

CBPF-NT-001/89  
PROGRAMADOR DE PULSO PARA BARRAMENTO S100 APLICADO A  
RESSONÂNCIA MAGNÉTICA NUCLEAR PULSADA

por

Geraldo R.C. CERNICCHIARO, Marcelo G. RUDGE  
e Marcelo P. ALBUQUERQUE

Centro Brasileiro de Pesquisas Físicas - CBPF/CNPq  
Rua Dr. Xavier Sigaud, 150  
22290 - Rio de Janeiro, RJ - Brasil

Resumo: Neste trabalho descrevemos o projeto e a construção de um gerador de pulsos de quatro canais em uma placa padrão S100 e seu software de controle por microcomputador.

O microcomputador tem total controle sobre o gerador de pulsos, que possui sete parâmetros programáveis, definindo a posição de quatro pulsos e a largura dos três primeiros. Permite, também, a repetição automática de um intervalo selecionado.

Este gerador de pulsos é controlado por um software desenvolvido na linguagem C, e é usado em experiências de Ressonância Magnética Nuclear pulsada.

**PALAVRAS CHAVE :**

Programador de Pulso; Interface; Barramento S100;  
Hardware RMN; Gerador de Pulso.

## INTRODUCAO

Numa medida de Ressonância Magnética Nuclear (RMN) pulsada [2,5,6,8], a amostra é submetida a campos magnéticos oscilantes obtidos através de pulsos de rádio frequência (RF) com amplitude, largura e separação definidos pelo programador de pulsos, de acordo com a necessidade. A seleção desses parâmetros que caracterizam os pulsos era feita manualmente e com poucas opções para cada um deles. O programador de pulsos descrito neste trabalho foi desenvolvido com o objetivo de automatizar a seleção destes parâmetros no laboratório de RMN.

Anteriormente, em uma experiência típica, para se gerar "N" pulsos, era necessária a utilização de um microcomputador especificamente projetado para controlar a separação entre estes, e mais "N" módulos, programados manualmente, para controlar a largura desses respectivos pulsos.

Com o emprego do programador de pulsos, torna-se dispensável a utilização dos módulos e tanto a separação quanto a largura passam a ser definidos por um único programa de controle, economizando tempo e simplificando a operação.

Firmas especializadas em instrumentação eletrônica já possuem equipamentos complexos e sofisticados para tais experiências. Mas o alto custo desses equipamentos, aliado às dificuldades de importação e manutenção, nos levou a considerar a viabilidade de desenvolvê-lo em nossos laboratórios.

Um aparelho que satisfizesse somente às exigências das experiências de RMN desenvolvidas no laboratório do CBPF implicaria em uma redução considerável do custo do projeto, e mais, um conhecimento completo do circuito desenvolvido, a facilidade de manutenção e a disponibilidade de dispositivos eletrônicos no mercado, para reposição, foram fatores determinantes para sua execução.

## HARDWARE

O programador de pulsos é compreendido basicamente por duas funções: o posicionamento de pulsos no tempo, definido como geração de pulsos, e a largura dos respectivos pulsos. Os três primeiros pulsos excitam a amostra, enquanto que o quarto e último ativa o sistema de aquisição do sinal de resposta, dispensando para tal, o controle da variação da largura.

O programador de pulsos é composto por 35 componentes, dispostos em grupos, em um cartão padrão e100, de forma organizada e eficiente a fim de possibilitar uma fácil e rápida localização de qualquer um de seus componentes.

A figura 1 nos mostra o diagrama em blocos da interface constituída pelos seguintes módulos: decodificação, geração dos pulsos e largura dos pulsos.

A figura 2 nos mostra o diagrama de estado de operação onde numa primeira etapa é obtida a formação de quatro pulsos a partir de um sinal de sincronismo, que significa a taxa de repetição dos pulsos. E na outra etapa é definido a largura de três desses pulsos.

No módulo de decodificação é usado o circuito integrado 74LS138 (3/8 line decoders multiplexers ) [7] que é responsável pela habilitação das portas de entrada e saída, através da seleção do endereço associado a cada uma delas.

A Interface de Comunicação Paralela 8255A (PPI) [1] possui três portas de oito bits cada, cujo o sentido ( entrada e saída ) pode ser escolhido através do software. No programador de pulsos, as três portas operam somente como saídas de dados, utilizando-se, portanto, do modo 0 (figura 3). Esse modo permite a programação para entrada ou, como no caso, para a saída de dados.

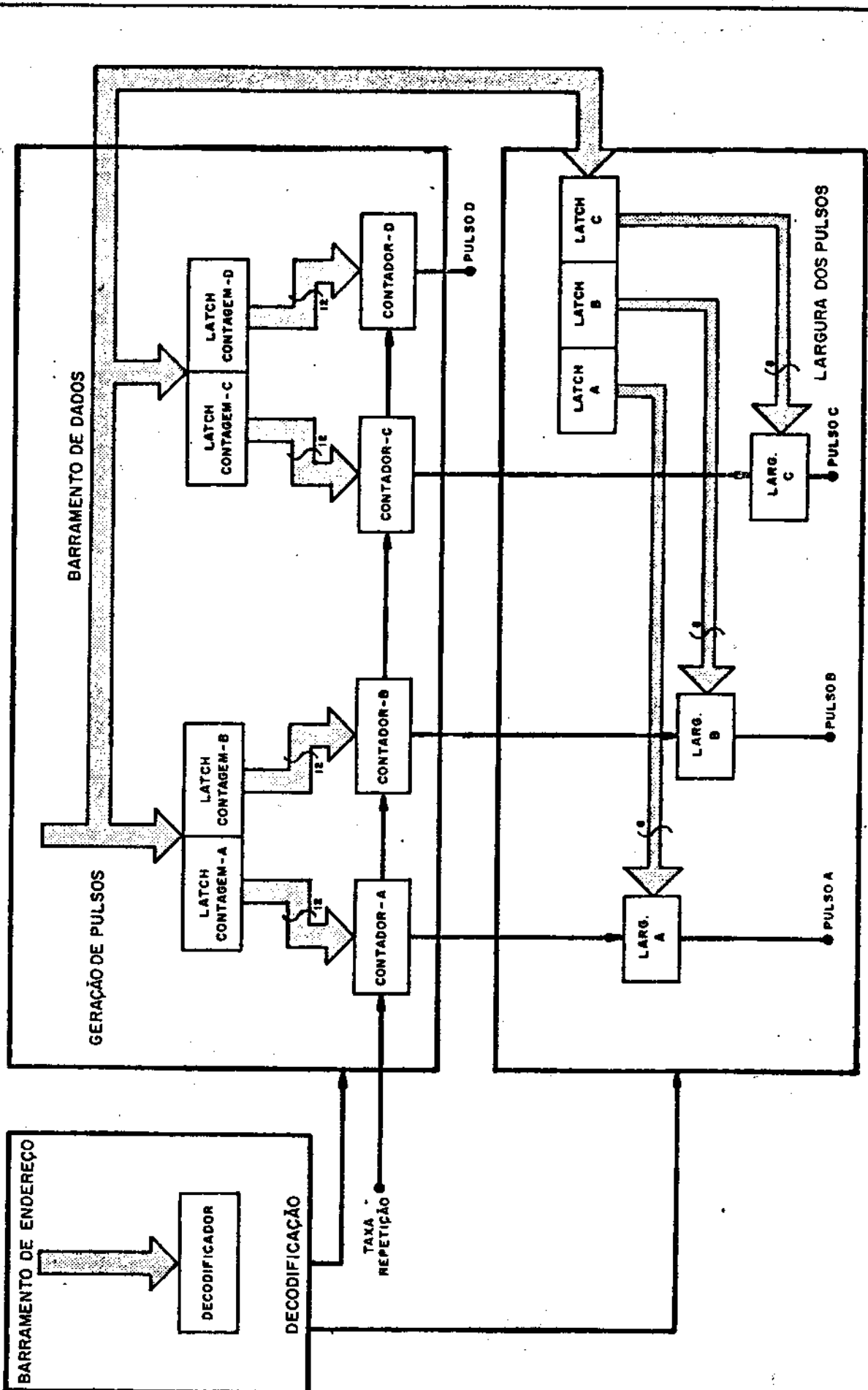


Fig. 1 DIAGRAMA EM BLOCOS

DIAGRAMA DE ESTADO PARA GERACAO DE PULSOS

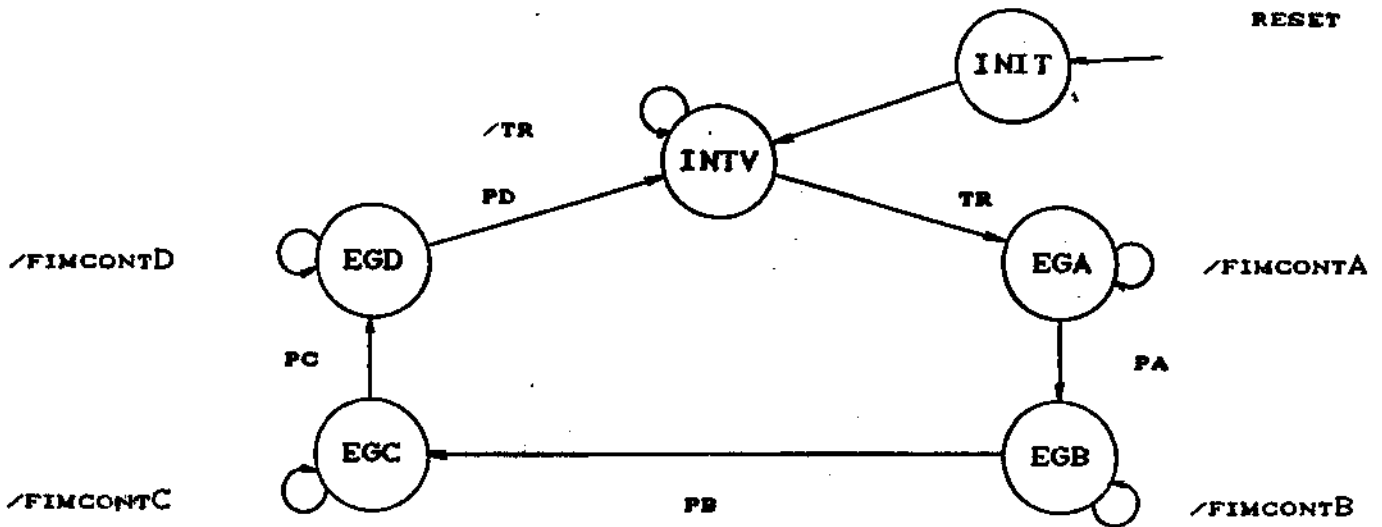
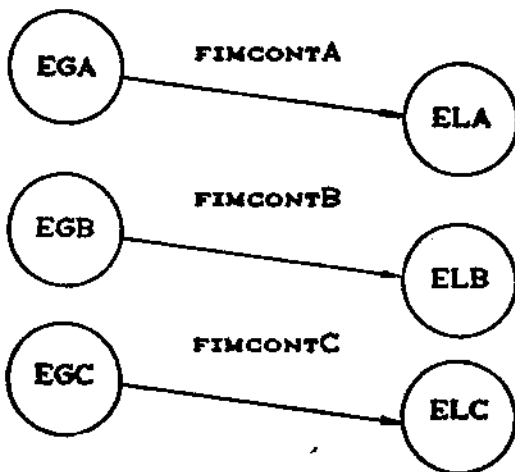


figura 2A

ESTADO DA INTERFACE	MENSAGENS
INIT - Inicialização	TR - Taxa de Repetição
INTV - Inativo	PA - Pulso A
EGA - Estado Gerador de A	PB - Pulso B
EGB - Estado Gerador de B	PC - Pulso C
EGC - Estado Gerador de C	PD - Pulso D
EGD - Estado Gerador de D	

DIAGRAMA DE ESTADO PARA ALARGAMENTO DOS PULSOS



ESTADO DA INTERFACE
ELA - Estado para Larg. A
ELB - Estado para Larg. B
ELC - Estado para Larg. C

figura 2B

Configuração adotada:

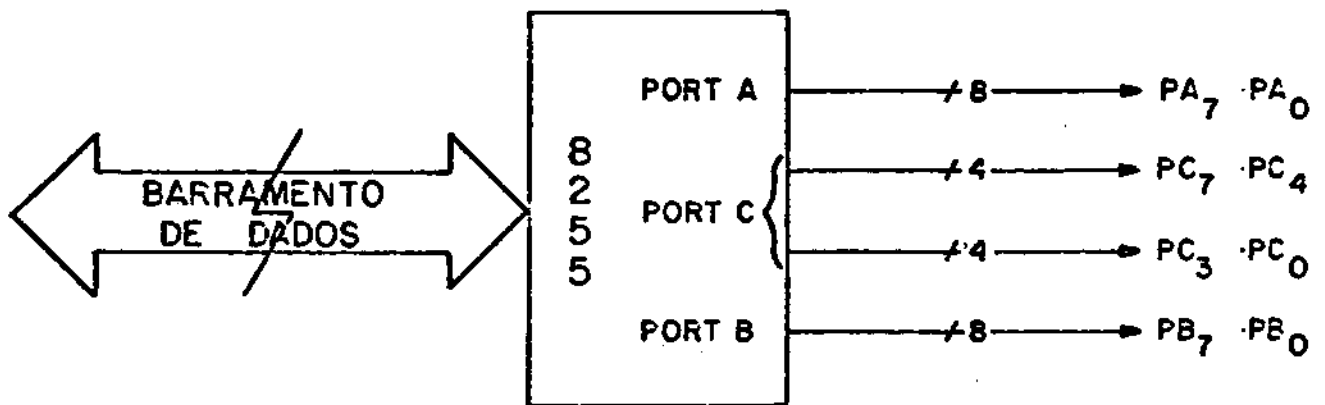


FIGURA 3

Os dados são transmitidos ou recebidos dos PPI 8255A por um "buffer" interno mediante uma execução de uma instrução de entrada ou saída pela CPU.

Existem ainda dois sinais de entrada, conectados aos bits menos significativos do barramento de endereços, que em conjunto com as entradas "read" e "write", controlam a seleção de uma das três portas existentes (figura 4a e 4b).

SINAIS	PORT A	PORT B	PORT C	PALAVRA DE CONTR.
A <sub>1</sub>	0	0	1	1
A <sub>0</sub>	0	1	0	1

FIGURA 4a

Endereços atribuídos às portas:

INTEGRADO	PORT A	PORT B	PORT C	PALAVRA DE CONTR.
8255 AA	A0	A1	A2	A3
8255 BB	A4	A5	A6	A7
8255 CC	A8	A9	AA	AB

FIGURA 4b



No módulo de geração dos pulsos duas PPI são utilizadas, as quais as denominamos de 8255AA e 8255BB que são divididas em dois grupos de doze bits cada. Esses grupos são conectados às entradas de um conjunto formado por três contadores, dispostos em cascata. As saídas desses contadores (74LS101-Synchronous Up/Down) [7] são "trigados" na transição para o nível alto da entrada do sinal de "clock", estando o "enable" no nível baixo. A seqüência da contagem é decrescente, determinada pelo nível da entrada "Up/Down", ligado a Vcc.

Esses contadores são completamente programáveis, isto é, as contagens podem ser modificadas para novos valores, os quais somente estarão atuando ao final da contagem decorrente. Na transição para o nível baixo da entrada "load" os novos valores são assumidos, a saída irá trocar de acordo com a entrada de dados, independente do nível de entrada do "clock".(figura 5)

Os "ripple clocks" produzem um pulso de saída igual a metade do período de um ciclo de clock, quando o contador zera a contagem. Esse pulso é utilizado para a função cascata, alimentando a entrada "clock" seguinte.

Os "max/min's" produzem um pulso de saída igual a um ciclo completo de "clock" quando o contador zera a contagem. Esses pulsos são ligados às entradas de uma porta AND (74LS11) [7] que dá origem a um "pulso" somente quando todos os contadores findam a contagem.

Operando em sincronismo com cada conjunto de contadores, existem dois "flip-flops" tipo D (74LS74) [7], que controlam a inibição da contagem. Todo esse processo é repetido em quatro diferentes blocos, gerando a formação dos quatro pulsos desejados.

No módulo de largura de pulso, uma terceira PPI que foi denominada de 8255CC, é dessa vez dividida em três grupos de oito bits, e é utilizada para controlar a largura dos três primeiros pulsos.

Cada um desses grupos são conectados às entradas de um conjunto formado por dois contadores, dispostos em cascatas.

O processo de alargamento dos pulsos se inicia quando um desses alimenta a entrada de "clock" de um "flip-flop" tipo D (74LS74) [7], levando ao estado lógico 1 a sua saída e, conseqüentemente, habilitando a contagem dos referidos contadores.

Os contadores (74LS163 - Synchronous 4-bit counters) [7] são completamente programáveis, semelhante aos já mencionados anteriormente, porém, utilizando-se de uma contagem crescente. Quando o contador completa a contagem, os "ripple carries" produzem um pulso de saída igual a um ciclo completo de "clock", que é utilizado para a função cascata, alimentando a entrada do "enable P" seguinte, cujo "clock" é paralelo. O outro "enable", T, encontra-se permanentemente habilitado, no nível alto. Esses pulsos são também ligados à entrada de uma porta NAND (74LS00), que dá origem a um pulso quando os contadores findam a contagem. Esse pulso alimenta a entrada "clear" do "flip-flop" (74LS74) [7], mudando a sua saída para o nível baixo, alargando dessa maneira, o pulso de entrada do "clock". (figura 6)

De modo idêntico, dois outros blocos controlam a largura de dois outros pulsos. (figura 7)

Na figura 8 é apresentado o esquema de ligações do programador de pulsos, que é composto por 35 componentes, dispostos em grupos, em um cartão padrão S-100 [4,3], de forma organizada e eficiente a fim de possibilitar uma fácil e rápida localização de qualquer um de seus componentes.

# 74LS191

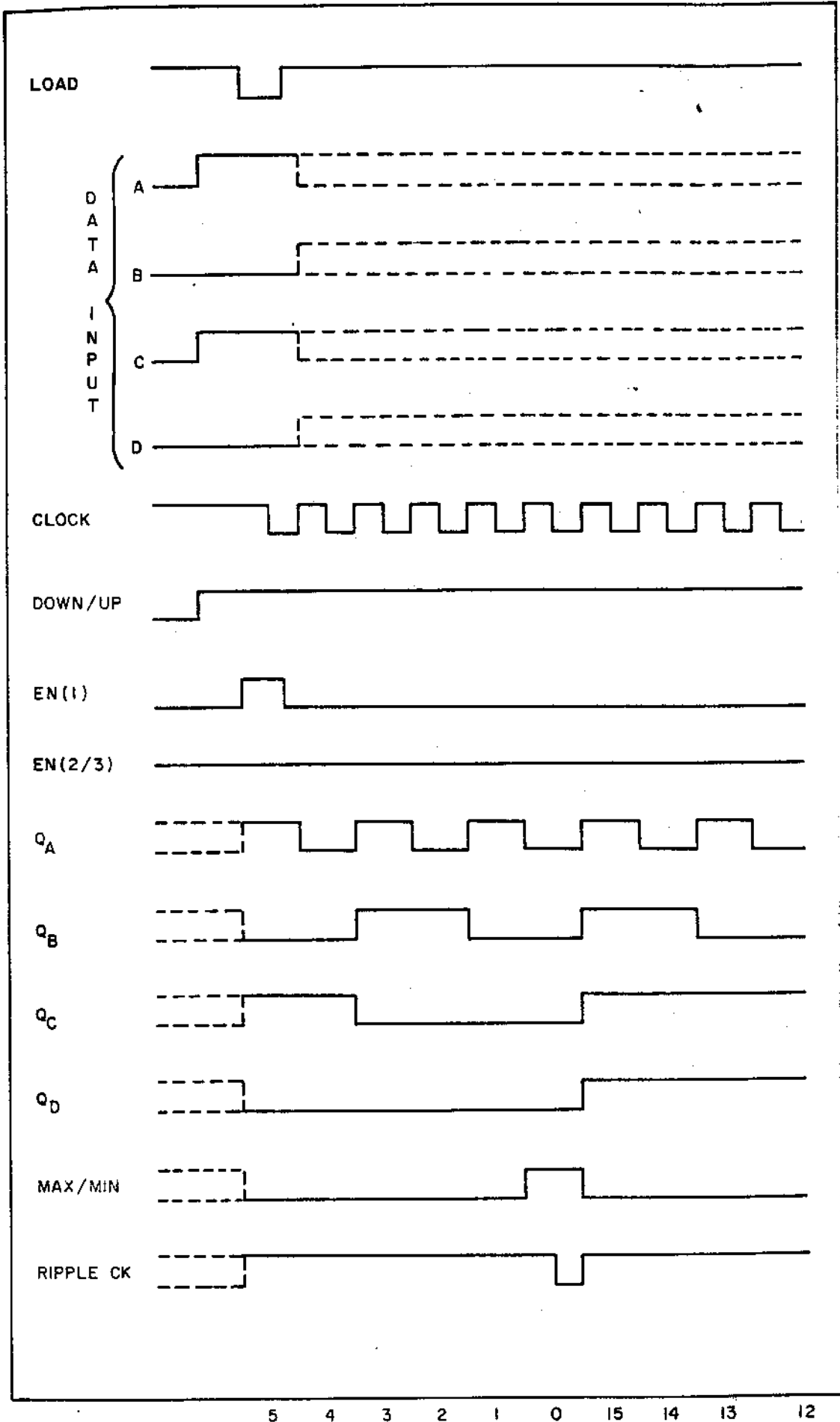


Figura 5

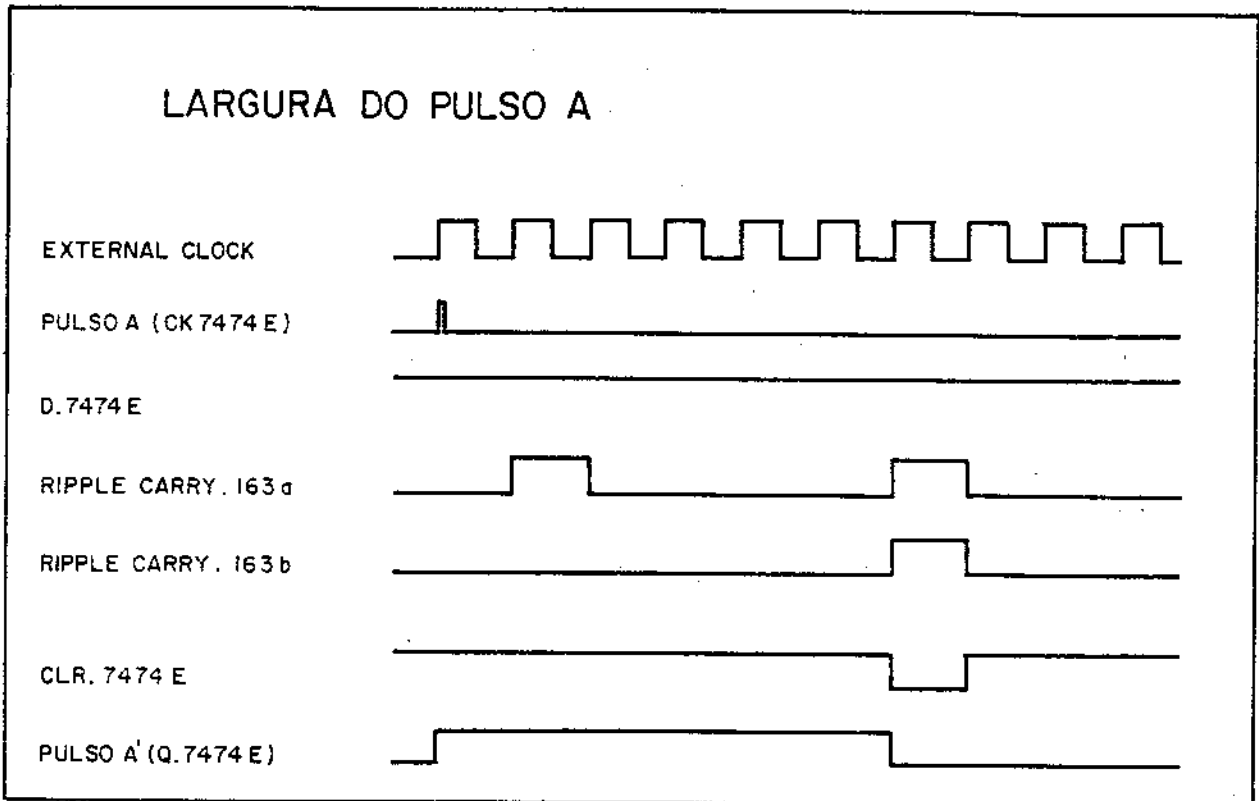


Figura 6

### DISPOSIÇÃO DOS QUATRO PULSOS

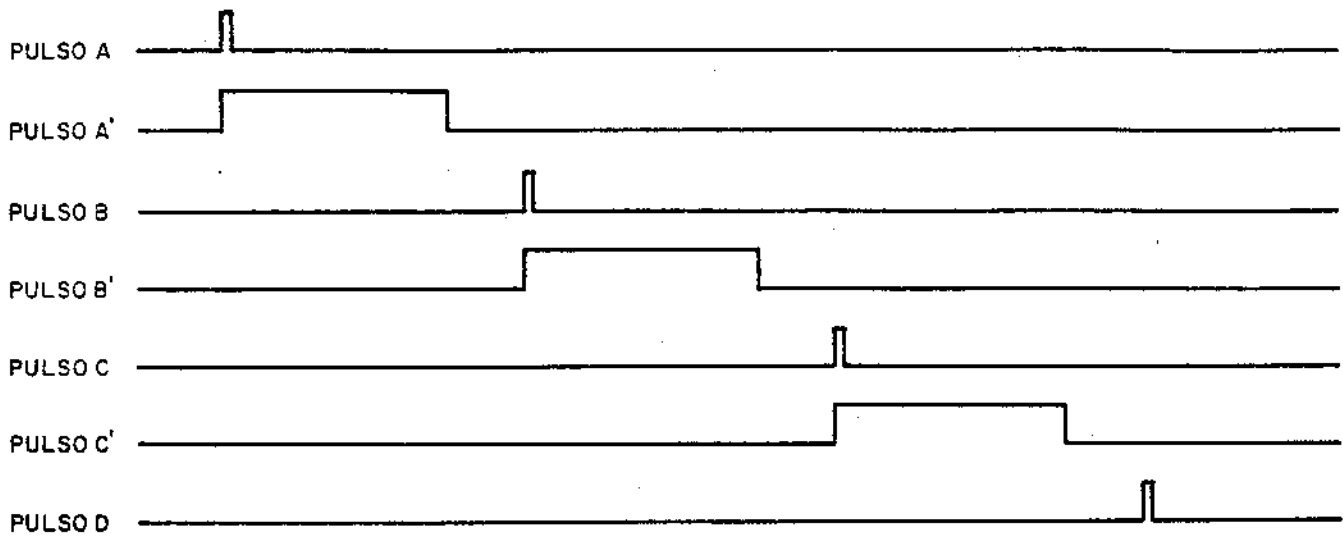


Figura 7

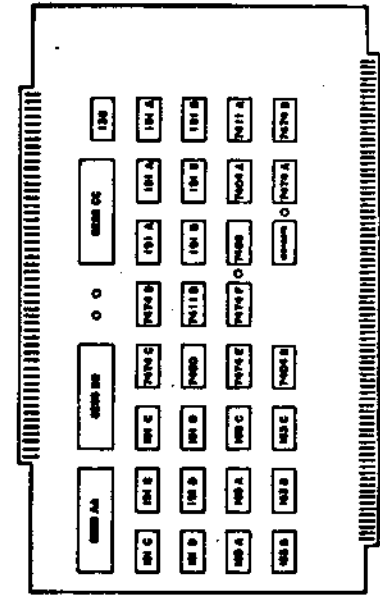
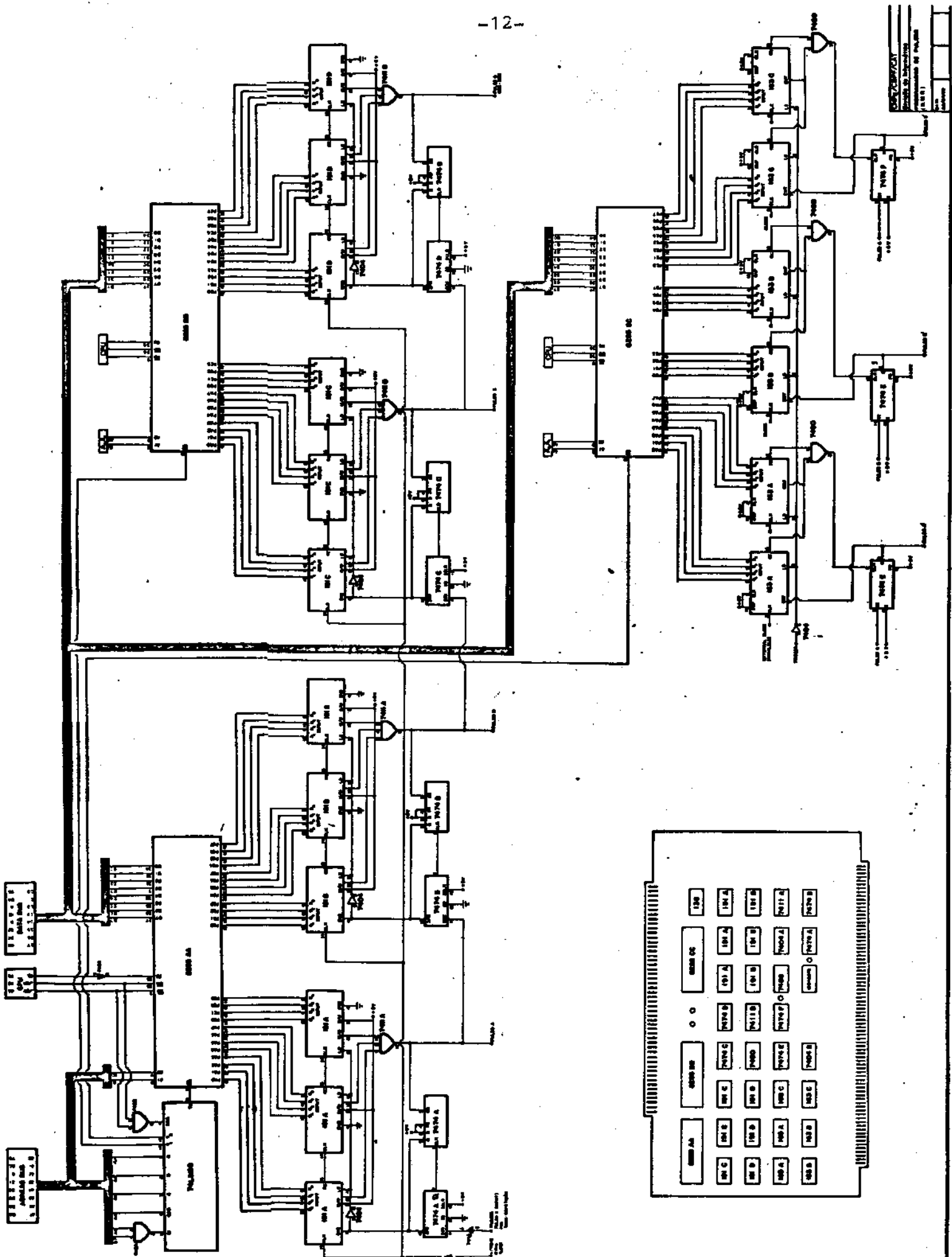


Figura 8

SOFTWARE

A figura 9a descreve o fluxograma simplificado do software que faz a programação dos pulsos e a figura 9b apresenta uma parte resumida deste software, desenvolvido na linguagem C.

FLUXOGRAMA SIMPLIFICADO

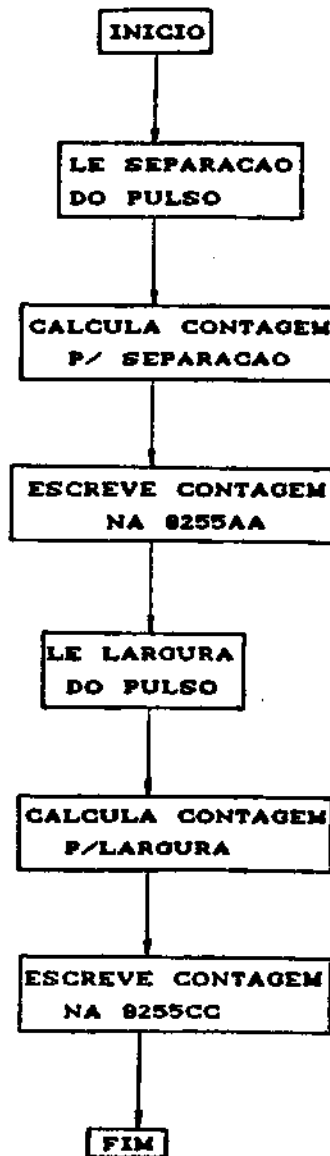


FIGURA 9a

SOFTWARE DE CONTROLE PARA PROGRAMADOR DE PULSO

```

#include "STDIO.H"

#define PORT_1A      0xA0      /* Porta A 8255AA      */
#define PORT_1B      0xA1      /* Porta B 8255AA      */
#define PORT_1C      0xA2      /* Porta C 8255AA      */
#define CNTR_1       0xA3      /* Porta de controle 8255AA */
#define CNTR_2       0xA7      /* Porta de controle 8255BB */
#define CNTR_3       0xAB      /* Porta de controle 8255CC */
#define PALAVRA_CNTRL 0X80     /* Palavra de controle  */
#define LARG_A       0xAB      /* Porta A 8255CC      */
#define LARG_B       0xA9      /* Porta B 8255CC      */
#define LARG_C       0xAA      /* Porta C 8255CC      */

int      separação();
int      largura();
double   res_separação,res_largura;
double   freq_sep =10e6,freq_larg=10e6;

main()
<
    inicializa_8255();          /* Inicializa todas as PPIs */
    res_separação = 1/freq_sep ; /* Calcula resolução da Sep.*/
    res_largura    = 1/freq_larg; /* Calcula resolução da Larg*/
    .
    .
    .
    prg_pulso_a();             /* Programa Pulso A      */
    prg_pulso_b();             /* Programa Pulso B      */
    .
    .
    .
>

inicializa_8255() /* Escreve a Palavra de Controle nas PPIs */
<
    out(CNTR_1,PALAVRA_CNTR);
    out(CNTR_2,PALAVRA_CNTR);
    out(CNTR_3,PALAVRA_CNTR);
>

prg_pulso_a()
<
    int contagem_a;

    printf("Entre a sep.da Taxa de Rep.-PULSO-A [seg]");
    contagem_a=separação();
    out(PORT_1A,contagem_a & 0xFF);
    out(PORT_1C,(in(PORT_1C) & 0xF0) | ((contagem_a & 0xF00)>>8));
    printf("Entre com a largura do PULSO A [seg]");
    contagem_a=largura();
    out(LARG_A,contagem_a);
>

```



```

prg_pulso_b()
{
    int contagem_b;

    printf("Entre a sep. PULSO B-PULSO A [seg]");
    contagem_b=separacao();
    out(PORT_B,contagem_b & 0xFF);
    out(PORT_C,((in(PORT_C) & 0x0F) | ((contagem_b & 0xF00)>>4)));
    printf("Entre com a larg. do PULSO B [seg]");
    contagem_b=largura();
    out(LARG_B,contagem_b);
}

int separacao()
{
    double separação=0;
    char s[50];

    separação=atof(gets(s));
    return((separação / res_separação)+1);
}

largura()
{
    double largura;
    char s[50];

    largura=atof(gets(s));
    return(254-(largura / res_largura));
}

```

FIGURA 9b

O programa é basicamente dividido em três etapas, que são: inicialização, definição da separação e da largura dos pulsos.

Na primeira etapa são inicializadas as três PPI 8255A da interface, que são programadas no modo 0 de operação, e com palavra de controle 80H, que configura todas as portas de E/S para escrita.

Na segunda etapa define-se os tempos de separação entre os pulsos, i.e., a contagem que se faz necessário enviar aos contadores, para que este gere o pulso. Esta contagem é calculada entrando com a separação na escala de tempo (segundo) e fazendo uma transformação para a contagem.

A seqüência de pulsos é gerada a partir de um sinal de sincronismo ( Taxa de Repetição ) que habilita a contagem de uma freqüência conhecida que denominamos de freqüência de separação. A resolução da contagem é, portanto, o período da freqüência de separação, onde resumimos a contagem na seguinte fórmula:

$$\text{contagem\_sep.} = \frac{\text{separacao\_escolhida}}{\text{resolucao\_separação}} + 1$$

Devido ao pulso ser gerado durante a última contagem faz-se necessário somarmos 1 ao valor calculado.

Na terceira etapa define-se a largura dos pulsos A, B e C a partir de um sinal de sincronismo para cada pulso, este sinal habilita a contagem de uma freqüência conhecida que denominamos de freqüência de alargamento.

A largura dos pulsos é calculada do mesmo modo que na etapa anterior, entrando com o valor na escala de tempo (segundo) e transformando-a para a contagem a ser enviada aos contadores. Este calculo se faz pela fórmula:

$$\text{contagem\_larg} = 255 - \left( \frac{\text{largura\_escolhida}}{\text{resolução\_larg.}} - 1 \right)$$

É necessário subtrair de 255 porque a contagem é crescente, e 1 porque o pulso é gerado durante a última contagem.

As rotinas, descritas na figura 5b, que programam o pulso A e o pulso B ( prg\_pulso\_a() e prg\_pulso\_b() ) foram feitas para permitir mudanças nos valores das contagens sem alterar os outros contadores, i.e., podem ser chamadas de qualquer parte para se realizar alterações nos tempos dos pulsos. Estas rotinas são as mesmas para programar os pulsos C e D diferindo apenas nas portas de E/S.

CONCLUSAO

Testes com um analisador lógico (1240-Tektronix) com definição de 20ns e com um osciloscópio de 60MHz (2213-Tektronix), demonstram a alta confiabilidade do Programador de Pulsos, operando em uma frequência máxima de até 35MHz.

Devido a frequência dos contadores não ser síncrona com a taxa de repetição dos pulsos, deve-se chamar a atenção para erros desta natureza, que equivalem a um período desta frequência. Para um melhor desempenho do programador de pulso a frequência dos contadores e a taxa de repetição devem ser geradas com a mesma referência.

A opção por uma estrutura modular e por um barramento padrão para esta interface proporcionam vantagens para um futuro acréscimo de módulos sem a necessidade de se alterar a estrutura do projeto, além da portatibilidade e facilidade de programação permitirem grandes perspectivas de aplicação na geração de diferentes seqüências de pulsos. Desse modo várias sofisticações podem ser adicionadas, seja através de um software mais complexo, de um hardware mais expandido ou de substituições dos componentes TTL utilizados no projetos para a família F ou H.

*Agradecimentos :*

- Ao desenhista Paulo Mendo, aos professores Alberto P. Guimarães e Nilton Alves Júnior pelo contato proporcionado no Laboratório de Ressonância Magnética Nuclear, aos companheiros da Divisão de Informática e ao chefe da Coordenadoria de Atividades Técnicas prof. Alfredo Marques.

REFERENCIAS

- 1) Intel - 1980 - Component Data Catalog - 8255A pg.8-61.
- 2) Costa, A.P.R.; Cernicchiaro, G.R.C. e Machado, R.M.C. - 1984 - Anais da 36 SBPC - pg. 404.
- 3) Poblet, J.M. - 1980 - Interconexión de Perifericos e Microprocessadores - Baixareu Editores.
- 4) Libe, S. e Garetz, M. - 1981 - Interfacing to S100/IEEE 696 Microcomputers.
- 5) Veludo de Paiva, M.S. - 1984 - Construção de um Gerador de Pulso Programavel para Experinciade RMNp - Tese de Mestrado, Instituto de Física e Química de São Carlos, Universidade de São Paulo.
- 6) Fukushima, E. e Roeder, S.B.W.; Experimental Pulse NMR (A nuts and bolts approach) - Advanced book Program.
- 7) Texas - The TTL Data Book - Texas Instruments Inc. - 1980.
- 8) Alves, N. - 1986 "Estudo por RMN das Interacoes Hiperfinas Transferidas em GdCo<sub>2</sub>Al<sub>2</sub>O<sub>7</sub>". Tese de Mestrado, Universidade Federal Fluminense.

**Abstracts:** In this work we describe the development and construction of a four channel generator on a S100 standard board and its software controls.

The microcomputer has full control of the pulse generator, that has seven programmable parameters defining the four pulses positions and the width of the first three pulse. Automatic repeat of a selected range is also provided.

This pulse generator is controled by a software development in the C language, and is used in pulsed Nuclear Magnetic Resonance experiments.

**KEY WORDS :**

Pulse Programmer; Interface; Bus S100; NMR Hardware;  
Pulse Generator.