

# Especificação Módulo Counter\_User

## 1 Especificação - Módulo Counter\_User

O módulo **Counter\_User** é responsável por incrementar uma sequência de números conforme o usuário avança nas rodadas de um jogo. Ele verifica se o limite da contagem foi atingido, e sinaliza quando o limite é alcançado.

## 2 Interfaces

### 2.1 INPUT

- **clk**
  - **Tipo:** wire
  - **Descrição:** Sinal de clock que sincroniza as operações do contador.
- **R**
  - **Tipo:** wire
  - **Descrição:** Sinal de reset. Quando ativado ('1'), o contador é reinicializado para zero.
- **E**
  - **Tipo:** wire
  - **Descrição:** Sinal de habilitação. Quando ativado ('1'), permite que o contador incremente seu valor.
- **data**
  - **Tipo:** wire [3:0]
  - **Descrição:** Define o limite de contagem. Quando o valor do contador atinge este limite, o sinal **tc** é ativado.

## 2.2 OUTPUT

- **tc**
  - **Tipo:** Reg
  - **Descrição:** Sinal de término da contagem. É ativado ('1') quando o contador atinge o limite definido por **data**.

## 3 Blocos e Sinais Internos

- **Bloco de Processamento**
  - **Processo Principal:** O processo principal é responsável por:
    - \* Reinicializar o contador (**total**) quando o sinal de reset (**R**) está ativado.
    - \* Incrementar o valor do contador (**total**) a cada borda de subida do clock (**clk**) quando o sinal de habilitação (**E**) está ativado.
    - \* Verificar se o contador atingiu o limite definido por **data** e ajustar o sinal **tc** adequadamente.
- **Sinal Interno**
  - **total**
    - \* **Tipo:** reg [3:0]
    - \* **Descrição:** Armazena o valor atual do contador. O valor máximo de contagem é "1111" (15 em decimal).

## 4 Fluxo de Operação

- **Reset:** Quando o sinal **R** é ativado ('1'), o contador **total** é reinicializado para "0000" e o sinal **tc** é desativado ('0').
- **Incremento:** A cada borda de subida do clock (**clk**) e com o sinal de habilitação (**E**) ativado ('1'), o contador **total** é incrementado.
- **Verificação de Limite:** Após cada incremento, o valor do contador **total** é comparado com o valor de **data**. Se os valores forem iguais, o sinal **tc** é ativado ('1'); caso contrário, **tc** permanece desativado ('0').

## 5 Exemplo de Aplicação

Este componente pode ser utilizado em jogos eletrônicos onde é necessário rastrear a sequência de ações do usuário em rodadas, verificando se um determinado objetivo foi atingido conforme definido por **data**.

## 6 Considerações

- **Reset Assíncrono:** O reset do contador é assíncrono, ou seja, ele pode ser acionado independentemente do clock.
- **Limite de Contagem:** A contagem é limitada a 4 bits, permitindo valores de 0 a 15. Ajustes podem ser necessários para suportar contagens maiores.

## 7 Possíveis Melhorias

- **Ajuste de Largura do Contador:** Aumentar a largura do vetor `total` para suportar contagens maiores, se necessário.
- **Contagem Bidirecional:** Adicionar funcionalidades para decremento, se necessário.
- **Sinais de Controle Adicionais:** Incluir sinais para pausa ou reinício de contagem sem reset completo.

## 8 Conclusão

O componente **Counter\_User** é essencial para aplicações que necessitam de um contador simples e eficiente com capacidade de verificar limites pré-definidos, como em jogos ou sequências controladas pelo usuário.