EA870 - Laboratório de Computação

Laboratório 4
Interfaces de Entrada e Saída - Revisão
Tempo de Execução das Instruções do MC68000
Interface Paralela MC68230 PI/T
Padrão Centronics

1. Introdução

Uma das características básicas de um computador é a sua capacidade de trocar os dados com o mundo exterior (dispositivos periféricos). Para tornar compatíveis, elétrica e temporalmente, os sinais internos à máquina com os dos mundo exterior, fazem-se necessários circuitos especiais denominados interfaces de entrada e saída. O objetivo principal deste laboratório é permitir ao aluno o contato com uma interface paralela disponível no mercado. Este contato será efetuado através da programação da interface paralela MC68230. Será também apresentado o padrão Centronics para interfaces paralelas. Adicionalmente serão apresentados os tempos de execução das instruções do MC68000, a serem utilizados na experimentação com a MC68230.

Este roteiro está organizado como se segue:

- Seção 2: Tempo de Execução das Instruções do MC68000
- Seção 3: Interfaces de Entrada e Saída
- Seção 4: Interface Paralela MC68230
- Seção 5: Padrão Centronics
- Seção 6: A interface MC68230 no SBC68K
- Seção 7: Exercícios.

2. Tempo de execução das instruções do MC68000

Para estimar o tempo de execução de um bloco de instruções pelo microprocessador MC68000 pode-se utilizar as tabelas de tempos de execução em anexo. Nestas tabelas os tempos são dados em função da quantidade de períodos do relógio interno do computador gastos para a execução das instruções. Foi considerado que cada ciclo de acesso à memória (seja acesso de escrita ou de leitura) necessita de quatro períodos.

O microcomputador SBC68K está configurado para trabalhar com a freqüência de 8 MHZ, implicando em um período de 125 ns.

A título de exemplo, apresenta-se na tabela 4.1 o tempo aproximado gasto para a execução de algumas instruções:

Instruções	Número de períodos	Tempo em µs
MOVE.B D0,D1	4	0,5
MOVE.L D0,D1	4	0,5
MOVE.W (A0),(A1)	12	1,5
MOVE.L (A0),(A1)	20	2,5
MOVE.W -(A0),D0	10	1,25
DIVU D0,D1	140	17,5

Tabela 4.1 - Tempo de execução de instruções

Observar que o tempo de execução de uma instrução leva em consideração o tempo de busca da instrução, do cálculo do endereço efetivo dos operandos, da busca dos operandos e finalmente da execução da ação associada à instrução.

Considere a primeira instrução da tabela 4.1. Esta instrução transfere o byte (8 bits) menos significativo contido em D0 para o registrador D1. Esta instrução exige apenas um ciclo de leitura (busca da instrução) e possui um tempo de execução igual a 4 períodos do relógio (acesso de leitura gasta 4 períodos de relógio).

Considere a segunda instrução da tabela 4.1. Esta instrução transfere a palavra longa (32 bits) residente em D0 para D1. Esta instrução, como a anterior, exige apenas um ciclo de leitura (a busca da instrução) e possui um tempo de execução igual a 4 períodos do relógio.

Considere a terceira instrução da tabela 4.1. Esta instrução transfere a palavra (16 bits) residente na posição de memória apontada pelo registrador A0 para a posição de memória apontada pelo registrador A1. Esta instrução exige, portanto, 3 ciclos de acesso à memória (1 ciclo de busca da instrução + 1 ciclo de leitura do operando fonte + 1 ciclo de escrita no endereço destino), e possui um tempo de execução igual a 12 (3 x 4 - cada acesso em memória gasta 4 períodos) períodos do relógio.

Considere a quarta instrução da tabela 4.1. Esta instrução transfere a palavra longa (32 bits) residente na posição de memória apontada pelo registrador A0 para a posição de memória apontada pelo registrador A1. Por ser uma instrução de transferência de 32 bits entre posições de memória, ela exige 5 ciclos de acesso à memória (1 ciclo de busca da instrução + 2 ciclos de leitura (16 bits cada) do operando fonte + 2 ciclos de escrita (16 bits cada) para a transferência dos 32 bits para o endereço destino). Esta instrução possui um tempo de execução igual a 20 (5 x 4) períodos do relógio

Nestes quatro exemplos o tempo de execução é definido diretamente pelo tempo associado aos ciclos de acesso externo (busca da instrução, leitura de operando em memória e escrita de operando em memória)

Já na quinta instrução da tabela 4.1 tem-se 2 ciclos de acesso à memória (1 ciclo de busca da instrução + 1 ciclo de leitura do operando fonte (16 bits)) e um tempo de execução de 10 períodos do relógio. Observe que, nesta instrução, somado aos tempos de busca da instrução e do operando fonte (2 ciclos de acesso à memória com duração de 4 períodos cada), tem-se 2 ciclos adicionais associados ao cálculo do decremento associado ao modo de endereçamento do operando fonte. Observar pelas tabelas do anexo A (Tempo de Execução das Instruções), que estes ciclos adicionais não se fazem necessários caso o modo de endereçamento indireto com pré-decremento esteja sendo utilizado para especificar o operando destino.

Considere agora a última instrução da tabela 4.1. Esta instrução efetua a divisão sem sinal do operando destino pelo operando fonte, deixando o resultado no endereço efetivo do operando destino. Esta instrução, segunda a tabela 8-4 do anexo, exige apenas um ciclo de leitura (busca da instrução), possuindo um tempo de execução de 140 períodos do relógio. Esta instrução é um exemplo extremo do caso em que o tempo de execução da operação associada à instrução é significativo no cálculo do tempo total de execução (4 períodos para a busca da instrução e 126 períodos para a realização da divisão).

3. Interfaces de Entrada e Saída

A comunicação de um computador com o mundo exterior pode se dar em dois sentidos: de dispositivos periféricos para o computador (entrada - E), e do computador para dispositivos (saída - S). O controle da transferência de dados pode ser realizada de três maneiras distintas: E/S programada, E/S acionada por interrupção e E/S por DMA (direct memory access). Neste curso não será discutido E/S por DMA.

Na **E/S programada**, o programador define os instantes das operações de E/S, inserindo em seu código as instruções que realizam a operação de E/S. Já na **E/S acionada por interrupção**, o instante da operação de E/S não está definido por programa, mas sim, é ditado pelo dispositivo externo através de um pedido de interrupção. E/S acionada por interrupção será assunto dos laboratórios 5 e 6.

Ainda com referência à E/S programada é possível identificar duas variações, dependendo se o instante de transferência está ou não condicionado ao estado do dispositivo. **Nas transferências incondicionais**, a operação de E/S é realizada independentemente do estado do dispositivo. Na **transferência condicional**, a operação de E/S só é realizada quando e se o dispositivo estiver pronto para tal. Em geral, em um programa que implementa uma transferência condicional tem-se um laço (loop) de espera, onde são efetuados a leitura e o teste do estado do dispositivo repetidas vezes até que o mesmo indique que a operação de E/S pode ser realizada. Essa transferência, por ser caracterizada pela consulta ao dispositivo, também é conhecida como transferência por **polling**.

De acordo com o número de bits que são transmitidos simultaneamente, distinguem-se ainda transmissões paralelas das seriais. Nas transmissões paralelas são enviados um ou mais bytes simultaneamente, enquanto que nas seriais , a transmissão é bit-a-bit. Neste laboratório discute-se a comunicação paralela e no laboratório 4 será tratada a comunicação serial.

Na comunicação paralela entre elementos com velocidades diferentes, como é o caso, por exemplo, de um computador e uma impressora, faz-se necessária a correta temporização entre o transmissor e o receptor para se evitar eventuais perdas de informação. Existem inúmeras maneiras de se garantir a temporização apropriada entre transmissor e receptor. Entretanto, é possível identificar duas estratégias básicas:

- **Síncrona:** computador e dispositivo compartilham uma base de tempo comum (sinal de relógio de referência)
- **Assíncrona:** computador e dispositivo não compartilham uma base de tempo comum. Em geral, esta falta de uma base de tempo comum entre transmissor e receptor pode levar a perda de dados uma vez que a transmissão pode ocorrer sem que o receptor esteja preparado para receber o dado. Uma forma de se garantir a correta temporização entre transmissor e receptor é através do mecanismo de *handshaking*. *Handshaking* condiciona a transferência dos dados a sinais de controle trocados entre transmissor e receptor. Estes sinais têm por função garantir que o transmissor e o receptor operem de forma interdependente. Interdependência significa, neste contexto, que uma ação só pode ser iniciada após o término da ação corrente. *Handshaking* pode ser implementado através de duas linhas de controle: uma linha é utilizada pelo transmissor para sinalizar que um dado está disponível; a outra utilizada pelo receptor para indicar que o dado foi aceito, e que, portanto, o transmissor pode enviar um novo dado. Por exemplo, numa operação de saída, a UCP envia um sinal ao dispositivo avisando que um dado está disponível; em resposta o dispositivo lê o dado e sinaliza que um novo dado pode ser enviado.

Os elementos internos ao computador responsável pela transferência de dados entre computador e os dispositivos periféricos são denominados interfaces de entrada e saída. Em geral, estas interfaces são constituídas de:

- **Registrador(es) de Controle** (utilizados para escrita). O conteúdo deste(s) registrador(es) define(m) o modo de operação da interface.
- **Registrador(es) de Dados.** Este(s) registrador(es) é(são) utilizado(s) para o armazenamento temporário de forma a melhor acomodar o fluxo de dados entre computador e dispositivo.

- **Registrador(es) de estado** (utilizados para a leitura). Este(s) registrador(es) indica(m) o estado do dispositivo e/ou da interface e pode(m) ser utilizado(s) para controlar a dinâmica da transferência.

É possível distinguir ainda duas organizações no que se refere ao espaço de endereçamento associado às interfaces de E/S:

- **E/S Isolada.** O espaço de endereçamento de E/S é disjunto do de memória. Como características marcantes desta organização têm-se Barramentos de Endereços de Memória e o de E/S logicamente distintos; e instruções específicas para E/S.
- **E/S Mapeada em Memória.** Memória e E/S compartilham o mesmo espaço de endereçamento. Como características marcantes desta organização têm-se um único (físico e lógico) Barramento de Endereços; e não há distinção entre instruções de acesso à memória e as interfaces de E/S.

4. Interface Paralela MC68230

A Interface Paralela/Temporizador (parallel interface/timer - PI/T) MC68230 é um circuito integrado formado dois grandes blocos funcionais: Interface Paralela e Temporizador. O bloco Interface Paralela agrupa as facilidades da MC68230 associadas às conexões paralelas. O bloco Temporizador oferece facilidades para a implementação de sinais de temporização das mais variadas formas. Neste laboratório será tratado apenas o bloco Interface Paralela.

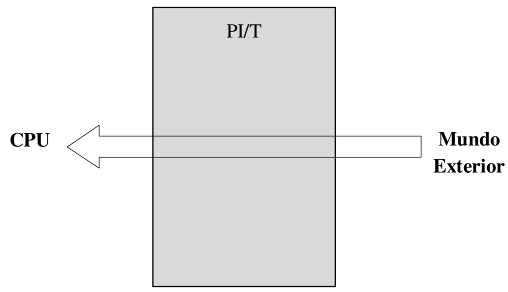
A interface paralela da MC68230 é composta de:

- Porta A com 8 bits
- Porta B com 8 bits
- 4 bits para *handshaking*
- Porta C com 8 bits. Os bits da porta C podem também ser configurados para operarem em funções de controle associadas às portas A, B, ou mesmo, do temporizador

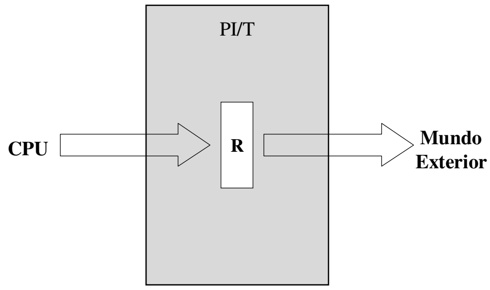
As portas da interface paralela podem ser programadas para atender requisitos distintos de uma ampla gama de dispositivos periféricos. Entre as características programáveis destacam-se:

- **Tamanho dos Dados.** As portas podem operar de maneira independente, definindo uma comunicação paralela de 8 bits. Alternativamente, pode-se configurar as Portas A e B para operarem em conjunto, definindo uma comunicação de 16 bits.
- **Direção do Fluxo dos Dados.** As portas podem ser programadas para a operação uni ou bidirecionalmente. No modo **unidirecional**, tem-se um registrador de direção de 8 bits (data direction register - DDR) associado a cada uma das portas, sendo que o valor de cada bit deste registrador define direção de transferência do pino corresponde da porta. No modo **bidirecional** a direção de transferência é dinamicamente definida pelos sinais de *handshake* H1, H2, H3 e H4.
- **Capacidade de Armazenamento Interno (buffering).** A interface MC68230 admite ainda que suas portas sejam programadas para operar nas seguintes configurações: **non-latched** (sem armazenamento), **single-buffered** (armazenamento simples) e **double-buffered** (armazenamento duplo). A configuração *non-latched* só é permitida para a entrada de dados (direção do fluxo da informação do mundo exterior para o processador). Nesta configuração, a leitura do valor de entrada corresponde ao valor presente fisicamente no pino de entrada da interface. A configuração *single-buffered* só é permitida para a saída de dados. Nesta configuração, o valor a ser transferido é armazenado temporariamente em um registrador interno à interface (R na Fig. 4.1.b). Já a configuração *double-buffered* pode ser

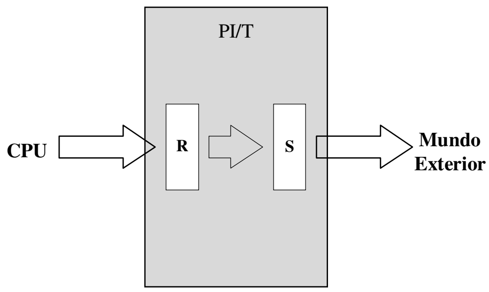
utilizada tanto para a entrada quanto para a saída de dados. Nesta configuração são utilizados dois registradores internos, em cascata, para o armazenamento temporário dos dados a serem transferidos (R e S na Fig. 4.1.c). É possível ilustrar o funcionamento dessas três configurações de armazenamento interno através de analogias com filas de 0, 1 e 2 posições respectivamente. À medida que um dado chega, ele ocupa a primeira posição livre, avançando na fila à medida que os seus antecessores forem sendo atendidos (consumidos).



a) Entrada *Non-Latched*



b) Saída *Single-Buffered*



c) Saída *Double-Buffered*

Fig. 4.1: Configurações de armazenamento interno (*buffering*)

Neste laboratório, a interface MC68230 será utilizada apenas no modo de transferência unidirecional com 8 bits. Esta configuração corresponde ao denominado **modo 0**. A tabela 4.2 apresenta os possíveis submodos da interface paralela MC68230 para a configuração em modo 0.

Submodo	Entrada	Saída
00	Double-buffered	Single-Buffered
01	Non-Latched	Double-Buffered
1X	Non-Latched	Single-Buffered

Tabela 4.2 - Submodos do modo 0

Na MC68230 os submodos que permitem *double-buffered* (submodos 00 e 01) estão ainda associado aos pinos de handshake disponíveis na interface: H1 e H2 são os pinos de handshake associados à Porta A; H3 e H4 os pinos de handshake associados à Porta B. Estes pinos/sinais de handshake são utilizados para controlar o fluxo de informação entre a interface e um eventual dispositivo a ela conectado. O significado dos sinais de handshake é programável e depende da direção de transferência (entrada ou saída):

Entrada de dados: O sinal H1(H3) é gerado pelo dispositivo externo para sinalizar à interface que os dados de entrada estão disponíveis. A interface utiliza este sinal para grampear os dados em um de seus registradores internos. A interface possui um registrador de estado PSR, cujo bit H1S(H3S) é *setado* toda vez que um dos dois registradores de entrada contenha dado que ainda não foi lido pela CPU. Já o significado de H2(H4) é programável pelo usuário, podendo ter as seguintes funções:

1. H2(H4) pode ser um sinal de entrada, independente de H1(H3). A transição de borda deste sinal torna verdadeiro o bit H2S(H4S) do registrador de estado (PSR).
2. H2(H4) pode ser um sinal de saída de propósito geral.
3. H2(H4) pode ser um sinal de saída que opera em conjunto com o sinal H1(H3) para a comunicação através de um mecanismo de handshake. O sinal H2(H4) pode ser entendido como a resposta da interface ao sinal H1(H3), indicando que o dado foi lido e que o dispositivo pode enviar o próximo dado. A PI/T suporta dois protocolos distintos de *handshake*: *interlocked* e *pulsed*. No protocolo *pulsed* o sinal H2(H4) nunca permanece VERDADEIRO mais do que quatro pulsos de relógio, enquanto que no *interlocked* o transição do estado VERDADEIRO para FALSO do sinal H2(H4) depende exclusivamente de H1(H3).

Saída de dados: O sinal H1(H3) é gerado pelo dispositivo externo para sinalizar a interface que os dados foram recebidos, liberando a interface para enviar um próximo dado. O bit H1S(H3S) do PSR pode ser programado para ser *setado* em uma das seguintes situações: quando apenas um dos registradores de saída está vazio; ou quando ambos os registradores de saída estão vazios.

O significado de H2(H4) é programável pelo usuário, podendo ter as seguintes funções:

1. H2(H4) pode ser um sinal de entrada, independente de H1(H3). A transição de borda deste sinal torna verdadeiro o bit H2S(H4S) do registrador de estado (PSR).
2. H2(H4) pode ser um sinal de saída de propósito geral.
3. H2(H4) pode ser um sinal de saída que opera em conjunto com o sinal H1(H3) para a comunicação através de um mecanismo de *handshaking*. O sinal H2(H4) é gerado pela interface para sinalizar ao dispositivo que um novo dado encontra-se disponível. Em resposta, o dispositivo gera o sinal H1(H3), indicando que o dado foi lido e que a interface pode enviar um próximo dado. A PI/T suporta dois protocolos distintos de *handshake*: *interlocked* e *pulsed*. No protocolo *pulsed* o sinal H2(H4) nunca permanece VERDADEIRO mais do que quatro pulsos de relógio, enquanto que no *interlocked* o transição do estado VERDADEIRO para FALSO do sinal H2(H4) depende exclusivamente de H1(H3).

Sob a ótica de quem vai programar, a interface paralela MC68230 é composta de um conjunto de registradores, que podem ser agrupados em 3 classes distintas: controle, dados e estado.

Os **registradores de controle** (veja tabela 4.3) permitem a configuração do modo e submodo de operação, a direção do fluxo da informação (entrada ou saída) nas portas e o controle de pedidos de interrupção (a ser discutido no laboratório 5).

Registrador	Descrição
registrador de controle geral das portas (PGCR)	controla as funções comuns às operações das portas, tais como modo de operação e o estado dos pinos de handshake
registrador de requisição de serviços das portas (PSRR)	controla outras funções comuns às operações de portas, tais como tipos de requisição, a corrente função dos pinos da porta C
registrador de direção de dados da porta A (PADDR)	determina a direção e a característica de *buffering* de cada pino da porta A
registrador de direção de dados da porta B (PBDDR)	determina a direção e a característica de *buffering* de cada pino da porta B
registrador de direção de	determina a direção e a característica de *buffering* de cada pino da porta C

dados da porta C (PCDDR)	
registrador do número de vetor de interrupção (PIVR)	determina a subrotina para tratamento de interrupção
registrador de controle da porta A (PACR)	controla o submodo de operação da porta A e os pinos H1 e H2
registrador de controle da porta B (PBCR)	controla o submodo de operação da porta B e os pinos H3 e H4

Tabela 4.3: Registradores de controle da PI/T MC 68230

Os **registradores de dados** (veja tabela 4.4) são elementos de armazenamento temporário dos dados que fluem através da interface.

Registrador	Descrição
registrador de dados da porta A (PADR)	armazena os dados a serem transferidos de ou para os pinos da Porta A
registrador de dados da porta B (PBDR)	armazena os dados a serem transferidos de ou para os pinos da Porta B
registrador de dados da porta C (PCDR)	armazena os dados a serem transferidos de ou para os pinos da Porta C

Tabela 4.4: Registradores de dados da PI/T MC 68230

O registrador de estado (PSR) contém informações sobre cada pino de *handshake* X (X = 1, 2 3 e 4). Os bits 4, 5, 6 e 7 refletem o nível de tensão presente nos pinos H1, H2, H3 e H4, respectivamente. E os bits 0, 1, 2 e 3 correspondem aos bits H1S, H2S, H3S e H4S, cujos significados dependem do modo/submodo de operação. O bit de estado de um pino de handshake permite, por exemplo, verificar se os dados foram corretamente transferidos de/para o periférico.

Observar que através dos bits 0, 1, 2 e 3 do registrador de controle PGCR é possível definir o tipo de lógica associada aos pinos de *handshake*. Dois tipos de lógica são possíveis: lógica positiva e lógica negativa. Na lógica positiva o sinal VERDADEIRO corresponde a um nível de tensão alto (~5V TTL) e o sinal FALSO corresponde a um nível de tensão baixo (~0V TTL). Na lógica negativa o sinal FALSO corresponde a um nível de tensão alto (~5V TTL) e o sinal VERDADEIRO corresponde a um nível de tensão baixo (~0V TTL). Observar ainda que os bits de estado HXS, quando programados para tal, refletem a condição VERDADEIRO/FALSO, e não o nível de tensão dos pinos.

5. Padrão Centronics

O padrão Centronics de comunicação paralela é bastante difundido no mercado de computadores pessoais para a comunicação destes sistemas com impressoras. Este padrão é universalmente aceito pelos fabricantes de impressoras, sendo suportado pela maioria do fabricantes de computadores pessoais, incluindo os da linha IBM-PC e Apple.

O padrão Centronics define um conector padrão, denominado conector Centronics, cujo conector-fêmea é localizado na impressora (veja Fig. 4.2).

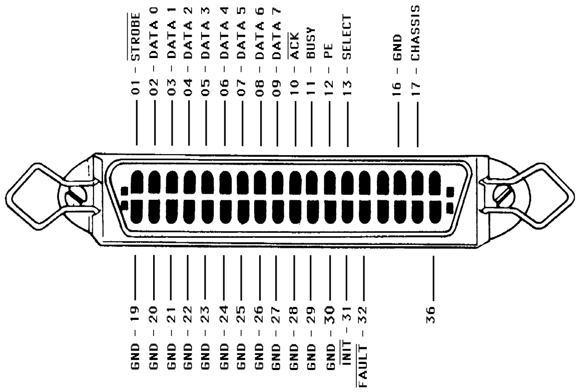


Fig. 4.2: Conector Centronics

Os pinos deste conector estão associados aos seguintes sinais:

Pino	Sinal	Direção	Descrição
01	/STB	Computador -» Impressora	Pulso negativo fornecido pelo computador como clock de transmissão dos dados.
02	DATA 0	Computador -» Impressora	Bit 0 do dado a ser impresso.
03	DATA 1	Computador -» Impressora	Bit 1 do dado a ser impresso.
04	DATA 2	Computador -» Impressora	Bit 2 do dado a ser impresso.
05	DATA 3	Computador -» Impressora	Bit 3 do dado a ser impresso.
06	DATA 4	Computador -» Impressora	Bit 4 do dado a ser impresso.
07	DATA 5	Computador -» Impressora	Bit 5 do dado a ser impresso.
08	DATA 6	Computador -» Impressora	Bit 6 do dado a ser impresso.
09	DATA 7	Computador -» Impressora	Bit 7 do dado a ser impresso.
10	/ACK	Impressora -» Computador	Pulso negativo que indica ao computador que a impressora recebeu o dado.
11	BUSY	Impressora -» Computador	Em nível alto, indica que a impressora não pode receber dados.
12	PE	Impressora -» Computador	Paper Empty. Em nível alto, indica que acabou o papel da impressora.
13	SELECT	Impressora -» Computador	En nível alto, indica que a impressora está em modo remoto. Caso contrário, está em modo local. No modo remoto a impressora pode receber dados do computador.
14-15	NC	-	-
16	GND	-	Terra lógico. Usado como referência.
17	CHASSIS	-	Terra da carcaça.
18	NC	-	-
19-30	GND	-	Terra de retorno para par trançado.
31	/INIT	Computador -» Impressora	Pulso negativo que pode ser utilizado para reinicializar a impressora.
32	/FAULT	Impressora -» Computador	Em nível baixo, indica falha na impressora.
33-36	NC	-	-

Tabela 4.5: Pinos/Sinais do conector Centronics

O diagrama de tempo da interface Centronics, apresentando apenas os sinais que serão considerados neste laboratório, é mostrado na Fig. 4.3.

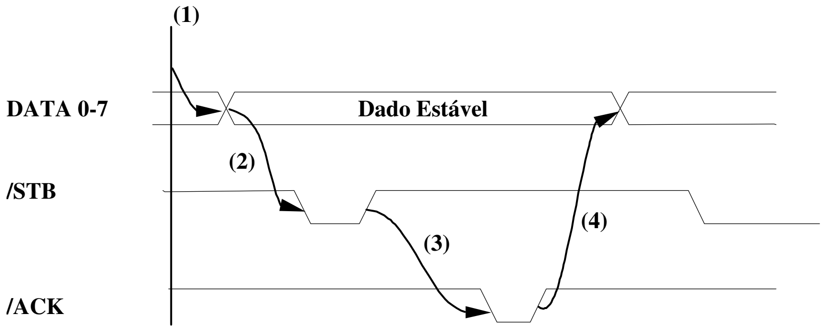


Fig. 4.3: Diagrama de tempo

Resumidamente este diagrama pode ser explicado da seguinte maneira:

1) O computador deve fornecer os dados através das 8 linhas de dados DATA 0 até DATA 7.
2) Após a estabilização dos dados, o computador deve acionar o sinal /STB (o símbolo / antes do nome do sinal indica que o mesmo é ativo baixo), indicando para a impressora que um dado está disponível.
3) A impressora responde com o sinal /ACK, informando que o dado foi lido.
4) Somente após o sinal de /ACK é que o computador pode mandar um próximo dado.

O diagrama de tempo apresentado à página 2-4 do manual da PI/T refere-se ao protocolo *handshake pulsed* da PI/T, considerando que H2 corresponde ao /STB e H1 ao /ACK. Observar que o protocolo *handshake pulsed* da PI/T atende as especificações do protocolo Centronics, podendo, portanto, ser utilizado para a implementação deste.

6. A interface MC68230 no SBC68K

No SBC68K a interface MC68230 é utilizada para funções de entrada/saída no formato paralelo. Nesta máquina, que utiliza o esquema E/S Mapeada em Memória, os registradores da interface MC68230 estão alocados nos seguintes endereços:

Registrador	Endereço em SBC68K
PGCR	$FE8001
PSRR	$FE8003
PADDR	$FE8005
PBDDR	$FE8007
PCDDR	$FE8009
PIVR	$FE800B
PACR	$FE800D
PBCR	$FE800F
PADR	$FE8011
PBDR	$FE8013
PAAR	$FE8015
PBAR	$FE8017
PCDR	$FE8019
PSR	$FE801B

Tabela 4.6: Endereços dos registradores da MC68230 do SBC68K.

A interface MC68230 é utilizada no ambiente SBC68K/TUTOR para a conexão do computador com uma impressora através do conector PRINTER. O ligação física dos pinos deste conector com os pinos da interface MC68230, conjuntamente com a indicação do sinal Centronics associado, é apresentado na tabela 4.7:

Pino do Conector PRINTER	Pino da MC68230	Sinal Centronics
01	H2	/STB
02	Porta A - Bit 0	DATA 0
03	Porta A - Bit 1	DATA 1
04	Porta A - Bit 2	DATA 2
05	Porta A - Bit 3	DATA 3
06	Porta A - Bit 4	DATA 4
07	Porta A - Bit 5	DATA 5
08	Porta A - Bit 6	DATA 6
09	Porta A - Bit 7	DATA 7
10	H1	/ACK
11	Porta B - Bit 2	BUSY
12	Porta B - Bit 1	PE
13	Porta B - Bit 0	SELECT
14	-	-
15	H3	/FAULT
16	H4	/INIT
17	-	-
18	-	-
19	-	-
20	-	-
21	-	-
22	-	-
23	-	-
24	-	-
25	-	-

Tabela 4.7: Associação pinos do conector PRINTER - pinos da MC68230 - sinal Centronics.

O canal de comunicação com a impressora é referenciada no TUTOR como porta 3. Observar que o jumper J7 deve estar fechado para a comunicação com a impressora através do conector PRINTER.

7. Exercícios

OBS:

1. Nos exercícios 2 e 3 programar a MC68230 desconsiderando eventuais inicializações já realizadas pelo TUTOR (É necessário programar a PI/T).
2. No relatório justificar cada uma das instruções responsáveis pela inicialização/configuração da PI/T (É necessário documentar e justificar a programação da PI/T).

1. (3,0 pts) Considere o programa abaixo:

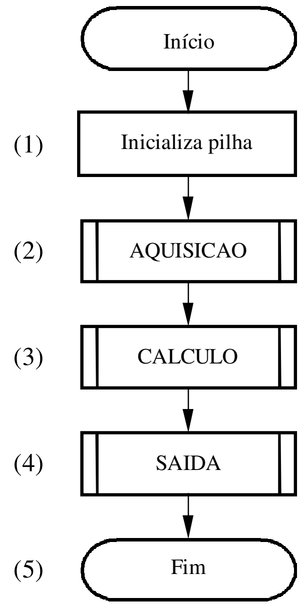
```
            ORG       $1000
INICIO:     LEA.L     $1000,A7     ; (1)
            BSR       AQUISICAO    ; (2)
            BSR       CALCULO      ; (3)
            BSR       SAIDA        ; (4)
            MOVE.B    #228,D7      ; (5)
            TRAP      #14          ; (5)
```

```
AQUISICAO:  MOVEM.L   D2-D7,-(A7)
            MOVE.B    #0,$FE8001
            MOVE.B    #0,$FE8007
            MOVE.B    #$60,FE800F
            CLR.L     D0
            CLR.L     D1
            MOVE.B    $FE8013,D6      ; Ponto A
            AND.B     #1,D6
ESPERA1:    MOVE.B    $FE8013,D7      ; Ponto B
            AND.B     #1,D7
            CMP.B     D6,D7
            BEQ.L     ESPERA1
            MOVE.B    D7,D6
AMOSTRA1:   MOVE.B    $FE8013,D7      ; Ponto C
            AND.B     #1,D7
            CMP.B     D6,D7
            BNE.L     TRATA2
            ADD.W     #1,D0
            BRA.L     AMOSTRA1
TRATA2:     MOVE.B    D7,D5
AMOSTRA2:   MOVE.B    $FE8013,D7      ; Ponto D
            AND.B     #1,D7
            CMP.B     D5,D7
            BNE.L     ARRUMA
            ADD.W     #1,D1
            BRA.L     AMOSTRA2
ARRUMA:     MOVE.W    D1,D2
            ADD.W     D0,D2
            CMP.B     #1,D6
            BEQ.L     FIM
            MOVE.W    D1,D0
FIM:        MOVE.W    D2,D1
            MOVEM.L   (A7)+,D2-D7
            RTS

CALCULO:    MULU.W    #10000,D0
            DIVU.W    D1,D0
            AND.L     #$FFFF,D0
            RTS

SAIDA:      MOVEM.L   D0-D7/A0-A6,-(A7)
            LEA.L     BUFFER,A6
            MOVE.W    #236,D7
            TRAP      #14
            MOVE.W    #3,D7
            LEA.L     MENSAGEM+$13,A5
COPIA:      MOVE.B    -(A6),-(A5)
            SUB.W     #1,D7
            CMPA.L    #BUFFER,A6
            BNE.L     COPIA
            CMP.W     #0,D7
            BLT.L     SAI
ZERO:       MOVE.B    #$30,-(A5)
            DBF       D7,ZER0
SAI:        LEA.L     MENSAGEM,A5
            LEA.L     MENSAGEM+$13,A6
            MOVE.B    #243,D7
            TRAP      #14
            MOVEM.L   (A7)+,D0-D7/A0-A6
            RTS

MENSAGEM:   DC.B      'DUTY CYCLE = 0,0000'
BUFFER:     DS.B      4
```



Este programa calcula o *duty cycle* de um trem de pulsos injetado no *bit* 0 da Porta B da PI/T. O *duty cycle* é definido por:

$$\frac{th}{th + tl}$$

onde: th (time high) é tempo em que o sinal permanece no nível alto, e

tl (time low) é o tempo em que o sinal permanece no nível baixo

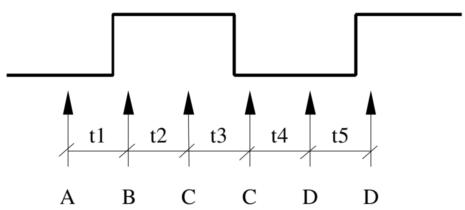
O programa é formado por três subrotinas:

1. AQUISICAO: Inicializa a interface paralela MC68630 e faz a aquisição dos tempos amostrando o sinal presente no bit 0 da porta B.
2. CALCULO: Efetua o cálculo do *duty cycle*.
3. SAIDA apresenta na tela do terminal o valor do *duty cycle*.

Insira, teste, analise o programa e responda:

a) Quais são as instruções responsáveis pela programação da interface MC68630? Quais são os registradores inicializados nesta programação? Justifique e comente os valores atribuídos para cada bit dos registradores inicializados.

b) Considerando a subrotina AQUISICAO, indique os valores dos tempos t1, t2, t3, t4 e t5, assumindo a amostragem do trem de pulsos como apresentada abaixo. As letras A, B, C e D representam os pontos da subrotina AQUISICAO, como apresentados na listagem, nos quais o sinal é amostrado. (Dica: Para o cálculo dos tempos, utilizar os tempos de execução das instruções). Qual o tipo de transferência (veja seção 3 do roteiro) implementado pela subrotina AQUISICAO ?



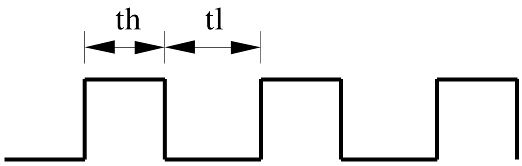
c) Modifique a subrotina AQUISICAO para que t2 = t3 = t4 = t5 (Dica: inserir instruções que gastem tempo e não alterem o comportamento do programa). Por que a condição t2 = t3 = t4 = t5 é desejável ? Qual é a frequência mínima do sinal mensurável pela subrotina? E a máxima ?

d) Por definição o *duty cycle* é sempre menor do que 1. Qual foi o expediente utilizado na subrotina CALCULO para o cálculo do *duty cycle* utilizando a instrução DIVU (divisão inteira sem sinal) ? Justifique

e) Na subrotina SAIDA foi utilizada a função utilitária do TUTOR HEX2DEC (função 236) para a conversão hexadecimal para decimal. Esta função, entretanto, descarta eventuais zeros que residam na frente do número. Qual foi o expediente utilizado na subrotina SAIDA para contornar este problema e sempre apresentar na tela do terminal um valor de *duty cycle* com 4 casas decimais.

OBS.: Para o teste do programa utilizar gerador de trem de pulsos nível TTL, osciloscópio e o conector especialmente confeccionado para esta experiência.

2. (3,0 pt) Fazer um programa que gere, no bit 0 da porta A da MC68230 (pino 2 do conector PRINTER), o seguinte sinal periódico:



onde: os valores de **th** e **tl**, em milisegundos, devem ser definidos pelo usuário através do teclado.

Justificar a programação da PI/T, comentando cada um dos *bits* dos regitradores de controle inicializados.

OBS.: Utilizar um osciloscópio para monitorar o sinal gerado.

3. (4,0 pts) Fazer uma subrotina que imprima caracter ASCII contido em D0 (observar que esta subrotina é funcionalmente equivalente à função CHRPRINT do TUTOR). Testar esta subrotina com programa principal que (observar que são dois programas principais distintos):

a) Imprima e mostre na tela do terminal caracteres teclados pelo usuário. Utilizar as funções INCHE e OUTCH para a comunicação com o terminal. O caracter CTRL-A ($01), ao ser digitado, deve terminar o programa.

b) Imprima em uma linha 7 vezes a seqüência "ABCDEFGHIJ”. A linha deve ser terminada com CR (carriage return - $0D) seguido de LF (line feed - $0A). Repetir indefinidamente a impressão desta linha.

OBS.:

1) Utilizar o protocolo de *handshake pulsed* da PI/T
2) A impressora está configurada para imprimir uma linha apenas após o recebimento do caracter LF.

8. Referências

[1] Arnewsh Inc.: *SBC68K User’s Manual Revison 1.1*, 2nd Edition, 1990.

[2] Clements, Alan: *Microprocessor Systems Design - 68000 hardware, software, and interfacing*, 2nd Edition, PWS-Kent Publishing Company, 1992.

ANEXO A
TEMPO DE EXECUÇÃO DAS INSTRUÇÕES

ANEXO B
INTERFACE PARALELA MC68230