CONTADOR SÍNCRONO EM VHDL

Mathaus C. Huber, Felipe Bueno

[1]Universidade Federal de Pelotas (UFPEL) – Discentes do Curso Superior de Ciência da Computação
R. Gomes Carneiro, 1 - Centro – 96075-630 – Pelotas – RS – Brazil

mchuber@inf.ufpel.edu.br, felipe13579@yahoo.com.br

Abstract. *This paper describes the second work of the digital techniques course, given to the second semester of the Computer Science course at the Federal University of Pelotas, which refers to the creation of an architectural project, vhdl description, and prototyping of a synchronous counter. 6 (six) bits, with some functions, are: clear, hold, load, sense, departure.*

Resumo. *Este artigo descreve o segundo trabalho da disciplina de técnicas digitais, conferida ao segundo semestre do curso de Ciência da Computação, da Universidade Federal de Pelotas, no qual se refere a criação de um projeto arquitetural, descrição em vhdl, e prototipação de um contador sincrono de 6 (seis) bits, com algumas funções, são elas: clear, hold, load, sentido, partida.*

1. Informação Geral

Neste trabalho foi utilizado uma FPGA (que consiste em um arranjo de células lógicas ou blocos lógicos configuráveis contidos em um único circuito integrado), a linguagem de descrição de hardware (VHDL) para implementar tais funções, além do software Quartus II da Altera.

1.1. Contador Síncrono

Nesses contadores, o sinal de relógio entra em todos os flip-flops simultaneamente. Para que haja mudanças de estado, deve-se então estudar o comportamento das entradas J e K dos vários flip-flops, para que se tenha nas saídas as sequências desejadas. O contador síncrono é o mais completo contador, ele tem condições de gerar qualquer tipo de sequência binária, ou seja, é um gerador de palavras e consequentemente de códigos binários.

Em teoria, todos tipos de contadoresem, levando em consideração o princípio de funcionamento teórico dos transistores, o qual diz que: ”os transistores não devem conduzir quando nenhum sinal é aplicado em seu terminal de controle, no caso, o terminal da base em transistores bipolares de junção, e o terminal da porta em transistores mosfets, salvo uma possível corrente de fuga, que é insignificante, a qual não alteraria os estados de funcionamento do circuito”. Partindo deste princípio, no estado inicial, todos os flip flops dos contadores deveriam apresentar idealmente em suas saídas Q o nivel lógico binário 0, e nas saídas Q’ o nível lógico binário 1, contudo na prática isto é bem difícil de acontecer pelos flip flops por sí só, se não forem tomadas medidas de inicialização.

2. Implentação

O projeto de um contador síncrono crescente / decrescente necessita da inserção de uma variável de controle. No nosso caso, vamos chamar essa variável de controle de variável ”cont”, que será inserida no circuito combinacional para executar uma etapa de controle. Optamos por não usar a pinagem do clock em 27MHz, como está habitualmente usado no manual da placa da Altera, o que seria possível somente pela criação de um divisor de frequência, para conseguir obter o resultado desejado na placa, dessa forma, utilizamos um dos switches para realizar a operação de clock, sendo que, as entradas clear, load, sentido e hold, são assíncronos, ou seja, não dependem do clock.

Criamos também, um arquivo vhdl como nosso top-level, para fazer os port maps do arquivo 'contador' para as saídas dos leds, ou seja, nosso display de sete segmentos, sendo eles, um arquivo para a vizualização das unidades denominado ”seg71”, e um arquivo para a vizualição das dezenas, denominado ”seg72”. Foi feito também a pinagem das entradas e saídas em seus respectivos switches, assim como está indicado em nosso manual do contador síncrono, também disponível na pasta do trabalho.

3. Hold

O hold é a função que mantém o contador recebendo o seu valor atual, ou seja, funcionaria como uma função de um relógio parado, sem pilhas, onde a única informação resultante nos leds do display de sete segmentos seria o resultado atual quando o hold recebe a parâmetro lógico '1', ou ligado. Mostramos na figura 1 como é implementada essa função:

4. Clock

O clock é a nossa função que faz o contador somar um, no caso do sentido progressivo, ou diminuir um, no caso do sentido regressivo, como dito antes, ao invés de o usar o pulso de clock ligado a uma das entradas da placa, que operam em 27Mhz, optamos pelos switches, ou seja, o clock operando manualmente, a cada subida no pino equivale a um pulso de clock. Mostramos na figura 2 como é implementada essa função:

5. Clear

O clear é a função que limpa o nosso contador, ou seja, colocando zeros na saída e na variável cont, fazendo com que o display de sete segmentos mostre algo referente a uma reinicialização do contador, sendo ele começado do zero, independente do sentido da contagem (crescente ou decrescente). Mostramos na figura 3 como é implementada essa função:

6. Load

O load é a função que carrega um valor de 6 bits para dentro do nosso contador, mantendo essa informação atual, independente do clock, seria como deixar nosso relógio parado, porém com a informação desejada. Mostramos na figura 4 como é implementada essa função:

```
        cont:= partida;
    if (hold or clear)='1' then --se hold ou clear for =1, então

        if hold= '1' then   --se hold =1, então
                cont:=cont;
        end if;
```

Figura 1. hold vhdl

```
else if clock = '1' then   --se não, se clock for=1,entao

    if sentido= '1' then    --se sentido for = 1 , entao

        if load= '1' then   -- se load for = 1, entao
                cont:= valor;
        else                                    --se não...
                cont:= cont + 1;
        end if;

        if cont > 59 then       --se cont for maior que 59, entao
                cont:="000000";
        end if;
```

Figura 2. clock vhdl

7. Projeto arquitetural

O circuito é composto por um arquivo top level, nomeado segmento7, onde conecta as saidas dos dois displays de sete segmentos, onde as entradas clock, clear, sentido, hold, valor, load, e as operações entram no contador, saem com os resultados e os respectivos, passam pelo arquivo segmento7 que posteriormente os envia para suas saídas. Assim como pode se observar na figura 5 e 6.

```
if clear = '1'then  --se clear =1, então
        cont :- "000000";
end if;
        saida <= cont;
```

Figura 3. clear vhdl

```
        cont:= partida;
if (hold or clear)='1' then --se hold ou clear for =1, então

        if hold= '1' then   --se hold =1, então
                cont:=cont;
        end if;
```

Figura 4. Load vhdl

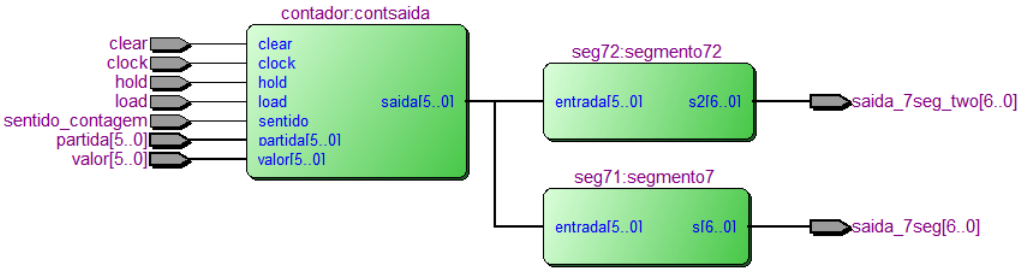

Figura 5. arquitetura vhdl

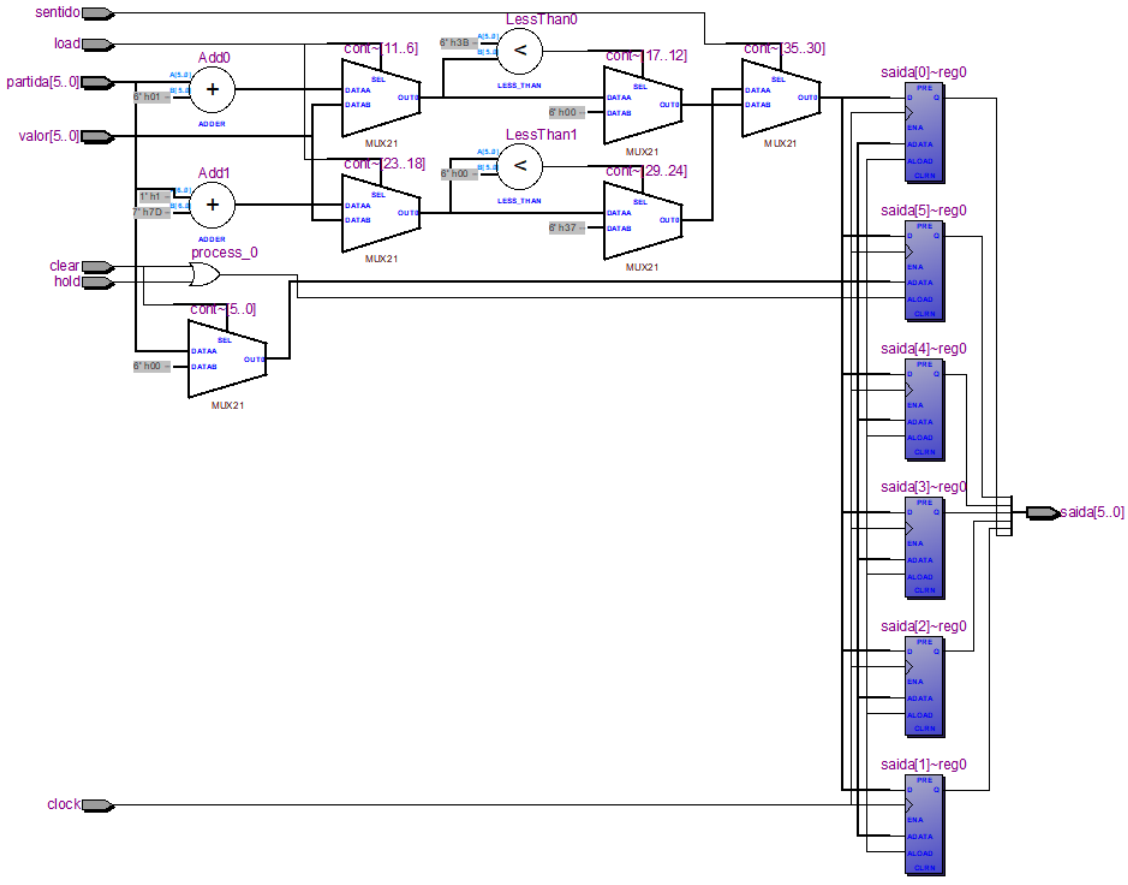

Figura 6. arquitetura vhdl