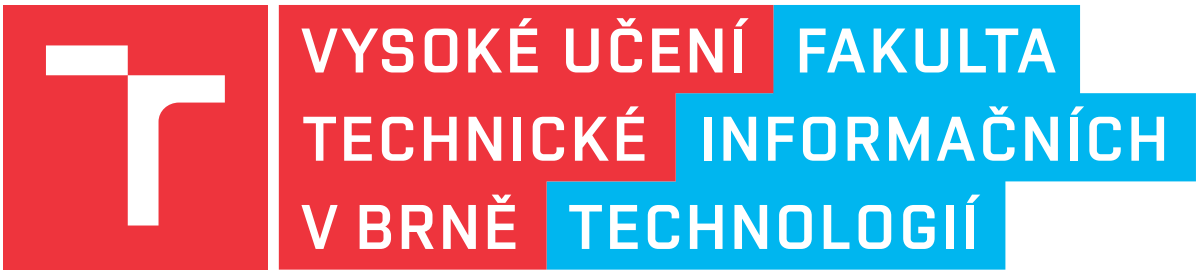


Grafické a multimediální procesory (GMU)

Příprava na zkoušku

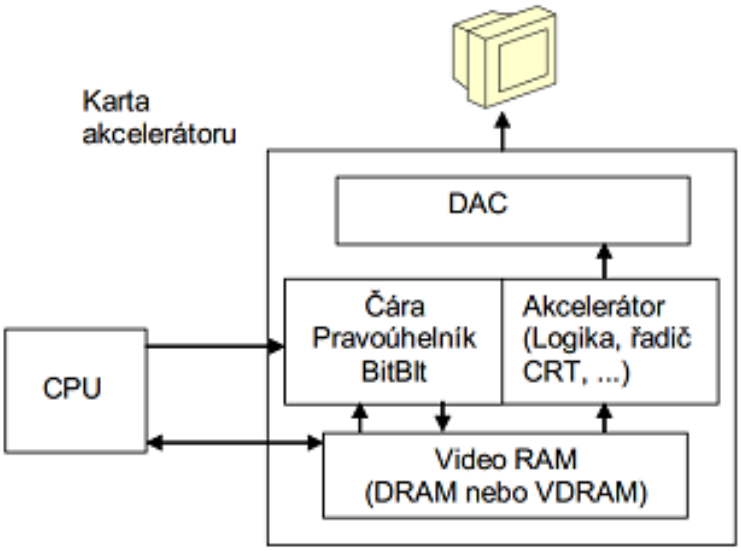
git

Obsah

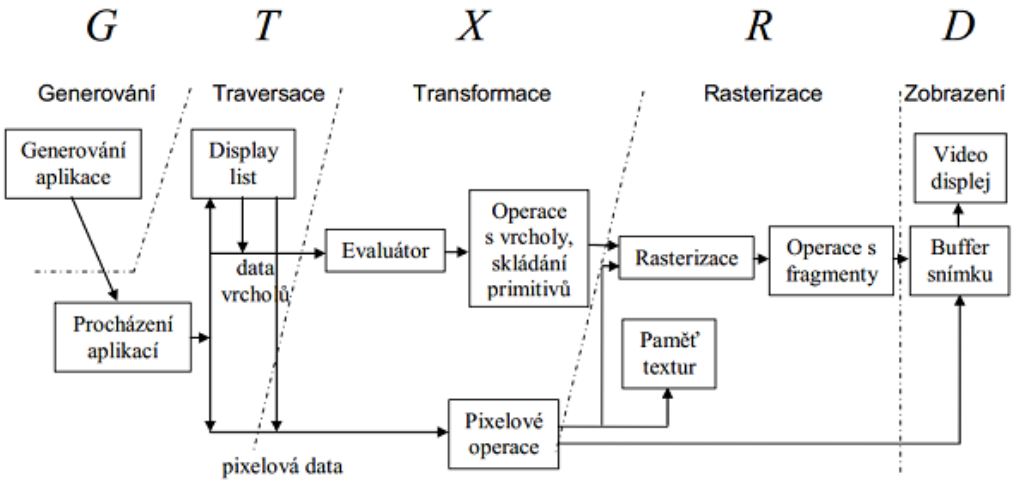
1 Popište základní principy návrhu energeticky úsporného procesoru CPU/GPU, jak se vyhodnocuje energetická úspornost.

TODO

2 Popište vývoj 2D/3D grafického řetězce s různým stupněm zřetězení.



Obrázek 1: Základní 2D zpracování (IBM 8514)



Obrázek 2: Akeleyho pipeline

generovanie → traverzacia → transformacia → rasterizadia → zobrazenie (display) Implementovana napr. u i740, Savage3D

3 Popište 1- až n-rozměrné multiprocesorové propojovací struktury (Origin/Onyx) a typy propojovacích sběrnic Silicon Graphics.

Origin

Propojovací struktura architektury koncepce Origin2000 umožnuje vytváret až 5D krychle. Díky její pružnosti a koncepcní pokrokovosti byla prevzata i do navazujících architektur SGI, jako napríklad Onyx2, a další.

Architektura Origin2000 je tvorena množstvím výpoctových uzlu, propojených navzájem propojovací sítí CrayLink. Každý výpoctový uzel obsahuje jeden nebo dva procesory s pametí cache, sdílenou pamet, adresár pro rízení koherence cache s hlavní pametí a dve ruzné jednotky interface – jeden, XIO, který pripojuje I/O jednotky, a druhý, CrayLink, propojující uzly pres jednotku Router.

Onyx2

Každý procesor je spolu se svou pamětí cache a částí operační paměti počítače (o kapacitě od 128 MB do 8 GB) součástí jednoho procesorového uzlu, který obsahuje čtyři procesory. Obousměrný přenos dat mezi procesory v jednom uzlu dosahuje hodnoty 1,6 GBs-1, resp. 800 MBs-1 v každém směru zvlášť. Jednotlivé uzly jsou mezi sebou propojeny pomocí propojovací sítě, jejíž topologie sice vychází z ideální hyperkrychle, ale z důvodu optimalizace datových přenosů jsou mezi uzly vytvořeny i další přídavné datové spoje (již z principu musí jít o diagonály), zejména mezi procesory a zobrazovacími subsystémy.

4 Popište principy výstavby, činnosti a použití GPGPU.

Jde o techniku využití GPU k obecným (negrafickým) výpočtům, které běžně probíhají na CPU. Zatímco dříve bylo zpracování dat na GPU pevně dané (static pipeline), dnes už je převážná část programovatelná, a tudíž nabízí možnost obecných výpočtů. K tomu slouží např. rozhraní OpenCL, CUDA či ATI Stream.

Z povahy grafických čipů vychází také způsob práce s daty, který je silně orientován na paralelní proudové zpracování dat. Data jsou často rozdělena do 2D mřížky, kde nad každou buňkou pracuje jedna výpočetní jednotka. Tyto jednotky sdílí kód programu, nazývaný také kernel. Možnost synchronizace či sdílení dat mezi jednotkami v průběhu výpočtu je omezená a značně snižuje výsledný výkon. (Alespoň v OpenCL možnost sdílení paměti existuje.) Z běžných součástí GPU lze použít například texturovací jednotky jako vstupní rozhraní a framebuffer jako výstupní.

Syntaxe programovacího jazyka může být podmnožinou jazyka C (v OpenCL), ovšem existují omezení plynoucí právě z paralelního proudového zpracování. Není možné používat rekurzi, volání vnořených funkcí má svá omezení, jsou zavedeny speciální funkce a datové typy pro práci s vektory, je žádoucí znovuvyužívat použité proměnné a obecně musí optimalizace probíhat jinak než u běžných (CPU) programů - s ohledem na povahu HW.

Vhodnými úlohami pro zpracování na GPU je například zpracování videa, zvuku či řeči, provádění fyzikálních simulací (kapaliny, osvětlení, počasí) anebo třeba úlohy z kryptografie.

5 Popište koncepci a vlastnosti různých typů Streaming Multiprocesoru

TODO

6 Popište principy komprese dat v systému Pascal (delta-komprese a další).

TODO

7 Popište architekturu grafických multiprocesorů a nové principy činnosti, jako komprese dat, preempce a její typy.

TODO

8 Popište princip tensorového jádra Turing, k čemu slouží, jaké formáty dat se používají.

TODO

9 Popište principy a funkce preempce Pascal.

TODO

10 Popište vývoj unifikovaného adresového prostoru, principy činnosti, a jeho hardwarovou podporu.

TODO

11 Popište principy návrhu energeticky úsporného GPU pro mobilní zařízení.

TODO

12 Popište hlavní vývojové etapy a principy grafických systémů Mali pro mobilní zařízení.

TODO

13 Vysvětlete a zdůvodněte koncepci kachliček (tiles) 16x16 pixelů v grafice Mali.

TODO

14 Popište formáty dat podporované v grafice Mali, skalární i vektorové.

TODO

15 Popište a zdůvodněte principy tří základních typů komprese textur - ztrátové, bezeztrátové a adaptivní.

TODO

16 Popište alespoň 3 rozdíly mezi architekturami GPU a CPU.

TODO

17 **Co je to kernel?**

18 **Co je to vlákno?**

19 **Co je to divergence vláken a kdy vzniká?**

20 **Co je to warp/wavefront?**

21 **Co je to multiprocesor (streaming multiprocessor/compute unit) GPU a k čemu slouží.**

22 **Co je to fronta příkazů (command queue)?**

23 **Jakým způsobem je možné synchronizovat úlohy ve frontě příkazů s vykonáváním mimo pořadí (out-of-order execution) nebo mezi frontami v OpenCL?**

24 **Jakým způsobem se eliminuje/zmírňuje dopad aritmetických/paměťových latencí.**

25 **Co je to konflikt banků v lokální paměti a kdy vzniká?**

26 **Co je to zarovnaný přístup do paměti?**

27 **Popište konstantní/uniformní paměť.**

28 **Popište lokální/sdílenou paměť.**

29 **K čemu slouží texturovací jednotky?**

30 **Jaké jsou rozdíly mezi bufferem a texturou v OpenCL?**

31 **Jaký je rozdíl mezi normalizovanými a nenormalizovanými texturovacími koordináty?**

32 **Co je to register spilling?**

33 **Jaké jsou způsoby komunikace mezi vlákny ve skupině a mezi skupinami?**

34 **K čemu slouží atomické instrukce? Popište jejich vlastnosti a příklad algoritmu kde je možné je využít.**

35 **Jaké mohou být příčiny nízké výkonnosti kernelů?**

36 **Co je to obsazenost multiprocesoru (occupancy), na čem je závislá?**

37 **Co je to aritmetická a paměťová latence?**

38 **Co je to separabilní filtr a jakým způsobem lze přistupovat k**