AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI, INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ

SYSTEMY DEDYKOWANE W UKŁADACH PROGRAMOWALNYCH

Algorytm *Fast inverse square root*

Autorzy: *Łukasz Orzeł; Jakub Świebocki*

Kierunek studiów: *Mikroelektronika w technice i medycynie*

Kraków, 2023

Spis treści

Wstęp

W niniejszym raporcie przedstawiamy wyniki projektu, którego celem było zaimplementowanie algorytmu Fast Inverse Square Root na układzie FPGA (Field-Programmable Gate Array). Jako, że algorytm Fast Inverse Square Root, ze względu na szybkość działania, jest szeroko stosowany w dziedzinie grafiki komputerowej oraz obliczeń naukowych, prezentujemy również przykładowe aplikacje, które korzystają z obliczeń naszej implementacji.

Pierwszym zastosowaniem algorytmu Fast Inverse Square Root było zaimplementowanie w grze komputerowej Quake III Arena. Algorytm ten znalazł zastosowanie w przyspieszaniu obliczeń związanych z oświetleniem i renderingiem graficznym. Stąd też nazwa tego algorytmu to Quake Fast Inverse Square Root algorithm. Działa on na podstawie magii bitowej (bitwise magic) oraz manipulacji liczbami zmiennoprzecinkowymi w reprezentacji binarnej. Podstawowy schemat działania algorytmu Fast Inverse Square Root można przedstawić w kilku krokach:

- Konwersja liczby zmiennoprzecinkowej na jej binarną reprezentację.
- Zastosowanie pewnych operacji bitowych na tej reprezentacji, które mają na celu przybliżone obliczenie wartości odwrotności pierwiastka kwadratowego.
- Ostateczne dostosowanie wyniku, aby uzyskać bardziej precyzyjną wartość odwrotności pierwiastka kwadratowego.

Algorytm wykorzystuje technikę zwana metodom Newtona-Raphsona do iteracyjnego obliczania przybliżonej wartości pierwiastka kwadratowego. Jednak kluczową innowacją algorytmu Fast Inverse Square Root jest wykorzystanie operacji bitowych, które pozwalają na przyspieszenie tego procesu.

Ważnym aspektem algorytmu jest również tzw. "magic number"(liczba magiczna), która jest używana w manipulacji bitowej i jest zależna od wartości liczby, dla której obliczamy odwrotność pierwiastka kwadratowego. To właśnie ta liczba magiczna pozwala na przyspieszenie obliczeń i uzyskanie przybliżonego wyniku.

W kolejnych sekcjach raportu szczegółowo omówimy proces implementacji algorytmu Fast Inverse Square Root na układzie FPGA oraz przedstawimy wyniki naszych badań i analizę efektywności implementacji.

1. Model behawioralny algorytmu

1.1. Implementacja modelu w języku C

```
#include <stdio.h>
#include "platform.h"
#include "xil_printf.h"
#include "str_acc.h"
#include "xuartps.h"

#define NBR_OF_DATA 11 //All possible values for the range

u32 data_in[NBR_OF_DATA]; //Accelerator input buffer
u32 data_out[NBR_OF_DATA]; //Accelerator output buffer
float data_in_c[NBR_OF_DATA];
float data_out_c[NBR_OF_DATA];
float init_d = 0.0001;
char line[33];
```

1.2. Testbench modelu behawioralnego w języku C

2. Model syntezowalny potokowy algorytmu

2.1. Układ mnożący

2.2. Układ odejmujący

2.3. Implementacja kodu

2.4. Schemat blokowy

2.5. Testbench

3. AXI

3.1. Wrapper

4. Uruchomienie na sprzęcie

4.1. Test układu przez konsolę

5. Zastosowanie

5.1. Struktura blokowa

5.2. Komunikacja między PC, a Zedboard

5.3. Dostosowanie kodu na płytce Zedboard

5.4. Implementacja kodu z interakcją użytkownika

5.4.1. FigureMoveIn3D

5.4.2. VectorMoving

Spis tabel

Spis rysunków