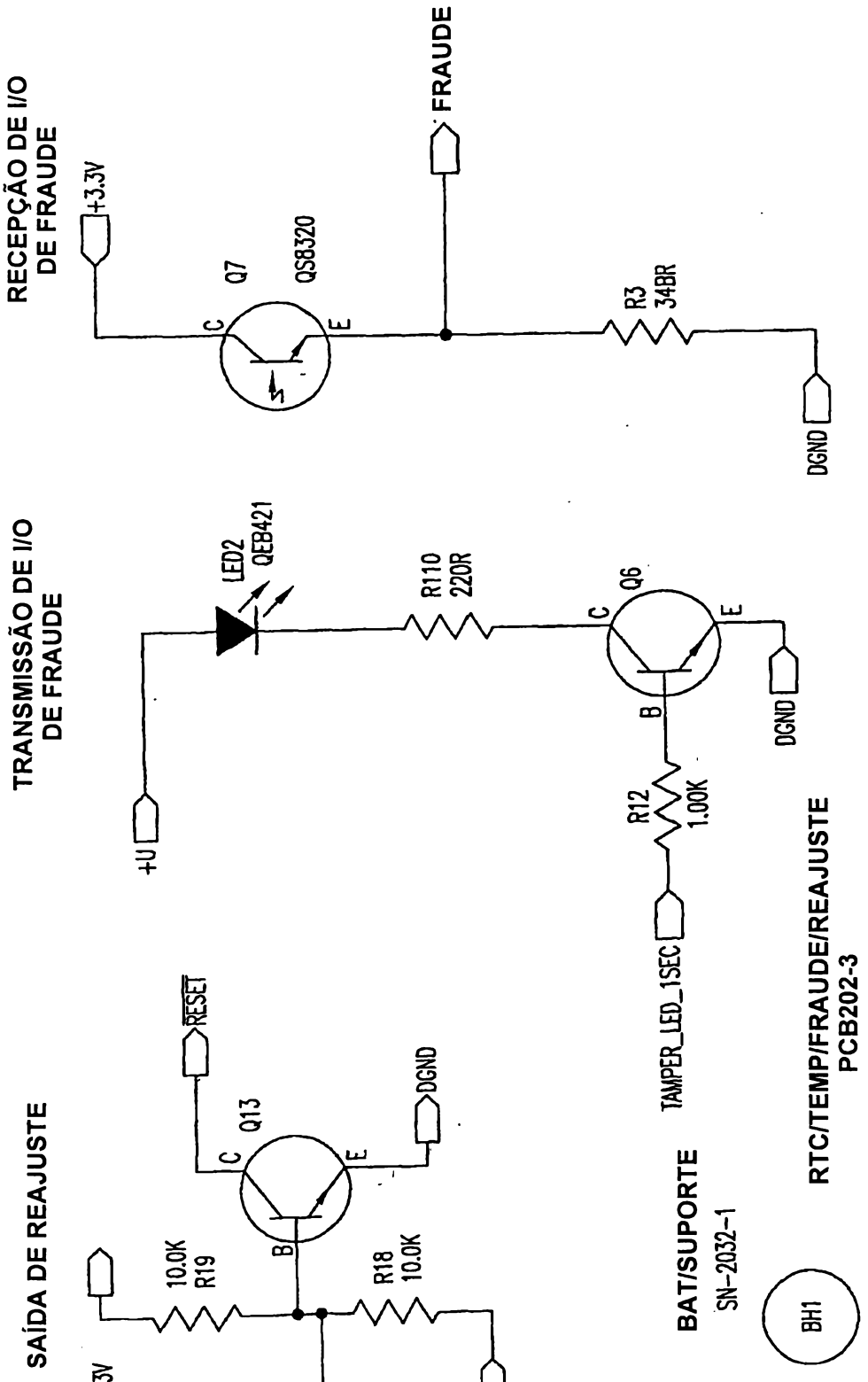

RECEPÇÃO DE I/O DE FRAUDE
+3.3V
Q7
QS8320
C
E
FRAUDE
R3
34BR
DGND
TRANSMISSÃO DE I/O DE FRAUDE
LED2
QEB421
R110
220R
Q6
C
E
B
+U
R12
1.00K
DGND
TAMPER_LED_1SEC
SAÍDA DE REAJUSTE
RESET
Q13
C
E
B
DGND
10.0K
R19
R18
10.0K
+3.3V
RESET
DGND
BAT/SUPORTE
SN-2032-1
BH1
RTC/TEMP/FRAUDE/REAJUSTE
PCB202-3

FIG.33E-2

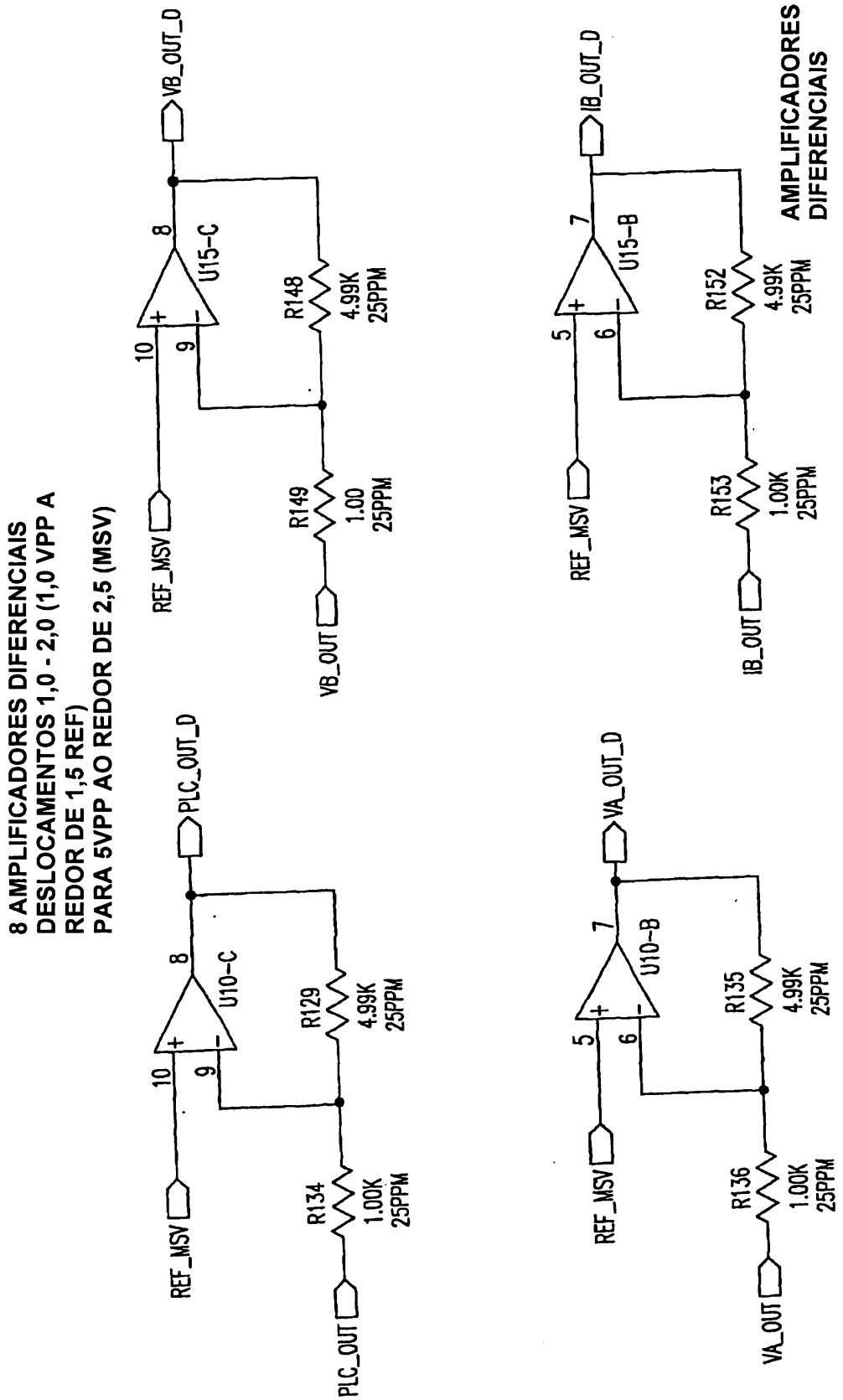

8 AMPLIFICADORES DIFERENCIAIS
DESLOCAMENTOS 1,0 - 2,0 (1,0 VPP A
REDOR DE 1,5 REF)
PARA 5VPP AO REDOR DE 2,5 (MSV)
AMPLIFICADORES
DIFERENCIAIS
VB_OUT_D
U15-C
R148
4.99K
25PPM
REF_MSV
R149
1.00
25PPM
VB_OUT
PLC_OUT_D
U10-C
R129
4.99K
25PPM
REF_MSV
R134
1.00K
25PPM
PLC_OUT
IB_OUT_D
U15-B
R152
4.99K
25PPM
REF_MSV
R153
1.00K
25PPM
IB_OUT
VA_OUT_D
U10-B
R135
4.99K
25PPM
REF_MSV
R136
1.00K
25PPM
VA_OUT

FIG.33F-1

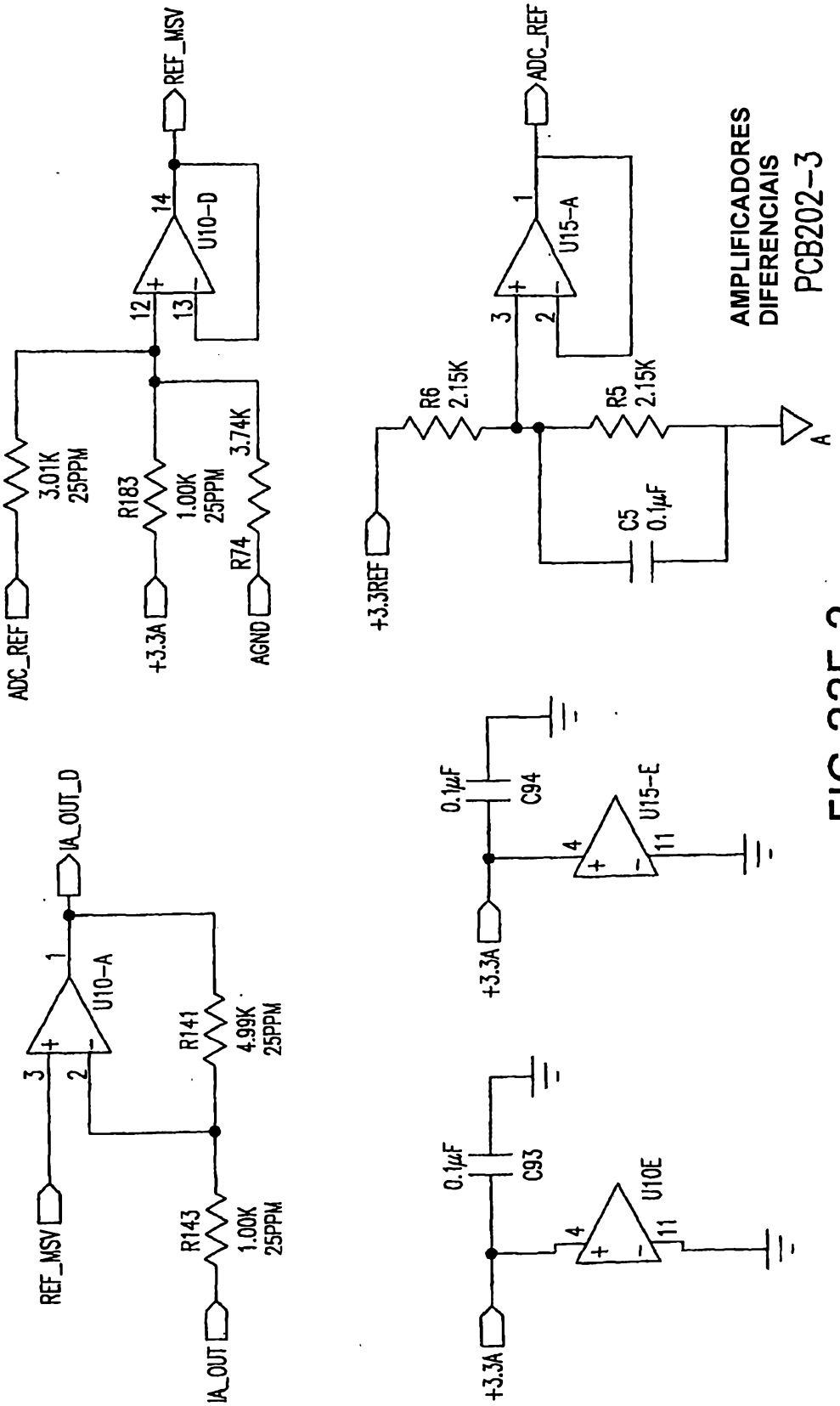

AMPLIFICADORES
DIFERENCIAIS
PCB202-3
ADC_REF
REF_MSV
IA_OUT_D
IA_OUT
+3.3A
+3.3REF
AGND
A
R184
3.01K
25PPM
R183
1.00K
25PPM
R74
3.74K
R143
1.00K
25PPM
R141
4.99K
25PPM
R6
2.15K
R5
2.15K
C5
0.1µF
C93
0.1µF
C94
0.1µF
U10-A
U10-D
U15-A
U15-E
U10E

FIG.33F-2

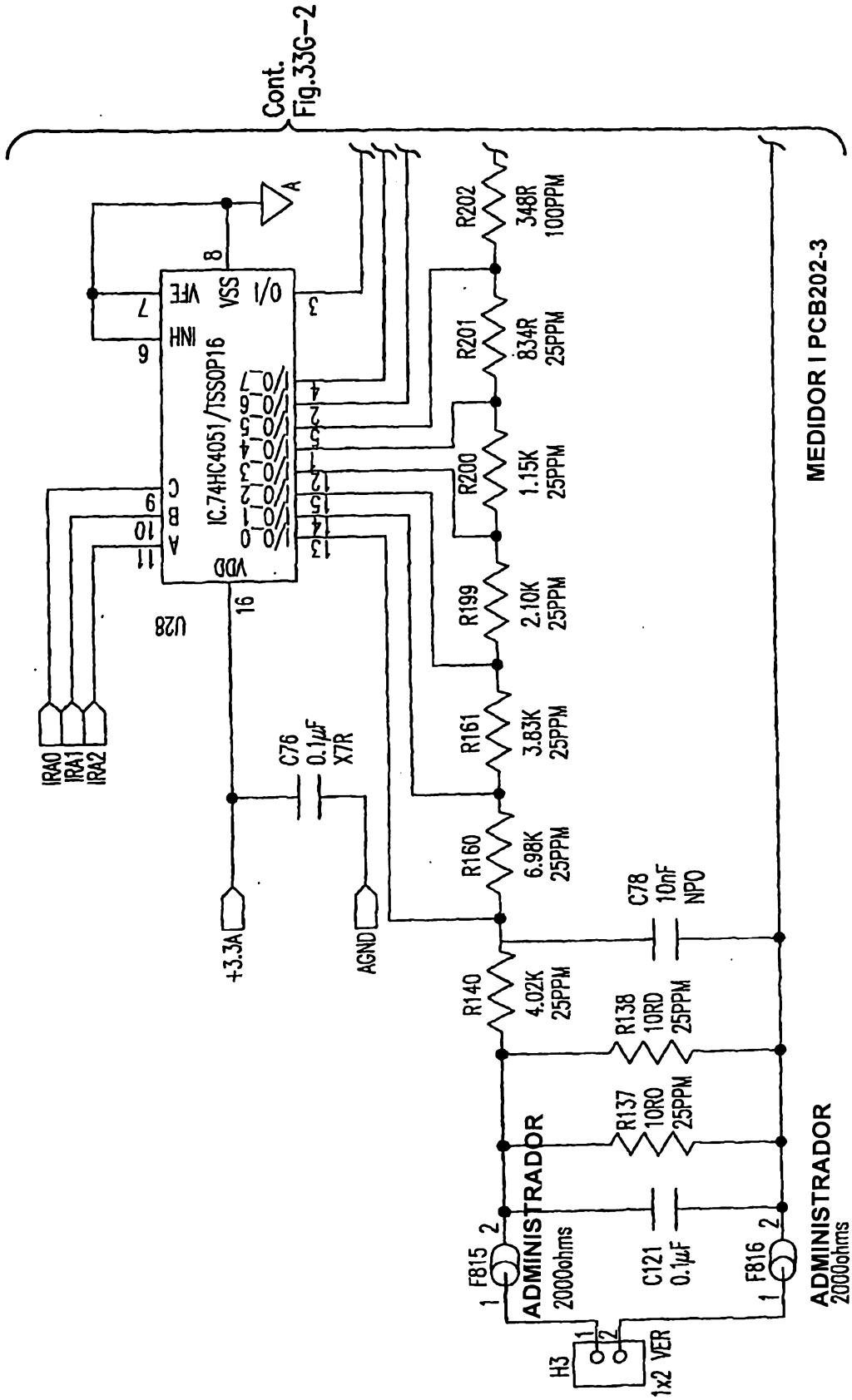

Cont.
Fig.33G-2
IC.74HC4051/TSSOP16
U28
VFE
INH
VSS
O/I
VDD
A
B
C
IRA0
IRA1
IRA2
C76
0.1µF
X7R
+3.3A
AGND
R202
348R
100PPM
R201
834R
25PPM
R200
1.15K
25PPM
R199
2.10K
25PPM
R161
3.83K
25PPM
R160
6.98K
25PPM
C78
10nF
NPO
R140
4.02K
25PPM
R138
10RD
25PPM
R137
10RO
25PPM
C121
0.1µF
F815
F816
ADMINISTRADOR
2000ohms
ADMINISTRADOR
2000ohms
H3
1x2 VER
MEDIDOR I PCB202-3

FIG.33G-1

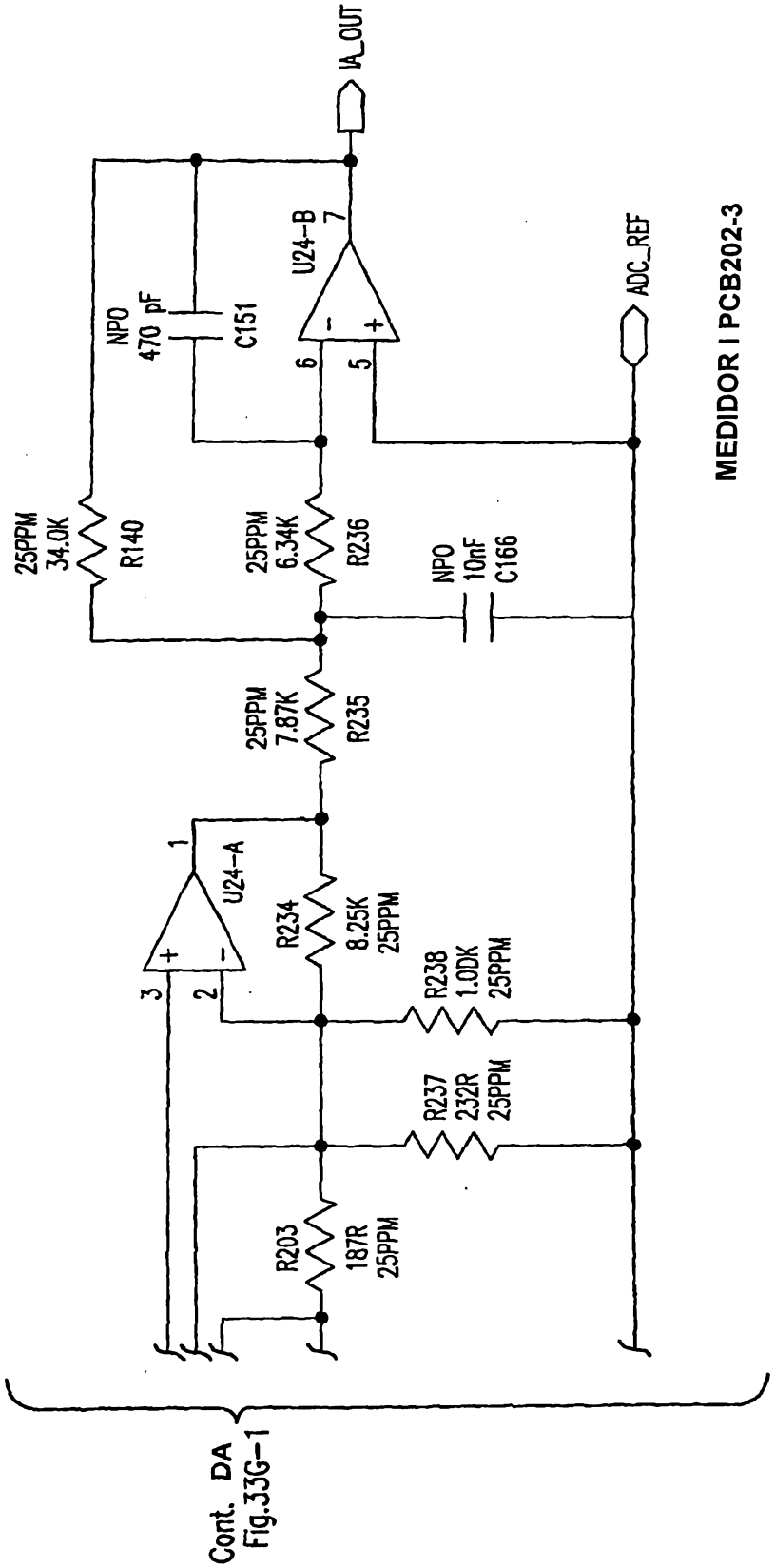

IA_OUT
U24-B
7
6
5
NPO
470 pF
C151
ADC_REF
25PPM
34.0K
R140
25PPM
6.34K
R236
NPO
10nF
C166
MEDIDOR I PCB202-3
25PPM
7.87K
R235
1
U24-A
3
2
R234
8.25K
25PPM
R238
1.0DK
25PPM
R237
232R
25PPM
R203
187R
25PPM
Cont. DA
Fig.33G-1

FIG.33G-2

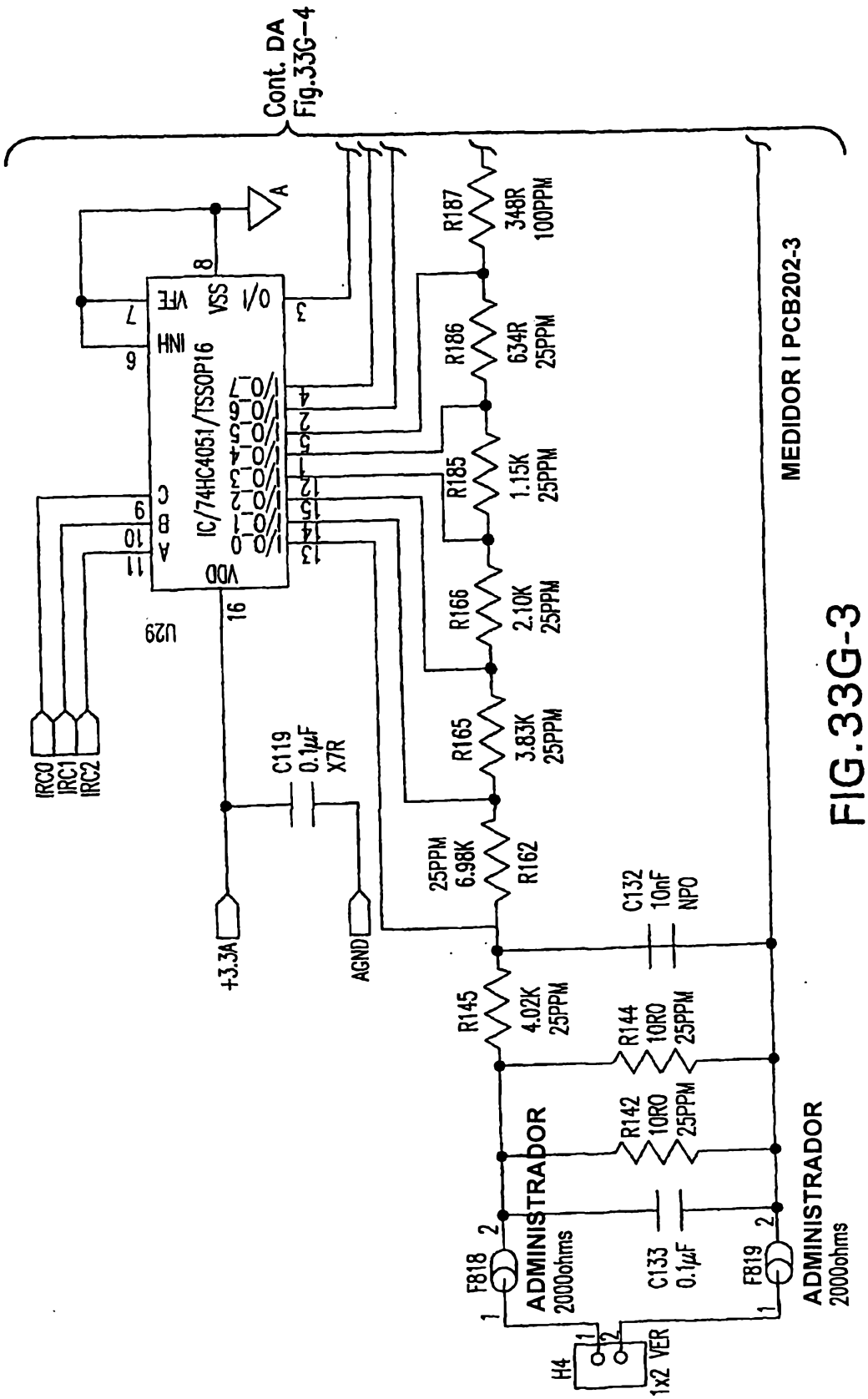

Cont. DA
Fig.33G-4
MEDIDOR I PCB202-3
U29
IC/74HC4051/TSSOP16
VDD
VSS
VEE
INH
A
B
C
R187
348R
100PPM
R186
634R
25PPM
R185
1.15K
25PPM
R166
2.10K
25PPM
R165
3.83K
25PPM
R162
6.98K
25PPM
R145
4.02K
25PPM
R144
10R0
25PPM
R142
10R0
25PPM
C119
0.1µF
X7R
C132
10nF
NPO
C133
0.1µF
IRC0
IRC1
IRC2
+3.3A
AGND
F818
F819
ADMINISTRADOR
2000ohms
ADMINISTRADOR
2000ohms
H4
1x2 VER

FIG.33G-3

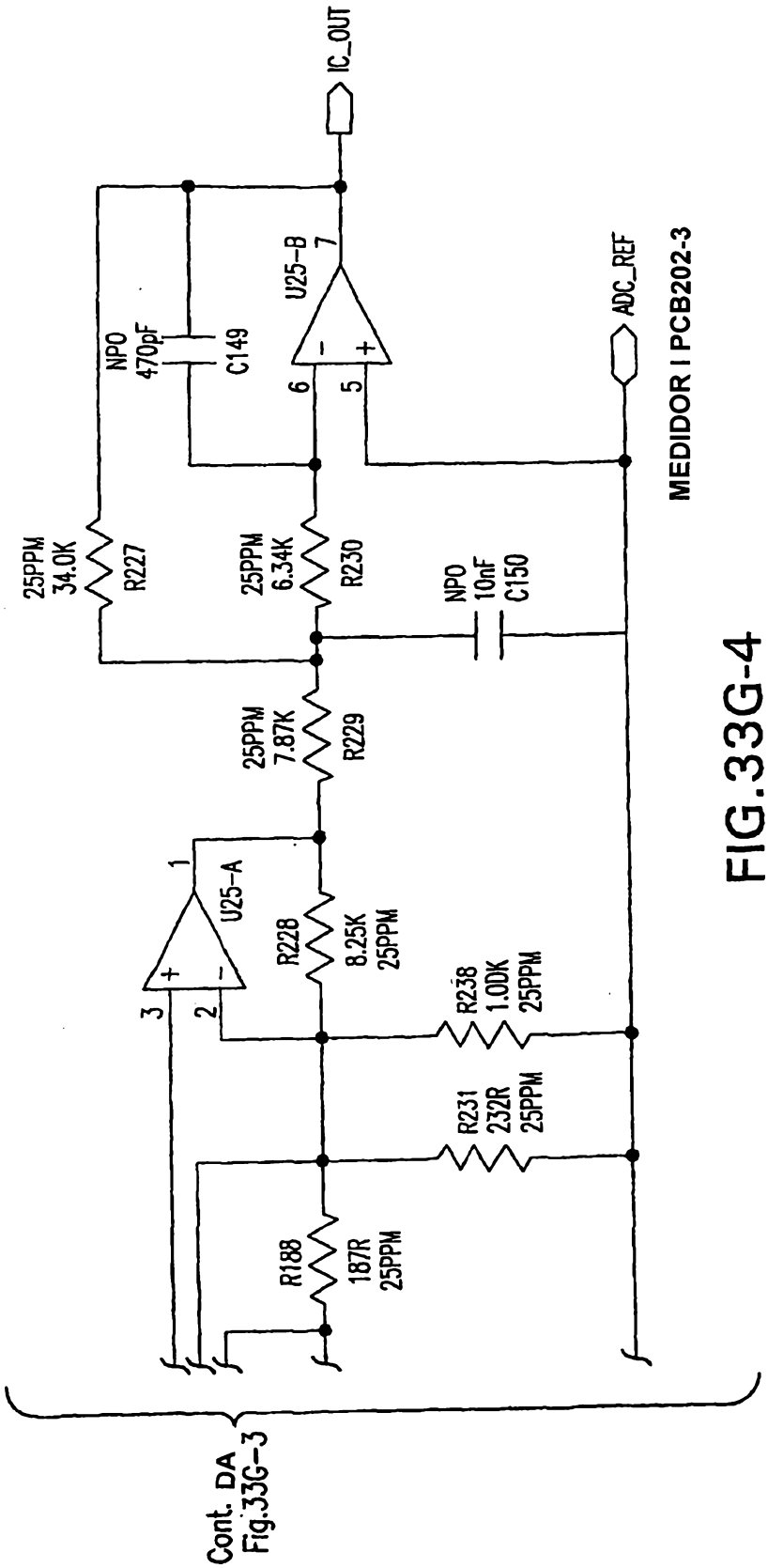

IC_OUT
U25-B
7
6
5
NPO
470pF
C149
ADC_REF
MEDIDOR I PCB202-3
25PPM
34.0K
R227
25PPM
6.34K
R230
NPO
10nF
C150
25PPM
7.87K
R229
U25-A
1
3
2
R228
8.25K
25PPM
R238
1.00K
25PPM
R231
232R
25PPM
R188
187R
25PPM
Cont. DA
Fig.33G-3

FIG.33G-4

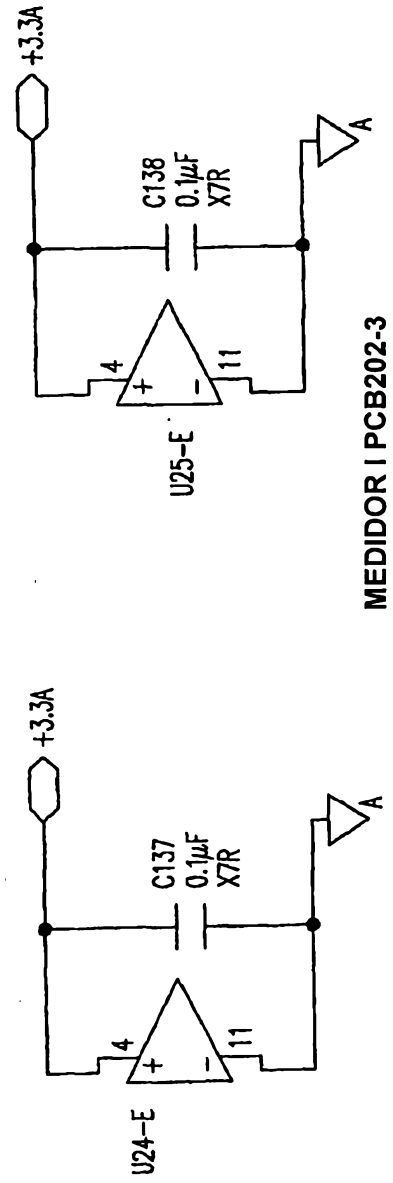

+3.3A
C138
0.1µF
X7R
A
4
11
U25-E
+3.3A
C137
0.1µF
X7R
A
4
11
U24-E
MEDIDOR I PCB202-3

FIG.33G-5

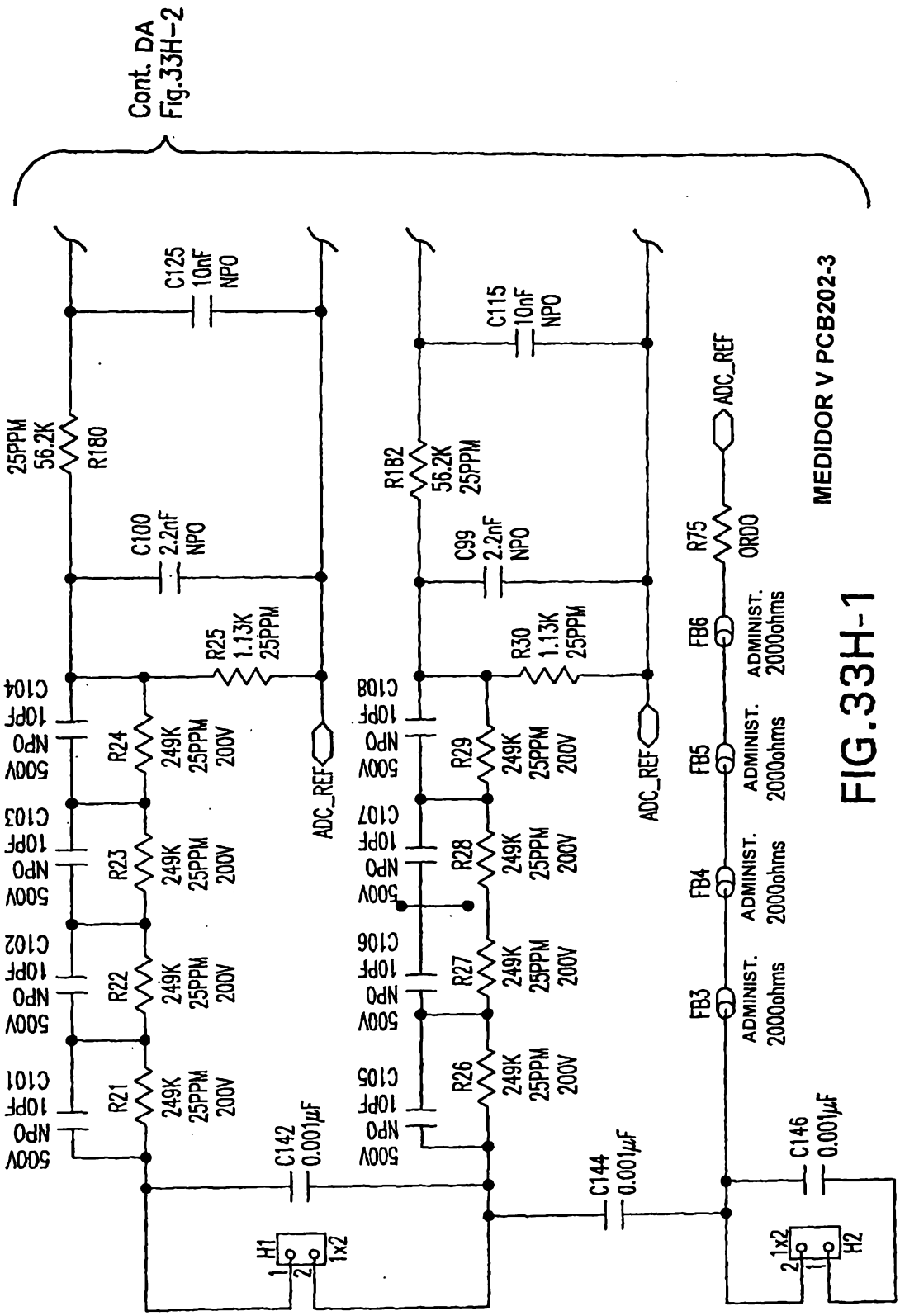

Cont. DA
Fig.33H-2
MEDIDOR V PCB202-3
C125
10nF
NPO
C115
10nF
NPO
25PPM
56.2K
R180
R182
56.2K
25PPM
C100
2.2nF
NPO
C99
2.2nF
NPO
R25
1.13K
25PPM
R30
1.13K
25PPM
ADC_REF
ADC_REF
ADC_REF
R75
ORDO
FB6
FB5
FB4
FB3
ADMINIST.
2000ohms
C104
C103
C102
C101
C108
C107
C106
C105
10PF
NPO
500V
R24
R23
R22
R21
R29
R28
R27
R26
249K
25PPM
200V
C142
0.001μF
C144
0.001μF
C146
0.001μF
H1
1x2
H2
1x2

FIG.33H-1

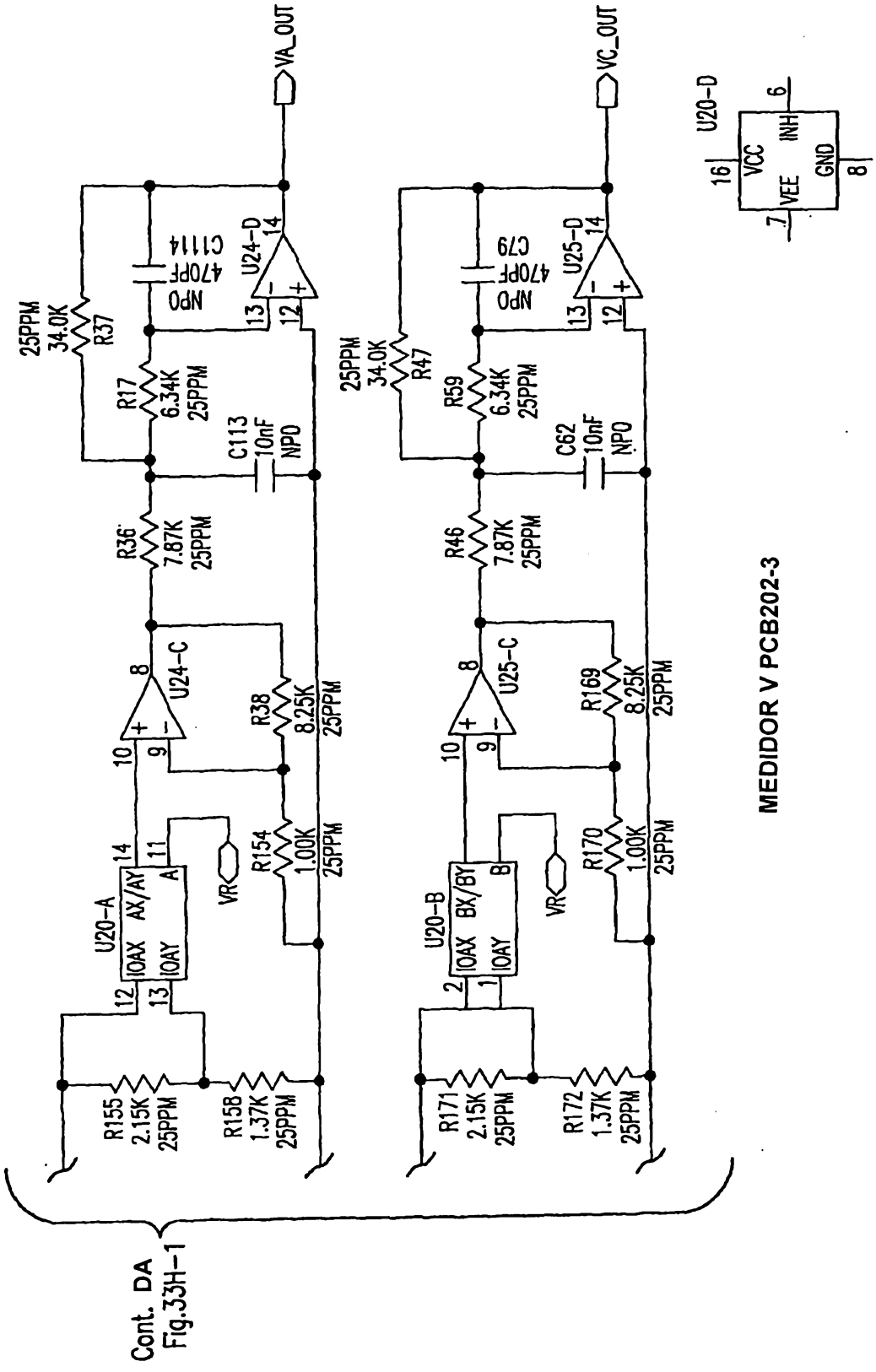

VA_OUT
VC_OUT
U20-D
VCC
VEE
INH
GND
16
7
6
8
25PPM
34.0K
R37
R47
C114
470PF
NPO
C79
U24-D
U25-D
R17
6.34K
25PPM
R59
C113
10nF
NPO
C62
R36
7.87K
25PPM
R46
U24-C
U25-C
R38
8.25K
25PPM
R169
U20-A
U20-B
IOAX
AX/AY
IOAY
BX/BY
VR
R154
1.00K
25PPM
R170
R155
2.15K
25PPM
R171
R158
1.37K
25PPM
R172
Cont. DA
Fig.33H-1
MEDIDOR V PCB202-3

FIG.33H-2

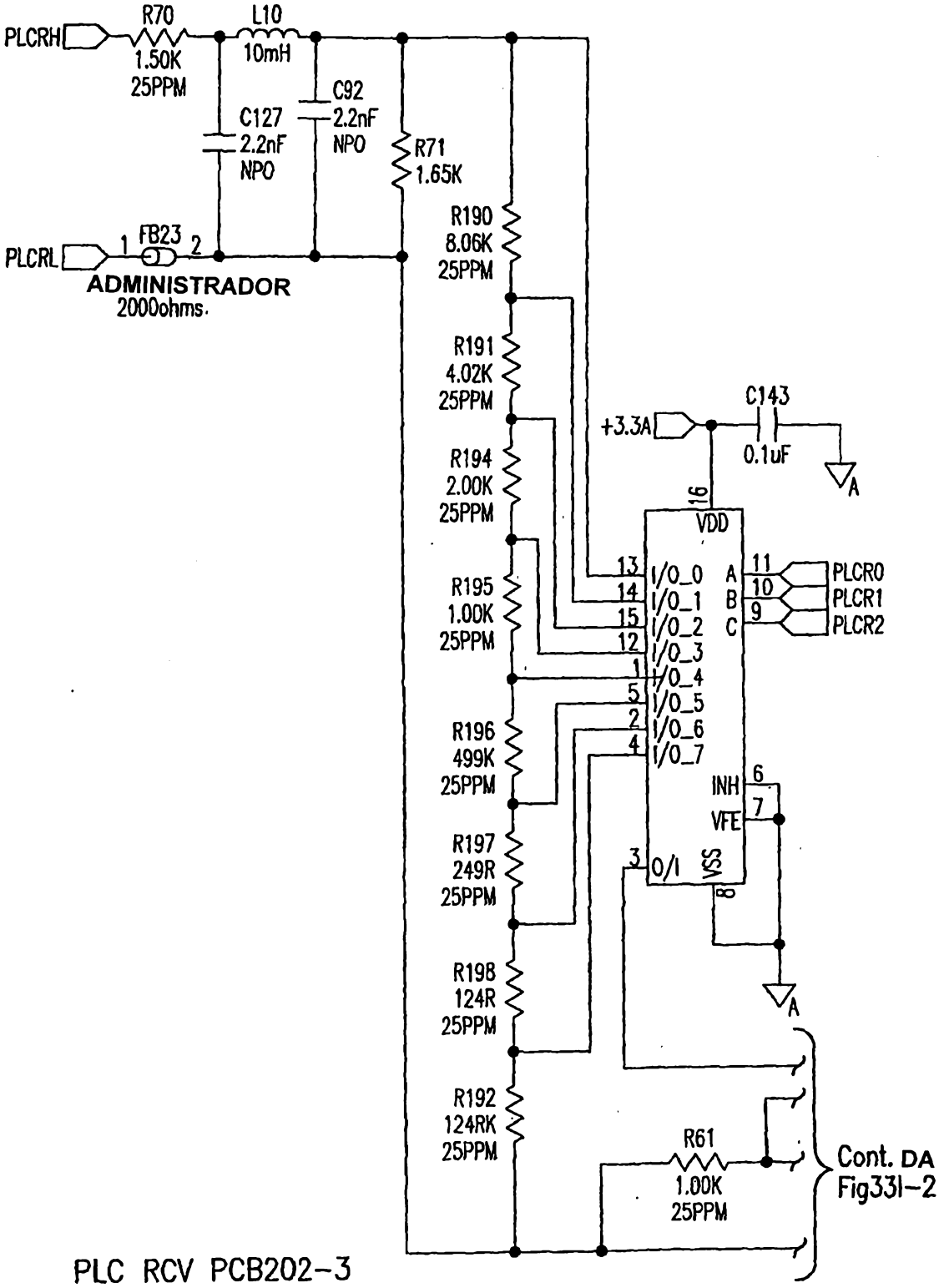
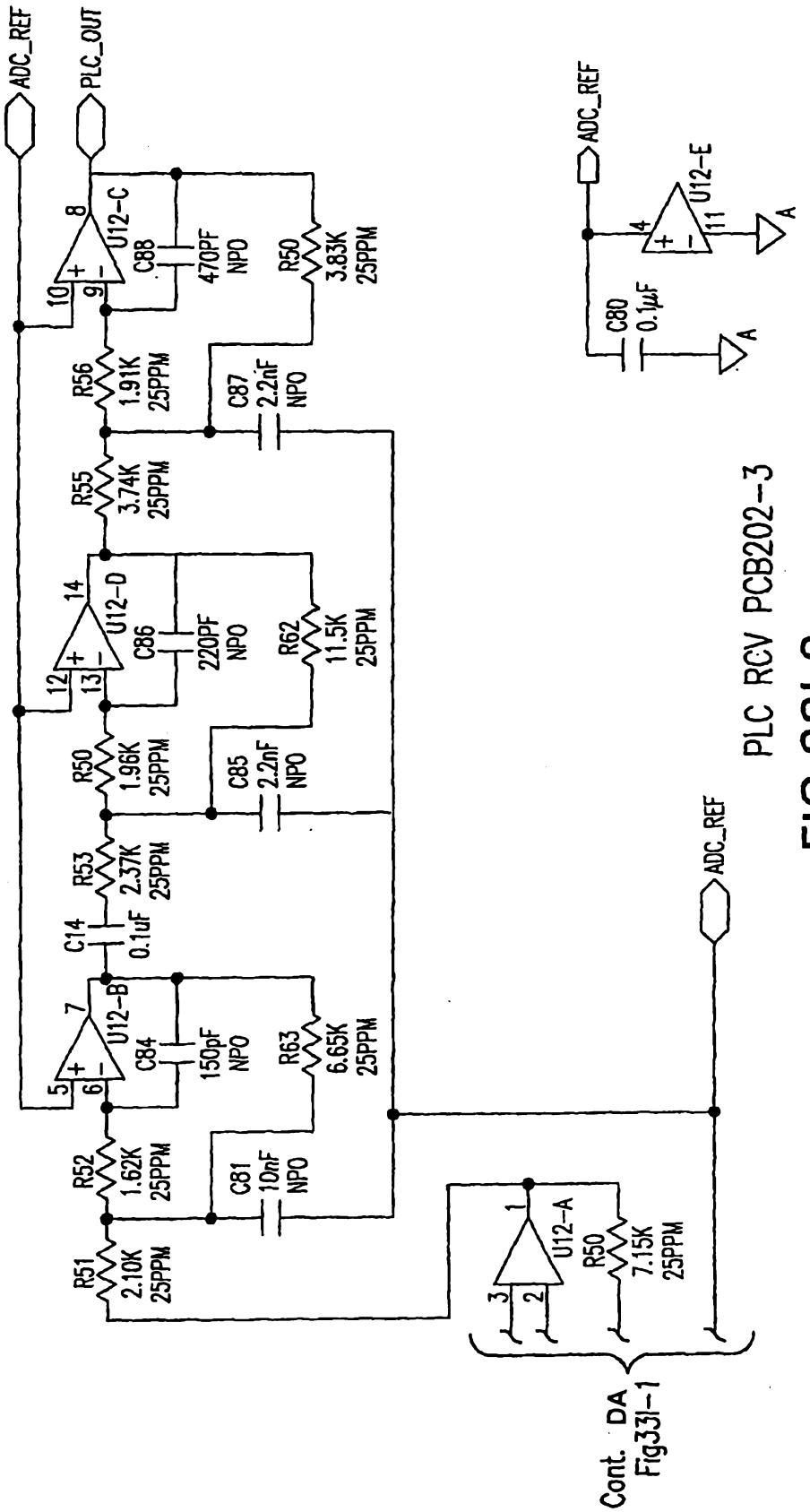
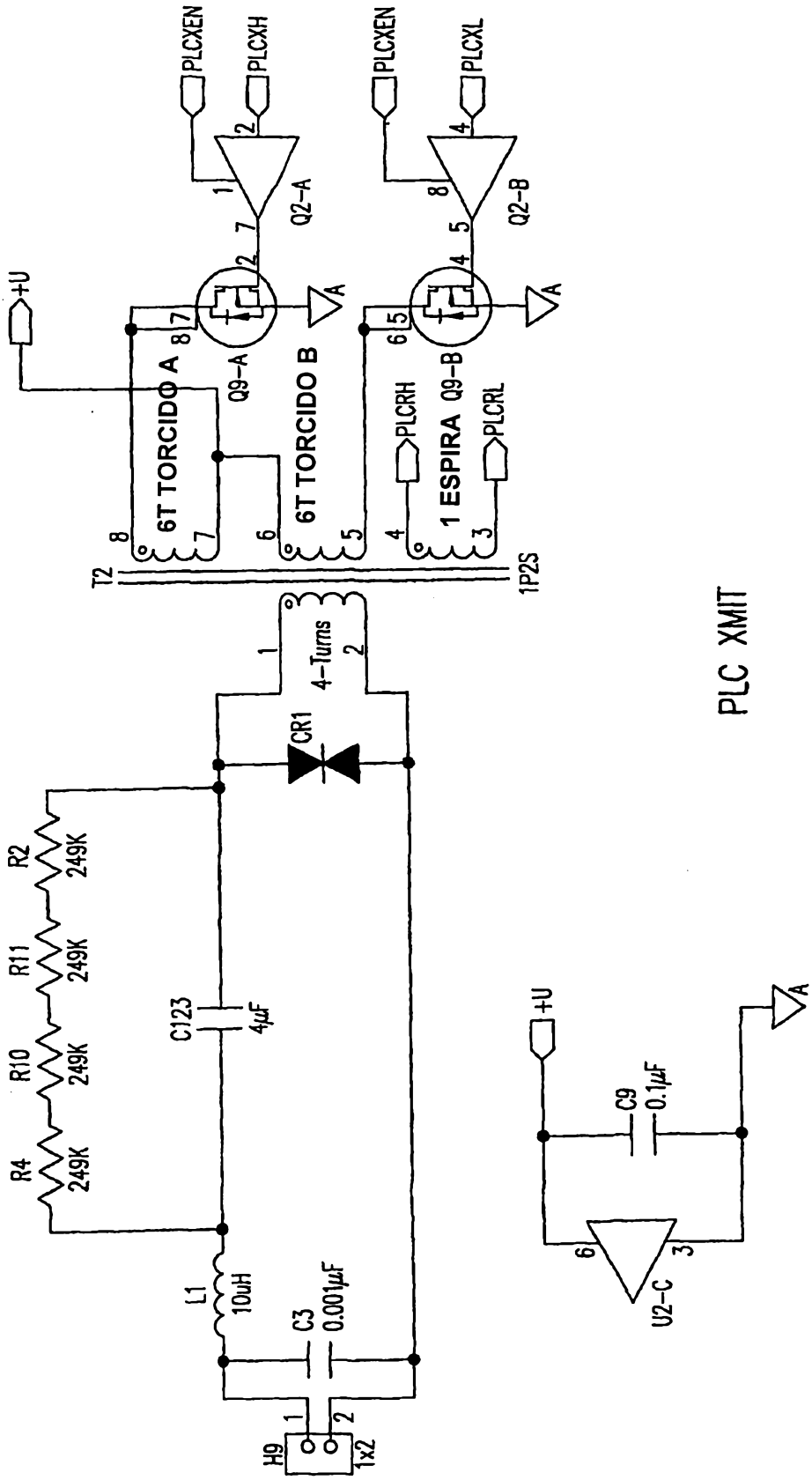

PLCRH
R70
1.50K
25PPM
L10
10mH
C92
2.2nF
NPO
C127
2.2nF
NPO
R71
1.65K
PLCRL
FB23
ADMINISTRADOR
2000ohms.
R190
8.06K
25PPM
R191
4.02K
25PPM
R194
2.00K
25PPM
R195
1.00K
25PPM
R196
499K
25PPM
R197
249R
25PPM
R198
124R
25PPM
R192
124RK
25PPM
+3.3A
C143
0.1uF
VDD
I/O_0
I/O_1
I/O_2
I/O_3
I/O_4
I/O_5
I/O_6
I/O_7
INH
VEE
O/I
VSS
PLCR0
PLCR1
PLCR2
R61
1.00K
25PPM
Cont. DA
Fig331-2
PLC RCV PCB202-3

FIG.331-1



PLC RCV PCB202-3

FIG.33I-2



PLC XMIT

FIG.33J

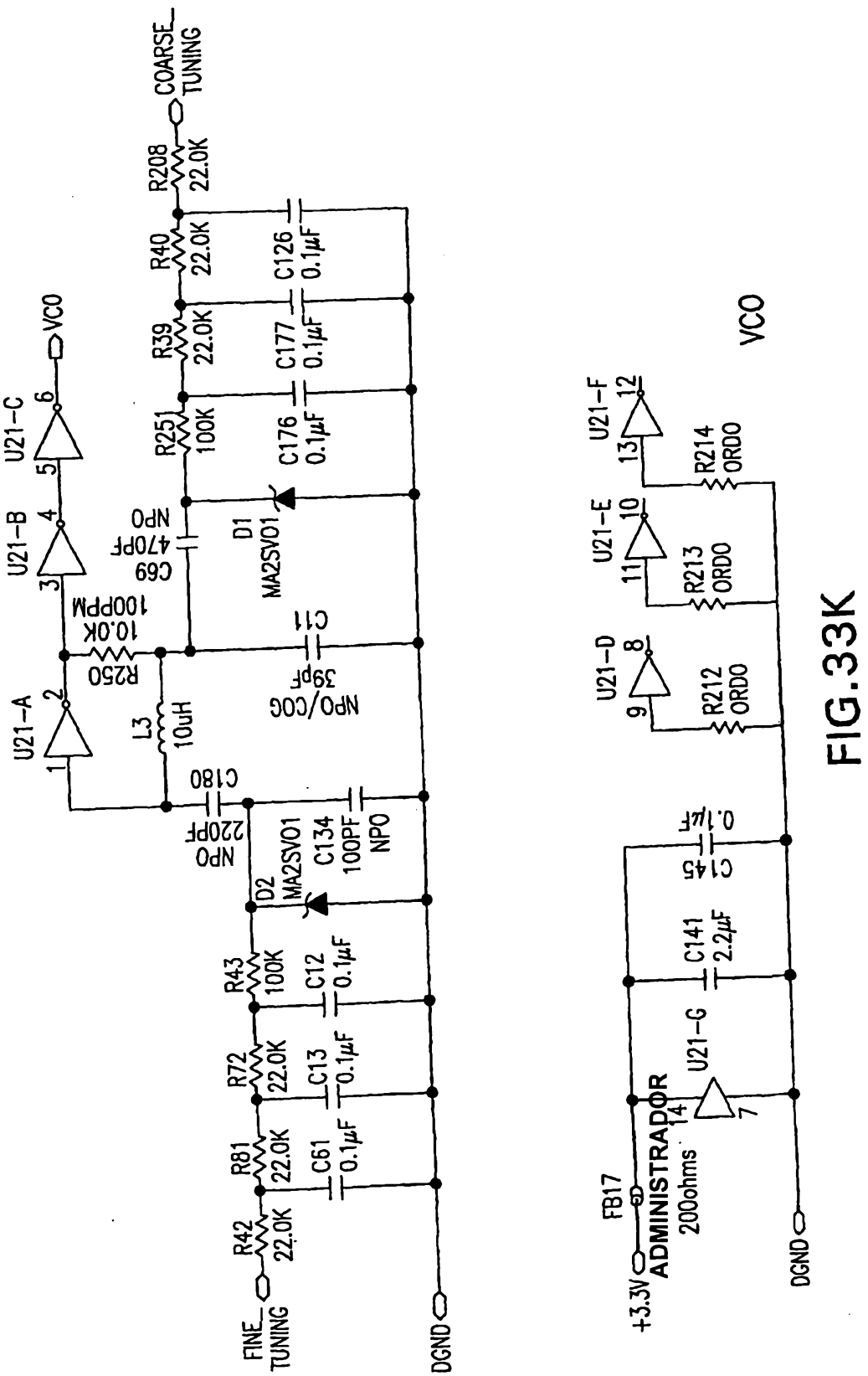

VCO
U21-A
U21-B
U21-C
U21-D
U21-E
U21-F
U21-G
R208 22.0K
COARSE TUNING
R40 22.0K
R39 22.0K
R251 100K
R250 10.0K 100PPM
R214 ORDO
R213 ORDO
R212 ORDO
R43 100K
R72 22.0K
R81 22.0K
R42 22.0K
FINE TUNING
C126 0.1µF
C177 0.1µF
C176 0.1µF
C69 470PF NPO
C11 39pF NPO/C0G
C180 220PF NPO
C134 100PF NPO
C12 0.1µF
C13 0.1µF
C61 0.1µF
C145 0.1µF
C141 2.2µF
D1 MA2SV01
D2 MA2SV01
L3 10uH
FB17 200ohms
ADMINISTRADOR
+3.3V
DGND

FIG.33K

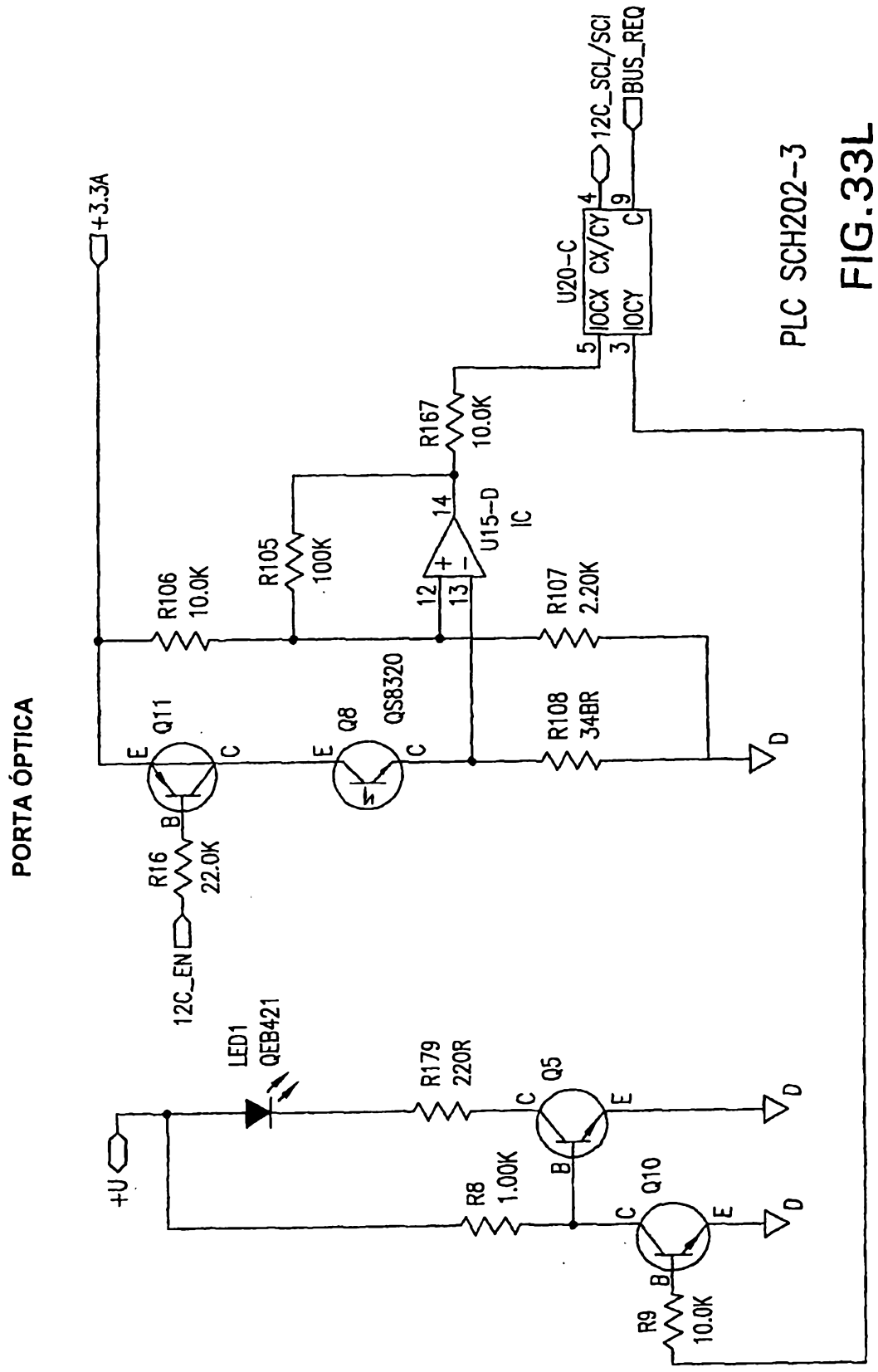

PORTA ÓPTICA
+3.3A
U20-C
IOCX CX/CY
IOCY
5
3
4
9
12C_SCL/SCI
BUS_REQ
R167
10.0K
R105
100K
R106
10.0K
14
12
13
U15-D
IC
R107
2.20K
Q11
Q8
QS8320
R108
34BR
R16
22.0K
12C_EN
LED1
QEB421
R179
220R
Q5
+U
R8
1.00K
Q10
R9
10.0K
PLC SCH202-3

FIG.33L

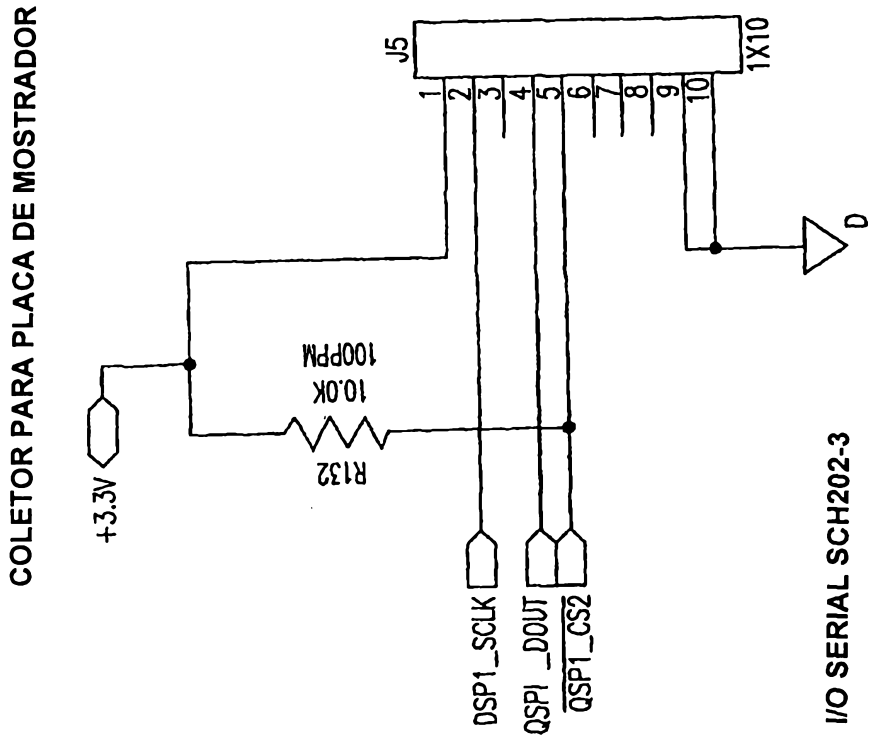

COLETOR PARA PLACA DE MOSTRADOR
+3.3V
R132
10.0K
100PPM
J5
1X10
1
2
3
4
5
6
7
8
9
10
DSP1_SCLK
QSPI _DOUT
QSP1_CS2
D
I/O SERIAL SCH202-3

FIG.33M

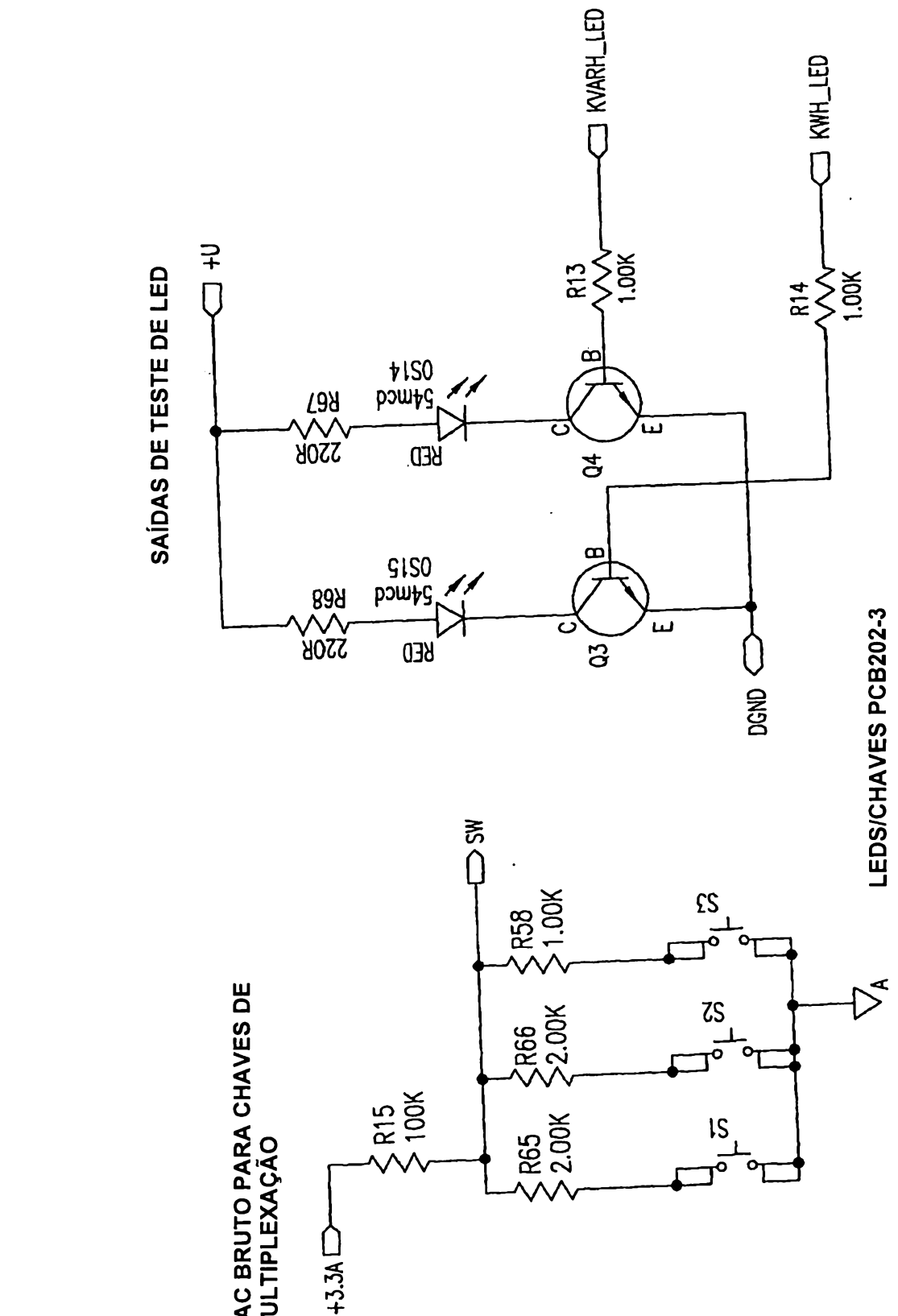

SAÍDAS DE TESTE DE LED
+U
R67
220R
OS14
54mcd
RED
R68
220R
OS15
54mcd
RED
Q4
Q3
B
C
E
R13
1.00K
R14
1.00K
KVARH_LED
KWH_LED
DGND
DAC BRUTO PARA CHAVES DE MULTIPLEXAÇÃO
+3.3A
R15
100K
R65
2.00K
R66
2.00K
R58
1.00K
SW
S1
S2
S3
A
LEDS/CHAVES PCB202-3

FIG.33N

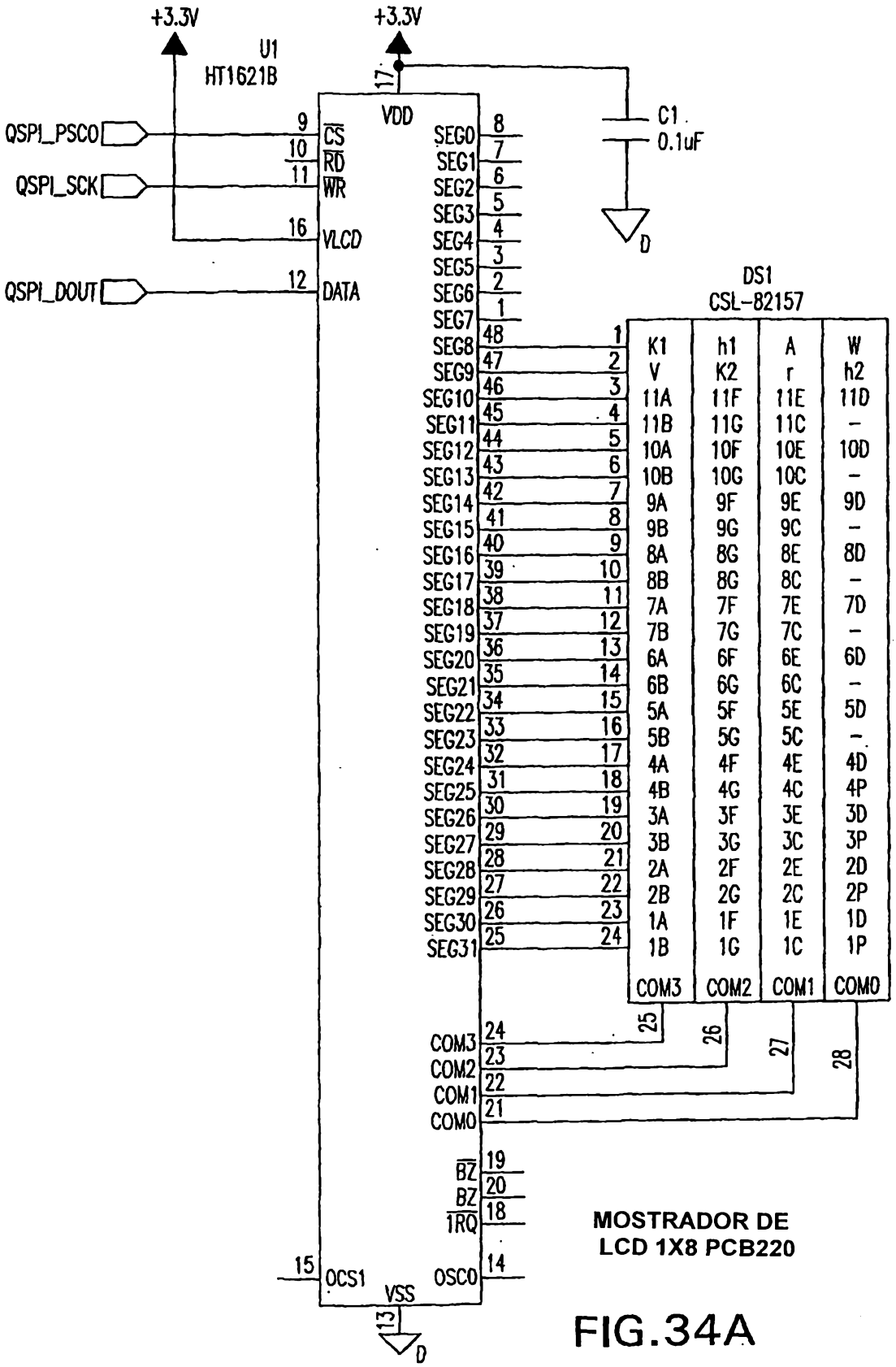

+3.3V
U1
HT1621B
+3.3V
17
VDD
C1
0.1uF
QSPI_PSCO
9
CS
QSPI_SCK
10
RD
11
WR
16
VLCD
QSPI_DOUT
12
DATA
SEG0 8
SEG1 7
SEG2 6
SEG3 5
SEG4 4
SEG5 3
SEG6 2
SEG7 1
SEG8 48
SEG9 47
SEG10 46
SEG11 45
SEG12 44
SEG13 43
SEG14 42
SEG15 41
SEG16 40
SEG17 39
SEG18 38
SEG19 37
SEG20 36
SEG21 35
SEG22 34
SEG23 33
SEG24 32
SEG25 31
SEG26 30
SEG27 29
SEG28 28
SEG29 27
SEG30 26
SEG31 25
DS1
CSL-82157
1 K1 h1 A W
2 V K2 r h2
3 11A 11F 11E 11D
4 11B 11G 11C -
5 10A 10F 10E 10D
6 10B 10G 10C -
7 9A 9F 9E 9D
8 9B 9G 9C -
9 8A 8G 8E 8D
10 8B 8G 8C -
11 7A 7F 7E 7D
12 7B 7G 7C -
13 6A 6F 6E 6D
14 6B 6G 6C -
15 5A 5F 5E 5D
16 5B 5G 5C -
17 4A 4F 4E 4D
18 4B 4G 4C 4P
19 3A 3F 3E 3D
20 3B 3G 3C 3P
21 2A 2F 2E 2D
22 2B 2G 2C 2P
23 1A 1F 1E 1D
24 1B 1G 1C 1P
COM3 COM2 COM1 COM0
25 26 27 28
COM3 24
COM2 23
COM1 22
COM0 21
BZ 19
BZ 20
IRQ 18
15 OCS1
OSC0 14
VSS
13
MOSTRADOR DE
LCD 1X8 PCB220

FIG.34A

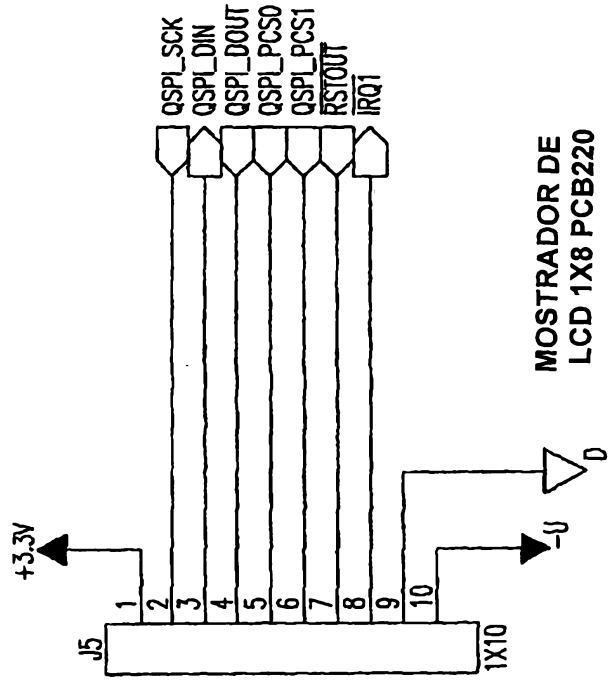

QSPI_SCK
QSPI_DIN
QSPI_DOUT
QSPI_PCS0
QSPI_PCS1
RSTOUT
IRQ1
MOSTRADOR DE
LCD 1X8 PCB220
+3,3V
-U
D
J5
1X10
1
2
3
4
5
6
7
8
9
10

FIG.34B

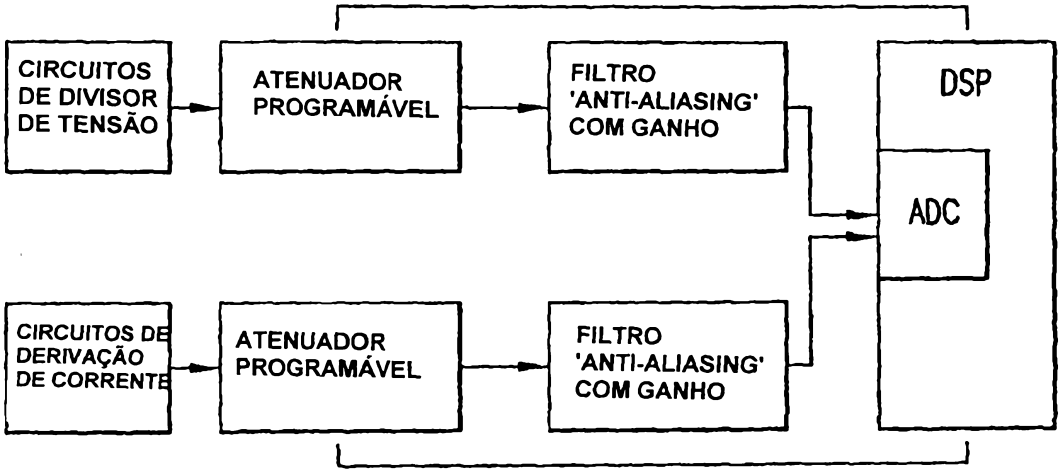

CIRCUITOS DE DIVISOR DE TENSÃO
ATENUADOR PROGRAMÁVEL
FILTRO 'ANTI-ALIASING' COM GANHO
DSP
ADC
CIRCUITOS DE DERIVAÇÃO DE CORRENTE
ATENUADOR PROGRAMÁVEL
FILTRO 'ANTI-ALIASING' COM GANHO

FIG.35

DIAGRAMA DE BLOCO DE HARDWARE DE ID

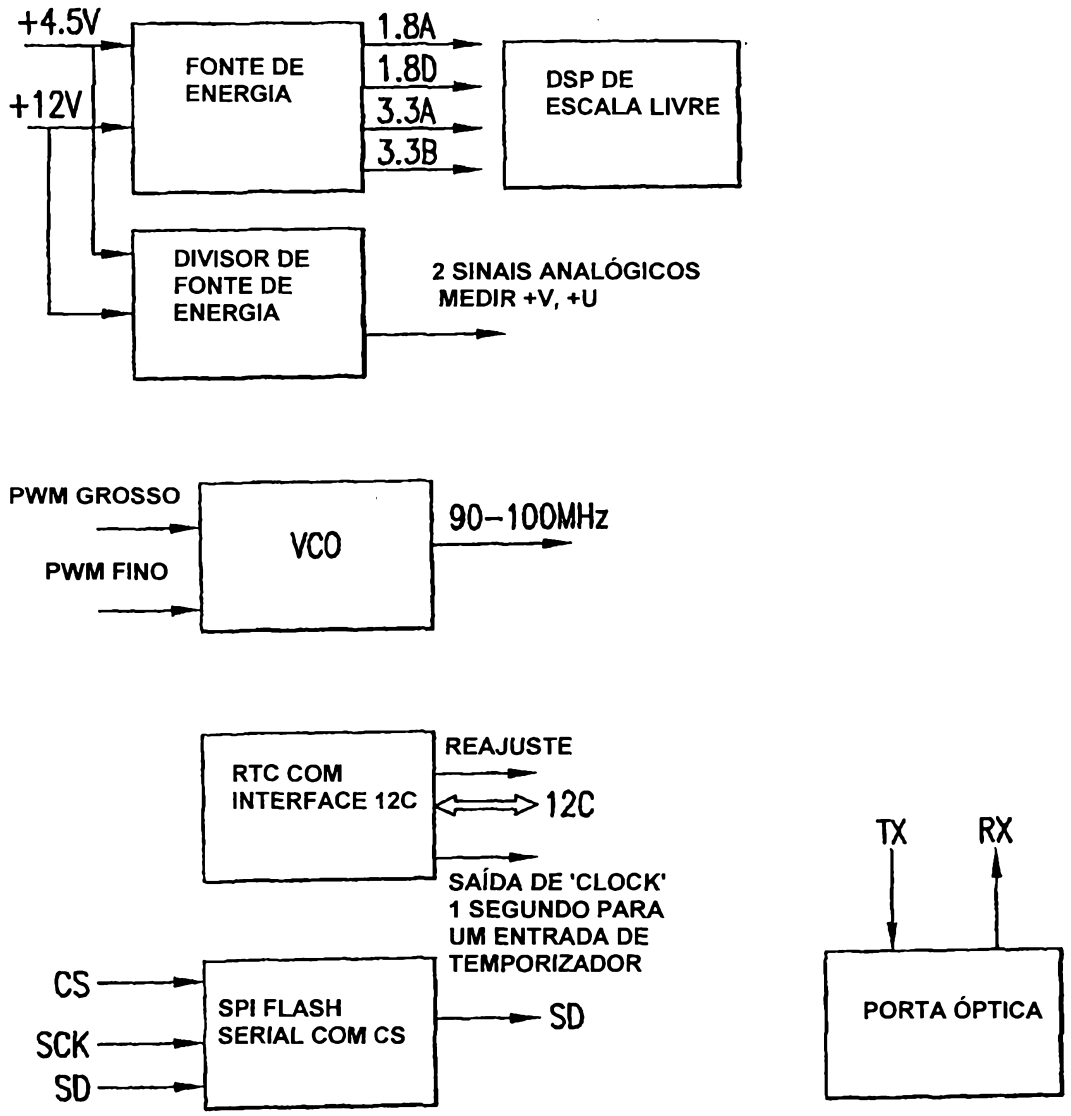


FIG.36

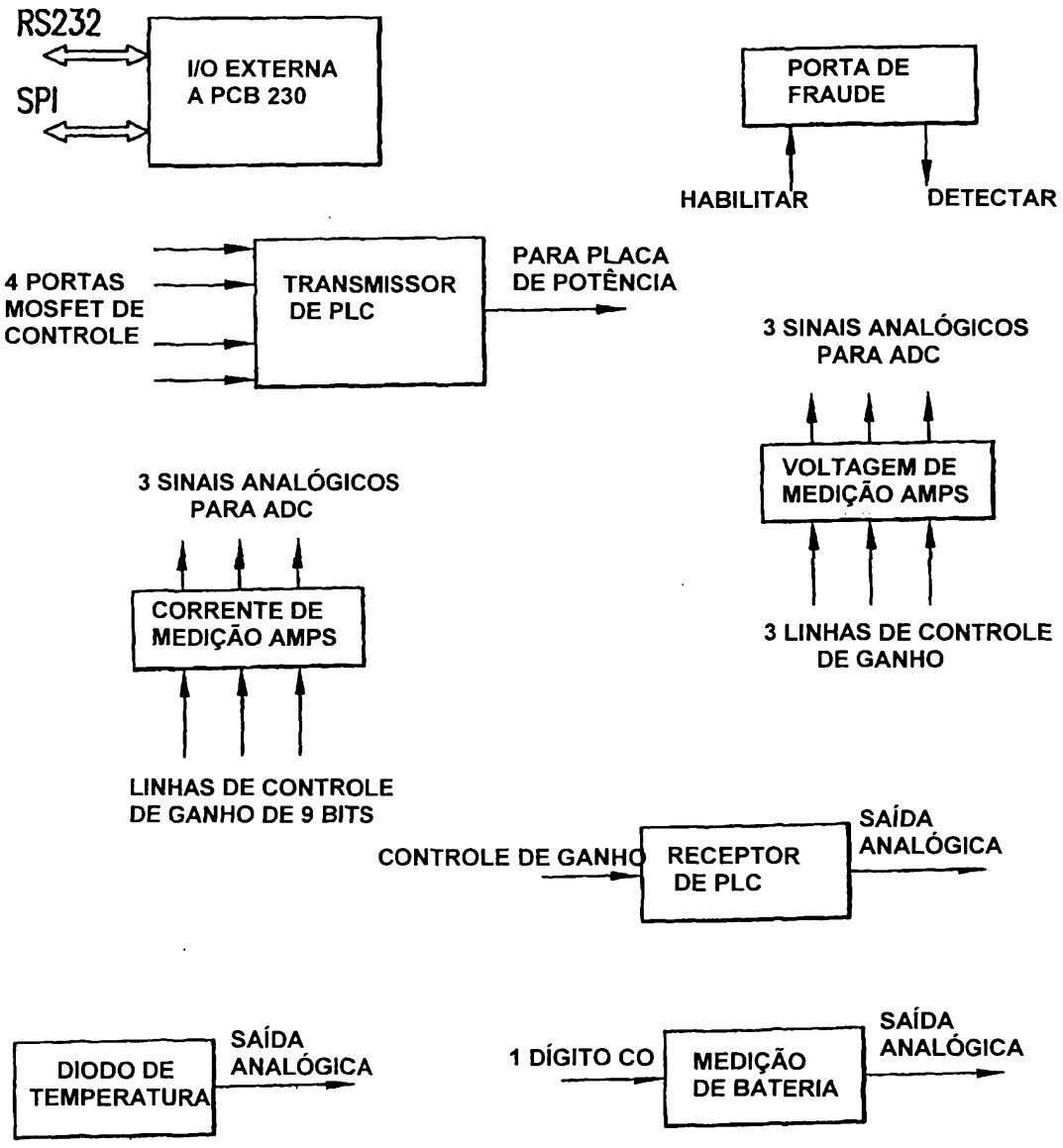

RS232
SPI
I/O EXTERNA A PCB 230
PORTA DE FRAUDE
HABILITAR
DETECTAR
4 PORTAS MOSFET DE CONTROLE
TRANSMISSOR DE PLC
PARA PLACA DE POTÊNCIA
3 SINAIS ANALÓGICOS PARA ADC
VOLTAGEM DE MEDIÇÃO AMPS
3 LINHAS DE CONTROLE DE GANHO
3 SINAIS ANALÓGICOS PARA ADC
CORRENTE DE MEDIÇÃO AMPS
LINHAS DE CONTROLE DE GANHO DE 9 BITS
CONTROLE DE GANHO
RECEPTOR DE PLC
SAÍDA ANALÓGICA
DIODO DE TEMPERATURA
SAÍDA ANALÓGICA
1 DÍGITO CO
MEDIÇÃO DE BATERIA
SAÍDA ANALÓGICA

FIG.37

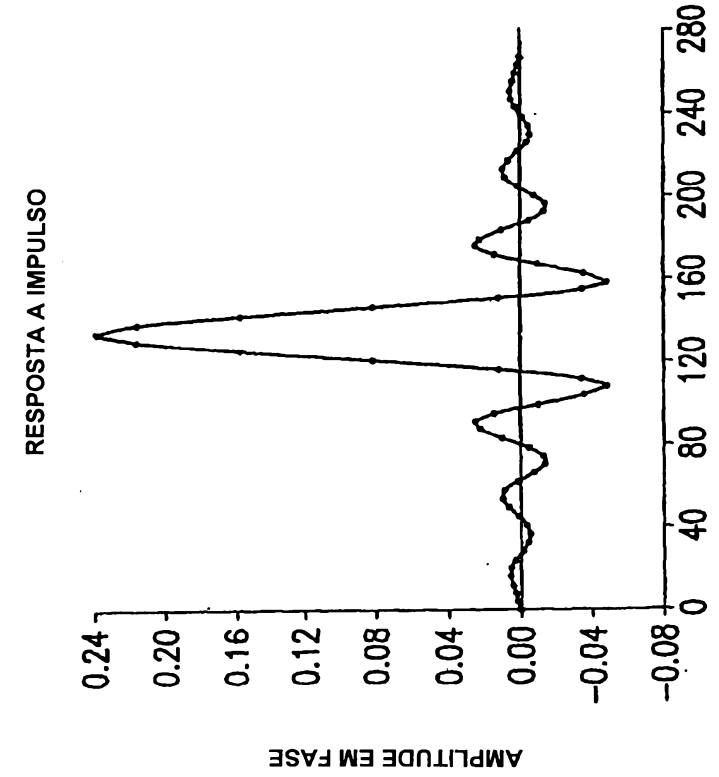


FIG.38B

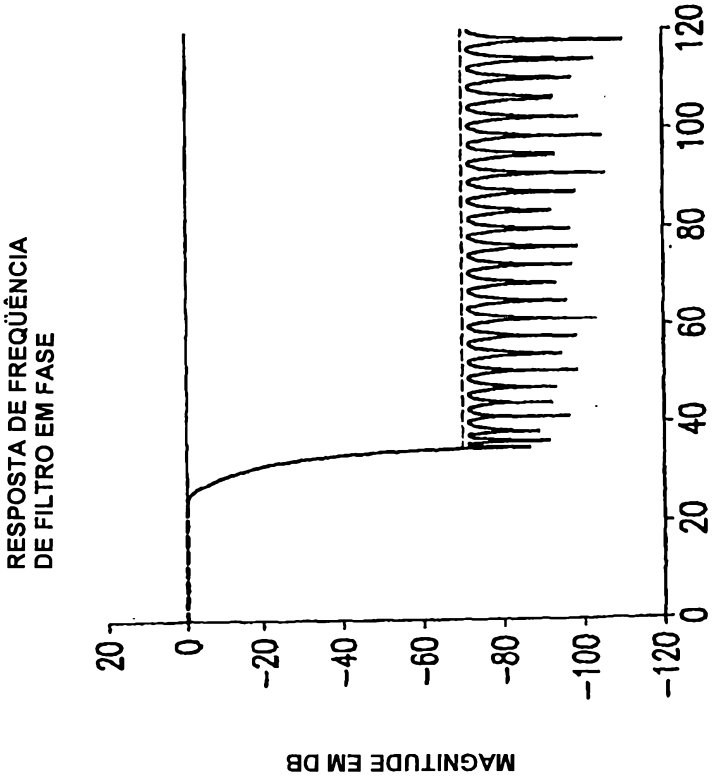


FIG.38A

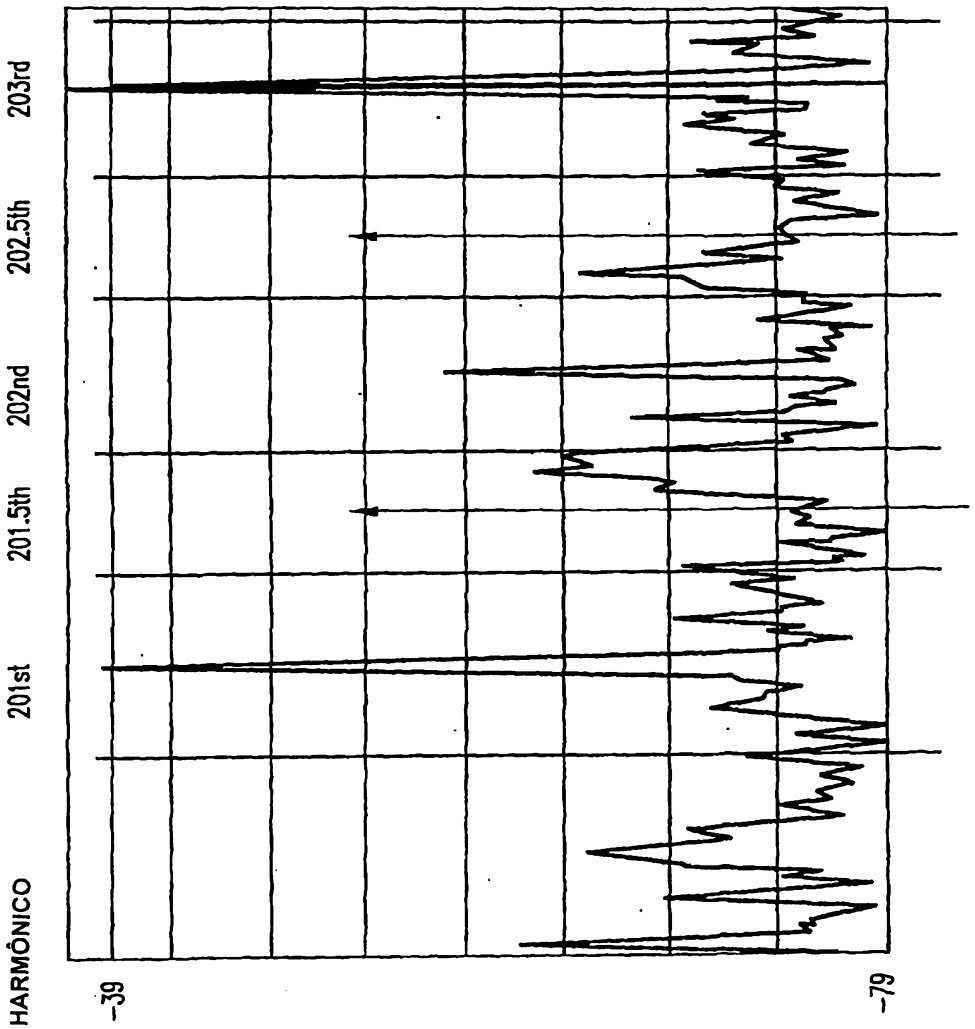


FIG.39

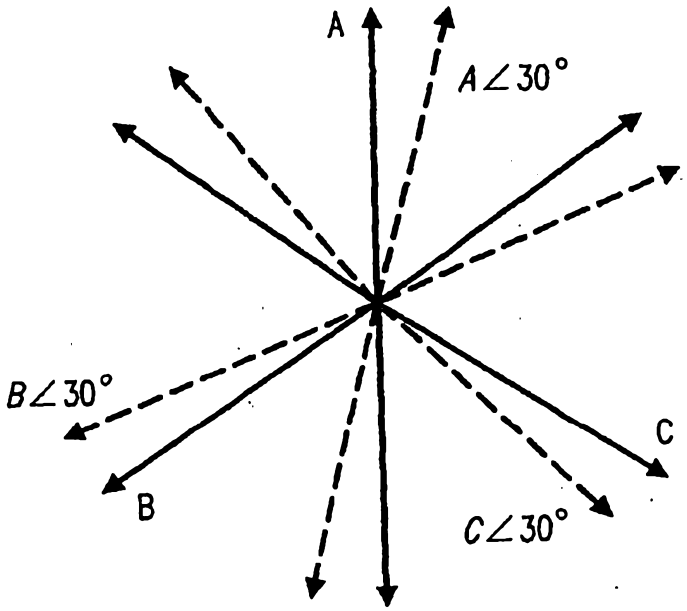


FIG.40

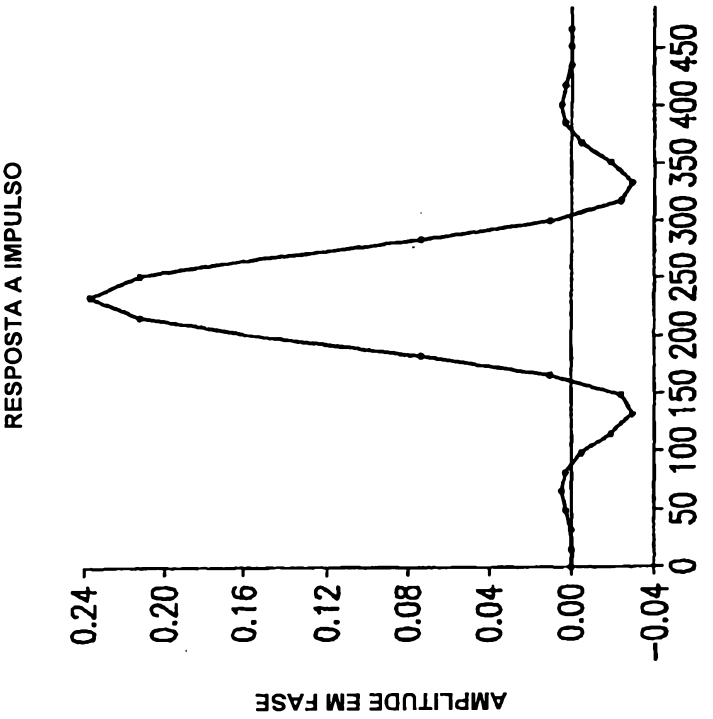


FIG.41B

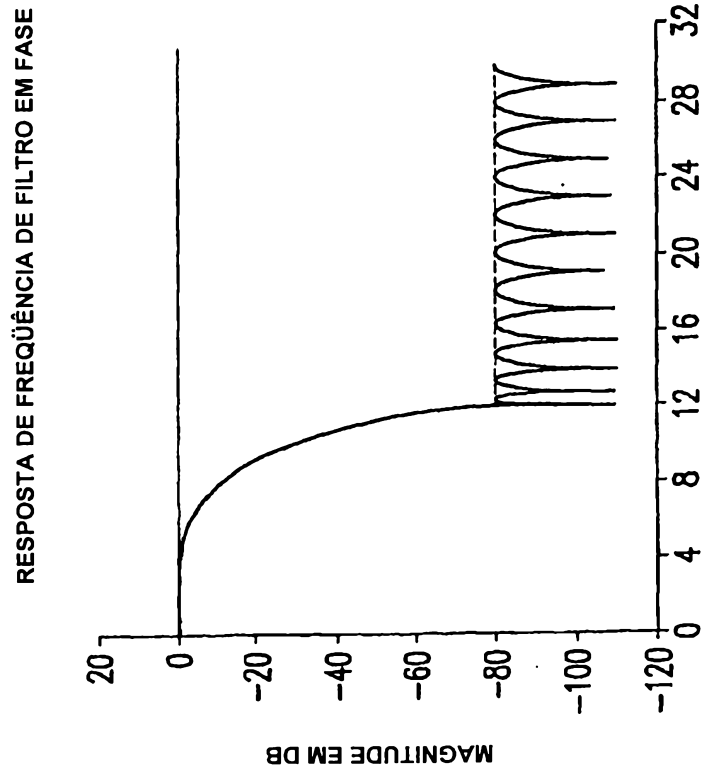


FIG.41A

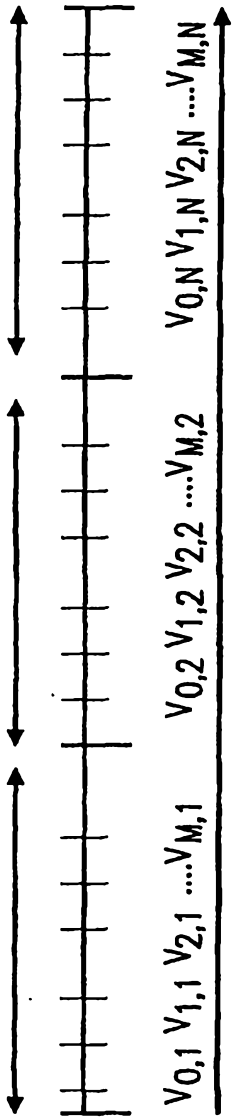


FIG.42