

INSTITUTO SUPERIOR TÉCNICO

MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE
COMPUTADORES

CO-PROJECTO HW/SW

Data Input/Output - Parte 1

Trabalho realizado por:
Ion Ciobotari
Mihir Odhavji

Número
84075
84151

Turno: Quarta-Feira

05 de Maio de 2019

Conteúdo

1	Introdução	2
2	Sw Only	2
3	Sw-Hw	2
3.1	Testbench	2
3.2	ILA	3
4	Otimização	4
5	Conclusão	4

1 Introdução

O âmbito principal deste projeto era criar um programa que realiza a convolução de matrizes. Este programa será seguidamente executado em paralelo com um parte em Hw (fornecida pelo professor). Numa última fase é esperado que os alunos façam a otimização do IP fornecido pelo docente.

2 Sw Only

Numa primeira versão deste projeto desenvolveu-se um programa que faz a leitura do ficheiro de imagens. Para facilitar o cálculo da convulsão das imagens, primeiro determinou-se uma matriz "transformada" cujas linhas contêm os elementos da matriz que compõe a imagem que têm de ser multiplicadas pela matriz Kernel. Desta forma conseguimos facilitar os acessos à memória onde está a ser guardada as matrizes que são usadas. De seguida, basta efetuar a multiplicação da matriz "transformada" com a matriz Kernel, obtendo uma matriz final 24 por 24.

3 Sw-Hw

De forma a tornar o programa mais rápido podemos optar por usas componentes de hardware que efetuam a multiplicação de matrizes. No nosso caso, o professor responsável pela cadeira disponibilizou um IP que efetua a multiplicação de matrizes. Assim, basta enviarmos as matrizes que queremos multiplicar ao IP e obtemos o resultado pretendido, igual ao resultado obtido no ponto anterior.

3.1 Testbench

Para verificar o funcionamento do IP disponibilizado, realizou-se a simulação para um caso simples onde é enviado dois vetores, um que representa o Kernel e o outro que representa a matriz da imagem, de tamanhos 4 e 16, respetivamente.

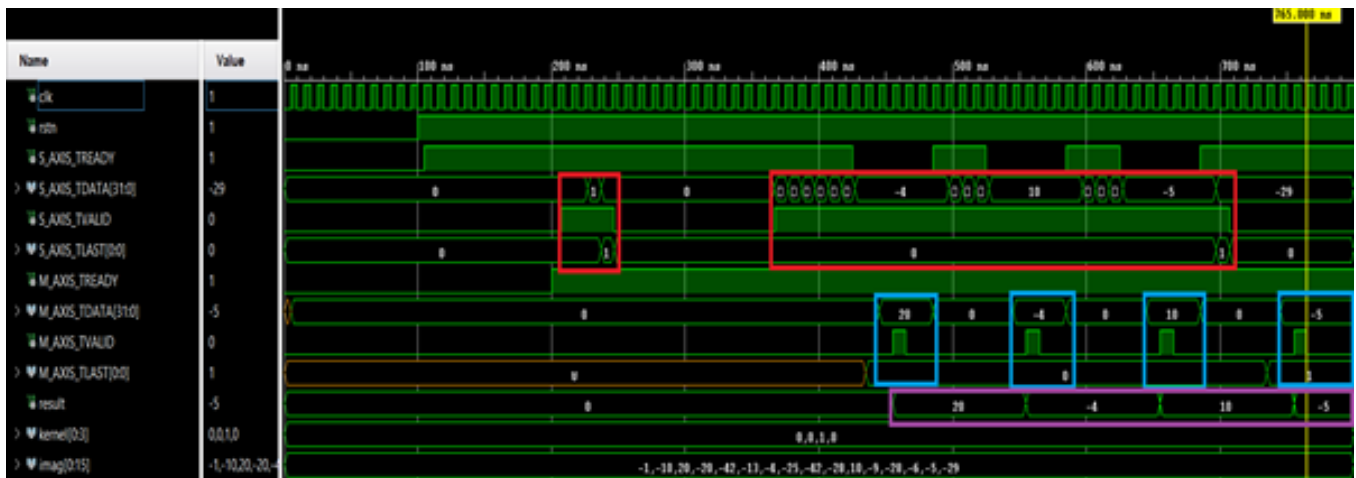


Figura 1: Resultado da simulação

Da figura1, podemos observar que apenas quando S'AXIS'TREADY possui valor lógico positivo é permitido o envio de dados para o IP. Durante a leitura de dados à entrada, S'AXIS'TVALID indica se a informação lida se encontra correta. Esta leitura de dados acaba quando o sinal S'AXIS'TLAST muda de valor lógico negativo para positivo. São esperados o envio de dois conjuntos de dados de tamanhos diferentes. No instante em que é efetuada a segunda leitura de dados, observamos que não é realizada toda a leitura de uma vez. Isto deve-se à forma como foi programado o IP. Neste caso, são lidos uma quantidade de dados igual ao primeiro envio. Após a esta leitura é realizada a multiplicação dos dados adquiridos, cujo resultado

é enviado de imediato. O envio do resultado é efetuado no instante quando M'AXIS'TVALID possui valor lógico positivo. Este processo de leitura e calculo continua até S'AXIS'TLAST possuir valor lógico positivo, que indica o fim do envio de dados. Da mesma forma que é finalizada a leitura de dados, M'AXIS'TLAST muda para valor lógico positivo para finalizar o envio do resultado da multiplicação. Podemos concluir que o funcionamento do IP disponibilizado é o esperado.

3.2 ILA

Acabado de verificar o funcionamento do IP em simulação, podemos recorrer ao uso de um debugger para verificar o funcionamento do IP durante a execução do programa previamente desenvolvido.

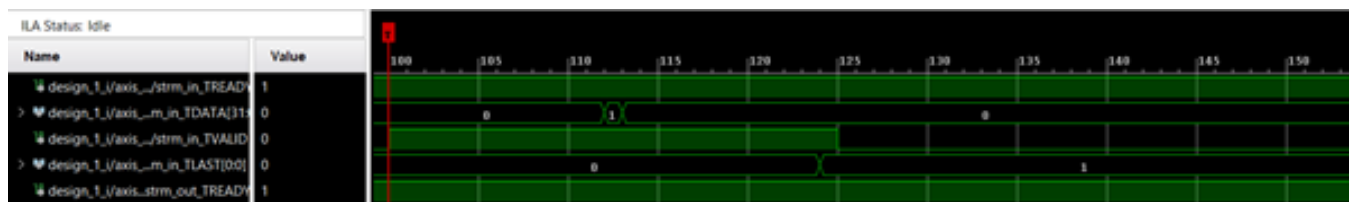


Figura 2: Primeira leitura de dados

No princípio do programa é enviado para o IP um vetor correspondente à matriz Kernel usada para o cálculo do resultado. A figura 2 expõe a leitura desta matriz. Podemos observar que o intervalo de tempo no qual strm'in'TVALID se encontra com valor lógico positivo corresponde à dimensão da matriz Kernel, que no nosso caso é 25 elementos. Tal como no ponto anterior a leitura acaba quando strm'in'TLAST possui valor lógico positivo.

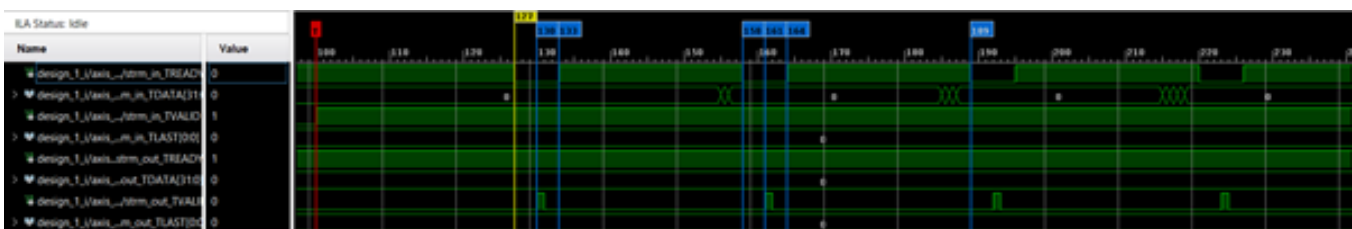


Figura 3: Início da segunda leitura de dados

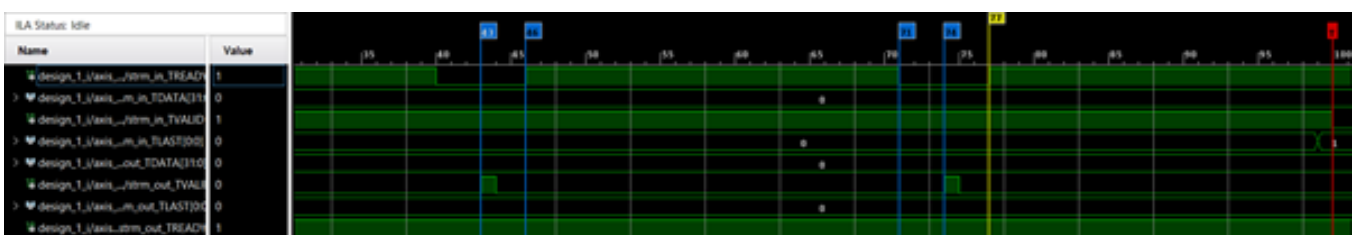


Figura 4: Fim da segunda leitura de dados

As figuras 3 e 4 expõem a leitura da segunda matriz, que corresponde ao envio da matriz da imagem “transformada”. Observamos que é efetuada a leitura de 25 elementos desta matriz de cada vez. No fim de cada uma destas leituras é efetuada multiplicação com os dados da primeira leitura e é enviado o resultado obtido. Este comportamento é idêntico ao comportamento observado no ponto anterior. Tal como no ponto anterior, a leitura de dados e envio do resultado é terminada quando strm'in'TLAST possui valor lógico positivo.

4 Otimização

Em geral, a combinação de software e hardware proporciona uma execução mais rápida de programas do que o uso único de software. Contudo, pela observação da tabela 1 observamos que isso não se aplica neste caso. Uma das possíveis do tempo de execução alto da combinação de software e hardware deve-se ao grande número de envio e leitura de dados que ocorrem. Uma solução para diminuir o número de transmissões será diminuir os bytes de leitura e envio do IP de inteiro (4 bytes) para caracteres (1 byte). Esta solução é viável uma vez que ambas as matrizes do Kernel e da imagem são matrizes de caracteres. Após a realização desta otimização, verificou-se que o tempo de execução é inferior a ambos os casos anterior, confirmando assim que o uso da combinação de software e hardware é benéfica.

	Apenas SW	Sw-Hw	Sw e Hw otimizado
Tempo de execução [μ s]	1533.86	3537.17	1242.85

Tabela 1: *Tempo de execução do programa*

5 Conclusão

Comparando os três resultados obtidos foram algum diferentes do esperado. Neste caso não se verificou o facto de a combinação de hardware e software proporcionar um programa mais rápido que só o uso de software. Todavia após a realização de otimizações verifica-se que a combinação software e hardware é mais rápida que qualquer umas das anteriores.