

Temporizadores de hardware

Sondagem de temporizadores de hardware

AULA 7

IOUL IIA SKL I AROVA

IPs de temporizador de hardware

Temporizador AXI

- AXI Timer/Counter é um módulo de temporizador de 32/64 bits que faz interface com a interface AXI4-Lite.

Temporizador de intervalo fixo (FIT)

- O núcleo FIT é um periférico que gera um sinal estroboscópico (interrupção) em intervalos fixos e não está conectado a nenhum barramento.

Temporizador AXI

Dois temporizadores de intervalo programáveis com recursos de interrupção, geração de eventos e captura de eventos.

Cada módulo de temporizador possui um registro de carga associado que é usado para armazenar o valor inicial do contador para geração de eventos ou um valor de captura, dependendo do modo do temporizador.

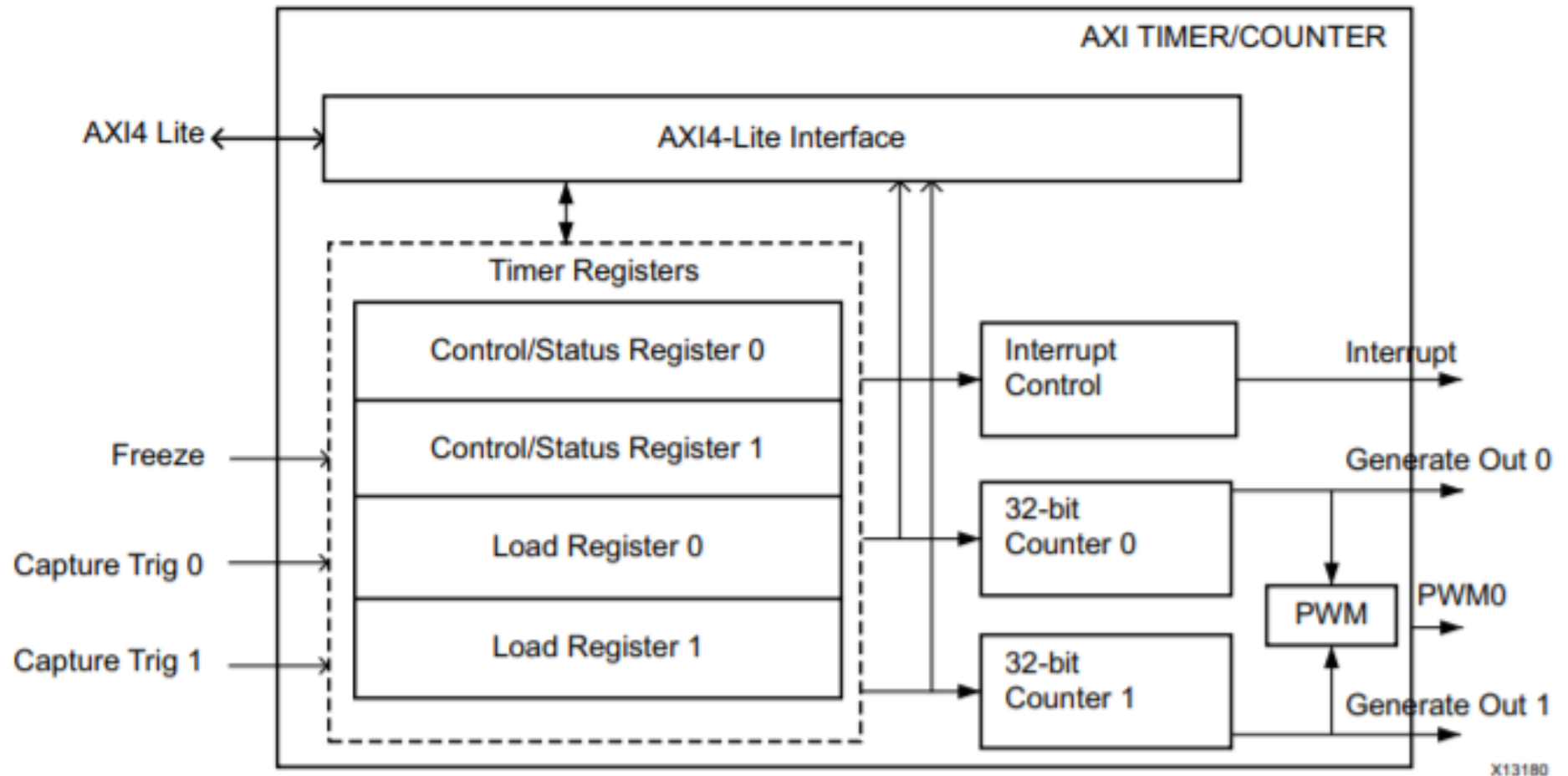
No modo Gerar, o valor do registrador de carga é carregado no contador. O contador, quando habilitado, começa a contagem crescente ou decrescente, dependendo da seleção do bit Up/Down Count Timer (UDT) no Timer Control Status Register (TCSR).

Na transição do carry out do contador, o contador para ou recarrega automaticamente o valor gerado do registrador de carga e, após atingir o valor de timeout, continua contando conforme selecionado pelo bit Auto Reload/Hold (ARHT) no TCSR.

O bit Timer Interrupt Status (TINT) é definido no TCSR e, se habilitado, o sinal GenerateOut externo é levado a 1 por um ciclo de clock.

Se habilitado, o sinal de interrupção do temporizador é levado a 1 ao atingir o valor de timeout. Limpe a interrupção escrevendo 1 no registro de interrupção do temporizador. Use este modo para gerar interrupções repetitivas ou sinais externos com um intervalo especificado.

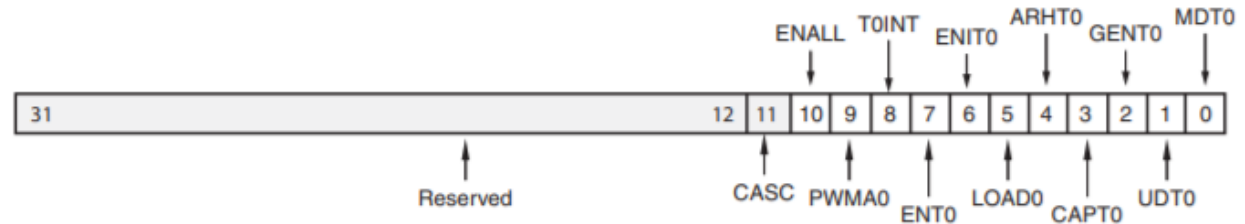
Diagrama de blocos do temporizador AXI



Registro de controle/status do temporizador AXI

T0INT - Interrupção do Timer 0 Indica que ocorreu a condição de interrupção deste timer.

- Ler:
 - 0 = Nenhuma interrupção ocorreu
 - 1 = Ocorreu interrupção
- Escrever:
 - 0 = Nenhuma alteração no estado de T0INT
 - 1 = Limpar T0INT (limpar para 0)



ENT0 - Habilitar temporizador 0

- 0 = Desativa o temporizador (o contador para)
- 1 = Habilita temporizador (contador funciona)

LOAD0 - Temporizador de carga 0

- 0 = Sem carga
- 1 = Carrega temporizador com valor em TLR0

ARHT0 - Auto Reload/Hold Timer 0 (quando o temporizador está no modo Generate, este bit determina se o contador recarrega o valor gerado e continua funcionando ou retém no valor de terminação).

- 0 = Contador de retenção
- 1 = Recarregar gera valor

UDT0 - Temporizador de contagem crescente/regressiva 0

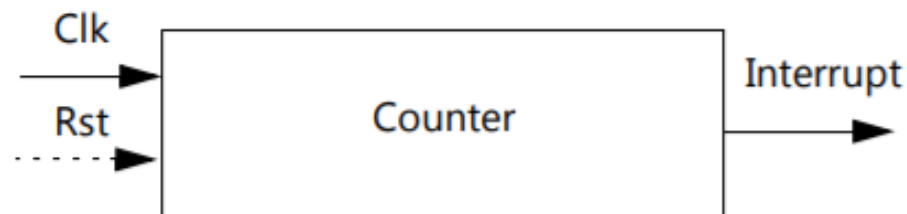
- 0 = O temporizador funciona como contador crescente
- 1 = O temporizador funciona como contador decrescente

Temporizador de intervalo fixo (FIT)

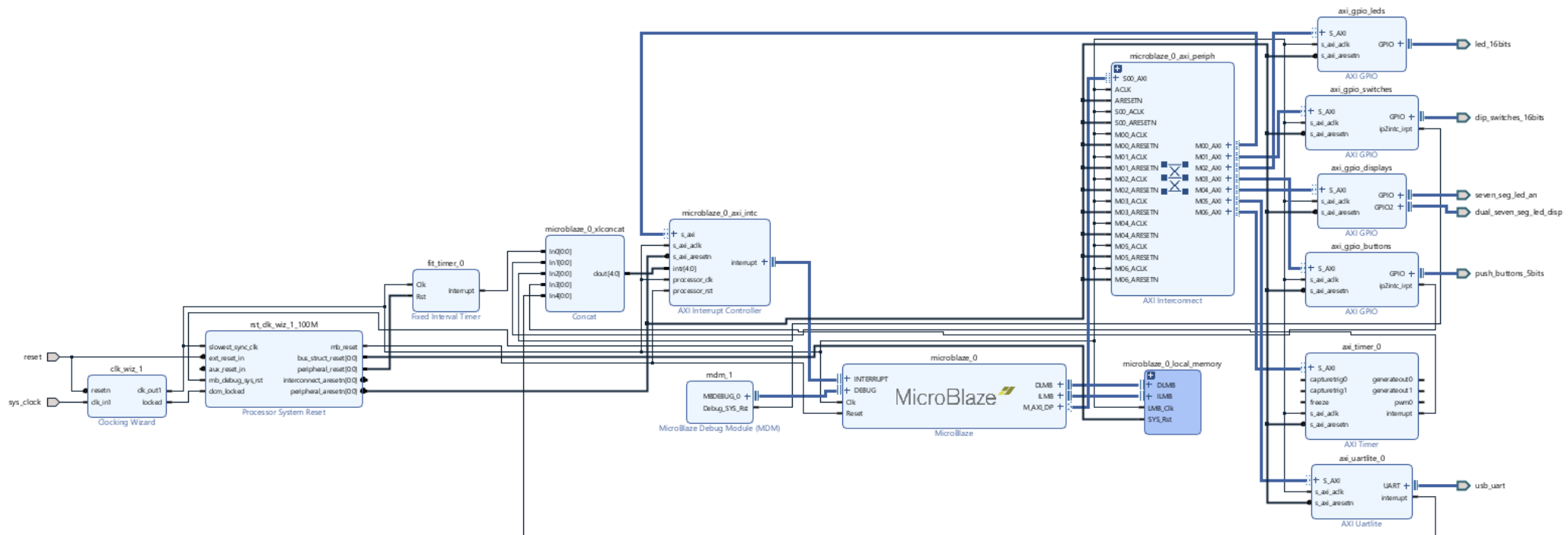
O núcleo FIT é um periférico que gera um sinal estroboscópico (interrupção) em intervalos fixos e não está conectado a nenhum barramento.

O núcleo FIT gera uma interrupção a cada C_NO_CLOCKS. O sinal de interrupção é mantido em nível alto por um ciclo de clock.

O núcleo começa a operar imediatamente após a configuração do dispositivo se o reset não estiver conectado.



Design de Bloco (BD)



Votação

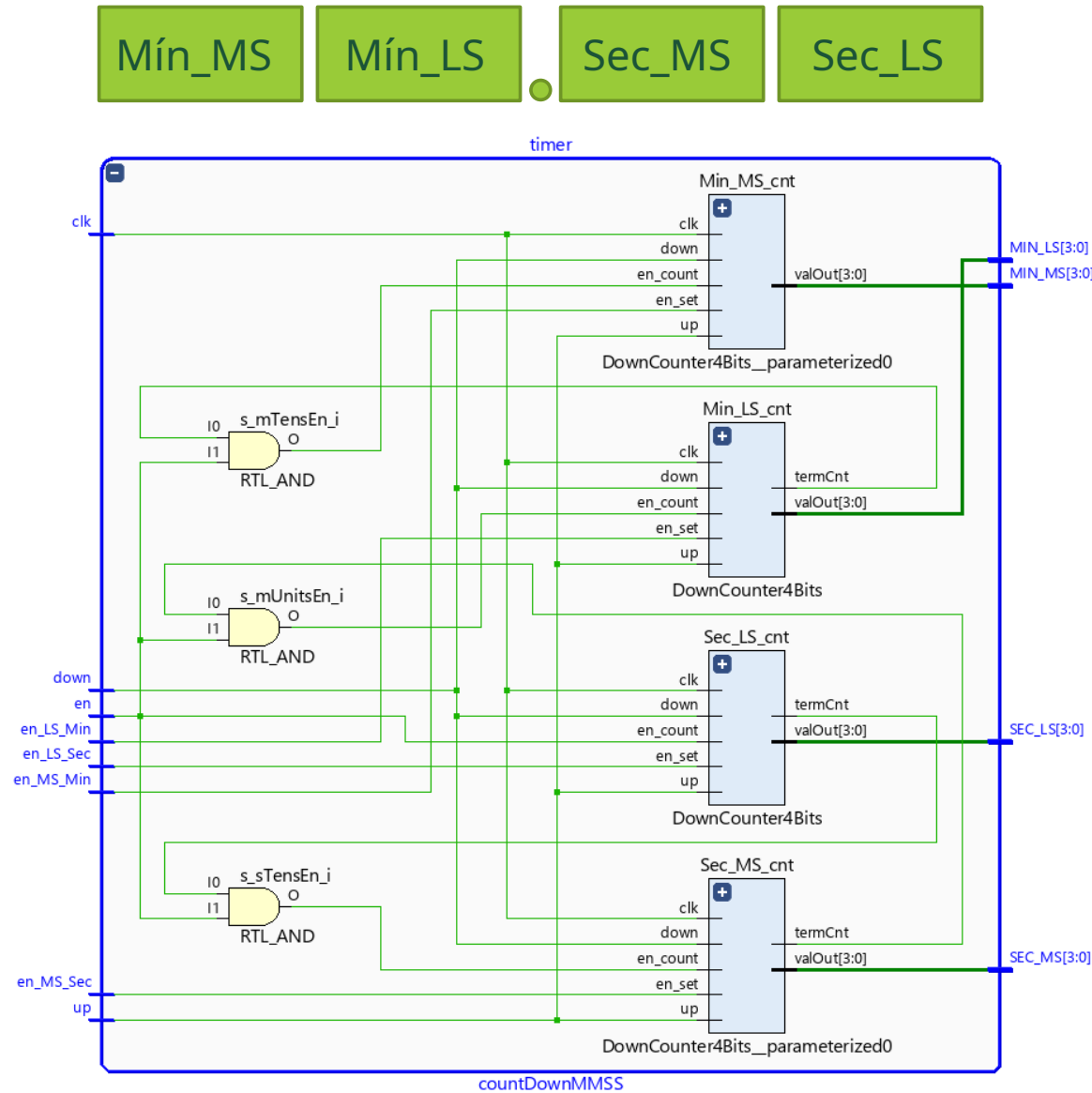
Votação significa ler repetidamente um registro de temporizador e testar o valor dos dados de entrada.

- simples de codificar
- não é muito eficiente porque a votação acontece mesmo que a leitura do temporizador não tenha atingido o valor requerido

Funções de driver de baixo nível (ou macros) que podem ser usadas para acessar o temporizador:

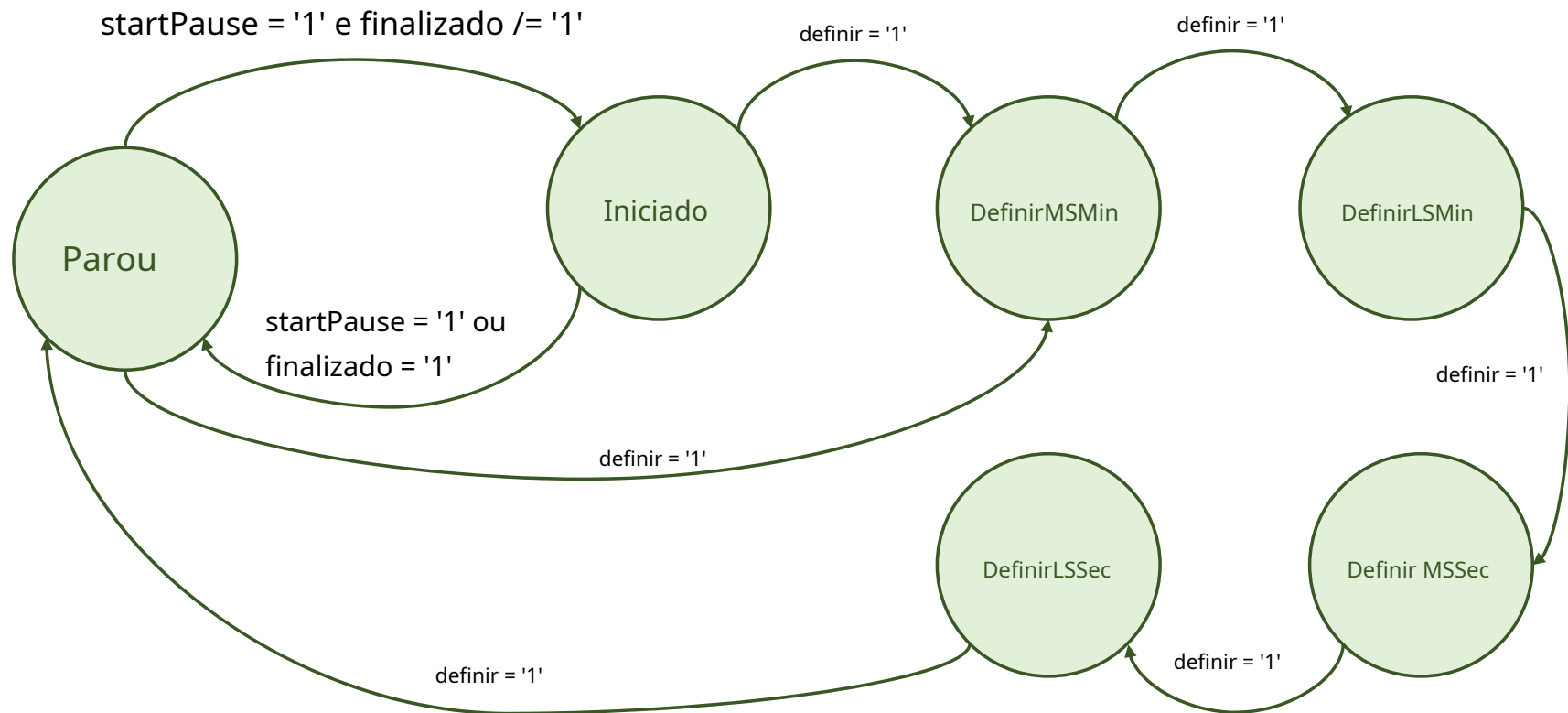
- XTmrCtr_SetControlStatusReg
- XTmrCtr_GetControlStatusReg
- XTmrCtr_SetLoadReg
- XTmrCtr_GetTimerCounterReg

Caminho de dados do temporizador de contagem regressiva



Caminho de controle do temporizador de contagem regressiva

```
conjunto = btnR
startPause = btnC
```



Temporizador de contagem regressiva em software

Que operações periódicas devem ser executadas e com que frequência? 1 hertz

- para diminuir o cronômetro de contagem regressiva

800Hz

- para atualizar as exibições de 7 segmentos

2Hz

- para definir a hora
- piscar o ponto que separa minutos e segundos (com frequência de 1Hz)

4Hz

- para fazer o dígito que está sendo definido piscar (com frequência de 2 Hz)

8Hz

- para ler o status dos botões

Código C para o cronômetro de contagem regressiva

```
// State machine data type
typedef enum {Stopped, Started, SetLSec, SetMSSec, SetLSMin, SetMSMin} TFSMState;

// Buttons GPIO masks
#define BUTTON_UP_MASK      0x01
#define BUTTON_DOWN_MASK   0x04
#define BUTTON_RIGHT_MASK  0x08
#define BUTTON_CENTER_MASK 0x10

// Data structure to store buttons status
typedef struct SButtonStatus
{
    bool upPressed;
    bool downPressed;
    bool setPressed;
    bool startPressed;

    bool setPrevious;
    bool startPrevious;
} TButtonStatus;

// Data structure to store countdown timer value
typedef struct STimerValue
{
    unsigned int minMSValue;
    unsigned int minLSValue;
    unsigned int secMSValue;
    unsigned int secLSValue;
} TTimerValue;
```



Teste 1

Estrutura

- Perguntas verdadeiro/falso – sem penalidade por respostas erradas
- Questões de múltipla escolha – 33% de desconto para respostas erradas; sem desconto por não responder
- Perguntas Cloze (preencher um determinado código, calcular alguns valores) – sem penalidade

Organização

- Durante a aula de **8 de abril**
- 1h
- É permitido o uso de calculadoras
- Prepare uma caneta e papel
- Nenhuma consulta é permitida

Material

- Tudo considerado nas aulas anteriores
- No entanto, não haverá dúvidas sobre desenvolvimento de software

Considerações finais

Ao final desta palestra você deverá ser capaz de:

- escrever programas C que pesquisam um temporizador de hardware

Pendência:

- Laboratório. 6 parte 1