

HARDWARE ACCELERATION OF GRAPHICS ALGORITHMS USING LOW-COST FPGA

Čapka Ladislav

Master Degree Programme (5), FIT BUT

E-mail: xcapka01@stud.fit.vutbr.cz

Supervised by: Vašíček Zdeněk

E-mail: vasicek@fit.vutbr.cz

ABSTRACT

This article deals with a method and a implementation of graphic acceleration of rasterization based on field programmable gate arrays (FPGAs). FPGA can be used as alternative way to acceleration specific method. Their computation efficiency is higher than efficiency of DSP processors. These FPGAs can be programmed by high-level programming languages, e.g. VHDL. Main condition of solved implementation was realization of rasterization block chain using low-cost hardware.

1 ÚVOD

Cílem tohoto článku je seznámit čtenáře s architekturou grafického hardware, který realizuje základní rasterizační řetězec. Hlavním prvkem návrhu architektury rasterizačního řetězce je část zajišťující vlastní rasterizaci. Tento blok vyžaduje implementaci vhodné rasterizační metody. Pro popis grafických výstupů je možno využít několik metod (viz [2]), které vyžadují v FPGA více či méně dostupného prostoru. Cílová HW platforma by měla být řazena do kategorie levnějších, a proto by měla metoda rasterizace využívat co nejmenší množství zdrojů. Proto je zvoleným tvarem je trojúhelník. Metodu rasterizace trojúhelníku lze provádět v oboru celých čísel. Trojúhelník je definován třemi řídicími body, tudíž nevyžaduje tedy mnoho zdrojů. Z těchto důvodů je trojúhelník pro zpracování v malých FPGA vhodný.

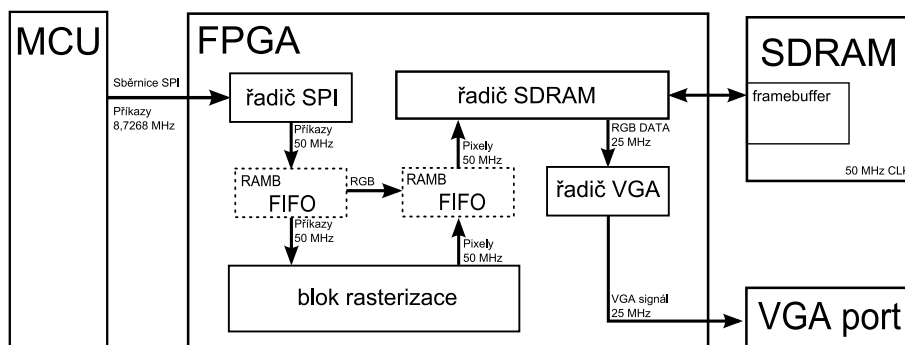
2 CÍLOVÁ HW PLATFORMA

Jako cílová platforma byl zvolen vývojový kit FITkit obsahující FPGA [1], který je určen pro potřeby výuky studentům fakulty informačních technologií Vysokého učení technického v Brně. FITkit je osazen čipem FPGA Spartan 3, který byl zvolen jako kompromis mezi cenou a dosažitelným výkonem. Uvedené programovatelné hradlové pole obsahuje čtyři blokové paměti (RAMB) a čtyři rychlé vestavěné násobičky s 18-bitovým rozsahem vstupních operandů, které mohou pracovat na vysoké frekvenci a ušetří cenné zdroje. Nevýhodou použitého FPGA je sdílení některých signálů RAMB s násobičkami. Využití paměti RAMB tedy omezí počet dostupných násobiček a naopak. Z těchto důvodů je nutné s těmito zdroji šetřit. Kromě FPGA kit

obsahuje i výkonný mikrokontrolér a řadu periférií. Mezi tyto periférie patří paměť SDRAM s kapacitou 8 MB a grafický port VGA, s jejichž pomocí je možno demonstrovat funkčnost navržené architektury grafického hardware.

3 NAVRŽENÁ ARCHITEKTURA

Pro rasterizační řetězec zahrnující blok rasterizace trojúhelníku byla navržena vhodná architektura, kterou lze na platformě FITkit provozovat. Tato architektura je zobrazena na obrázku 1. Vstupní data rasterizačního řetězce v FPGA jsou získávána z mikrokontroleru MCU pomocí sběrnice SPI. Grafický řetězec je k této sběrnici připojen pomocí řadiče SPI, který zpracovává komunikaci na sběrnici. Přijaté požadavky na rasterizaci trojúhelníku jsou ukládány do paměti RAMB, která slouží jako fronta požadavků. Každý požadavek je poté postupně zpracován hlavním rasterizačním blokem, jenž generuje polohy výsledných bodů v obraze, které patří rasterizovanému trojúhelníku. Tyto polohy je nutné opět vystavovat do paměti RAMB, která slouží jako fronta výsledných poloh a to z důvodu zvýšení efektivity v čase, kdy není přístup k paměti SDRAM pro tento blok možný.



Obrázek 1: Blokové schéma kompletní architektury

Přístup k paměti pro čtení a zápis dat je v pevném intervalu na určitou dobu omezen, protože je nutné provádět pravidelné obnovovací cykly. Omezením je také dostupnost pouze jednoho portu paměti SDRAM, který je nutné v průběhu využívání sdílet.

Zobrazovací blok, který generuje výstupní signál na port VGA, je řadič grafického portu VGA. Tento řadič čte zobrazované body z paměti SDRAM. Tato operace vyžaduje přesné časování a rychlou dostupnost dat v konkrétních časech. Jelikož paměť vyžaduje pravidelný obnovovací cyklus, který musí být zajištěn, je pro tuto funkci vhodné najít časový interval. Blok rasterizace má v přístupu k paměti z uvedených bloků nejnižší prioritu. Výpočet rasterizačního bloku musí být v případě zaplnění výstupní fronty pozastaven.

4 BLOK RASTERIZACE

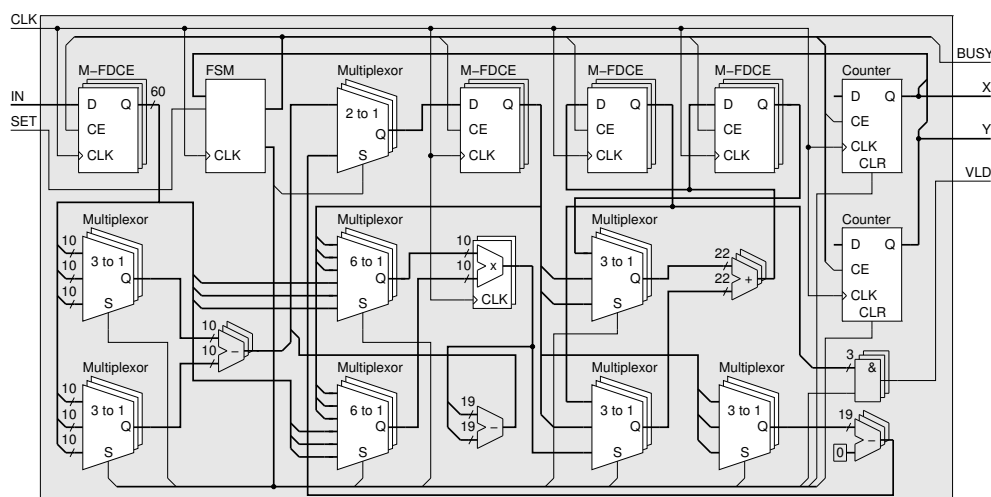
Blok rasterizace zajišťuje vlastní výpočet a generuje polohy rasterizovaných bodů na základě poloh řídicích vrcholů trojúhelníku načtených z fronty příkazů. Realizace výpočetního bloku je založena na metodě hranové detekce, pro kterou jsou důležité rovnice (1).

Vnitřní blokové schéma navrženého bloku rasterizace trojúhelníku je ukázáno na obrázku 2. Rasterizační blok využívá dvě dostupné násobičky, které jsou v průběhu výpočtu sdílené. Pro

řzení jednotky je v architektuře navržen konečný automat. První kroky algoritmu slouží k výpočtu a korekci koeficientů A1-3, B1-3 a C1-3, které jsou následně uloženy do registrů. Vlastní testování je založeno na akumulaci hodnot, ze kterých je možno přímo definovat, zda-li bod patří trojúhelníku. Na začátku každého řádku jsou 3 akumulátory naplněny určitou hodnotou, která je charakteristická pro testovaný řádek. V každém bodě je k akumulátorům přičtena určitá hodnota. Pokud jsou ve všech akumulátorech kladná čísla, patří testovaný bod trojúhelníku a jeho pozice je tedy vystavena na sběrnice X a Y a potvrzena signálem VLD.

$$A \cdot x + B \cdot y + C = d \quad (1)$$

$$A = y_0 - y_1 \quad B = x_1 - x_0 \quad C = x_0 \cdot y_1 - x_1 \cdot y_0$$



Obrázek 2: Schéma rasterizační jednotky

Doba rasterizace je dána velikostí testovaného prostoru a frekvenci, na které je rasterizační blok schopen jednotlivé body prostoru vyhodnocovat. Doba ohodnocení jednoho bodu je vždy jeden takt hodinového signálu. Architektura otestovaná na platformě FITkit je schopna pracovat bez problémů na frekvenci 50 MHz. Při této frekvenci může dosahovat rychlost až 190 trojúhelníků za sekundu. Faktorem znemožňujícím dosažení této hranice je sdílená paměť SDRAM.

5 ZÁVĚR

Navržená architektura demonstruje, jak lze rozvrhnout a realizovat rasterizační jádro v FPGA, která patří k levnějším na trhu. Efektivita řešení je limitována sdílením přístupu k paměti SDRAM mezi výpočtem a generováním výstupního obrazového signálu. Vhodnější by bylo využít například paměť typu SRAM s více dostupnými porty. Dalším možným rozšířením by mohla být realizace bloku interpolace barev, které by zvýšilo možnosti využití této architektury.

LITERATURA

- [1] *Výuková platforma FITkit*, <http://merlin.fit.vutbr.cz/FITkit/> (citováno 2007)
- [2] *Foley, J., D., van Dam, A., Feiner, S., K., Hughes, J. F.: Computer Graphics: Principles & Practice - 2nd edition.* Addison-Wesley, 1996