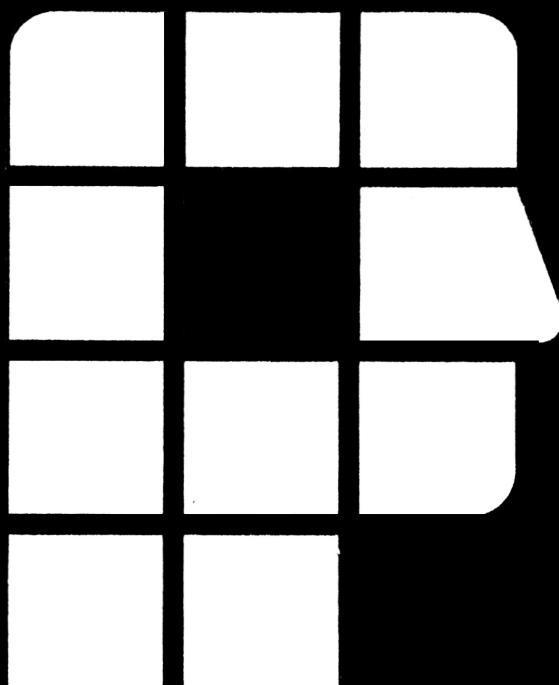


061



PROLOGICA

microcomputadores

MANUAL DO CP 400

MANUAL TÉCNICO

CP 400

ÍNDICE

1. INTRODUÇÃO
2. DESCRIÇÃO DOS PRINCIPAIS COMPONENTES
 - a. CPU MC6809E
 - b. SINAIS DE CONTROLE
 - c. ACUMULADORES
 - d. REGISTRADOR DE PÁGINA DIRETA
 - e. REGISTRADORES INDEXADOS
 - f. STACK POINTER
 - g. CONTADOR DE PROGRAMA
 - h. REGISTRADOR DE CÓDIGO DE CONDIÇÃO
3. MODOS DE ENDEREÇAMENTO DE 6809
4. DESCRIÇÃO DE CADA INSTRUÇÃO INDIVIDUAL DO 6809E
5. ADAPTADOR DE INTERFACES E PERIFÉRICOS MC6821
6. DESCRIÇÃO DA PINAGEM
7. MODOS DE APRESENTAÇÃO DO VÍDEO
8. DESCRIÇÃO GERAL
9. DESCRIÇÃO CIRCUITO ELÉTRICO PROLÓGICA 12
 - a. MAPEAMENTO DE MEMÓRIA
10. ADVERTÊNCIA

1. INTRODUÇÃO

Este manual visa explicar o funcionamento do CP 400 e sua consequente manutenção. Para melhor entendimento, é necessário que o usuário tenha conhecimento prévio de lógica digital, microprocessadores; além de conhecimento básico sobre TV a cores e já ter lido os manuais de operação e programação do equipamento.

Sugerimos ainda a leitura da seguinte bibliografia:

- Motorola

 Microprocessors

 Data Manual

- Microprocessors Applications Manual

 Motorola Semiconductor Products INC

- M6800 Microprocessor

 Programming Manual

 Motorola

- 6809 Assembly Language Programming

 Lance A. Leventhal

 OSBORNE/MC GRAW HILL

2. DESCRIÇÃO DOS PRINCIPAIS COMPONENTES

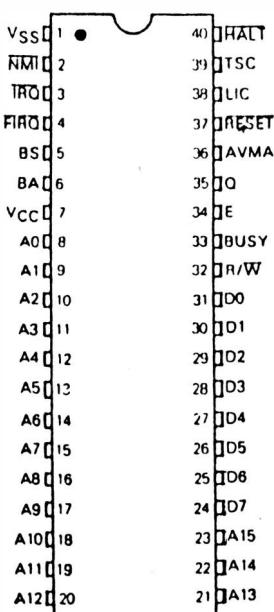
a) CPU MC6809E

A CPU 6809, fabricada pela Motorola, trabalha com barramento de dados de 8 bits e 16 linhas de endereçamento , tendo como frequênciá máxíma um clock de 1,0MHz de fase dupla. Abaixo as principais características:

- Compatibilidade com a família de microprocessadores MC6800
- Características da arquitetura:
 - . Dois registradores indexados de 16 bits.
 - . Dois indicadores de pilha de 16 bits, indexáveis.
 - . Dois acumuladores de 8 bits, que podem ser concatenados para formar um acumulador de 16 bits.
 - . Endereçamento direto de toda memória.
- Características do Hardware:
 - . Entradas de clock externa, E e Q.
 - . Entrada de controle do buffer do barramento interno-TSC.
 - . Indicador de busca do código de operação-LIC.
 - . Recursos para uso em multiprocessamento-VMA.
 - . Linha de Status para uso em multiprocessamento BUSY.
 - . Entrada de requisição de interrupção rápida.
 - . Alimentação de + 5V e terra.
- Características do software:
 - . 10 modos de endereçamento.
 - . 1464 instruções com um único modo de endereçamento.
 - . Operações aritméticas de 16 bits.
 - . Troca/transferência de todos registradores.

Abaixo temos a pinagem da CPU 6809 e a descrição de cada pino:

FIGURE 1 – PIN ASSIGNMENT



Alimentação

1 - VSS - ØV

7 - VCC - +5V

Barramento de endereços:

A₀ - A₁₅ - 16 linhas de endereço, ativas em nível lógico alto se a CPU não está efetuando endereçamento, essas linhas ficarão em nível lógico alto (FFFF), R/W será 1 e BS=Ø. Todas as linhas de endereço ficarão no estado three-state se a saída BA for igual a 1 ou se a entrada TSC estiver em nível lógico alto.

Barramento de dados:

D₀ - D₇ - Oito linhas bidirecionais, para a comunicação da CPU com o meio exterior.

b) SINAIS DE CONTROLE

R/W (read/write) - Esse sinal indica a direção do barramento de dados. Um nível baixo nesta saída indica que a CPU irá escrever os dados no dispositivo endereçado pelas linhas A0-A15. R/W irá para um estado de alta impedância enquanto BA ou TSC estiverem em nível lógico alto.

RESET - em nível lógico baixo nesta entrada, terá que ficar presente por um tempo maior que o ciclo do barramento, para resetar a CPU. Durante o tempo de reset, o sinal de reconhecimento de interrupção (BA e BS iguais a 1) será ativo e a CPU fará busca do vetor de inicialização nos endereços FFFF e FFFE, sendo que o endereço FFFE conterá o byte mais significativo desse vetor.

HALT - em nível baixo nesta entrada fará com que a CPU pare ao fim da instrução em execução, ficando assim indefinidamente, sem perda de dados. Enquanto no estado de Halt, a linha BA ficará em nível lógico alto, indicando que o barramento de dados está em estado de alta impedância. A linha BS também vai a nível lógico alto, indicando que a CPU está em estado de Halt. Durante o estado de Halt, a CPU ignorará requisições externas de interrupção (FIRQ e IRQ), exceto os sinais NMI e RESET que estão armazenados para resposta posterior.

BUS AVAILABLE, BUS STATUS (BA,BS)- A saída Bus available (barramento disponível), quando em nível lógico alto, indica que o barramento está em three-state. Quando BA vai a nível lógico baixo, um ciclo passará antes que a CPU comece a utilizar o barramento. O sinal de saída Bus Status (estado do barramento), quando decodificado com o sinal BA, irá fornecer o estado da CPU (válido com o sinal Q).

Estado CPU	Definição estado CPU	
BA	0	Normal (processando)
BA	1	Reconhecimento Interrupção ou Reset
BS	0	Reconhecimento SYNC
BS	1	Reconhecimento HALT

O reconhecimento de interrupção é indicado durante ambos ciclos de uma busca de um vetor de hardware (RESET, NMI, FIRQ, IRQ, SWI, SWI2, SWI3). Esse sinal pode ser decodificado para indicar em que nível de interrupção a CPU se encontra. A localização dos vetores de interrupção é a seguinte:

Localização dos vetores no mapa de memória		Descrição do Vetor de interrupção
MS	LS	
FFFFE	FFFF	RESET
FFFC	FFFD	NMI
FFFA	FFFB	SWI
FFF8	FFF9	IRQ
FFF6	FFF7	FIRQ
FFF4	FFF5	SWI2
FFF2	FFF3	SWI3
FFF0	FFF1	Reservado

Reconhecimento de sincronismo serve para indicar que a CPU está esperando por um sincronismo externo ou uma interrupção.

Reconhecimento de Halt indica que a CPU está em estado de Halt.

NON MASKABLE INTERRUPT (NMI) - Um nível lógico baixo nessa entrada indica à CPU que está sendo solicitada uma interrupção não mascarada. Uma NMI não pode ser inibida por programa e tem prioridade maior que FIRQ, IRQ ou interrupções por software. Durante o reconhecimento de uma NMI, o ponto atual de processamento é salvo no stack de hardware, para posterior retorno ao programa principal. Para que o sinal NMI seja reconhecido, este deve permanecer em nível lógico aíxo, no mínimo, o tempo de um ciclo do clock E.

FAST INTERRUPT REQUEST (FIRQ) - Um nível lógico baixo nessa entrada irá iniciar uma sequência de requisição de interrupção rápida. Porém, isso sómente ocorrerá se o flag F, do registrador do código de condição for zero. Essa sequência possui uma prioridade maior que o sinal IRQ.

INTERRUPT REQUEST (IRQ) - Um nível lógico baixo nesta entrada inicia uma requisição de interrupção desde que o flag I do registrador de código de condição esteja em nível lógico zero.

ENTRADA DE CLOCK E e Q - E e Q são os sinais de clock requeridos pela CPU MC809E. Q deve anteceder E; isto é, uma transição em Q deve ser seguida por uma transição similar em E, após um mínimo atraso. O endereçamento será válido no máximo 200 ns após a borda de descida do clock E e os dados serão colocados no barramento de dados após a borda de descida do clock E.

BUSY - Busy estará alto durante leitura e ciclos modificados de instruções de leitura-modificação-gravação e durante o acesso ao primeiro byte de uma operação de dois bytes (por exemplo: LDX, STD, ADDD).

O sinal Busy também estará em nível alto durante o primeiro byte de qualquer instrução indireta ou busca de vetor (por exemplo jump extendido, SWI indireto etc.).

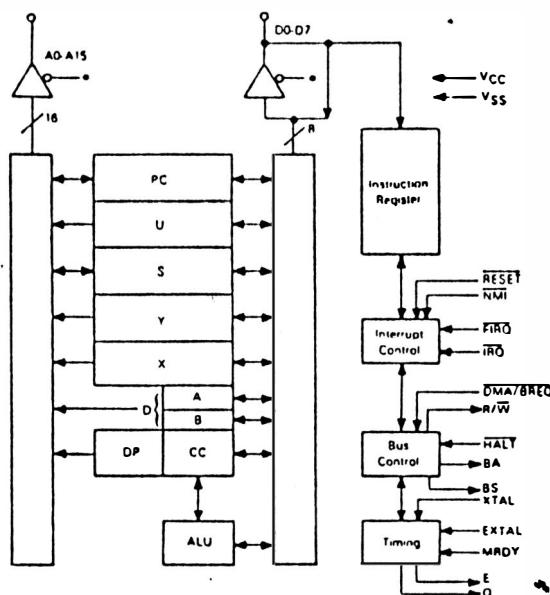
Em um sistema de multi-processamento, Busy indica a necessidade de adiar a retomada de controle do barramento pela CPU.

AVMA - Este sinal indica que a CPU usará o barramento no ciclo seguinte do barramento. Também é usado para sistemas de multiprocessamento.

LAST INSTRUCTION CYCLE - (LIC) Esse sinal é ativo em nível lógico alto no último ciclo de cada instrução e sua transição de alto para baixo indicará que o primeiro byte de um novo código de operação poderá ser colocado ao fim do presente ciclo do barramento.

THREE STATE CONTROL (TSC) - Esse sinal de entrada quando em nível lógico alto fará com que o barramento de endereços, dados e a saída R/W assumam um estado de alta impedância.

Descrição e diagrama de blocos do 6809E:



c) Acumuladores: São para os registradores A e B, de 8 bits, uso geral em cálculos aritméticos e manipulação de dados.

Certas instruções concatenam os registradores A e B para formar um registrador de 16 bits, chamado de registrador D, onde o registrador A fica sendo o byte mais significativo.

d) Registrador de Página direta - PD: Esse registrador é utilizado no modo de endereçamento direto. O conteúdo desse registrador é colocado na parte mais significativa do barramento de endereços (A8-A15) durante a execução de instruções de endereçamento direto.

e) Registradores indexados - X,Y: Esses registradores são usados no modo indexado de endereçamento, sendo ambos registradores de 16 bits.

f) Stack Pointer - U,S: São dois registradores de 16 bits cada um, sendo que o registrador S é o "hardware stack pointer" e o registrador U é o "software stack pointer".

O registrador S é usado automaticamente pela CPU durante instruções e armazena o endereço para o qual a CPU deve retornar após a execução da sub-rotina.

g) Contador de Programa - PC: o contador de programa é usado pelo processador para apontar para o endereço da própria instrução a ser executada ou o endereço do próximo dado a ser lido.

h) Registrador de código de condição - CC: O registro de código de condição define a condição da CPU em um determinado momento, de acordo com a tabela abaixo:

Bit	Nome	Condição definida
D7	E	Flag de entrada?
D6	F	Máscara do FIRQ
D5	H	Half Carry
D4	I	Máscara do IRQ
D3	N	Negativo
D2	Z	Zero
D1	V	"Overflow"
D0	C	"Carry"

D7 - E: Esse bit é setado em nível lógico 1 para indicar um estado de máquina completo.

D6 - F: Esse é o flag do sinal "fast interrupt request" (FIRQ). Interrupções vindas da entrada FIRQ não serão reconhecidas se este bit estiver setado em nível 1.

D5 - H: Esse bit é responsável pelo "half carry" e é usado para indicar que houve um "carry" do bit 3 da unidade lógica e aritmética, durante operações de soma, sómente.

D4 - I: Máscara do sinal "interrupt request". O microprocessador não reconhecerá o sinal IRQ enquanto este bit estiver setado em nível 1.

D3 - N: Esse flag serve para indicar, quando em alto, que a operação precedente resultou em um número negativo.

D2 - Z: Quando em 1, esse flag indica que a operação precedente resultou em zero.

D1 - V: Flag de "overflow", indica que a operação executa da excedeu a capacidade dos registradores internos, quando em nível 1.

D0 - C: É o "carry" flag e quando em nível lógico 1 indica que houve um "carry".

3. MODOS DE ENDEREÇAMENTO DE 6809

- Modo Intrínseco ou Inerente

Nesse modo de endereçamento o código da instrução contém todas informações necessárias à sua execução.

Por exemplo: ABX, DAA, SWI, ASRA, etc.

- Modo Imediato

Nesse modo, o endereço efetivo dos dados é aquele imediatamente posterior ao código de operação.

Por exemplo:

```
LDA #$20
LDY #$F000
LDY #L1
```

- Modo Extendido Direto

Nesse modo, o endereço do dado é especificado pelos dois bytes subsequentes ao código de operação.

Por exemplo:

```
LDA L1
STX L2
LDD $2000
```

- Modo Extendido Indireto

Esse é um caso especial do endereçamento indexado.

Por exemplo:

```
LDA [L1]
LDX [$FFFE]
STU [L2]
```

- Modo Direto

Esse modo é similar ao modo extendido, exceto que sómente um byte do endereço segue o código de operação. Esse byte especifica a metade menos significativa do endereçamento. Os outros oito bits do endereçamento são conseguidos através do registrador DP.

- Modo de Endereçamento por Registrador

Alguns códigos de operação são seguidos por um byte que define um registrador ou alguns registradores que serão usados pela instrução.

Por exemplo:

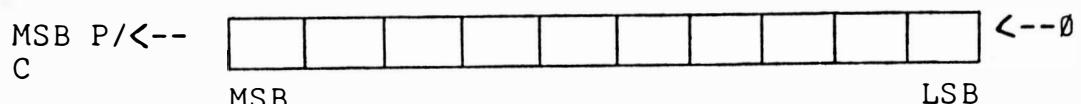
```
TFR X,Y
EXG A,B
PSHS A,B,X,Y
```

- Modo Indexado

Em todos endereçamentos indexados, um dos registradores de apontamento (X,Y,S,U e algumas vezes o próprio PC) são usados no cálculo do endereço efetivo do operando.

4. DESCRIÇÃO DE CADA INSTRUÇÃO INDIVIDUAL DO 6809E

- | | |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ABA | Adiciona acumulador B ao acumulador A.
Essa instrução adiciona o valor dos dois acumuladores e armazena o resultado no acumulador A. |
| ABX | Adiciona acumulador B ao Registrador de Índice X.
Essa instrução adiciona o valor do acumulador B ao registrador de índices X atende armazenar o resultado no registrador de índice X. ABX trata o valor do acumulador B como se esse não tivesse sinal. |
| ADCA | Essa instrução adiciona ao valor do acumulador o "carry flag" e o valor contido no endereço subsequente do código de operação, armazenando o resultado no acumulador A |
| ADCB | Idem anterior, usando acumulador B. |
| ADCD | Idem anteriores, usando o registro de 16 bits, D, formado pelos acumuladores A e B. |
| ADDA | Adiciona uma posição da memória ao acumulador A. Essa posição será endereçada pelo byte posterior ao código da operação ADDA e pelo byte contido no registrador de página direta (DP), sendo este o byte mais significativo. O resultado ficará armazenado em A. |
| ADDB | Idem anterior, com o acumulador B. |
| ADDD | Idem, usando o registrador de 16 bits D. Neste caso será somada ao registrador D a palavra constituída pelos dois bytes contíguos, sendo o primeiro endereçado da mesma forma que nas instruções ADDA e ADDB. |
| ANDA | Essa instrução efetua um "E" booleano entre o acumulador A e o valor da posição de memória endereçada pelo registrador de índice X. |
| ANDB | Idem acima, com o acumulador B. |
| ANDCC | Idem acima, com o registrador de código de condição CC. |
| ASLA | Desloca o acumulador A um bit à esquerda, colocando o bit mais significativo no "carry flag" e introduzindo 0 no bit menos significativo, conforme diagrama abaixo: |



ASLB	Idem acima, com o acumulador B.
ASL	Idem acima, usando o byte endereçado pelo "software stack pointer" (U). O registrador U é automaticamente incrementado com a execução desta instrução.
ASRA	Desloca o acumulador A um bit à direita, colocando o bit menos significativo no "carry flag" e conservando o valor do bit mais significativo.
ASRB	Idem acima, com o acumulador B.
ASR	Como ASL, efetuando deslocamento à direita.
BCC	Essa instrução efetua um salto (jump) para endereço especificado pela fórmula PC + nn + 2, onde nn é o valor do byte subsequente ao código de operação de BCC. Porém isso sómente ocorrerá se o "carry flag" for zero. NN deve estar contido no intervalo 127 (10000000) < nn < 127 (01111111). O valor anterior do PC é perdido, quando é executada esta instrução.
BCS	Idem anterior, utilizando a condição "carry flag"(C)=1.
BEQ	Idem anterior, utilizando o "zero flag" (Z)/ Será executada se Z=1.
BGE	Idem anterior, usando os flags N e V. Só será executada quando N=1 e V=1 ou N=0 e V=0.
BGT	Idem anterior, só sendo executada quando N=1 e V=1 ou N=0 e V=0 com Z=0 em ambas as condições.
BHI	Idem anterior, só executada se C=0 e Z=0/
BHS	Idêntica a BCC
BIT A	Essa é uma instrução de três bytes que efetua a operação booleana "E" do valor do acumulador A com o valor da posição de memória endereçadas pelos dois bytes subsequentes ao seu código de operação. Essa instrução só altera o registrador de código de condição.
BIT B	Idem acima, usando o acumulador B.
BLE	Igual a BCC, só que usando as condições Z=1 ou N=1 e V=0 ou N=0 e V=1.
BLO	Idem acima, usando a condição C=1.
BLS	Idem acima, usando as condições C=1 ou Z=1.

BLT	Igual a BCC, só que usando as condições N=1 e V=0 ou N=0 e V=1.
BMI	Idem anterior, usando a condição N=1.
BNE	Idem anterior, usando a condição Z=0
BPL	Idem, usando N=0
BRA	Essa instrução, sempre que encontrada, fará com que a CPU salte para o endereço PC+nn+2, onde nn é o valor do byte subsequente à instrução BRA. Essa instrução não altera em nada o processamento, sendo idêntica à instrução BRA, exceto no fato que o valor anterior do PC é salvo no "Hardware Stack Pointer", da seguinte forma: 1. Decrementa o registro S ("Hardware stack pointer") e armazena o byte menos significativo do PC no endereço atualmente apontado por S. 2. Decrementa novamente o registro S e armazena o byte mais significativo do PC neste endereço (S-2). O retorno à instrução BSR se dará quando for encontrada uma instrução RTS ou PULS PC.
BVC	Essa instrução opera similarmente à instrução BCC com a condição V=0.
BVS	Idem acima, usando a condição V=1.
CBA	Essa instrução compara os acumuladores A e B, efetuando a subtração B-A e setando os flags correspondentes.
CLC	Apaga o flag C.
CLF	Apaga o flag F (Fast interrupt mask).
CLI	Apaga o flag I (Interrupt mask).
CLIF	Apaga os flags I e F.
CLRA	Apaga o acumulador A.
CLRB	Apaga o acumulador B.
CLR	Apaga uma posição específica da memória, determinada pelos dois bytes subsequentes à instrução.

CLV	Apaga o flag V.
CMPA	Essa instrução compara o valor do acumulador com o byte subsequente à instrução, efetuando sua subtração (A é o subtraendo), alterando sómente o registro de código de condição (CC).
CMPB	Idem acima, usando o acumulador B.
CMPD	Neste caso é efetuada uma comparação de 16 bits. O valor a ser comparado situa-se nos dois bytes subsequentes à instrução.
COMA	Esta instrução complementa o acumulador A, ou seja, substitui 1 por 0 e 0 por 1.
COMB	Idem anterior, com o acumulador B.
COM	Complementa o endereço de memória especificado pelos dois bytes subsequentes à instrução.
CWAI	Essa instrução efetua a operação booleana "E" entre o registrador de código de condição e o byte subsequente ao byte do código da instrução, salva o conteúdo de todos registradores no "Hardware stack pointer" e paralisa todas operações até que acorra uma interrupção externa. Essa instrução serve para sincronizar a CPU com dispositivos externos.
DAA	Essa instrução converte o conteúdo do acumulador para a forma decimal.
DECA	Decrementa o acumulador A de uma unidade.
DEC B	Decrementa o acumulador B de uma unidade.
DEC	Decrementa um byte de memória, endereçado pelos dois bytes subsequentes ao byte da instrução.
DES	Decrementa o "Hardware stack pointer" de uma unidade.
DEX	Decrementa o registrador X de uma unidade.
DEY	Decrementa o registrador Y de uma unidade.
E ORA	Essa instrução executa um "OU EXCLUSIVO" entre o conteúdo do acumulador A e o byte posterior ao byte da instrução.
E ORB	Idem acima.
EXG r,r'	Troca o conteúdo do registrador r com r' e vice-versa.

INCA	Incrementa o acumulador A de uma unidade.
INC B	Incrementa o acumulador B de uma unidade.
INC	Incrementa a posição de memória endereçada pelos dois bytes da instrução.
INS	Incrementa o "Hardware Stack Pointer" de uma unidade.
INX	Incrementa registrador X de uma unidade.
INY	Incrementa registrador Y de uma unidade.
JMP	Efetua um desvio incondicional para o endereço fornecido pelos dois bytes subsequentes ao código da instrução.
JSR	Desvia para a sub-rotina endereçada pelos dois bytes subsequentes ao código da instrução salvando o valor do contador de programa no "Hardware Stack Pointer".
LBRA	Essa instrução pega o valor de 16 bits correspondente aos dois bytes posteriores ao código da instrução, efetua a soma desse valor com o PC+3 e dá continuidade à execução.
LBCC	Idem acima, porém só será efetuado se C=0.
LBCS	Idem, com a condição C=1.
LBEQ	Idem, com a condição Z=0.
LBGE	Idem, com a condição N=1 e V=1 ou N=0 e V=0.
LBGT	Idem, com a condição Z=0 e (N=1 e V=1 ou N=1 e V=0).
LBHI	Idem, com a condição Z=0 e C=0.
LBHS	Idêntica à LBCC.
LBLE	Idêntica à LBGT.
LBL0	Idêntica à LBCS.
LBLS	Idem acima com a condição Z=1 ou C=1.
LBLT	Idem acima, com a condição V=0 e N=1 ou V=1 e N=0.
LBMI	Idem acima, com a condição N=1.
LBNE	Idem, com a condição Z=1.
LBPL	Idem, com a condição N=0.
LBRN	Constitui um NOP de 4 bytes.
LBSR	Efetua um salto para a subrotina endereçada por PC+3+

	nnnn, onde nnnn são os dois bytes subsequentes ao código da instrução.
LBVS	Idem LBRA, com a condição V=1.
LDA	Carrega acumulador A com o conteúdo de memória endereçado pelos dois bytes posteriores ao código da operação.
LDB	Idem acima, usando o acumulador B.
LDI	Idem acima, usando o acumulador D.
LDS	Idem acima, usando o registrador S.
LDU	Idem acima, usando o registrador U.
LDX	Idem acima, usando o registrador X.
LDY	Idem acima, usando o registrador Y.
LEAS	Carrega um endereço de 16 bits no "Hardware stack pointer".
LEAU	Idem, para o registrador U.
LEAX	Idem, para o registrador X.
LEAY	Idem, para o registrador Y.
LSLA	Essa instrução é idêntica à ASLA.
LSLB	Idem, para ASLB.
LSL	Idem, para ASL.
LSRA	Essa instrução é idêntica à ASRA.
LSRB	Idem, para ASRB.
LSR	Idem, para ASR.
MUL	Essa instrução efetua a multiplicação de dois números , sem considerar o sinal.
NEGA	Efetua o complemento de dois do acumulador A.
NEG B	Idem, para o acumulador B.
NEG	Idem, para um byte de memória.
NOP	"No operation", não altera o processamento.
ORA	Similar à instrução ANDA, executando a operação OU.
ORB	Similar à ANDB, executando a operação OU.

OR	Similar à instrução AND, executando a operação OU.
ORCC	Similar à instrução ANDCC, executando a operação ON.
PSHU	Salva um ou todos registradores no "software stack pointer".
PSHS	Salva um ou todos registradores no "hardware stack pointer".
PULU	Recoloca nos registradores o valor que estiver armazenado no "software stack pointer".
PULS	Recoloca nos registradores o valor que estiver armazenado no "hardware stack pointer".
ROLA	Gira o conteúdo do registrador A à esquerda, através do "carry flag".
ROLB	Idem acima, usando o registrador B.
ROL	Gira byte da memória à esquerda, usando o "carry flag".
RORA	Igual à ROLA, só que gira à direita.
RORB	Idem acima, usando acumulador B.
ROR	Idem acima, usando a memória.
RII	Essa instrução efetua o retorno de uma subrotina de interrupção.
RTS	Essa instrução efetua o retorno de uma subrotina.
SBA	Essa instrução efetua a subtração do acumulador B do acumulador A.
SBCA	Subtrai o conteúdo do byte de memória endereçado pelos dois bytes subsequentes à instrução do valor contido no acumulador, gerando bit de transporte.
SBCB	Idem acima, usando acumulador B.
SEC	Essa instrução coloca o "carry flag" em nível 1.
SEF	Idem acima, para o "fast interrupt flag".
SEI	Idem acima, para o "Interrupt flag".
SEIF	Idem acima, para os flags I e F.
SEV	Idem acima, para o "Overflow flag" V.
SEX	Essa instrução transforma um número de 8 bits.

	(complemento de 2), armazenado no acumulador B, em um número de 16 bits (complemento de 2) no registrador D.
STA	Armazena o valor do acumulador A em um endereço de <u>memória</u> .
STB	Idem, para o acumulador B.
STD	Idem, para o registrador D.
STX	Idem, para o registrador X.
STY	Idem, para o registrador Y.
STU	Idem, para o registrador U.
STS	Idem, para o registrador S.
SUBA	Subtrai o byte de memória endereçado pelos dois bytes subsequentes à interrupção do valor do acumulador A.
SUBB	Idem acima, para o acumulador B.
SUBD	Idem acima, para o registrador D.
SWI	Essa instrução incrementa o PC, coloca em um o flag E e salva todos os registradores, com exceção do registrador S. A partir daí o programa será executado a partir do endereço colocado nas posições FFFA e FFFF, conforme definição da tabela 2.
SWI2	Análoga à instrução acima, com o vetor de interrupção localizado nos endereços FFF4 e FFF5.
SWI3	Idem, com vetor localizado nas posições FFF2 e FFF3.
SYNC	Essa instrução efetua o sincronismo da CPU com um evento externo, fazendo com que a CPU fique aguardando por uma interrupção.
TAB	Transfere conteúdo do acumulador A para o B.
TAP	Transfere conteúdo do acumulador A para o registrador de código de condição.
TFR	Essa instrução efetua a transferência de dados entre dois registradores quaisquer.
TBA	Transfere conteúdo do acumulador B para o acumulador A.
TPA	Transfere conteúdo do registrador de código de condição para registrador B.

TSTA	Testa o conteúdo do acumulador A, setando os flags N e Z de acordo com seu valor.
TSTB	Idem acima, com o conteúdo do acumulador B.
TST	Idem, com o valor da memória endereçado pelos dois bytes subsequentes.
TSX	Transfere o valor do registrador S para o registrador X.
TXS	Transfere o valor do registrador X para o registrador S.
WAI	Força o processador a esperar por uma interrupção.

Abaixo damos um exemplo de programa escrito em essembler o 6809:

PIACA	EQU	\$8001
PIADDA	EQU	\$8000
PIADA	EQU	\$8000
INTRP	EQU	\$0100
	ORG	\$0000
	LDS	\$0100
	CLR	\$40
	CLR	PIACA
	CLR	PIADDA
	LDA	#%00000101
	STA	PIACA
	ANDCC	#%11101111
WTRDY	TST	\$40
	BEQ	WTRDY
	SWI	

O símbolo % indica números binários.

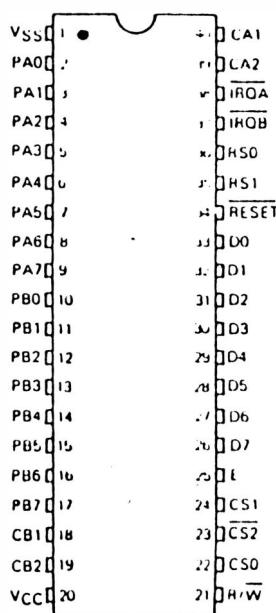
O símbolo \$ indica números hexadecimais.

O símbolo # indica modo de endereçamento direto.

5. ADAPTADOR DE INTERFACES E PERIFÉRICOS MC6821

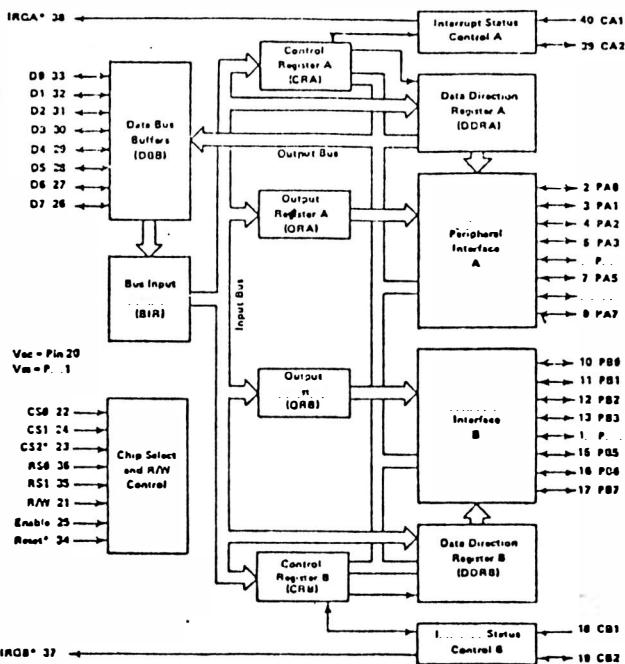
Esse dispositivo efetua o interfaceamento entre os diversos dispositivos da família 6800, possuindo para isso um barramento de dados e duas portas de entrada/saída.

Abaixo vemos sua pinagem e respectiva descrição.



- D0 a D7 - Barramento de dados, bidirecional.
- E - Entrada de habilitação, único sinal de temporização para a PIA.
- R/W - Entrada de leitura/gravação, indica a direção do fluxo de dados.
- RESET - Essa entrada, quando em nível zero, força o apagamento de todos os registradores internos da PIA.
- CS0,CS1 - Três entradas de "chip select", são usadas para seleção da PIA. Para que isso ocorra é necessário que CS0=1; CS1=1 e CS2=0.
- RS0/RS1 - Linhas de seleção do registro. Essas linhas são usadas para selecionar os vários registradores da PIA.
- IRQA/IRQB - Linhas de requisição de interrupção, ativas em nível lógico baixo.

Abaixo temos o diagrama de blocos da PIA:



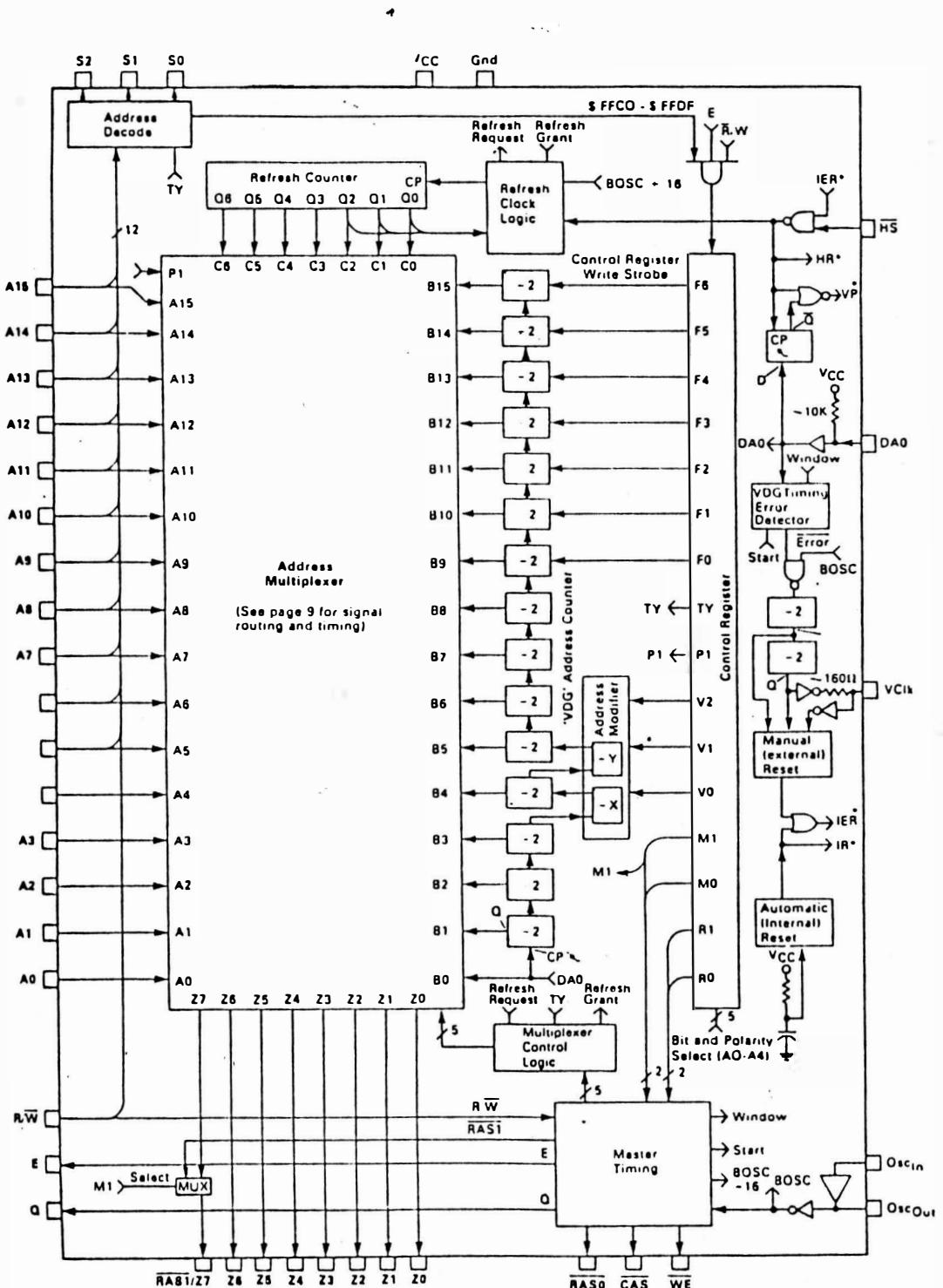
- PA₀-PA₇ - A PIA possui duas portas (PA₀-PA₇ e PB₀-PB₇) com 8 linhas bidirecionais cada, podendo ser programada de forma que cada bit dessas portas atue como uma entrada ou saída. A principal diferença entre as duas portas é que a porta B tem a capacidade de ficar em "three state".
- CA1/CB1 - Entradas de interrupção, controlam os flags de interrupção internos à PIA. O nível ativo desses sinais pode ser controlado através da programação do registrador de controle.
- CA2 - Esse pino pode ser programado para agir como uma entrada de interrupção ou como uma saída de controle para periféricos.
- CB2 - Idem acima, podendo ficar no estado "three state".

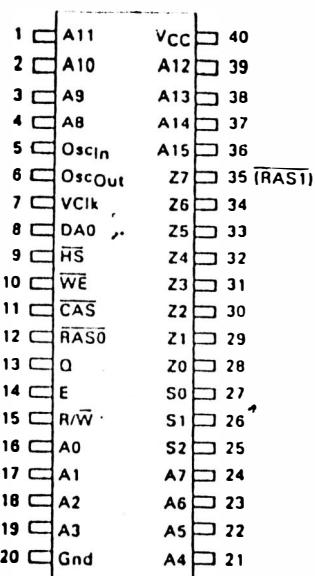
INICIALIZAÇÃO DA PIA

Durante a inicialização da PIA (RESET), todos os registradores são zerados, forçando as portas PA₀-PA₇; PB₀-PB₇. CA2 e CB2 a atuarem como entrada, desabilitando todas interrupções.

MC6883 ou 74LS783N - CONTROLADOR DE MEMÓRIA DINÂMICA

Abaixo vemos o diagrama de blocos e a pinagem do MC6883 ou 74LS783.





- A₀-A₁₅ Entradas de endereçamento, vindas da CPU.
- E,Q Entradas de clock que alimenta a CPU).
- Z₀-Z₇ Saídas de endereçamento para as RAM's dinâmicas. O MC6883 pode ser programado para endereçar memórias de 4K, 16K ou 64K. No caso de memórias de 4K poderão ser endereçados 2 bancos, usando as saídas de Z₀ a Z₅, ficando o bit Z₇ reservado para gerar o sinal de RAS1, para o 2º banco de RAM. O mesmo processo é usado para endereçar 2 bancos de 16K de RAM, usando as linhas de Z₀ a Z₆.
- R/W Entrada de leitura/gravação, deve ser conectada a CPU.
- WE Essa entrada, quando em nível lógico baixo zero, habilita a CPU a efetuar uma leitura ou gravação nas RAM's.
- CAS Sinal de saída, informa às RAM's que a parte mais significativa do endereço está presente nas saídas Z₀ a Z₇.
- S₂,S₁,S₀ Três saídas binárias, podem ser usadas em associação com um 74LS138 para efetuar mapeamento e seleção de memória.
- OSCIN e OSCOUT Entradas para o clock de sincronização que pode ser de 14,31818 MHz ou 16MHz.
- DA₀ Entrada do bit menos significativo do endereço de vídeo.
- HS Sincronismo horizontal; entrada; serve para reestar os 4 bits menos significativos do contador de vídeo.

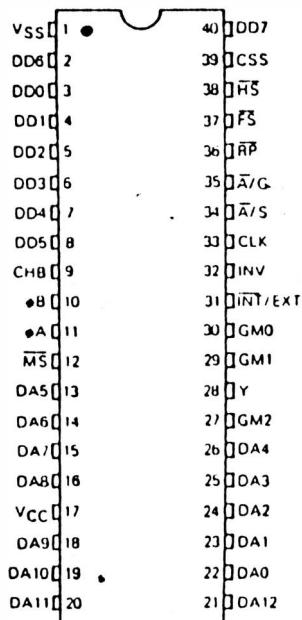
VCLK Saída, com frequência de 3,579545 MHz para uso no clock do VDG.

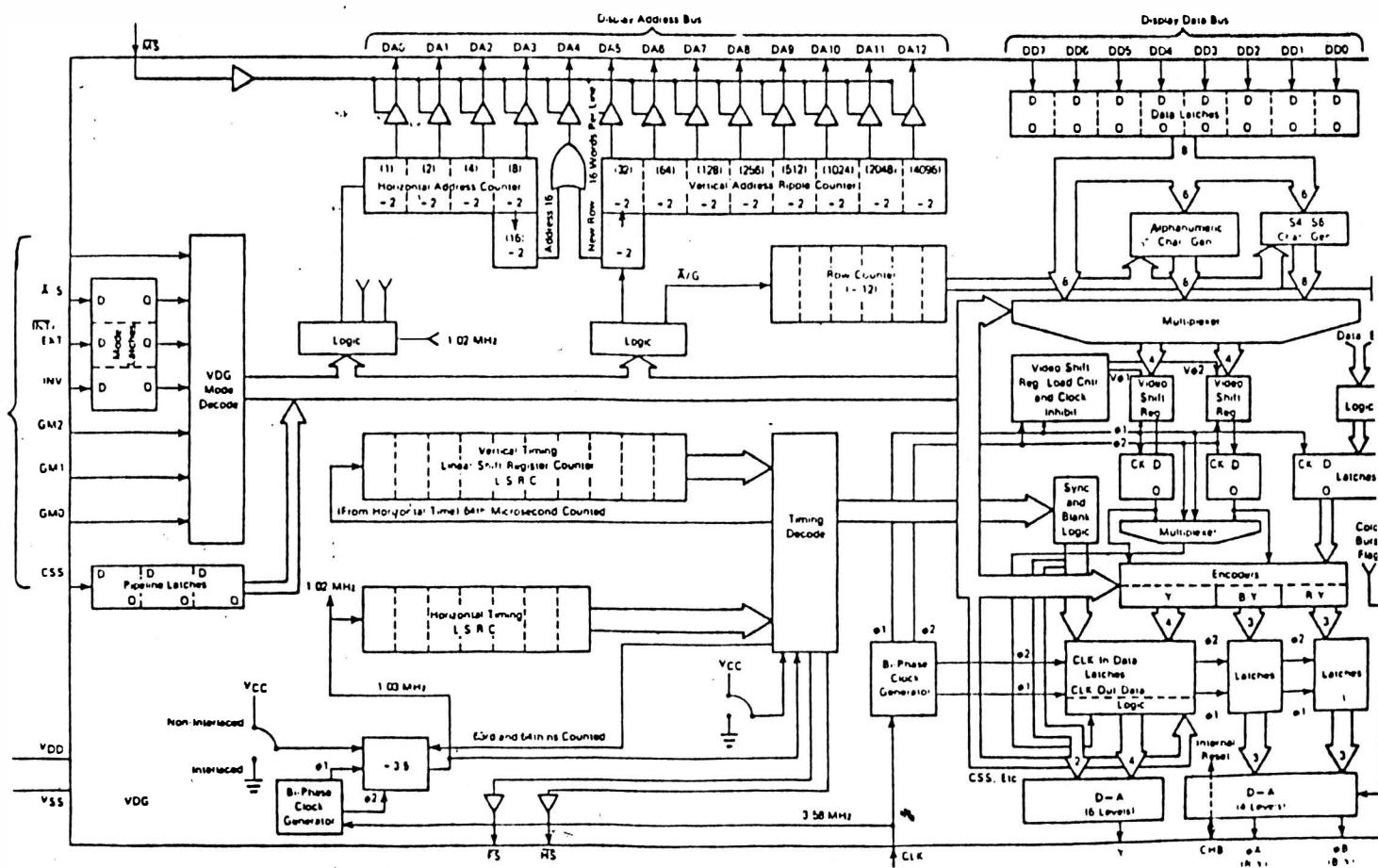
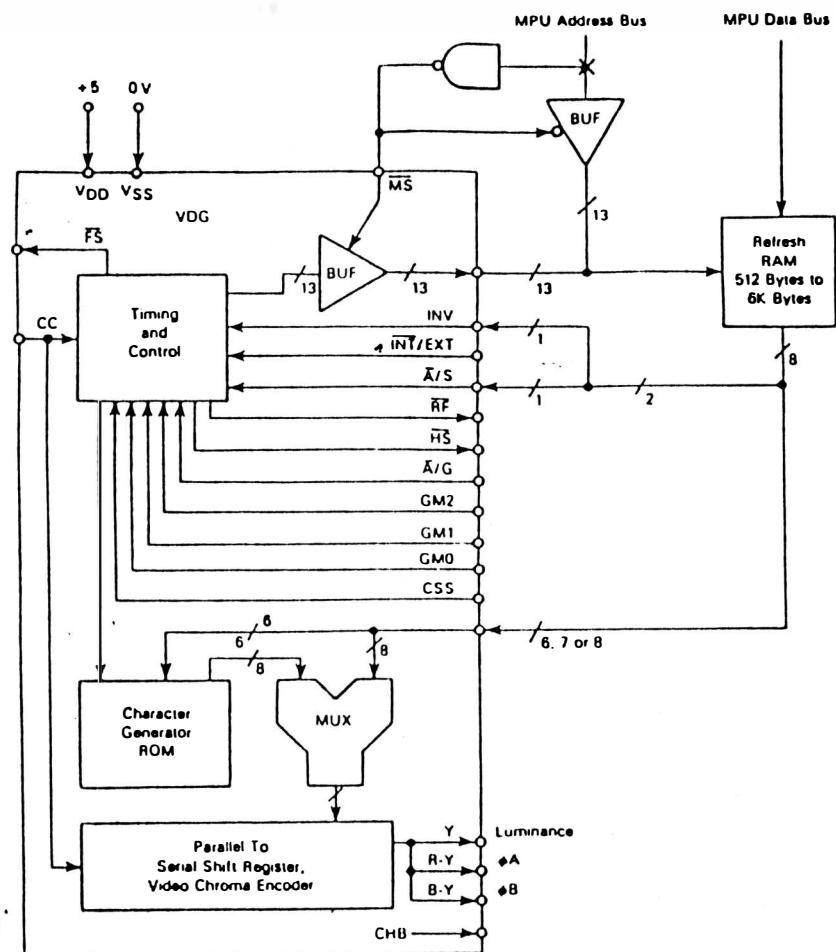
A função do 6883 é de efetuar o controle das RAM's dinâmicas (refresh, multiplexação de endereços e endereçamento) economizando dessa forma vários integrados MSI que seriam usados para essa função.

Além disso, esse chip também pode efetuar o controle do vídeo, se for utilizado o VDG 6847.

GERADOR DE VÍDEO MC6847

Abaixo vemos a pinagem, o diagrama de blocos simplificado e o diagrama de blocos detalhados do gerador de vídeo VDG MC6847.





6. DESCRIÇÃO DA PINAGEM

DD0-DD7	Entrada de dados para o VDG.. No caso de informação alfanumérica, esses bits referem-se ao código ASCII padrão. Nesse código, somente 6 bits são usados para definir o carácter, podendo os outros dois serem usados para chaveamento de cõr, vídeo reverso, carac teres alternativos, etc.
DA0-DA12	Linhas de endereçamento, servem para efetuar a var redura da memória de vídeo.
ØA, ØB, Y, CHB	Quatro saídas analógicas de vídeo ØA, ØB e Y sele cionam 1 entre 8 possíveis cõres, levam as informa ções de luminância, sincronismo e apagamento CHB é uma saída analógica que providencia um nível DC de referência para as saídas ØA e ØB, eliminando assim possíveis diferenças entre o VDG e o modulador.
CLK	Entrada do clock de 3,579545KHZ, padrão para TV a cõres.
FS	Saída, usada para sincronização. A descida desse si nal coincide com o fim da área efetiva do vídeo. A subida coincide com o retraço vertical.
HS	Saída de sincronismo horizontal.
RP	Entrada, serve para selecionar o uso de um gerador de carácter externo.
	GM0, GM1, GM2, CSS, INV, A/G.A/S, INT, EXT - Entradas de seleção de modo, de acordo com a tabela abaixo:-

A/G	A/S	INT/EXT	INV	GM2	GM1	GMØ	MODO	Número cores
0	0	0	0	X	X	X	Alfanumérico interno	2
0	0	0	1	X	X	X	Alfanumérico interno	2
0	0	1	0	X	X	X	Alfanumérico externo	2
0	0	1	1	X	X	X	Alfanumérico externo	2
0	1	0	X	X	X	X	Semigráfico 4 (SG4)	8
0	1	1	X	X	X	X	Semigráfico 6 (SGG)	8
1	X	X	X	0	0	0	Gráfico 64x4 (CG2)	4
1	X	X	X	0	0	1	Gráfico 128x64 (RG1)	2
1	X	X	X	0	1	0	Gráfico 128x64 (CG2)	4
1	X	X	X	0	1	1	Gráfico 128x96 (RG2)	2
1	X	X	X	1	0	0	Gráfico 128x96 (CG3)	4
1	X	X	X	1	0	1	Gráfico 128x192(RG3)	2
1	X	X	x	1	1	0	Gráfico 128x192(CG6)	4
1	X	X	X	1	1	1	Gráfico 256x192(RG6)	2

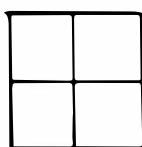
7. MODOS DE APRESENTAÇÃO DO VÍDEO

Há basicamente dois modos de operação do VDG: o modo 1, que contém a apresentação de caracteres alfanuméricos e gráficos limitados e o modo 2, que contém a apresentação de caracteres gráficos.

No modo 1, há 4 submodos alfanuméricos e 2 gráficos e no modo 2, há 8 submodos gráficos. A maneira de selecionar os diversos modos está descrita na tabela anterior.

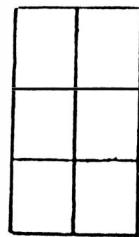
No modo 1 o vídeo é dividido em 32 colunas e 16 linhas, requerendo 512 bytes de memória. Cada carácter é formado por uma matriz 8x12. Os caracteres podem ser verdes sobre um fundo verde-escuro ou laranja sobre fundo laranja-escuro.

O modo semi gráfico 4 apresenta caracteres constituídos por 4 pixels, como mostrado abaixo:



ELEMENTO DE IMAGEM
(PIXEL)

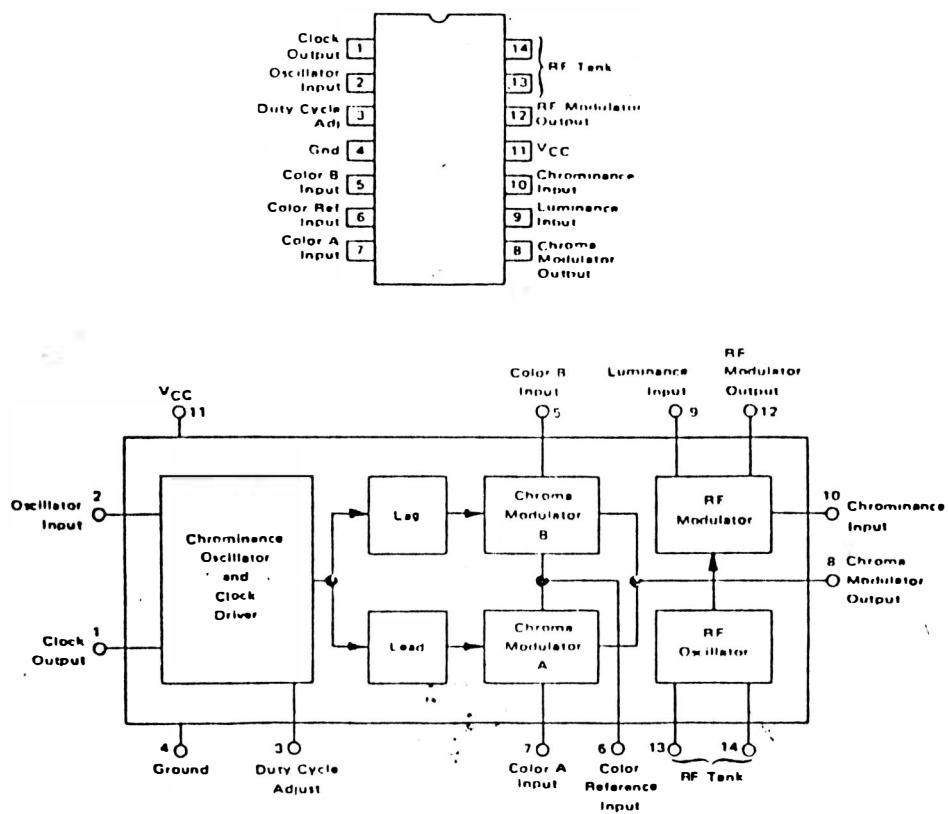
Os caracteres do modo semigráfico 6 encontram-se descritos a seguir:

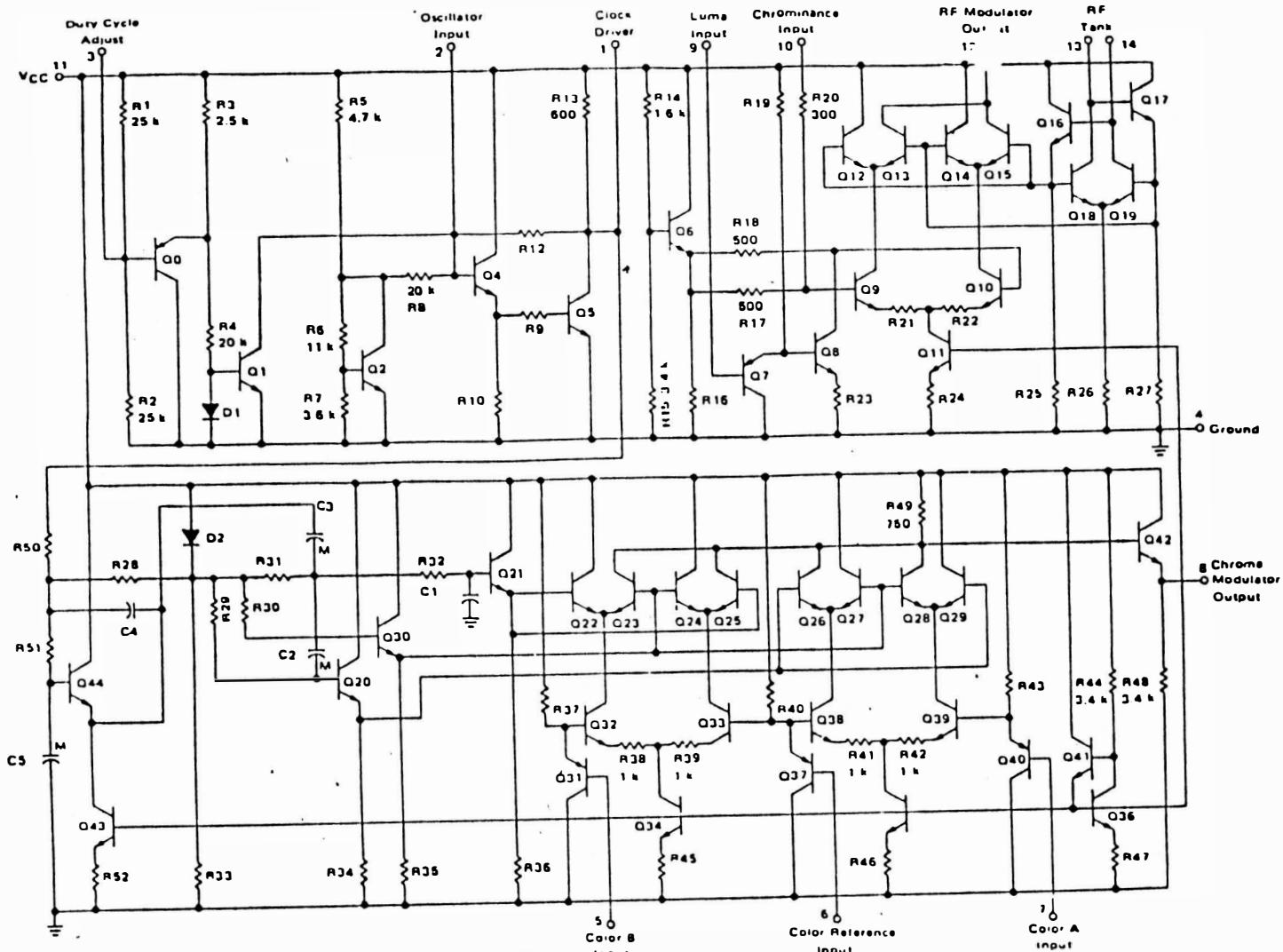


ELEMENTO DE IMAGEM

MC 1372 - CIRCUITO MODULADOR DE VÍDEO

Abaixo vemos a pinagem, diagrama de blocos e circuito elétrico do MC1372:





A função do MC1372 é de tornar os sinais fornecidos pelo MC6847 compatíveis com o vídeo, efetuado a mistura desses sinais e a sua modulação nos canais 3 ou 4.

Descrição da pinagem:

1- Saída de clock - Essa saída fornece uma onda quadrada cuja frequência é igual à frequência da subportadora de crominância.

2- Entrada do oscilador - Entrada de realimentação do oscilador da subportadora de cor.

3- Ajuste do ciclo de trabalho - Um nível DC nesta entrada ajusta o ciclo de trabalho do sinal de saída de clock. Se este pino estiver desconectado, o ciclo de trabalho deste sinal será de aproximadamente 50%.

4- Terra

5- Entrada de côr B - Entrada para o modulador de crominância B, cuja fase deve estar adiantada de 100 graus da entrada de côr A. A amplitude e polaridade da saída do modulador corresponde à diferença entre este sinal e a entrada de referência de côr.

6- Entrada de referência de côr - Um nível DC nessa entrada estabelece a tensão de referência para a comparação das entradas de côr A e B.

7- Entrada de côr A - Análogo à entrada de côr B, usando o modulador de crominância A.

8- Saída do modulador de crominância - Essa saída corresponde à soma vetorial das entradas de côr A e B.

9- Entrada de luminância - Entrada para o modulador de RF.

10- Entrada de crominância - Entrada para o modulador de crominância.

11- Alimentação - +5V

12- Saída do modulador de RF - Saída do sinal já misturado e modulado.

13 e 14 - Tanque RF - Essas entradas determinam a frequência do oscilador RF.

8 . DESCRIÇÃO GERAL

O CP 400 em sua versão básica é constituído por um módulo CPU que possui as seguintes características:

16K de EPROM com sistema monitor e interpretador basic.

16 ou 32K de RAM dinâmica.

55 teclas.

1 conector RCA para ligação à TV.

1 conector DIN para ligação com monitor de vídeo.

2 conectores DIN para ligação JOYSTICK.

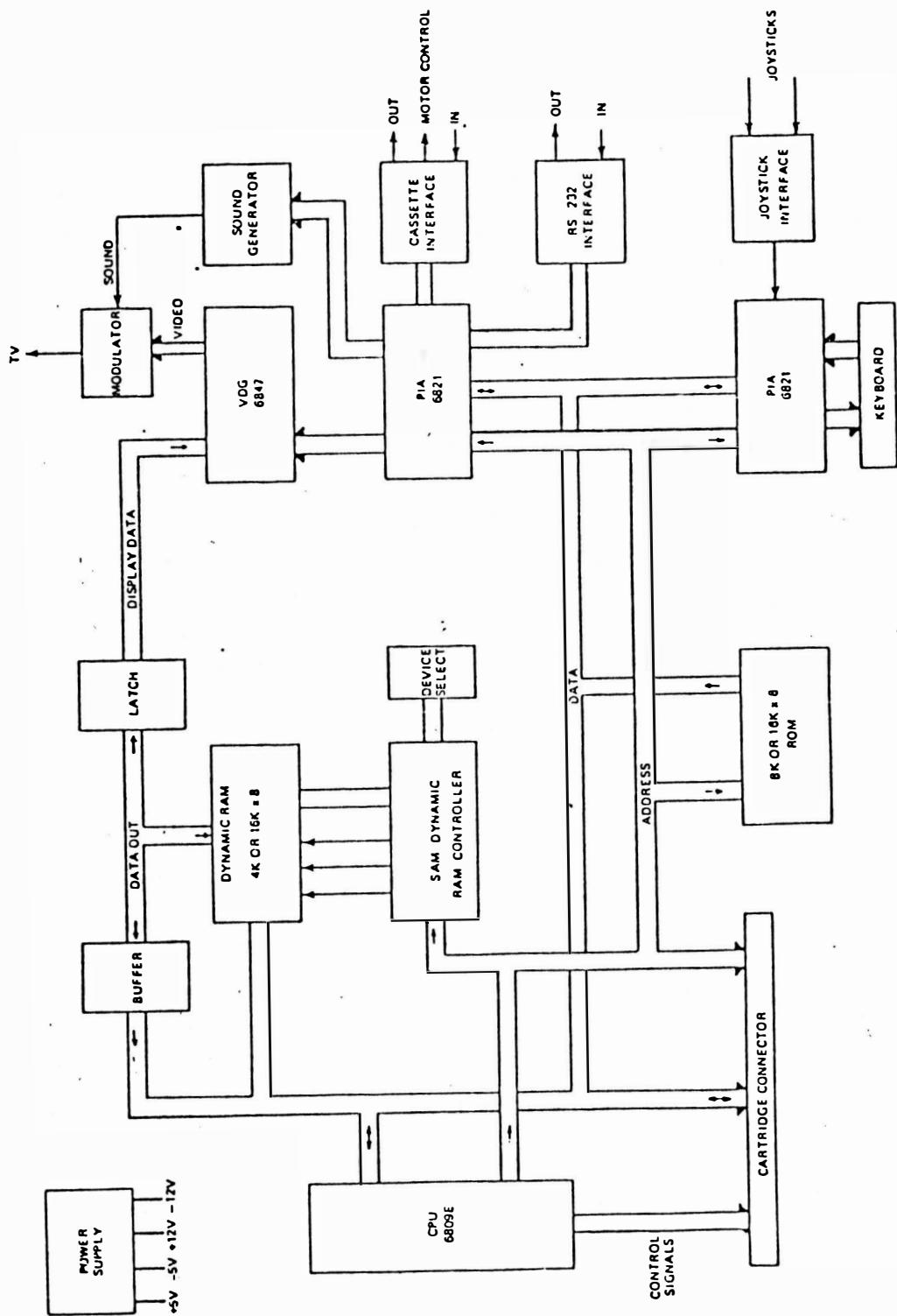
1 conector DIN para saída/entrada gravador.

1 conector DIN para saída/entrada serial (RS232C)

Conecotor para ligação de cartuchos de EPROM.

9. DESCRIÇÃO CIRCUITO ELÉTRICO PROLÓGICA 12

Abaixo vemos o diagrama de blocos da placa P12:



a) Mapeamento de Memória

A memória do CP 400 em sua configuração padrão máxima está mapeado de acordo com a tabela abaixo:

0000 a 8000	32K RAM
8001 a A000	8K EPROM
A001 a C000	8K EPROM
C001 a FF00	16K cartucho expansão
FF00 a FFBF	Controle das PIA'S
FFC0 a FFDF	Registradores de controle da SAM
FFE0 a FFFF	Vetores de interrupção e inicialização da CPU

Os endereços de controle das PIA'S (FF00 a FFBF) são os seguintes:

PIA CI 14

FF00 - Bit 0	- linha 1 do teclado ou joystick direito
Bit 1	- linha 2 do teclado ou joystick esquerdo
Bit 2	- linha 3 do teclado
Bit 3	- linha 4 do teclado
Bit 4	- linha 5 do teclado
Bit 6	- linha 6 do teclado
Bit 7	- entrada de comparação do joystick

FF01 - Bit 0	controle da entrada de interrupção
Bit 1	do sincronismo horizontal (63,5 microssegundos)
Bit 2	- normalmente 1
Bit 3	- SELA, CI15
Bit 4	- sempre 1
Bit 5	- sempre 1
Bit 6	- não usado
Bit 7	- flag de interrupção do sincronismo horizontal

FF02 - Bit 0 - coluna 1 do teclado
Bit 1 - coluna 2 do teclado
Bit 2 - coluna 3 do teclado
Bit 3 - coluna 4 do teclado
Bit 4 - coluna 5 do teclado
Bit 5 - coluna 6 do teclado
Bit 6 - coluna 7 do teclado
Bit 7 - coluna 8 do teclado

FF03 - Bit 0

controle da entrada de interrupção do sincronismo vertical (16,67 milissegundos)
Bit 1 - normalmente 1
Bit 3 - SELB, CI15
Bit 4 - sempre 1
Bit 5 - sempre 1
Bit 6 - não usado
Bit 7 - flag de interrupção do sincronismo vertical

Os endereços de FF04 a FF20 não são usados. Entre os endereços FF20 e FF23 temos a segunda PIA, CI17, controlada como segue:

FF20 - Bit 0 - entrada de dados do cassete
Bit 1 - saída de dados da RS232C
Bit 2 - LSB do conversor D/A 6 bits
Bit 3 - saída do conversor D/A
Bit 4 - saída do conversor D/A
Bit 5 - saída do conversor D/A
Bit 6 - saída do conversor D/A
Bit 7 - MSB do conversor D/A 6 bits

FF21 - Bit 0

controle de status da RS232C
Bit 1
Bit 2 - normalmente 1
Bit 3 - controle do motor do cassete 0-desligado 1-ligado

FF22 - Bit 0 - entrada de dados da RS232C
Bit 1 - saída de som
Bit 2 - entrada de extensão da RAM 0 - 16/4KB
1 - 16/48KB
Bit 3 - saída controle VDG CSS
Bit 4 - saída controle VDG GM0 e INT/EXT
Bit 5 - saída controle VDG GM1
Bit 6 - saída controle VDG GM2
Bit 7 - saída controle VDG A/G

FF23 - Bit 0 - controle da interrupção gerada pelo dispositivo
Bit 1 positivo conectado ao soquete do cartucho
Bit 2 - normalmente 1
Bit 3 - habilitação saída som
Bit 4 - sempre 1
Bit 5 - sempre 1
Bit 6 - não usado
Bit 7 - flag de interrupção do dispositivo do cartucho

Os endereços de FF40 a FFBF não são usados.

Os últimos endereços de RAM contém os vetores de interrupção e reset, conforme descrito abaixo:

OBS:- Tais endereços também podem ser acessados pela faixa FFF2 - FFFF.

FFFF ou BFFF	LSB vetor reset
FFFE ou BFFE	MSB vetor reset
FFFD ou BFFD	LSB vetor NMI
FFFC ou BFFC	MSB vetor NMI
FFFB ou BFFB	LSB vetor SWI1
FFFA ou BFFA	MSB vetor SWI1
FFF9 ou BFF9	LSB vetor IRQ
FFF8 ou BFF8	MSB vetor IRQ
FFF7 ou BFF7	LSB vetor FIRQ
FFF6 ou BFF6	MSB vetor FIRQ
FFF5 ou BFF5	LSB vetor SWI2
FFF4 ou BFF4	MSB vetor SWI2
FFF3 ou BFF3	LSB vetor SWI3
FFF2 ou BFF2	MSB vetor SWI3

Os endereços de FFF1 a FFF0 não são usados.

Para a seleção dos diversos dispositivos de memória é utilizado o CI28 (74LS138), controlado pelas saídas S0, S1 e S2 da SAM. Assim, para as diversas combinações dessas saídas teremos os seguintes CI'S acessados:

S2	S1	S0	Dispositivo acessado:
0	0	0	buffer bidirecional das RAM'S (CI26)
0	0	1	CI12 (8K inferior das EPROM'S)
0	1	0	CI11 (8K superior das EPROM'S)
0	1	1	seleção do cartucho externo
1	0	0	chip select PIA CI14
1	0	1	chip select PIA CI17
1	1	0	seleciona entre os endereços FF40 e FF5F do cartucho
1	1	1	não usado

A seleção entre memórias de 16K ou 32/48K é feito através dos jumps 11,12,13,14,15,16,17 e 18.

Dispositivos de I/O

Entrada/saída RS232C

A saída para RS232C é controlada diretamente pela porta PA1, sendo usado o comparador CI20 para fornecer o nível correto de saída (entre - 12V e +12V). Para entrada é usado o mesmo procedimento, sendo que esta é feita pela porta PB0. A porta CA1 é usada para verificar o status da RS232C.

Conversor digital/analogico

O CP 400 possui um conversor D/A de 6 bits, constituído pelo buffer CI18 e os resistores R52 a R57 e o capacitor de efeito C34.

A tensão de saída pode ser calculada, aproximadamente pela fórmula.

$$V_s = (N \cdot 0,0715) + 0,25$$

onde N é o valor decimal dos 6 bits (de 0 à 63)

A saída desse conversor é usada com saída sonora, saída de cassete e saída para controle de joystick.

Entrada/saída cassette

Para a saída de cassete é usado o sinal vindo do conversor D/A, que fornece uma forma de onda senoidal, com frequências variando entre 2400 ou 1200 Hertz, sendo que um ciclo em 1200Hz representará um "zero" e um ciclo em 2400Hz representará um "um". Como o nível de saída do conversor D/A pode ser de até 6V, este nível é atenuado para 1V pelos resistores R24 e R25.

O circuito de leitura é constituído por um "detetor de pinagem por zero", CI16. R66 é o resistor de resistor de realimentação usado para evitar que o comparador comece a oscilar.

Componentes utilizados (principais)

CI13 - MC6809E

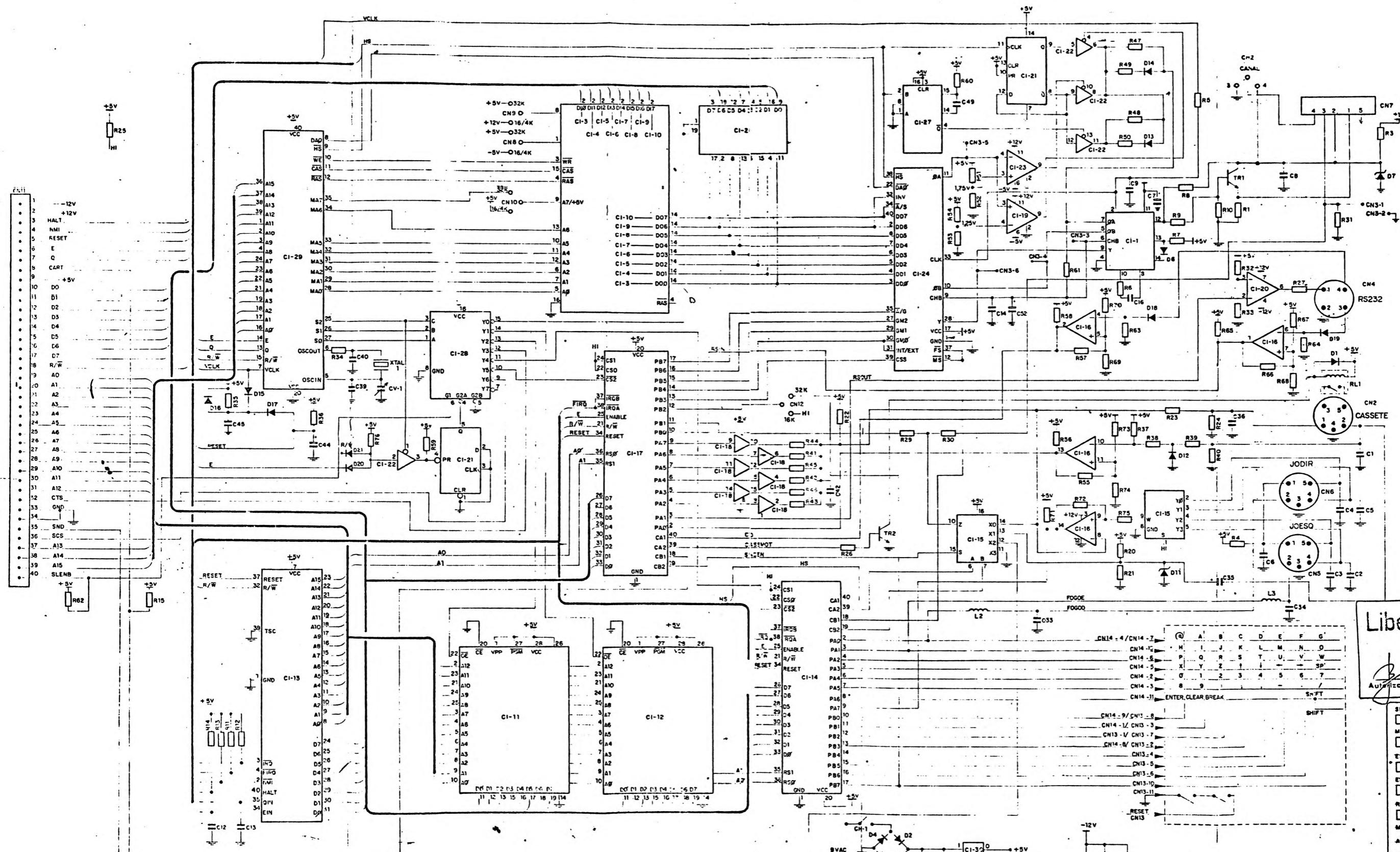
CI17 e CI14 - MC6821

CI11 e CI12 - EPROM 2764

CI 1 - MC1372

CI29 - MC6883 ou 74LS783N

CI24 - MC6847



Liberado por FSO

B D.A.T. 24'40'84
Autenticado Departamento Data

SISTEMA ADOTADO	<input type="checkbox"/> RÍGIDO	<input type="checkbox"/> FLEXÍVEL	<input type="checkbox"/> HÍBRIDO	
MATERIAL ADOTADO	<input type="checkbox"/> FENOLITE	<input type="checkbox"/> FIBRA DE VIDRO	<input type="checkbox"/>	
ESPESSEURA DO MATERIAL EM MM	16	20	30	<input type="checkbox"/>
ESPESSEURA DO COBRE	7,5 μ	35 μ	70 μ	<input type="checkbox"/>
RELAÇÃO ENTRE FACES	<input type="checkbox"/> SIMPLES	<input type="checkbox"/> DUPLA	<input type="checkbox"/>	
METALIZAÇÃO SUPERFICIAL	<input type="checkbox"/>			
ALTURA DO REVESTIMENTO	<input type="checkbox"/>			
METALIZAÇÃO DOS Furos	<input type="checkbox"/>			
TOLERÂNCIA DE FURAÇÃO	<input type="checkbox"/>			
TOLERÂNCIA DE CORTE	<input type="checkbox"/>			
DOURADAÇÃO DE CONECTORES	<input type="checkbox"/>			
	SIM	NÃO		

PROLÓGICA		ESTE DESENHO E TODAS AS INFORMAÇÕES DAS CONTAS CONSTITUÍM-SE EM PRO- PEITADO DA PROLÓGICA IND E COM DE MI- CROCOMPUTADORES LTD, NÃO PODENDO SER EMBODIZADO, COPIADO, NIÉ PODE SER DIFUSO A TERCEIROS, SEM AUTORIZAÇÃO PRA FICHTO	DES.Nº 030 FAZ-0-00
TOL. GERAL (MILIMETRO)	EQUIPAMENTO	DESPON R MURGATTE	DATA 14/07/84
0,0-0,3	CP-400	VERIF PUN	DATA
0,0-0,3	0,0-0,202		
0,0-0,3	0,0-0,250		
SUB CL. MUNTO NT		DESC DA PLACA PROLÓGICA 12	
ESCALA	ESTE DESENHO ESQUEMA ELETTRICO DA PLACA PROLÓGICA 12		