Relatório do Projeto 4: Modelagem de um multiplicador matriz vetor em nível funcional

Aluno: Jessé Barreto de Barros
Matricula: 17/0067033
Sistemas Mecatrônicos - PPMEC-UnB
Disciplina: Visão Computacional
Turma: A
Data: 24/04/2017

I. Objetivos

Descrever o modelo funcional de um sistema de multiplicação de matrizes 3x3 por vetores 3x1 através do uso de modelagem funcional.

O comportamento dos módulos são definidos por *threads* e os módulos são conectados por filas bloqueantes e a sincronização entre os módulos segue o comportamento que os dados fazem no interior do sistema.

II. Introdução

A. O Problema

Na modelagem de sistemas utilizando o paradigma *dataflow* não há a necessidade de sincronização utilizando um pulso de relógio, pois, o sistema funciona através do consumo de dados disponíveis e o fluxo de processamento é efetuado conforme os dados trafegam pelo sistema.

Na Figura 1, e possível visualizar a estrutura do sistema para efetuar a multiplicação para um sistema com uma matriz de dimensões 3x3 e o vetor de dimensão 3x1. Nessa figura os elementos do vetor são conectados aos módulos multiplicadores como sinais, representados por *sc_signal* enquanto os outros canais são representados por filas bloqueantes do tipo *sc_fifo*.

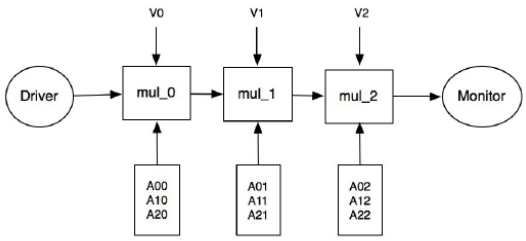

Figura 1. Ilustração do sistema.

III. Metodologia

A. Modelagem do Sistema

O sistema funciona a partir do seguinte fluxo de eventos:

- O componente *Driver* inicializa o sistema escrevendo o valor zero na fila e esperando um tempo $Deltat$.
- O primeiro componente multiplicador lê o valor da fila do *driver* e outro da fila de elementos da matriz e escreve na próxima fila o resultado. O resultado é basicamente a soma do resultado anterior, zero no caso do primeiro componente multiplicador, com o produto do elemento do vetor com o elemento da matriz. Como a fila do *driver* está vazia a *thread* é bloqueada.
- O segundo componente multiplicador lê o valor da fila do resultado do primeiro e efetua um processamento similar ao primeiro.
- O terceiro componente multiplicador lê o valor da fila do resultado do segundo multiplicador e calcula o valor final do elemento do vetor resultante e escreve esse resultado na fila do monitor.
- O monitor que estava bloqueado pela falta de dados lê e imprime o resultado e logo em seguida é bloqueado novamente. Dessa forma o fluxo é repetido até os componentes multiplicadores esvaziarem as filas de elementos das matrizes.

O sistema foi projetado para ser escalável, isto é, basta passar os valores da matriz e do vetor para o sistema para que o mesmo aloque dinamicamente o número de componentes multiplicadores necessários para efetuar a operação com matrizes de qualquer tamanho NxN.

Caso as dimensões sejam incompatíveis o sistema é interrompido com um erro alertando o usuário sobre o problema dimensional.

Com o intuito de trabalhar com diferentes tipos de dados o sistema foi implementado utilizando *template* com isso basta alterar o tipo de dado da matriz e do vetor para que o sistema trabalhe com tipos de dados que possuem as mesmas operações numéricas de um tipo inteiro, por exemplo, multiplicação e adição.

IV. Resultados

Os resultados mostram que o sistema consegue calcular corretamente diferentes matrizes 3x3 e vetores 3x1 como, por exemplo, a Figura 2 e 3

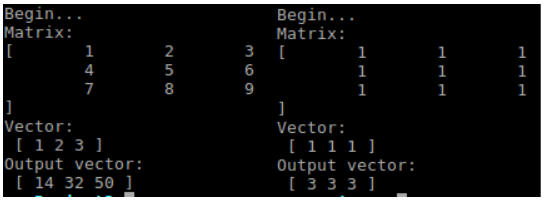

Figura 2. Resultados no terminal utilizando dimensão 3 e o tipo de dado é inteiro.

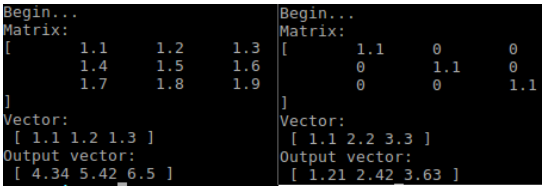

Figura 3. Resultados no terminal utilizando dimensão 3 e o tipo de dado *float*.

A diferença entre um software convencional e a descrição de hardware implementada é o paradigma *dataflow* utilizado para descrever a funcionalodade do *hardware*. Esse paradigma não é suportado normalmente pelo C++.

A diferença entre esse sistema e um *hardware* real é a abstração. A implementação de um sistema RTL necessita uma menor abstração.

O sistema já está generalizado para matrizes NxN um exemplo de resultado para matrizes NxN está presente na Figura 4

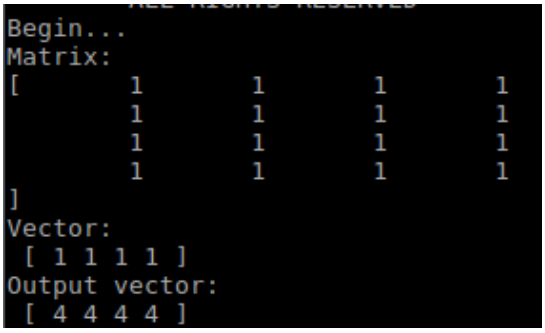

Figura 4. Resultados no terminal utilizando dimensão 4 e o tipo de dado *integer*.