

Trabalho prático 8

Neste trabalho prático deverá implementar e testar coprocessadores de hardware especializados para aceleração das operações "*reverse endianness*" (Parte 1) e "*population count*" (Parte 2). Ambos os coprocessadores serão ligados a um processador MicroBlaze através de interfaces AXI4-Stream (Master e Slave).

A operação "*reverse endianness*" inverte a ordem dos bytes numa palavra, neste caso de 32 bits, convertendo uma representação *little endian* numa *big endian*.

A operação "*population count*" conta o número de bits a "1" numa palavra, neste caso também de 32 bits.

Ambas as operações podem ser realizadas completamente em software, mas também podem ser aceleradas com o auxílio de hardware especializado. As duas abordagens, para ambas as operações, vão ser implementadas e comparadas neste trabalho prático.

Parte 1

1. Crie no Xilinx Vivado um novo projeto para a placa Digilent Nexys-4 (sugestão de nome: "StreamCopDemo").
2. Adicione o *block design* da plataforma base, ao projeto que acabou de criar (apesar da maioria dos periféricos dessa plataforma não ser usada neste trabalho prático).
3. Crie um novo IP *core* com o nome "ReverseEndiannessCop", cuja função será trocar o *endianness* de uma palavra de 32 bits, com:
 - uma interface AXI-Stream Master (com o nome "M00_AXIS");
 - uma interface AXI-Stream Slave (com o nome "S00_AXIS").

Algumas notas:

- Respeite escrupulosamente todos os nomes indicados;

- O novo IP *core* não deverá possuir qualquer outra interface além das duas indicadas ("M00_AXIS" e "S00_AXIS");

Questão: O que distingue uma interface AXI-Stream de uma *Memory Mapped*?

4. Modifique a configuração do MicroBlaze de forma a disponibilizar um par Master-Slave de interfaces AXI-Stream.

5. Instancie, no *block design*, o IP *core* "ReverseEndiannessCop" e ligue-o ao MicroBlaze. As ligações das interfaces AXI-Stream têm de ser realizadas manualmente: a interface AXI-Stream Master do MicroBlaze deve ligar à AXI-Stream Slave do IP *core* e vice-versa. Seguidamente, as ligações de *clock* e *reset* são realizadas pelo "Connection Automation" do Xilinx Vivado.

6. Usando a opção "Edit IP in Packer" sobre o módulo "ReverseEndiannessCop" edite o código fonte VHDL do IP *core*, substituindo-o integralmente pelo conteúdo dos 3 ficheiros incluídos neste RAR (é fundamental que tenha usado os mesmos nomes indicados acima). O código VHDL exemplo que é gerado aquando da criação do IP *core* é demasiado complexo e irrelevante neste caso, pelo que deve ser completamente substituído pelo código fornecido.

7. Analise cuidadosamente e interprete o código VHDL fornecido de forma a compreender a implementação da operação "*Reverse Endianness*", assim como das interfaces do coprocessador.

Algumas notas sobre o protocolo AXI-Stream:

- Todos os sinais são comutados no flanco ascendente do sinal de relógio;

- O sinal VALID é ativado pelo Master quando possui dados para enviar para o Slave;

- O sinal READY é ativado pelo Slave quando está pronto a receber esses dados;

- O sinal STRB (Strobe) é composto por 4 bits (1 por cada byte da palavra de dados, funcionando como byte enable) - em transferências de 32 bits todos os 4 bits estão ativos;

- O sinal LAST só é relevante na transferência de pacotes de informação compostos por múltiplas palavras, sendo usado para sinalizar a transferência da última palavra;

- Os últimos dois sinais (STRB e LAST) não são usados neste exemplo.

8. Grave todos os ficheiros do IP *core*, realize o seu *repackage* e feche a instância do Xilinx Vivado que usou para o editar.

Questão: Qual o papel desempenhado por cada um dos três ficheiros/módulos VHDL do IP *core*?

9. No projeto principal atualize o IP *core*, sintetize, implemente o projeto e gere a respetiva *bitstream* com a configuração da FPGA (hardware).

10. Exporte o hardware (incluindo a *bitstream*).

11. Abra o Xilinx Vitis e crie uma nova aplicação de software a executar sobre a plataforma construída, do tipo *standalone*, escrita em C e baseada no template "*Hello World*".

12. Substitua o código C predefinido do exemplo "Hello World" pelo fornecido neste ficheiro. Analise cuidadosamente o código dado de forma a compreender completamente o papel de cada função do software.

Questão: Que funções de software são usadas na transferência de dados de/para o coprocessador? Porque razão são invocadas de forma intercalada?

13. Compile e teste na FPGA a aplicação de software a correr sobre a plataforma de hardware construída. A aplicação vai bloquear devido a *stack overflow*. Aumente, no *linker script*, o tamanho da *stack* para o valor correto. Quando tudo estiver a funcionar corretamente deverá obter um resultado semelhante ao apresentado nesta figura. Analise os resultados obtidos.

Questão: Que cuidados foram tidos em conta para que os tempos de processamento apresentados sejam credíveis e comparáveis?

Parte 2

1. Adicione ao projeto anterior um novo coprocessador/acelerador de hardware capaz de realizar a operação "*Population Count*" de uma palavra de 32 bits. Essa determinação deverá ser realizada combinatorialmente de forma a poder ser realizada num único ciclo de relógio, facilitando a sua integração no coprocessador com interfaces AXI-Stream.

2. Realize todas as tarefas de projeto necessárias sobre plataforma de hardware de forma a adicionar este novo coprocessador (criação, instanciação, interligação e edição do código VHDL do IP core), efetuando de seguida a síntese, implementação e geração da *bitstream* do projeto no Xilinx Vivado.

Algumas notas:

- É recomendável que construa, num ficheiro VHDL, um módulo (par entidade-arquitetura) autónomo, que realize a operação "Population Count", que possa ser desenvolvido e simulado (com uma *testbench*) separadamente, e só depois integrado no coprocessador. Para esse efeito é fornecido [este esqueleto em VHDL](#). De notar que o valor por omissão do parâmetro N é irrelevante - no momento da instanciação do módulo deve ser feita a atribuição de um valor concreto ao parâmetro N da seguinte forma:

```
... generic map(N => C_S_AXIS_TDATA_WIDTH) ...
```

- O MicroBlaze pode ter até 16 pares de Stream Links Master-Slave. Neste caso vai ter 2 (cada um dos pares liga a um dos coprocessadores) de forma a que a mesma plataforma de hardware inclua os dois aceleradores de hardware.

3. Verifique a frequência máxima de operação do sistema completo.

4. Realize todos os passos necessários para atualizar no Vitis a plataforma de hardware.

5. Adicione ao módulo em C da aplicação de software duas funções:

- **PopulationCountSw** - que realize a operação "*Population Count*" completamente em software, sobre um *array* de uma determinada dimensão (contando de forma cumulativa todos os bits a "1" de todas as palavras do *array*);

- **PopulationCountHw** - que realize a operação "*Population Count*" assistida por hardware, sobre um *array* de uma determinada dimensão (contando de forma cumulativa todos os bits a "1" de todas as palavras do *array*; a iteração de todos os elementos do *array* deve ser realizada por software; a contagem dos bits a "1" dentro de cada palavra deve ser realizada em hardware pelo coprocessador).

6. Utilize as funções **PopulationCountSw** e **PopulationCountHw** na função **main** da aplicação, seguindo a mesma abordagem usada no código fornecido para a Parte 1. Adicione o código necessário para confrontar os resultados obtidos na implementação unicamente em software e na assistida por hardware, assim como para obter os respetivos tempos de execução.

Nota: o *array* usado para a operação "Reverse Endianess" também pode ser usado para a "Population Count".

7. Compile e teste na FPGA a aplicação de software a correr sobre a plataforma de hardware que inclui os dois coprocessadores. Analise os resultados obtidos. Compare os tempos de processamento das duas implementações da operação "*Population Count*".

8. Volte a implementar a plataforma de hardware aumentando para o limite máximo a frequência do sinal de relógio. Teste adequadamente o sistema e analise os resultados obtidos.

9. As aplicações de software deste trabalho prático foram todas compiladas em modo "*Debug*". Troque a configuração ativa para "*Release*" e volte a compilá-las e a executá-las. Analise os resultados e compare com os obtidos anteriormente.