

# AGP (Accelerated Graphics Port)

## Bevezető

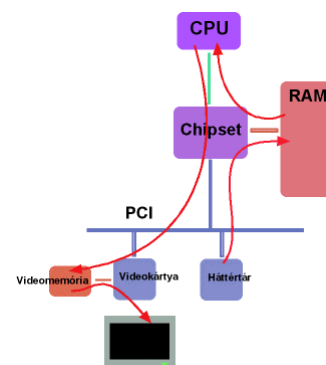
A 3D-s játékok, "interactive full-motion" videók korát éljük. Ezen alkalmazások (egy bizonyos szint felett) olyan nagy erőforrásokat igényelnek, amit a hagyományos rendszerek képtelenek biztosítani. A mai grafikus alkalmazások tehát egyre nagyobb követelményeket támasztanak az őket futtató hardware, többek között a megjelenítésért felelős videokártya iránt. Az új játékok textúrák sokaságát használják/használnák, viszont a ma szokásos 2-4 megabyte videomemória megbéklyózza a programfejlesztőket. Természetesen lehet bővíteni a videomemóriát (megléhetősen borsos áron), de ez nem igazi megoldás, hisz pazarlásnak tűnik olyan memóriába befektetni, amit az alkalmazásoknak csak egy része használ ki. Másik probléma a *PCI* busz, amely szűk keresztmetszetet jelent az adatforgalom számára, főleg, ha figyelembe vesszük, hogy a *PCI* buszt több nagysebességű eszköz is felhasználhatja, például UltraDMA meghajtók, vagy 100 Mbit/s sebességű LAN adapterek.

A fejlesztői szerint az *Accelerated Graphics Port* (rövidítve *AGP*) a megoldás a növekvő grafikus teljesítmény iránti igényre. Nagy sebességű átvitelt biztosít a PC grafikus gyorsítókártyája és a RAM-ja között, ez lehetővé teszi a videokártyának, hogy a textúrákat a kisméretű videomemória helyett a RAM-ból dolgozza fel, valamint segíti, hogy például a dekódolt videóinformáció eljusson a processzorból a grafikus vezérlőre. Nem tévesztendő össze az UMA (Unified Memory Architecture) nevű szabvánnyal, amely teljesen más célból készült. Az UMA célja, hogy a teljes videomemória a rendszer RAM-jába kerüljön költségcsökkentési célból.

## 3D-s grafika a mai PC-ken

Az *AGP* egyik fő előnyét a 3D-s grafika gyorsításában látják. Az előnyök megértéséhez lássuk először, hogy a hagyományos architektúrájú PC-ken (tehát hagyományos 3D gyorsítókkal) hogyan történik a 3D-s grafikák megjelenítése. Ehhez először is műveletigényes geometriai számítások szükségesek, amelyeket általában a processzor végez el, igen alkalmas lévén a szükséges lebegőpontos számítások elvégzésére. Ezzel egyidőben a grafikus vezérlő feladata a textúrák feldolgozása, hogy többé-kevésbé élethű felszíneket és árnyékokat állítson elő.

*A jobb oldali kép egy PCI buszos PC sematikus ábrája. Az ábrán a piros nyilak jelzik az adatforgalmat a háttértárolóról a monitorig a hagyományos, PCI buszos 3D-s gyorsítókártya esetén. (A képre kattintva nagyobbban is nézhető.)*



Ez -a textúrák feldolgozása- a 3D-s grafika legkritikusabb pontja, hisz a textúrák adják meg az objektumok felületét. A textúrák feldolgozása (igen leegyszerűsítve) abból áll, hogy egy bitmapből kell néhány ún. texel-t (texture element) kiolvasni, bizonyos matematikai közelítéseken alapulva átlagolni őket, majd kiírni az eredményként kapott pixelt a framebuffer-be. Az ún. texel koordináták a 3D-s nézőpont és az objektum (mely felszínére a bitmap vetítődik) geometriai felépítésének függvényei. A textúráknak tehát el kell jutniuk a videomemóriába, ahonnan a tényleges kirakás megtörténik.

Ezzel a módszerrel elég sok probléma adódik. Először is a redundáns tárolási mód: a textúrának mind a memóriában, mind a videomemóriában rendelkezésre kell állnia, felesleges memóriapazarlást eredményezve. Másodsor, a textúrák videomemóriában való tárolása (hacsak átmenetileg is) felső korlátot jelent a textúrák méretére. A videomemória elég drága dolog, a mai programok pedig egyre nagyobb és nagyobb, részletesebb textúrákat használnak, kényszerítve a gyártókat és felhasználókat az egyre több és több videomemóriával rendelkező rendszerek gyártására illetve vásárlására.

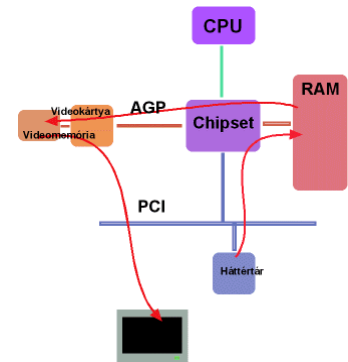
Végül, a *PCI* busz 132Mbyte/s-os átviteli sebessége behatárolja azt a textúramennyiséget, amelyet a videokártya fogadni tud, főleg, ha ezt még más eszközökkel is meg kell osztani. Ezt a problémát a mai programok úgy próbálják áthidalni, hogy a videomemória egy részét cache céljaira használják, amelyben mindig az adott képkocka kirajzolásához szükséges textúrák tárolódnak, míg a maradék videomemória fixen

tartalmazza a gyakran használatos adatokat (pl. az ég mintája egy repülőgépszimulátorban). Ha a hardware csak a videomemóriából tud textúrázni, akkor meg kell próbálni a jelenethez szükséges összes textúrát előre a videomemóriába juttatni, egyébként észrevehető lassulás lesz (ha a rajzolás közben kell a textúrának átkerülnie a videomemóriába).

## 3D-s grafika az AGP-s PC-ken

A 3D-s grafika megjelenítését lehet gyorsítani több módon is. Az első lehetőség egy gyors processzor beszerzése, amely felgyorsítja ugyan a geometriai számítások elvégzését, de nem tesz semmit a másik, szintén kulcsfontosságú feladatért, amely a textúrák eljuttatása a videomemóriába. Az AGP megoldást jelent erre a problémára. Eltávolítja a nagy adatforgalmat a PCI buszról, ami a szűk keresztmetszetet jelenti ebben az esetben, valamint lehetővé teszi a textúrák RAM-ban való tárolását is. A textúrákat tartalmazó memóriaterületeket, (melyet közös néven AGP memóriának, vagy nemlokális videomemóriának nevezünk) az operációs rendszer dinamikusan kezelheti.

A jobb oldali ábra egy AGP-s PC architektúrájának sematikus rajza. Látszik, hogy a videokártya közvetlenül a RAM-ból képes az adatok elérésére. Az adatforgalom a PCI buszt ebben az esetben teljesen kikerüli. (A PCI busz 132 Mbyte/s sebességéhez képest az AGP sebessége 500 Mbyte/s felett van.) (A képre kattintva nagyobb is nézhető.)



## Adatátviteli módok, memória mapping

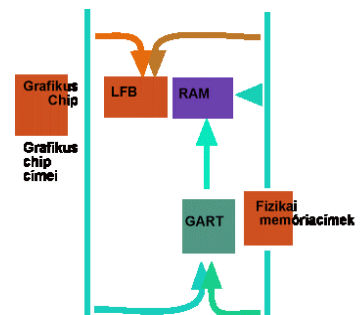
Míg a PCI busz maximum 132 Mbyte/s-os adatátviteli sebességet támogat, az AGP 66 Mhz-es frekvencia esetén is 533 Mbyte/s sebességgel röpti az adatokat rendeltetési helyükre. Az AGP két módot biztosít a videovezérlőnek arra, hogy a RAM-ban tárolt textúrákat közvetlenül elérje, az ún. "pipelining"-ot és a "sideband" címzést. Pipelining esetén az AGP képes átlapolni a memória, illetve a busz eléréséhez szükséges időtartamokat, míg a PCI busz erre nem képes. Ott minden újabb kérés teljesítésének megkezdését meg kell előznie az előző kérés befejezésének. Bár mind az AGP, mind a PCI képes "burst"-olni (több adatcsomagot folytatólagosan továbbítani válaszként egyetlen kérésre), ez a fajta "bursting" csak részben képes a PCI nem pipeline-olt architektúráján javítani. Az AGP pipelining mélysége függ a megvalósítástól, és átlátszó marad a software számára.

A sideband címzésnél az AGP még nyolc "sideband" címvonalat használ fel, melyek lehetővé teszik a videovezérlőnek, hogy az előző kérésekre (a 32 fő adat/címvezetően) érkező adatokkal egyidőben újabb címeket és kéréseket küldhessen.

Az ún. AGP memória tehát nem más, mint dinamikusan allokált területek a memóriában, amelyeket a videovezérlő gyorsan képes elérni. Ez a gyors elérés a 440LX chipset segítségével valósul meg, amely a címek átalakításával lehetővé teszi, hogy a grafikus kártya és software-e a memóriában szerteszét lévő adatokat egy folytonos területnek lássa. A chip neve GART, és funkciójában hasonlít a CPU lapozógységéhez.

A jobb oldali ábra azt próbálja reprezentálni, hogy a videokártya eléri a saját videomemóriáját, illetve a GART-on keresztül a rendszer RAM-ját.

A rendszer közvetlenül címez a RAM-ba, de bizonyos PCI eszközök, amelyek szintén az AGP memóriához nyúlnak (pl. videocapture), ezt a GART-on keresztül teszik. (A képre kattintva nagyobb is nézhető.)



## Szoftver támogatás

Az *AGP* szempontjából a software-fejlesztőknek két lehetőségük van: vagy támogatják az *AGP*-t, vagy nem. Ha a fejlesztők semmi külön erőfeszítést sem tesznek az *AGP* támogatása érdekében, a program akkor is gyorsabban futhat. Ez igaz a régebbi, tehát az *AGP* hardware-t közvetlenül támogatni nem képes alkalmazások esetén is. Az *AGP* hardware háromféle formában ölthet testet, ezek pedig a következők:

- A hardware rendelkezik *AGP* interface-szel, de nem valósítja meg a textúrázási lehetőségeket. Ekkor az előny a *PCI* buszhoz képest csupán a nagyobb sebességben jelentkezik.
- A hardware képes a textúrákat elérni az *AGP* memóriából, a saját videomemóriából pedig vagy igen, vagy nem. Ekkor az *AGP* memória használata gyorsabb lehet, mert nincsenek írási/olvasási konfliktusok a pixelek kiírása, texelbeolvasás stb. folyamán.
- A legjobb megoldás, ha képesek vagyunk mind a lokális videomemóriából, mind az *AGP* memóriából textúrázni. Ekkor a kisebb, gyakran használatos textúrák a videomemóriában helyezkednek el, a nagyobb méretűek pedig a RAM-ban. Ez csökkenti a CPU/videokártya által okozott konfliktust is a memória elérése során.

### DOS alkalmazások:

A textúrák direkt elérése a RAM-ban természetesen a *GART* chipet igényli (a mai operációs rendszerek virtuális címzési módja miatt). Mivel ez DOS-ban nincs, a *GART* nem is tesz itt jó szolgálatot. Tehát a régi DOS-os programok az *AGP* nagyobb sebességétől gyorsulnak, de külön driver szükséges ahhoz, hogy a videokártya a rendszer RAM-jában lévő textúrákat közvetlenül elérhesse.

### Windows alkalmazások:

A normál Windows alkalmazások kihasználhatják az *AGP* előnyeit, mert az oprendszer és a *DirectDraw*-t módosították annak érdekében, hogy alapból támogassa. A Microsoft információi szerint a Windows98 és az NT 5.0 natív *AGP* támogatással fog rendelkezni, míg a korábbi Windows verziókhoz is elérhetők az egyéb cégek által készített *AGP* kiegészítések. Az említett Windows verziókban tehát lesznek az *AGP* inicializálásáért, kezeléséért felelős kódrészletek. Az *AGP* kihasználása a *DirectDraw* 5.0-ás *API*-n keresztül valósulhat meg. A program a *DD*-n keresztül információt kaphat a rendelkezésre álló *AGP* memória mennyiségéről, illetve foglalhat le memóriaterületeket. A színtalpak mögött a *DirectDraw* foglalkozik ezen kérések *AGP*-n keresztüli lekezeléséről (ha a lefoglalt memória mérete már a rendszer stabilitását veszélyezteti, megtagadhat kéréseket). A *DirectDraw* felméri a hardware képességeit, és ettől függően engedélyezi, illetve tiltja a textúrák video/*AGP*-memóriában való elhelyezését.

Ha a program használja az ún. *mipmapping*-et (előre anti-aliasolt, kisebb méretű textúrák használata a távolabbi objektumokon), akkor (a nagyobb memóriaigény ellenére) javul a teljesítmény, mert a chip-nek nem kell nagyobb adatterületeket átugrani (tehát memóriacímeket nagy mértékben növelni), mikor mintát vesz a távoli objektumokra kerülő bitmap-ekből.

A jelenlegi megvalósításokban az oprendszer az *AGP* memóriát kizárja a cache-elésre kerülő memóriaterületekből, hogy ne adódjanak problémák a CPU cache és a videokártya között, egyébként az *AGP* memória elérése a cache ellenőrzését is maga után vonná. Persze a programnak ugyanúgy, mint a videomemória esetén is, kerülnie kell az olvasást ezekről a (cache-eletlen) memóriaterületekről. A Pentium II-es rendszerekben ezek a memóriaterületek WC (write combining) jelzöt kapnak a sima UC (uncacheable) jelző helyett, ami lehetővé teszi a processzornak, hogy "burst" write-ot alkalmazzon a memóriabuszon, ha az elérhető. Ez a módszer a nagyobb sebességet leszámítva átlátszó az alkalmazások számára.

## Az *AGP* előnyei röviden tehát így foglalhatók össze:

- A textúrák méretére, részletességére nem jelent korlátot többé a videomemória mérete.
- 3D-s alkalmazások gyorsulnak - a videomemóriát nem kell cache-elésre használni.
- összefoglalva, nem szükséges nagy méretű videomemória, ez nagyobb felbontások használatát teszi lehetővé, illetve pl. Z-bufferelést nagyobb képernyőméret mellett.
- A rendszer stabilabbá válik olyan értelemben, hogy a nagy sávszélességet lekötő grafikus és videoforgalomtól tehermentesíti a *PCI* buszt.