



Grafické a multimediální procesory (GMU)
Příprava na zkoušku

Obsah

1	Popište základní principy návrhu energeticky úsporného procesoru CPU/GPU, jak se vyhodnocuje energetická úspornost.	4
2	Popište vývoj 2D/3D grafického řetězce s různým stupněm zřetězení.	4
3	Popište 1- až n-rozměrné multiprocesorové propojovací struktury (Origin/Onyx) a typy propojovacích sběrnic Silicon Graphics.	5
4	Popište principy výstavby, činnosti a použití GPGPU.	5
5	Popište koncepci a vlastnosti různých typů Streaming Multiprocessoru	5
6	Popište principy komprese dat v systému Pascal (delta-komprese a další).	6
7	Popište architekturu grafických multiprocesorů a nové principy činnosti, jako komprese dat, preempce a její typy.	6
8	Popište princip tensorového jádra Turing, k čemu slouží, jaké formáty dat se používají.	6
9	Popište principy a funkce preempce Pascal.	6
10	Popište vývoj unifikovaného adresového prostoru, principy činnosti, a jeho hardwarovou podporu.	6
11	Popište principy návrhu energeticky úsporného GPU pro mobilní zařízení.	6
12	Popište hlavní vývojové etapy a principy grafických systémů Mali pro mobilní zařízení.	6
13	Vysvětlete a zdůvodněte koncepci kachlíček (tiles) 16x16 pixelů v grafice Mali.	6
14	Popište formáty dat podporované v grafice Mali, skalární i vektorové.	6
15	Popište a zdůvodněte principy tří základních typů komprese textur - ztrátové, bezztrátové a adaptivní.	7
16	Popište alespoň 3 rozdíly mezi architekturami GPU a CPU.	7
17	Co je to kernel?	7
18	Co je to vlákno?	7
19	Co je to divergence vláken a kdy vzniká?	7
20	Co je to warp/wavefront?	7
21	Co je to multiprocesor (streaming multiprocessor/compute unit) GPU a k čemu slouží.	7
22	Co je to fronta příkazů (command queue)?	7

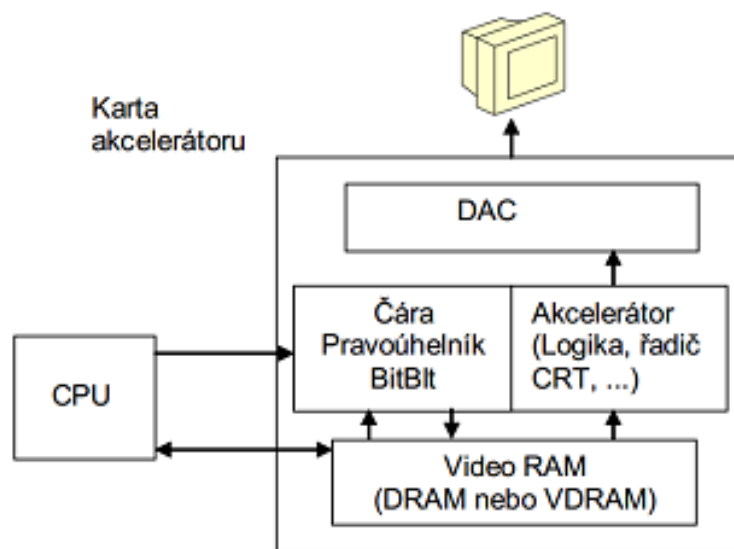
23 Jakým způsobem je možné synchronizovat úlohy ve frontě příkazů s vykonáváním mimo pořadí (out-of-order execution) nebo mezi frontami v OpenCL?	7
24 Jakým způsobem se eliminuje/zmírňuje dopad aritmetických/paměťových latencí.	7
25 Co je to konflikt banků v lokální paměti a kdy vzniká?	8
26 Co je to zarovnaný přístup do paměti?	8
27 Popište konstantní/uniformní paměť.	8
28 Popište lokální/sdílenou paměť.	8
29 K čemu slouží texturovací jednotky?	8
30 Jaké jsou rozdíly mezi bufferem a texturou v OpenCL?	8
31 Jaký je rozdíl mezi normalizovanými a nenormalizovanými texturovacími koordinátami?	8
32 Co je to register spilling?	8
33 Jaké jsou způsoby komunikace mezi vlákny ve skupině a mezi skupinami?	8
34 K čemu slouží atomické instrukce? Popište jejich vlastnosti a příklad algoritmu kde je možné je využít.	8
35 Jaké mohou být příčiny nízké výkonnosti kernelů?	8
36 Co je to obsazenost multiprocesoru (occupancy), na čem je závislá?	9
37 Co je to aritmetická a paměťová latence?	9
38 Co je to separabilní filtr a jakým způsobem lze přistupovat k jeho paralelizaci?	9
39 K čemu slouží paralelní redukce?	9
40 Co je to dynamický paralelismus?	9
41 Jaké jsou možné typy přenosů dat mezi CPU a GPU?	9
42 K čemu slouží P2P přístup a jakým způsobem ho lze využít.	9
43 K čemu slouží operace shuffle?	9
44 Vyjmenujte alespoň 3 api pro GP-GPU.	9
45 Co to je SpirV?	9
46 K čemu slouží operace ballot?	9
47 K čemu lze využít atomicCompSwap?	10
48 Jaký je vztah mezi NDRange/Grid/Dispatch, pracovní skupinou a vláknem/invokací?	10

- 49 Jaké paměti jsou na GPU a popište jejich vlastnosti (odkud je lze plnit, odkud je lze číst, relativní rychlost, relativní velikost). 10
- 50 Jaký je rozdíl mezi pracovní skupinou a warpem/wavefrontem. 10
- 51 K čemu slouží příkaz barrier? Jakou má souvislost s rozdělením vláken do skupin, warpů? Jaký to má vztah k větvení programu? 10

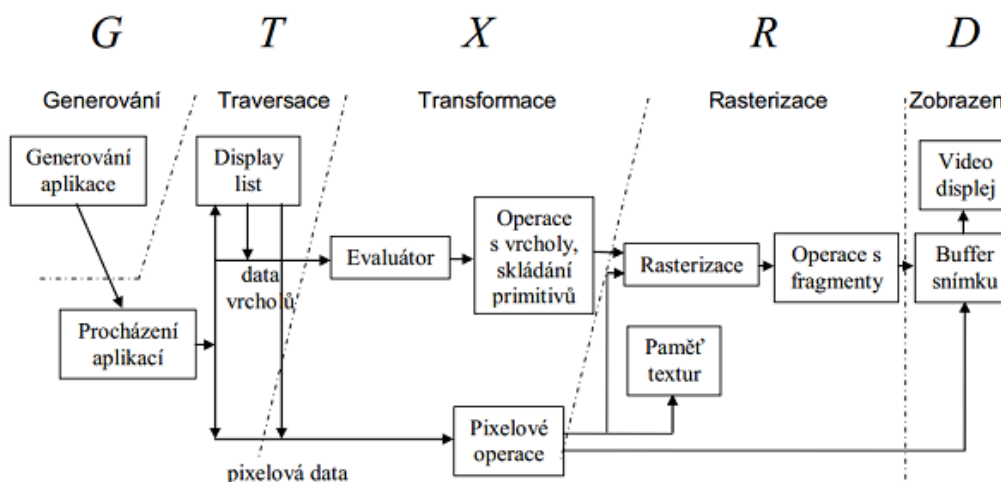
- 1 Popište základní principy návrhu energeticky úsporného procesoru CPU/GPU, jak se vyhodnocuje energetická úspornost.

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 → rasterizacia → zobrazenie (display) Implemento-
vana 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 umožňuje vytvářet až 5D krychle. Díky její pružnosti a koncepční pokrokovosti byla převzata i do navazujících architektur SGI, jako například Onyx2, a další.

Architektura Origin2000 je tvořena množstvím výpočtových uzlů, propojených navzájem propojovací sítí CrayLink. Každý výpočtový uzel obsahuje jeden nebo dva procesory s pamětí cache, sdílenou pamět, adresář pro řízení koherence cache s hlavní pamětí a dvě různé jednotky interface – jeden, XIO, který připojuje I/O jednotky, a druhý, CrayLink, propojující uzly přes 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 přídatné 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, bezztrátové a adaptivní.

TODO

16 Popište alespoň 3 rozdíly mezi architekturami GPU a CPU.

TODO

17 Co je to kernel?

TODO

18 Co je to vlákno?

TODO

19 Co je to divergence vláken a kdy vzniká?

TODO

20 Co je to warp/wavefront?

TODO

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

TODO

22 Co je to fronta příkazů (command queue)?

TODO

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

TODO

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

TODO

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

TODO

26 Co je to zarovnaný přístup do paměti?

TODO

27 Popište konstantní/uniformní paměť.

TODO

28 Popište lokální/sdílenou paměť.

TODO

29 K čemu slouží texturovací jednotky?

TODO

30 Jaké jsou rozdíly mezi bufferem a texturou v OpenCL?

TODO

31 Jaký je rozdíl mezi normalizovanými a nenormalizovanými texturovacími koordinátami?

TODO

32 Co je to register spilling?

TODO

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

TODO

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

TODO

35 Jaké mohou být příčiny nízké výkonnosti kernelů?

TODO

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

TODO

37 Co je to aritmetická a paměťová latence?

TODO

38 Co je to separabilní filtr a jakým způsobem lze přistupovat k jeho paralelizaci?

TODO

39 K čemu slouží paralelní redukce?

TODO

40 Co je to dynamický paralelismus?

TODO

41 Jaké jsou možné typy přenosů dat mezi CPU a GPU?

TODO

42 K čemu slouží P2P přístup a jakým způsobem ho lze využít.

TODO

43 K čemu slouží operace shuffle?

TODO

44 Vyjmenujte alespoň 3 api pro GP-GPU.

TODO

45 Co to je SpirV?

TODO

46 K čemu slouží operace ballot?

TODO

47 K čemu lze využít atomicCompSwap?

TODO

48 Jaký je vztah mezi NDRange/Grid/Dispatch, pracovní skupinou a vláknem/invokací?

TODO

49 Jaké paměti jsou na GPU a popište jejich vlastnosti (odkud je lze plnit, odkud je lze číst, relativní rychlost, relativní velikost).

TODO

50 Jaký je rozdíl mezi pracovní skupinou a warpem/wavefrontem.

TODO

51 K čemu slouží příkaz barrier? Jakou má souvislost s rozdělením vláken do skupin, warpů? Jaký to má vztah k větvení programu?

TODO