deti universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

Projeto CR

Acelerar a verificação e contagem de palíndromos com um Coprocessador de Hardware

Lúcia Sousa 93086
Raquel Pinto 92948

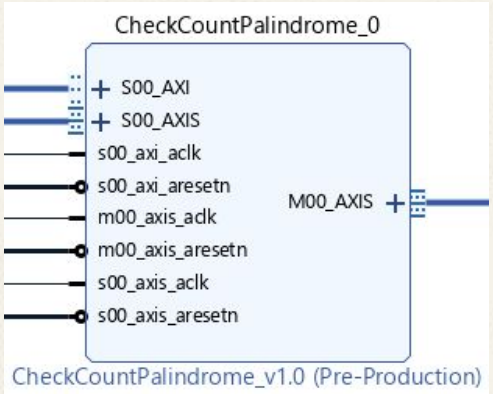
22/06/2022 CR - Computação Reconfigurável

Descrição do projeto

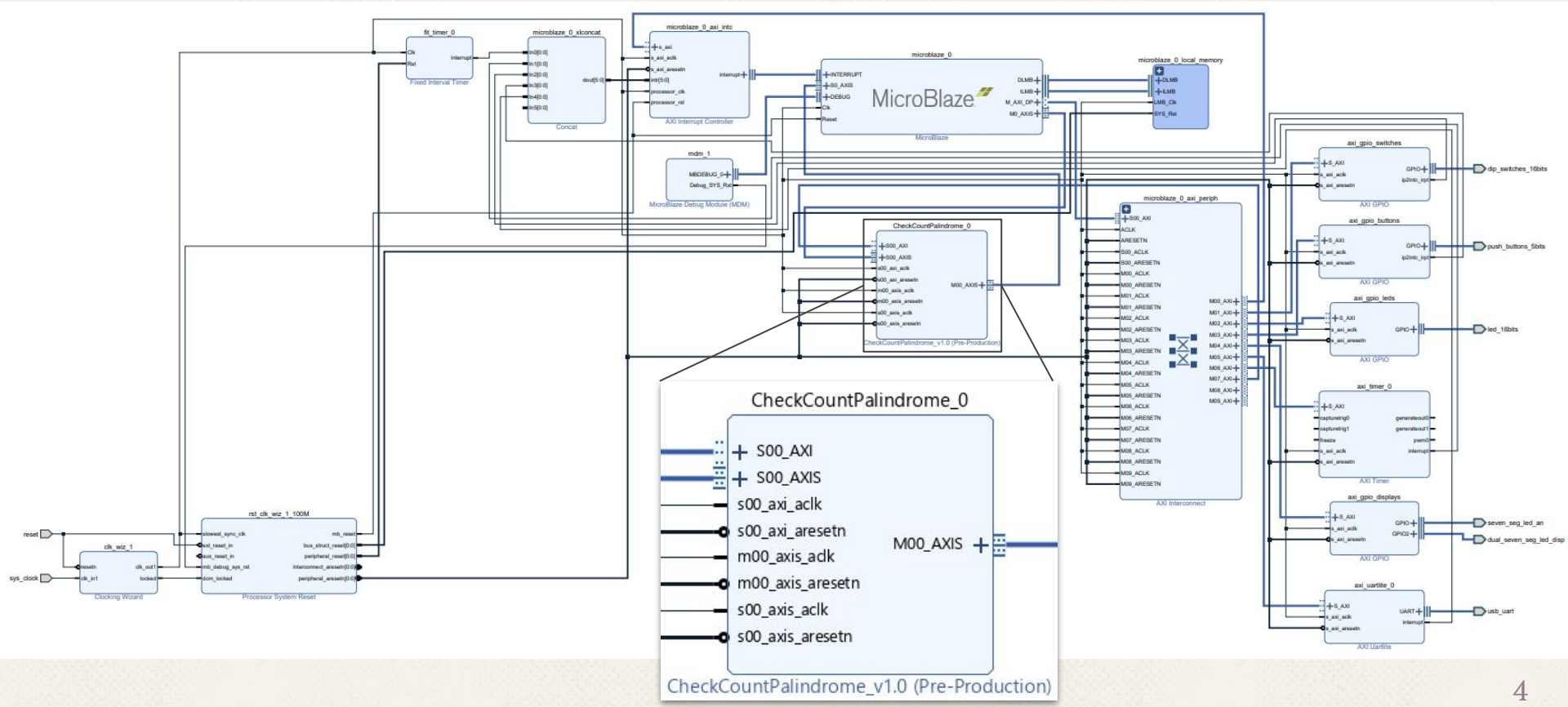
- Sistema com um acelerador hardware para verificar se cada vetor de 32 bits, lido da memória, é um palíndromo e contar quantos palíndromos existem na memória. (16 vetores de 32 bits cada um - entradas não aleatórias);
- Objetivo → resultado do contador igual no software e hardware e hardware ser mais rápido que software.

Descrição do projeto

- Módulo CheckCounterPalindrome:
  - Interface AXI-Stream Slave para receber dados do MicroBlaze que envia dados (vetores) para o novo módulo através da interface AXI-Stream Master. Algoritmo de verificação de palíndromos.
  - Interface AXI-Stream Master para enviar dados (resultado do algoritmo) ao Microblaze que os recebe através da sua interface AXI-Stream Slave.
  - Interface AXI-Lite Slave que comunica com AXI Interconnect para realizar um contador acumulativo feito por software (endereço base + 4). Escreve o resultado do algoritmo (endereço base).



Block Design



Algoritmo de verificação de Palíndromos

- O módulo PalindromoCheck foi implementado para verificar se um vetor de 32 bits é um palíndromo. Este contém uma entrada de 32 bits e uma saída de 32 bits.
- O vetor que recebe como entrada é invertido, e se a palavra que entrou é igual ao seu inverso, é um palíndromo e é concatenado a 31 zeros o bit a 1 do resultado, caso não seja igual, são enviados 32 bits a zero.

```
process(dataIn)
begin
    for i in 0 to dataIn'length-1 loop
        s_dataIn_reverse(i) <= dataIn(31-i);
    end loop;
end process;
result <= '1' when s_dataIn_reverse = dataIn else '0';
dataOut <= "00000000000000000000000000000000" & result;
```

Resultados e Demonstração

- Foram contados 9 palíndromos, tanto em hardware como em software.
- Software demorou cerca de 1,39 vezes mais em comparação ao hardware.

```
//palindromos = 9 -> indices 0, 3, 4, 6, 8, 9, 11, 13, 15
srcData[0] = 1704565158; //0110 0101 1001 1001 1001 1001 1010 0110
srcData[1] = 35648752;   //0000 0010 0001 1111 1111 0100 1111 0000
srcData[2] = 249224586;  //0000 1110 1101 1010 1101 1101 1000 1010
srcData[3] = 4294967295; //1111 1111 1111 1111 1111 1111 1111 1111
srcData[4] = 0;          //0000 0000 0000 0000 0000 0000 0000 0000
srcData[5] = 423077452;  //0001 1001 0011 0111 1010 0110 0100 1100
srcData[6] = 2702176389; //1010 0001 0000 1111 1111 0000 1000 0101
srcData[7] = 767213316;  //0010 1101 1011 1010 1011 1111 0000 0100
srcData[8] = 3536044875; //1101 0010 1100 0011 1100 0011 0100 1011
srcData[9] = 190593744;  //0000 1011 0101 1100 0011 1010 1101 0000
srcData[10] = 463115274; //0001 1011 1001 1010 1001 0100 0000 1010
srcData[11] = 3567625515;//1101 0100 1010 0101 1010 0101 0010 1011
srcData[12] = 520016216; //0001 1110 1111 1110 1101 0001 0101 1000
srcData[13] = 2483703849;//1001 0100 0000 1010 0101 0000 0010 1001
srcData[14] = 356733678; //0001 0101 0100 0011 0101 0010 1110 1110
srcData[15] = 463100376; //0001 1011 1001 1010 0101 1001 1101 1000
```

```
Filling memory with pseudo-random data...

Memory initialization time: 4 microseconds

Resultado software counter -> 9

Software only time: 33328 microseconds
Resultado counter -> 9

Hardware assisted reverse endianness time: 23954 microseconds

Checking result: OK
```