**3. The PC's Timer, the i8254 IC**

- Os timers operam de maneira idêntica, logo é apenas necessário descrever o modo de funcionamento de um.

- Um timer tem:

* Duas linhas de input: *Clock*, que é usada para mandar o pulso; *Enable*, que é usado para ativar/desativar o timer
* Um contador de 16 bits, que é decrementado em 1 para cada pulso da linha *Clock*
* Uma linha *Out* de output, que depende do valor do contador

- Cada relógio é programado independentemente dos outros. Para programar um timer, é necessário:

1. Especificar o modo de operação do timer, escrevendo uma control word (8bits) para o registo de controlo.
2. Definir o valor inicial do counter, escrevendo no registo de controlo.

- A seguinte tabela explicita a formatação de uma control word

|  |  |  |
| --- | --- | --- |
| **Bit** | **Value** | **Function** |
|  |  | **Select counter** |
| 7,6 | 00 | 0 |
| 01 | 1 |
| 10 | 2 |
|  |  | **Initialization Mode** |
| 5,4 | 01 | LSB |
| 10 | MSB |
| 11 | LSB followed by MSB |
|  |  | **Operating Mode** |
| 3,2,1 | 000 | 0 |
| 001 | 1 |
| 010 | 2 |
| 110\* |
| 011 | 3 |
| 111\* |
| 100 | 4 |
| 101 | 5 |
|  |  | **Counting base** |
| 0 | 0 | Binary (16 bits) |
| 1 | BCD (4 decades) |

* Os bits 6 e 7 identificam o timer a programar.
* Os bits 4 e 5 identificam como o valor inicial é carregado
  + LSB: lido apenas o byte menos significativo;
  + MSB: lido apenas o byte mais significativo;
  + LSB followed by MSB: é lido primeiro o LSB, seguido do MSB;
  + Isto é explicado porque o i8254 tem apenas linhas de 8bits. Assim, para carregar o valor inicial do counter, é necessário ler o LSB e o MSB separadamente. O i8254 permite carregar qualquer um dos bytes, ou ambos.
* Os bits 1, 2 e 3 especificam o modo de operação
* o bit 0 especifica se o contador é binário ou BCD.

- Para ler a configuração do timer, é necessário usar um Read-Back command. Este comando permite ler o modo programado, o valor do contador e o estado atual da linha *Out.*

|  |  |  |
| --- | --- | --- |
| **Bit** | **Value** | **Function** |
| 7,6 | 11 | **Read-Back Command** |
| 5 |  | **COUNT** |
| 0 | Read counter value |
| 4 |  | **STATUS** |
| 0 | Read Programmed Mode |
| 3 |  | **Select Counter 2** |
| 1 | Yes |
| 2 |  | **Select Counter 1** |
| 1 | Yes |
| 1 |  | **Select Counter 0** |
| 1 | Yes |
| 0 | 0 | **Reserved** |

* Como a *Control Word*, o *Read-Back command* também é escrito no *control register*
* Os bits 6 e 7 servem para distinguir um *Control Word* de um *Read-Back command*
* Os bits 4 e 5 identificam o que se quer ler: ou o valor do contador ou o modo programado.
* Os bits 1 a 3 identificam qual o bit a ser lido

- A seguinte tabela representa a formatação do *Status byte* do timer:

|  |  |
| --- | --- |
| **Bit** | **Function** |
| 7 | Output |
| 6 | Null Count |
| 5, 4 | Type of Access |
| 3,2,1 | Programmed Mode |
| 0 | BCD |

* Os bits 0 a 5 contêm os bits correspondentes à ultima *Control Word* escrita.
* O bit 6 está relacionado à leitura do valor do contador
* O bit 7 contém o valor atual da linha *Out*

- Todos os timers usam o mesmo clock signal com frequência *1193181* Hz.

- O output do timer2 está ligado às colunas do PC, e é gerado para gerar tons usando a *square wave*.

-(Não necessário saber) Furthermore, to enable the speaker, you must set to 1 both bits 0 and 1 of I/O port 0x61 (SPEAKER\_CTRL) . As shown in Figure 1, bit 0 is connected to the GATE of Timer 2 and if low, it will disable the timer. In addition, the OUT line of timer 2 is not connected directly to the speaker, instead it is gated via a NAND, whose other input line comes from bit 1 of port 0x61. Thus, unless that bit is set to 1, the input to the speaker will always be high, and no tone will be generated.

- O endereço do Control Register é 0x43, o do Timer0 é 0x40, o do Timer1 é 0x41 e o do Timer2 é 0x42.