Aceleração da identificação de números primos com um co-processador hardware

Computação Reconfigurável

Mestrado Integrado em Engenharia de Computadores e Telemática

universidade de aveiro
theoria poiesis praxis

Pedro Almeida - 89205
Renato Valente - 89077

Índice

1 - Introdução

A proposta inicial foi ligeiramente alterada depois de uma primeira tentativa mal sucedida. Assim, a proposta final é a seguinte:

Descrição:

Sistema de hardware assistido por DMA para calcular o número de números primos num array de comprimento configurável de entradas de 32 bits. Os números primos encontrados são apresentados ao utilizador.

Descrição da arquitetura:

O sistema irá conter um módulo hardware personalizado para verificar se uma entrada de 32 bits é um número primo. O módulo usa um controlador DMA e um acumulador para processar até 2^12 (4096) valores de 32 bits guardados na memória RAM externa.

Será comparado o tempo de execução das implementações de software e hardware.

Objetivo:

Reduzir o tempo de processamento.

Teste e Interação com utilizador:

Software verifica os resultados de hardware. Os dados serão auto-gerados por software. Interação com o utilizador será através de UARTLite e serial terminal.

2 - Descrição da funcionalidade do sistema

O sistema tem o objetivo de reduzir o tempo de processamento para verificar se um número é primo. Para isso, foi construído um módulo hardware especializado que com base numa entrada de 32 bits, verifica se o número primo. Caso seja um número primo, a saída tem o mesmo valor que a entrada (retorna o número primo), caso contrário é retornado o número 0.
O módulo usa um controlador DMA e um acumulador para processar até 2^12 (4096) valores de 32 bits guardados na memória RAM externa.

Durante a implementação do módulo hardware especializado, foi alterado o número de bits da entrada que são processados. Esta mudança foi feita depois de explicitar à professora que o tempo que levava a concluir a geração do *bitstream* era elevado (≈ 2 horas) . Assim, da entrada de 32 bits, apenas os 8 bits menos significativos são processados. Esta mudança, resolve o problema apresentado pois permite reduzir o número de iterações no algoritmo, uma vez que, em VHDL, é necessário saber o número de iterações de um ciclo *for* antes da síntese.

No lado do software, são gerados dados de entrada aleatórios que de seguida são processados tanto por software como por hardware e os respetivos resultados são verificados. É também apresentado o tempo de cada operação e o ganho obtido.

3 - Algoritmo implementado

Cálculo dos números primos. Um número é primo quando é divisível por 1 e por si próprio. Os números primos são números Naturais (N) maiores que 1.

O algoritmo genérico para verificar se um número é primo é o seguinte:

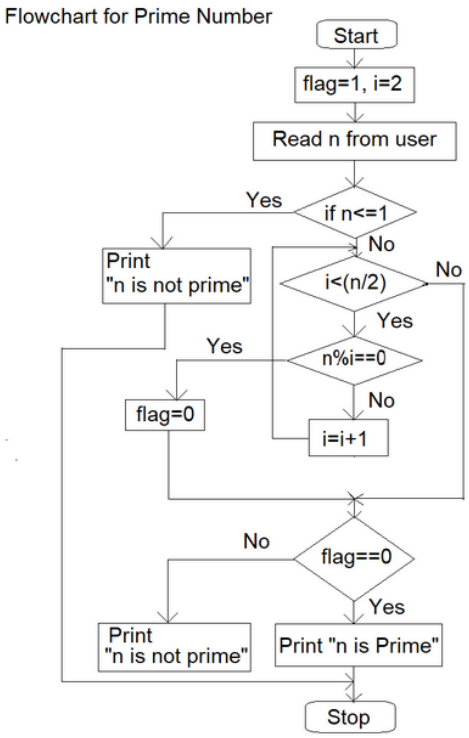

Figura 1 - Diagrama do algoritmo de números primos

Posteriormente, foi adicionada lógica para retornar o valor correto, isto é, retornar o número original caso seja primo, ou retornar zero, caso contrário.

Foi feito uma testbench para o algoritmo implementado em VHDL.

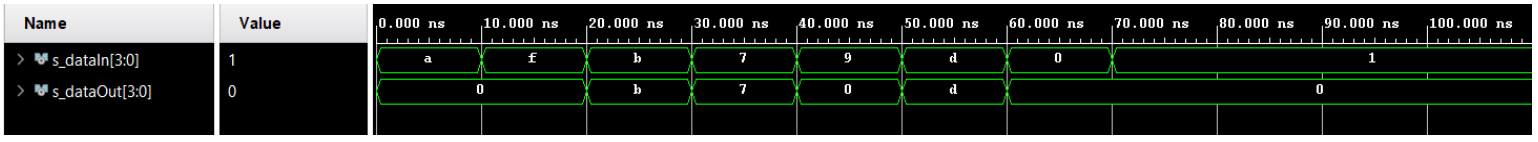

Figura 2 - Output da TestBench

Na implementação em software também foi gerado um ficheiro para testar o algoritmo implementado antes de o implementar no Vitis.

Os ficheiros de teste estão disponíveis no diretório do projeto, a testbench na pasta *primes* e o ficheiro de teste em *C* está no diretório raiz.

4 - Diagrama de blocos do coprocessador

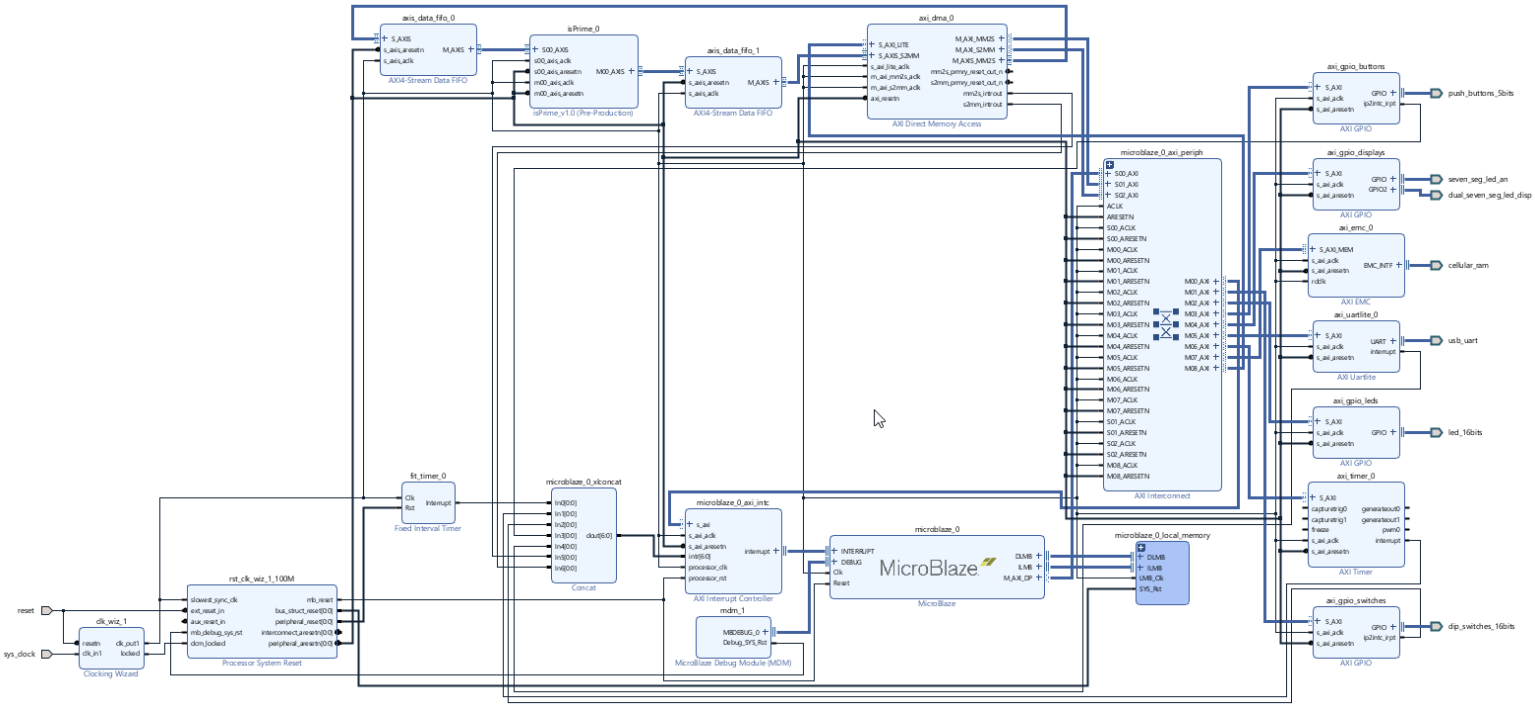

Figura 3 - Block Design

Como base para o projeto desenvolvido e de forma a aproveitar o trabalho desenvolvido nas aulas práticas, foi importado o *block design* com os módulos DMA e EMC. Posteriormente, foi adicionado o *custom IP* para fazer o processamento dos dados. Assim, o fluxo dos dados é o seguinte: o controlador DMA vai buscar os dados à memória RAM externa (EMC), os dados passam pelo *interconnect* e são recebidos no DMA, onde os dados são encaminhados para o módulo hardware especializado (custom IP) para serem processados, estes voltam ao DMA, que novamente, os encaminha de novo para a memória externa através do *interconnect*.

Entre o módulo DMA e o *custom IP*, são adicionadas duas *FIFO* para que os dados sejam enviados sem conflitos. Estas *FIFO* foram configuradas com uma *depth* de 4096.

A troca de dados é através de interfaces AXI Stream.

5 - Resultados

5.1 - Cálculo da frequência ideal

$$Fmax = \frac{1}{(t - WNS)}$$

$$f = \frac{1}{T}$$

Inicialmente estávamos a usar 70 MHz para a Fmax e tínhamos um WNS de -0.32ns.
Calculamos o T para 70 MHZ, e obtivemos 14.28ns.
Somamos os tempos, 14.28 + 0.32 = 15.0ns.
E por fim calculamos a frequência para 15.0ns, que deu 66 MHz.
Decidimos usar o valor de 62 MHz como frequência máxima para prevenir quaisquer erros.

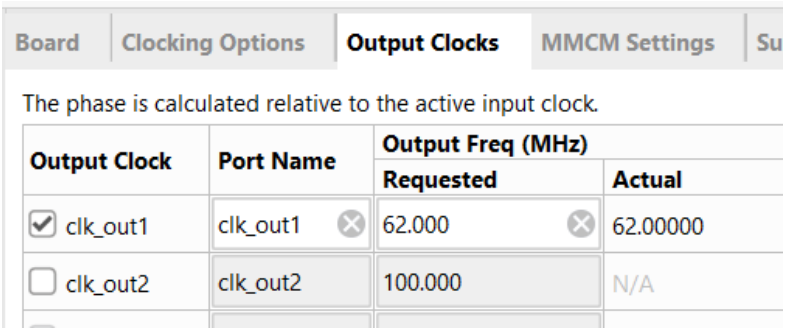

Figura 4 - Frequência do Clock

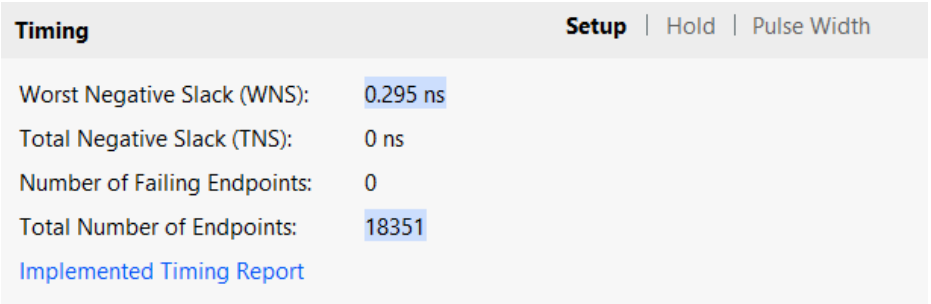

Figura 5 - Worst Negative Slack

5.2 - Resultados e Conclusões

Os resultados obtidos tanto por software como por hardware são verificados por software.
O cálculo do número de números primos identificados é também feito por software.
A versão de hardware tem um ganho de

$$\frac{376066}{1372} \approx 274$$

```
------Primes Program------

Filling memory with pseudo-random data. Seed is 50.

Primes Program


Memory initialization time: 105706 microseconds

216   209   10   184   206   103   27   17   23   160   218   137   85   193   64   15   85   235   247   236

-----------------------------------------------------------------

Software only

Execution time: 376066 microseconds

 Number of primes found Sw:  837

103   17   23   137   193   43   191   139   71   197   5   229   151   103   233   199   47   103   29   227

Checking Sw result: OK

-----------------------------------------------------------------

Hardware assisted

Configuring DMA...
DMA running...

Execution time: 1372 microseconds

 Number of primes found Hw:  836

103   17   23   137   193   43   191   139   71   197   5   229   151   103   233   199   47   103   29   227


Checking Hw result: OK
```

Figura 6 - Outputs e Resultados

6 - Auto-avaliação

De forma a não sobrecarregar nenhum dos alunos, a distribuição do trabalho foi uniforme e o mais justa possível. Assim sendo, pode-se dizer que a percentagem de contribuição de cada aluno foi de cinquenta por cento.
Visto que os requisitos propostos do projeto foram alcançados, a auto-avaliação do grupo é 16.