

On-chip buszok

Áramkörön belüli rendszerek kérdései

Mit jelent a System-on-a-Chip felépítés? Milyen a jellemző belső adatátviteli rendszer kialakítása? Hogyan csoportosítják/szervezik a belső egységek összekapcsolását?

Röviden: Az "egycsipes rendszerek" nagy bonyolultságú IC-k sok és sokféle belső egységgel, melyek jellemzően (sebesség szerint) hierarchikus buszrendszeren keresztül kapcsolódnak.

A System-on-a-Chip szabad fordításban "egycsipes rendszert" jelent és olyan integrált áramkört takar, melynek belső bonyolultsága "elérte a rendszer szintet". Gyakorlatiasabban fogalmazva, a csip önmagában is vagy minimális kiegészítéssel már képes komplex feladatok végrehajtására, mégpedig azért, mert egyetlen lapkára integráltan tartalmaz sok és sokféle egységet (processzormagot, speciális adatfeldolgozókat, perifériákat, vezérlőket, akár FPGA fabric-ot is), melyeket belső - akár hierarchikus - buszrendszer kapcsol össze. A fentiek alapján a mikrokontrollereket is SoC-nek nevezhetnénk, azonban a μC és az SoC között jellemzően nagy, bonyolultságbeli különbség van.

Ami a belső adatátvitelt illeti, integrált áramkörökben nem jellemző a közös buszvonalak és az arra kapcsolódó háromállapotú bufferek alkalmazása. A hiányuknak nem fizikai, hanem gyakorlati oka van: a régebben FPGA-k belső fabric-jában is alkalmazott háromállapotú bufferekkel a figyelmen kívül hagyható(k) könnyedén megsüthetik - és meg is sütötték - az IC-eket. Háromállapotú bufferek hiányában a belső buszok alapja lehet (ld. később részletesebben):

- kapcsolómátrix,
- master-slave multiplexelés,
- "ÉS-VAGY" elrendezés.

Az SoC-k belső buszrendszere jellemzően hierarchikus. Azt, hogy mely eszköz melyik buszra kapcsolódik, elsődlegesen az egység működési sebességétől függ: a főbuszra a leggyorsabb egységek (processzorok, memóriavezérlők), az albusz(ok)ra a lassabb egységek (perifériák) kapcsolódnak.

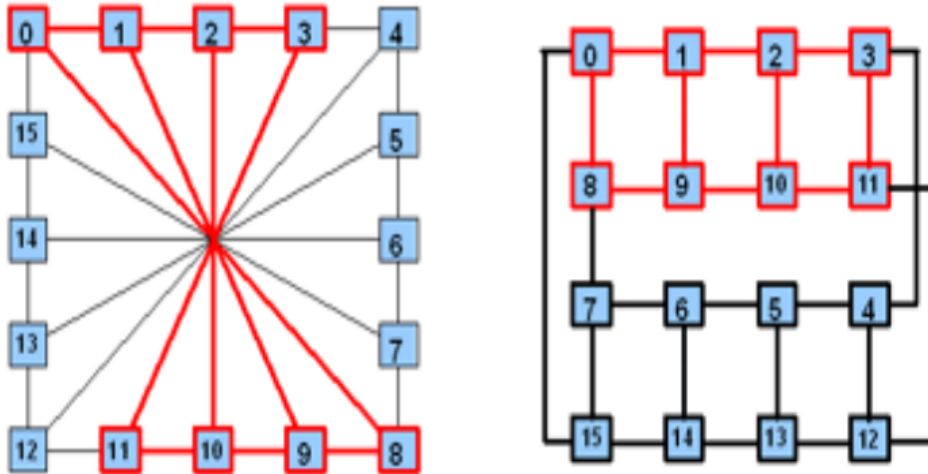
Mit jelent az NoC rövidítés? Az ST NoC hálózatának mi a topológiai előnye a VLSI áramkörtervezés szempontjából? Rajzolja le az elrendezést 15 egység esetén! Azonos elérési tulajdonságokat garantál minden egység között? Mennyi a leghosszabb út állomásainak száma?

Röviden: A Network-on-Chip koncepció a vonalkapcsolt hierarchikus buszok csomagkapcsolt hálózatokkal való kiváltását jelenti. Az STNoC topológia lényege, hogy az egy körre rendezett egységek a két szomszédjukkal és a velük átellenben lévővel vannak csak összekötve. Az STNoC jó teljesítmény/ár kompromisszum és majdnem síkba rajzolható.

A Network-on-Chip az egycsipes rendszereknek az a bonyolultsági szintje, amikor az egyes egységek közötti kommunikáció igényeit egy "hagyományos" buszrendszer már nem képes kielégíteni, ezért a számítógépes hálózatok ötletét "integrálva" egycsipes hálózatot hoznak létre,

mely a "nagy" hálózatokhoz hasonlóan csomagkapcsolt és útválasztók vannak benne. A hálózat alapú topológia az elérhető sebességen túl energiahatékonysági és skálázhatósági szempontból is előnyösebb.

Az STMicroelectronics network-on-chip topológiájának összeállítása: a rendszer egységeit gyűrűbe rendezzük és minden egységet összekötünk a két szomszédjával, valamint a gyűrűn vele átellenben lévő egységgel. Az STNoC topológia lényege a kapcsolatok számának és a csomópontokból kiinduló útvonalak számának csökkentése úgy, hogy közben bármely két egység közötti út se legyen túl hosszú. Az első szempontot hivatott szolgálni az, hogy minden egység mindössze három másikkal van kapcsolatban, utóbbit pedig az biztosítja, hogy a három kapcsolat közül egy éppen az adott egységtől legtávolabb eső másikhoz vezet.



1. ábra. Az STNoC előnye

A topológiának további előnye, hogy könnyedén átrendezhető olyan formára, melyben mindössze egyszer kell, hogy két vezeték messe egymást, egyébként síkba rajzolható.

A topológia, mivel teljesítmény/ár kompromisszumra törekszik, nem garantál azonos útvonalhosszt bármely két egység között, de a leghosszabb útvonal hossza is $N/4$, ami mindössze lineáris skálázódást jelent.

SoC rendszerek esetén mi a különbség a funkcionális modell, az architektúrális modell és a kommunikációs modell között?

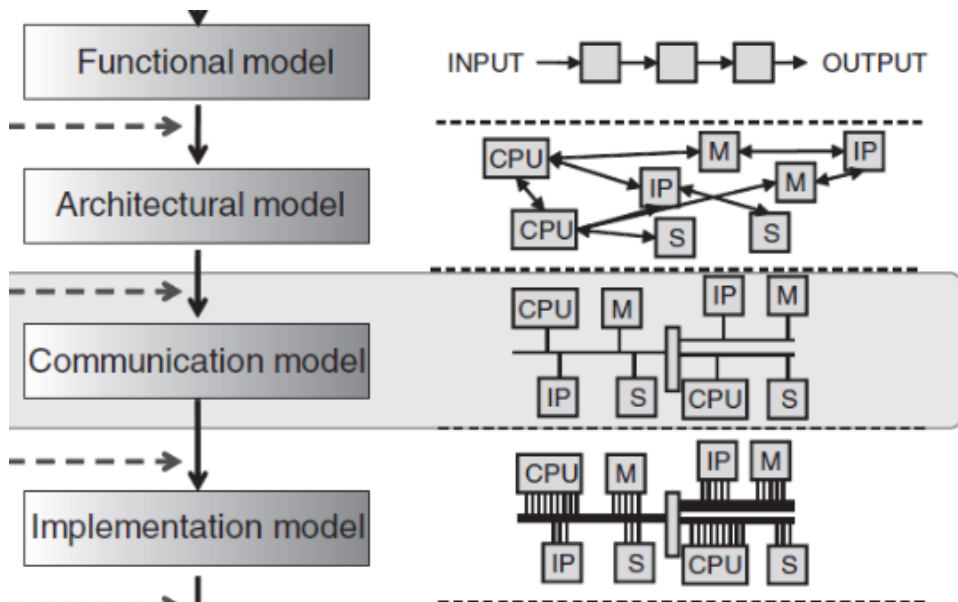
Röviden: A funkcionális modell a bemenet és a kimenet közötti feldolgozó lánc terve. Az architektúrális modell az egyes egységek logikai- és adatkapcsolatainak "spagettije". A kommunikációs modell az egységek busztopológiába rendezése.

Az egycsipes rendszerek tervezésének menetében előbb a rendszert felépítő egységek számbavétele és a bemenettől a kimenetig terjedő információáramlás felrajzolása történik. Ebben a fázisban még nem beszélhetünk buszokról vagy azok topológiájáról, mindössze "dobozokról" és azok feldolgozó láncban való elhelyezkedéséről: ez a funkcionális modell.

A feldolgozó lánc kidolgozásakor felszínre kerül, hogy az egységek milyen logikai viszonyban állnak egymással, milyen logikai- és adatkapcsolatok kell, hogy összekössék őket. Ekkor még csak egy "kapcsolathalmazunk" van, mely konkrét busztopológia helyett leginkább spagettire hasonlít.

A terv részletezésének következő lépése a logikai kapcsolatokat hatékonyan implementáló busztopológia megválasztása a konkrét "vezetékezés" és részletes interfészleírás nélkül: ez a kommunikációs modell.

A kérdés nem tartalmazza ugyan, de a teljesség igénye megkívánja: a busztopológia rögzítését követően, az adat- és vezérlő jelek és az azokra kapcsolódó interfészek kidolgozásának eredménye az implementációs modell.

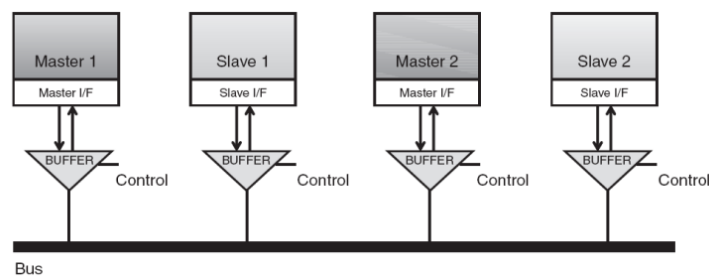


Áramkörön belüli buszokhoz kapcsolódó alapfogalmak: fizikai struktúrák, arbitráció, dekódolás, szinkron-aszinkron átviteli módok, egyszerű - pipe-line - burst átvitelek. Miért nem lehet kiaknázni az aszinkron adatátvitel által kínált elvileg legjobb sebességet?

Röviden: Fizikai buszstruktúrák: hagyományos, multiplexeres, ÉS-VAGY kapus. Arbitráció lehet központi vagy elosztott. Adatátvitel lehet egyszerű, átlapolódó, burst-ös és átlapolódó burst-ös. Az aszinkron átvitel versenyhelyzetekkel és ütközésekkel terhes, mely érvénytelen tranzakciókhoz és újrapróbálkozáshoz vezet.

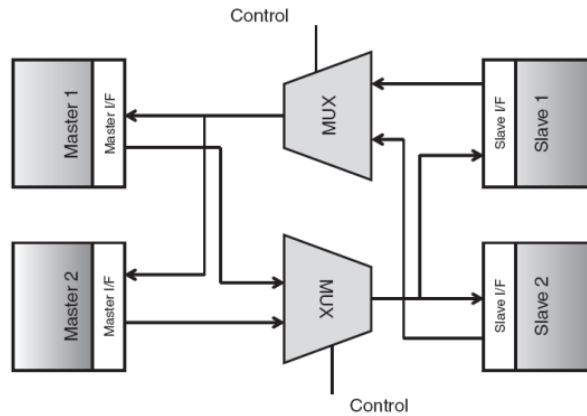
Fizikai struktúrák:

- Hagományos busz: közös vezetékes, három állapotú bufferekkel való kapcsolódás. Jellemzően NYÁK-on alkalmazott, csipen belül nem (ld. első kérdés), előnye a vezetékezési felülettel való takarékoskodás, hátránya a busz szűk keresztmetszetté válása nagy sebességek és adatmennyiség esetén.

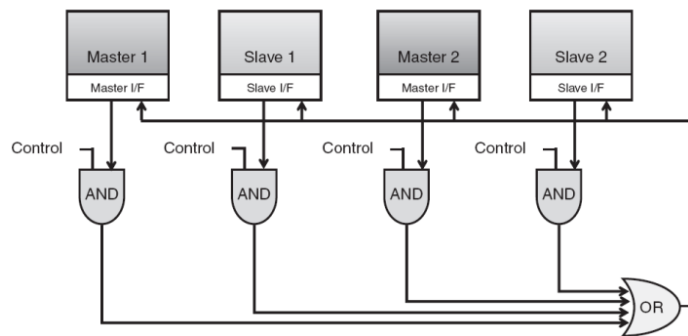


2. ábra. Hagományos busz

- Multiplexeléses busz, kétféle is lehetséges: 1) minden master egy multiplexerre, minden slave egy másikra kapcsolódik, így egy-egy nagy "kapu" van a master-slave és a slave-master irányokban, 2) minden masternek és minden slave-nek saját bemeneti multiplexere van, mely nagyobb párhuzamosságot enged meg, ugyanakkor nagyobb az erőforrás-igénye is.
- "ÉS-VAGY" jellegű busz: minden egység kimenete az azt engedélyező ÉS kapun megy keresztül, majd egy VAGY kapuba fut. Helyes vezérlés esetén csak egy kimenet engedélyezett és így a VAGY kapu bemenetére és magára a buszra is csak az az egy érték kerül.



3. ábra. Multiplexeres busz



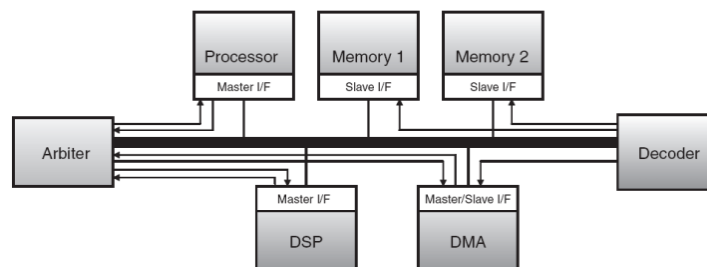
4. ábra. ÉS-VAGY jellegű busz

Arbitráció: a busz vezérléséért versengő master-ek közül eldöntjük, melyik kerülhet elsőként sorra.

Dekódolás: a kiválasztott master utasításainak értelmezése (pl. mely vezérlőjeleket kell figyelembe venni; címdekódolás).

Alapesetek:

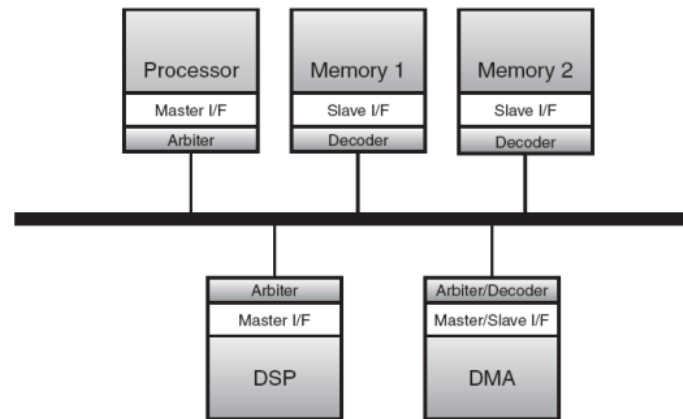
- Központi dekódolás és arbitráció: a busz vezérlési jogáért kérést küldő eszközök közül egy központi egység választ. Előnye, hogy az arbitrációs algoritmust egyetlen példányban kell implementálni, az eredmény pedig egyértelmű mindenki számára, illetve a slave-ek központilag vezérelhetők a kiválasztott master-től függően. Hátránya a külön arbiter és dekóder egységek erőforrás-igénye.



5. ábra. Központi arbitráció

- Elosztott dekódolás és arbitráció: az arbitrációt minden egység maga végzi a buszon lévő jelen megfigyelésével és értelmezésével. Előnye, hogy nem szükséges külön vezérlő egységeket

a rendszerbe építeni, hátránya azonban, hogy az arbitrációs algoritmust minden master egységnek implementálnia kell, mégpedig megbízható módon.



6. ábra. Elosztott arbitráció

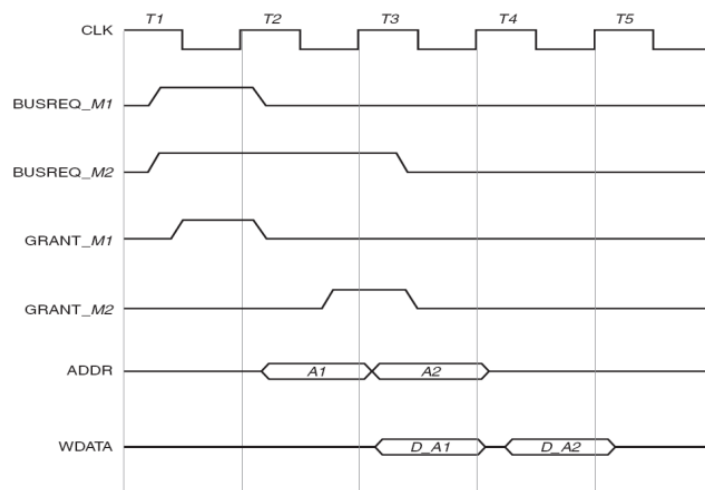
Átviteli módok szinkron-aszinkron csoportosítása:

- Szinkron átvitel: minden tranzakció egy közös órajelre ütemezve kezdődik és meg végbe, illetve egy órajel-periódusig vagy annak előre meghatározott többszöröséig tart.
- Aszinkron átvitel: a tranzakciók kezdete és időtartama az azokat vezérlő (általában handshake-) jelektől függ, a tranzakció lefolyását a meghatározott sorrendben egymást követő jelváltások vezénylik le. (Megj. szemi-szinkron kapcsolatnak nevezhető eset az, amikor bár a jelváltások az órajel ütemére történnek, a tranzakció lefolyását mégis handshake-jelek vezérlik.)

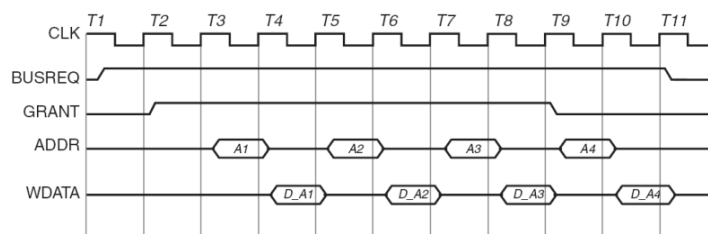
Átviteli módok tranzakciós felépítése szerinti csoportosítása:

- Egyszerű átvitel: az adatátvitel alapegységekben történik (pl. bájtokban vagy 32 bites szavakban), minden átvitel különálló tranzakció (pl. külön arbitrációs, címzési, adatátadási fázisa van), melyek egymástól teljesen elkülönülnek (az egyiknek be kell fejeződnie, mielőtt elkezdődik a másik). Előnye az egyszerűsége, hátránya azonban, hogy a busz egyes vonalai ilyenkor az idő nagy részében kihasználatlanok.
- Pipeline-osított átvitel: az adatátvitel továbbra is alapegységekben történik, az átvitelek önálló tranzakciók, melyek fázisai azonban átlapolódhatnak. Tehát pl. amikor az egyik tranzakció "elengedi" a címvonalakat, a következő már "megragadhatja" és kiteheti rájuk a saját átviteléhez tartozó címet.
- Burst-ös átvitel: az adatátvitel ebben az esetben nem alap egységekben, hanem annak többszöröseiben történik. Egy tranzakcióban nagy mennyiségű adat átvitelre kerülhet, a tranzakciók azonban egymással nem lapolódnak át.
- Pipeline-osított és burst-ös átvitel: Egyetlen tranzakción belül nagyobb mennyiségű adatot is átvihetünk, emellett a tranzakciók - egymást nem akadályozó mértékben - átlapolódhatnak.

Az aszinkron átvitel hatékonyságának kérdése: Bár az aszinkron átvitel tűnik a lehető leghatékonyabbnak, a jelváltások tetszőleges időpontokban való bekövetkezése versenyhelyzeteket és ütközéseket eredményez, melyek az adatátvitel érvénytelenségét vagy megszakítást és újrapróbálkozást vonnak maguk után.



7. ábra. Pipeline-osított (átfedésben lévő) tranzakciók



8. ábra. Burst adatátvitel

Busztopológiák és jellemzőik. Melyik mire képes, mi az előnye, mi a hátránya?

Röviden: Az egyszerű busz valóban egyszerű, de csak kis sebességekhez vagy kevés master-hez használatos. A hierarchikus busz csoportosítja az egységeket és bizonyos mértékben párhuzamosít, de az összekapcsoló bridge-ek erőforrás-igényesek. A megosztott busz az egyszerű busz több részre való szétkapcsolásával keletkezik, így párhuzamosít, de időnként össze kell kapcsolni a buszok közötti átvitelhez. A teljes pont-pont busz a legnagyobb sebességű, de nagy mértékű párhuzamosítás huzalozásigényes. A részleges pont-pont busz a teljesnek redukált verziója, worst case esetben lassabb a teljesnél.

Busztopológiák:

- Egyszerű busz: a rendszer minden alkotóeleme ugyanarra a buszra kapcsolódik. Egyszerű, de könnyedén szűk keresztmetszetté válhat a busz.
- Hierarchikus busz: a rendszer elemei több, különböző buszra kapcsolódnak adott szempont (pl. sebesség) szerint, a buszok közötti információáramlás áthidaló elemek (bridge-ek) segítségével történik. Az egyes buszok a maguk sebességén, protokollján kommunikálhatnak, egymástól függetlenül és egyszerre, de a buszokat összekapcsoló bridge-ek plusz erőforrás-igényt jelentenek.
- Megosztott busz: a rendszer elemei egynél több, azonos paraméterekkel rendelkező busz valamelyikére csatlakoznak, a két busz között pedig igény esetén van információátadás. A megosztott busz és a hierarchikus buszrendszer közötti különbségek: hierarchikus esetben más-más paraméterekkel (pl. sebességgel) rendelkező eszközök nekik megfelelő buszra kerülnek, és paraméter- és/vagy protokollkonverziót végző bridge-ekre van szükség az információcseréhez; megosztott esetben a két busz nem különbözik, a busz szétkapcsolása és az egységek szétosztása mindössze a párhuzamos működést hivatott elősegíteni, de ha egyik

buszról a másira információt kell átadni, a két busz összekapcsolásával a kommunikáció gond nélkül lefolytatható.

- Teljes pont-pont vagy mesh-busz: a rendszer minden egysége össze van kötve mindegyikkel. Előnye, hogy ez biztosítja a lehető legnagyobb adatátviteli sebességet, hátránya azonban, hogy nagy mennyiségű összeköttetést alkalmaz és az egységek számával négyzetesen skálázódik, ráadásul az összeköttetések nagy része a gyakorlatban nincs kihasználva.
- Részleges pont-pont busz: a teljes mesh-buszhoz hasonlóan nagy átviteli sebességet biztosít, azonban annál kevesebb erőforrást használ. Hogy mely összeköttetések kerülnek eldobásra, az az adott rendszertől függ.
- Körbusz: az egységek kör alakú buszra kapcsolódnak. Hasonló az egyszerű buszhoz, azonban a végek összekapcsolása felére csökkenti a buszon a maximális távolságot.

Busz topológiák: ARM AMBA rendszer APB - AHB - AXI Miért használnak hierarchikus felépítést? Mi a szerepe az AHB busznak? Mi a szerepe az APB busznak? Az átvitelek általános tulajdonságai (nem az idődiagram apró részletei, de pl. a fenti alapfogalmak alapján történő felsorolás) Mit jelent a split (megszakított) átvitel?

Röviden:

Az Arm cég AMBA (Advanced Microcontroller Bus Architecture) buszrendszere hierarchikus felépítésű topológia, melyben az egyes buszokra az eszközök sebességük és képességeik szerint csoportosítva kerülnek. Az AMBA rendszer három tagja:

- APB: Advanced Peripheral Bus

Kis sáv szélességű / adatsebességű perifériák csatlakoztatására szolgáló busz. A legegyszerűbb protokollal rendelkező busz az AMBA rendszerben, nem támogatja sem a burst-ös, sem az átlapolódó (pipelined) átvitelt. Az írási és olvasási műveletek ugyanazokon a busz vonalakon mennek végbe, a nagy sebességű buszok felé pedig bridge-ekel keresztül kapcsolódik.

- AHB: Advanced High-performance Bus

A nagyobb sáv szélességű / nagyobb sebességű eszközök csatlakoztatására szolgál. Olyan egységek értendők ide mint nagyobb sebességű belső- vagy külső- memóriák, DMA vezérlő, esetleg DSP processzor. A sebességnövekmény többek között a burst-ös adatátvitel és a tranzakciók átlapolódásának támogatásából adódik. Az AHB továbbra is közös vonalakat használ, ezért nagy számú eszköz esetén a busz továbbra is szűk keresztmetszetet jelenthet az igazán nagy sebességű eszközök számára. Az AHB támogatja a megszakított tranzakciókat.

Létezik Lite verziója, mely a master-ek számát egyre korlátozza. Az AHB-Lite előnye, hogy a busz vezérlése mindig egyértelmű, arbitráció és a velejáró hardver-költség elhagyható.

- AXI: Advanced eXtensible Interface

Az AMBA rendszer legnagyobb teljesítményű interconnect-je, a sebességnövekmény a közös busz elhagyásából és pont-pont kapcsolatokra cseréléséből, valamint a pont-pont összeköttetéseken belül kialakított öt külön csatorna alkalmazásából adódik. Minden AXI kapcsolatnak van külön írási cím, írási adat, írásválasz, olvasási cím és olvasási adat csatornája, melyek mind egymástól közel függetlenül, külön handshake-jelek által vezérelve működnek. Az egyes csatornákon burst-ös és átlapolódó átvitelek is támogatottak. Az AHB mellett az AXI is támogatja a megszakított tranzakciókat.

Az AXI-nak két alverziója is létezik. Az AXI-Lite verzióban nem támogatott a csatornákon történő burst-ös vagy átlapolódó adatátvitel, az ezekhez szükséges jeleket nem is

tartalmazzák az egyes csatornák. Az AXI-Stream verzió a pont-pont összeköttetések kétirányúságát szünteti meg: minden összeköttetésen egyetlen master és egyetlen slave található, az adatátvitel pedig mindig a mastertől a slave felé történik szakadatlan adatfolyam formájában.

A megszakított átvitel a busz jobb kihasználtságát hivatott elősegíteni. Ha egy slave úgy találja, hogy a master által kért tranzakció túl sok buszciklust igényelne, megszakított tranzakciót kezdeményezhet: ekkor megszakad az átvitel, az aktuális master-től az arbiter elveszi a busz vezérlésének jogát és a slave készenléti állapotának jelzéséig maszkolja a tranzakciót eredetileg kezdeményező master kéréseit.

Mik az AHB és az AXI közötti fontosabb eltérések? (Nyilván amiben az AXI többet nyújt, az az érdekes.)

Röviden: busz vs. pont-pont, többszörös vs. egyszeres címzés burst átvitelnél, egyszerű vs. fejlett SPLIT és memóriavédelem, plusz AXI-nál van fix burst mód, szemafor és fogyasztáscsökkentés.

- Az AHB közös busz topológiájú, megosztott írás/olvasás vonalakkal rendelkezik, az AXI pont-pont összeköttetéseket használ és különálló csatornákat az írási és olvasási műveletekhez.
- Az AHB burst átvitel igényli, hogy minden egyes átvitt adatot előtte megcímezzünk, míg AXI esetén csak a burst átvitelben elsőként küldött adat címét kell kiadnunk, a többi adat címét a fogadó fél kell hogy belsőleg előállítsa.
- Az AHB csak egyszerűbb támogatással rendelkezik a megszakított tranzakciókkal szemben.
- Az AHB nem rendelkezik az AXI-ban alkalmazott fix burst móddal, exkluzív hozzáférés-támogatással (szemafor), időzítés elválasztással, sem fogyasztáscsökkentő interfésszel.
- Az AHB mindössze egyszerű támogatással bír a memóriavédelem szempontjából, míg az AXI fejlett eszközökkel rendelkezik.

IBM CoreConnect általános felépítése, a buszrendszer részei (de semmi apró részlet).

Az IBM CoreConnect szolgáltatásaiban és komplexitásában az ARM AMBA-hoz hasonló. Általános célú SoC buszrendszer, háromféle busz alkotja:

- Processor Local Bus (PLB): általános célú processzor busz, a három közül a legnagyobb sebességű. Szinkron, nem multiplexált (hagyományos, közös), multimaster-es busz, külön írás- és olvasás vonalakkal, mely támogatja a pipeline- és burst átvitelt.
- on-chip Peripheral Bus (OPB): a lassabb sebességű (periféria) eszközökhöz. Szinkron, közös busz, külön írás- és olvasás vonalakkal, támogatja a pipeline- és burst átviteletet.
- Device Control Register (DCR) Bus: szinkron, közös busz, külön írás- és olvasás vonalakkal. Egyetlen master lehetséges, 10 bites címzést alkalmaz.

WishBone busz. Mi ez, honnan jön, előnye, hátránya? Milyen kommunikációs topológiákat támogat?

Röviden: Nyílt szabvány, egyszerű, nem hierarchikus. Nincs pipelined, sem split transzfer, de van burst. Multimaster-es, 64 bites címzéssel és változtatható adatmérettel, a topológia többféle is lehet.

Az OpenCores.org nyílt forráskódú buszrendszere. Nem hierarchikus (egyszintű), nagy sebességű, szinkron. Mivel nincs átlapolt működés (pipeline), sem megszakított átvitel (split transaction), ezért inkább közepes vagy kis teljesítményű rendszerekhez ajánlott, de támogatja az egyedi és blokkos (burst) átvitelt is. Multimaster-es, 64 bites címzést alkalmaz, az adatméret 8-tól 64 bitig támogatott. Nincs fix topológiája, többféle rendszer is kialakítható: pont-pont kapcsolatok és adatfolyam, normál (közös) busz vagy crossbar-jellegű interconnect.

Előnyei:

- nyílt, gyártófüggetlen,
- egyszerű,
- rugalmas, többféle topológia is kialakítható.

Hátrányok:

- nem hierarchikus,
- nem támogatja az átlapolt- és megszakításos átviteleket,
- nincs előírás az arbitrációra, sem a hibajelzésre.

Mit jelent a "socket" alapú kommunikáció, mi indokolta a javaslat megszületését? Miért sorakoztak fel mögötte az IP (Intellectual Property) fejlesztő cégek? Mi az előnye/hátránya ennek a megközelítésnek? Ábra az ilyen rendszer felépítéséről, több egységgel (M és S egységek és egy valamilyen busz). Milyen rétegek jelennek meg az egységeknél? Milyen alapvető átvitel típusok léteznek?

A socket alapú kommunikáció célja, hogy a rendszerépítés során az elkészítendő vagy felhasználandó (akár többféle gyártótól származó) IP blokkok egységes interfésszel rendelkezzenek, ne kelljen az adott buszra szabni a buszfüggetlen funkcionalitást, hordozhatóbbak legyenek a tervek.

Ezért az aktuális busz és az IP-k közé egy adatperréteget helyezünk: az adatterréteg a busz irányába a szükséges, buszspecifikus interfészt mutatja, az IP felé pedig szabványos socket interfészt biztosít.

OCP profilok, (csak az első 3 dia, további részletek nem) mi lehetett a profilok kialakításának oka és vajon miért lett a nevük H-bus profil, X-bus profil?

Röviden:

Az OCP az Open Core Protocol interfész rövidítése és socket alapú kommunikációs felületet definiál. A kommunikáció szinkron pont-pont kapcsolatot valósít meg, az adatfolyam- és kiegészítő jelek konfigurálhatók. Támogatja az átlapolt és blokkos átvitelt is.

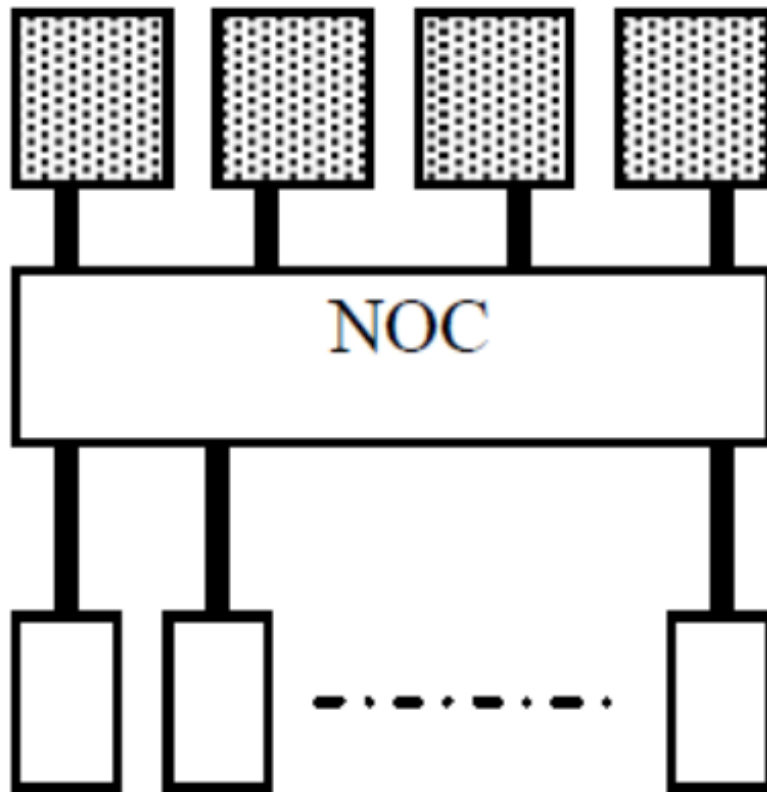
Előnye a szabványos kapcsolatok lehetősége, amivel csökken az inkompatibilitás veszélye, könnyebb a tesztkörnyezet fejlesztés.

A különböző képességű eszközökhöz profilok definiáltak. A natív profilok a 1) blokkos adatfolyam profil, 2) szekvenciális adatfolyam profil és 3) regiszter hozzáférés profil. Ezekon kívül híd profilok is definiáltak, melyek a 1) szimpla H-busz profil, 2) X-busz írás- és 3) olvasás profil. A H- és X-busz profil elnevezése valószínűleg az AHB és AXI buszokra utal.

NoC rendszerek, miért születtek, milyen a kialakított kommunikációs modell?
Az FPGAs példa rendszer különböző alternatív elrendezési lehetőségei 4 külső memória és 24 processzor esetére.

A Network-on-Chip interconnect rendszerek a buszrendszerek szűk keresztmetszetté válására adnak választ azzal, hogy a számítógépes hálózatokban alkalmazott, csomagküldéssel operáló topológiát valósítják meg szilíciumon.

24 processzoros, 4 memóriás példa lehetséges topológiái:



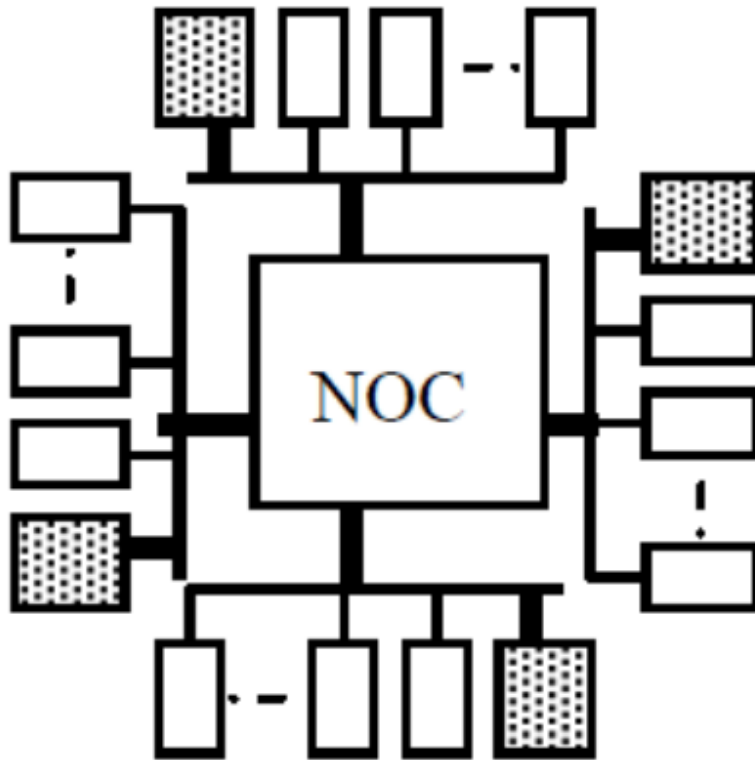
9. ábra. Teljesen globális összeköttetések

**GALS rövidítés jelentése. Miért izgalmas ez nagy ill. komplex rendszereknél?
Mit jelent az órajeltartományok közötti áthidalás?**

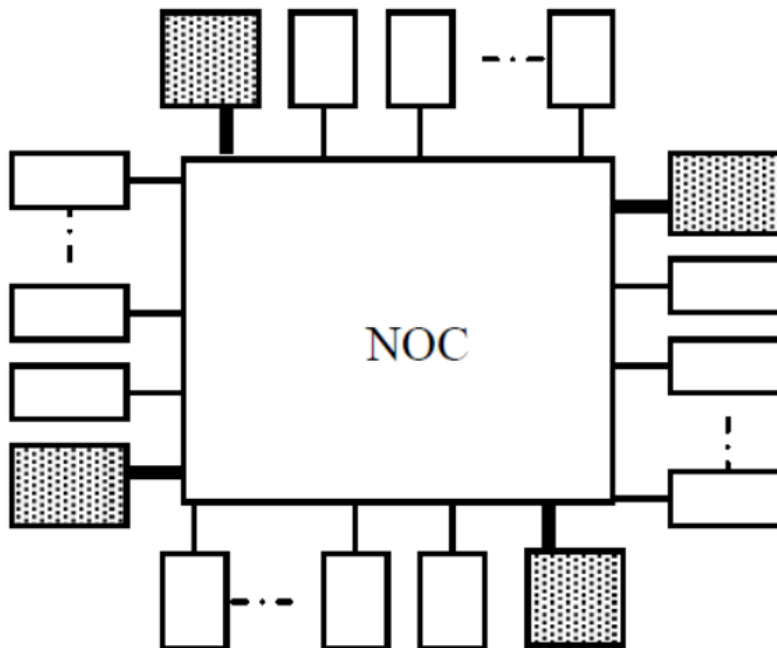
Röviden:

GALS = Globálisan Aszinkron, Lokálisan Szinkron.

Nagy és/vagy komplex rendszerek esetén az egyes alrendszerek, egységek számára más és más órajel-frekvencia lehet optimális (pl. több, különböző interfész kezelésekor). Az ilyen rendszereket órajel-tartományokra oszthatjuk, így minden egység a számára legkényelmesebb, legoptimálisabb sebességgel járhat. A rendszer így lokálisan (egyes egységeken, modulokon belül) szinkron marad, globális értelemben azonban aszinkron lesz. Az órajel-tartományok közötti átjárást jellemzően aszinkron, handshake-jellegű vezérlőjelekkel és FIFO adatcsatornákkal valósítják meg.



10. ábra. Lokális összeköttetések



11. ábra. Részben lokális kapcsolatok a globális interconnectben

Bus Introduction diaszor kérdései

Beágyazott rendszerek különböző felépítési szintjei (Rendszer-kártya-alkatrész)

- Rendszer szint: komplett rendszerekről, termékekről és kapcsolatokról beszélünk. A rendszer szint jellemző felépítésében egy központi egység áll kapcsolatban több perifériával,

periféria buszon keresztül.

- Kártya szint: a nyomtatott áramköri lapok szintje. A NYÁK-ok, mint alegységek, egymással való interakciója hozza létre a rendszer funkcionalitását. A kártyák mind egy-egy feladatot látnak el a rendszeren belül, olyanokat mint a tápellátás, az adatgyűjtés, -feldolgozás, periféria illesztés. A kártyák közötti kommunikáció belső buszokon keresztül folyik.
- Komponens vagy alkatrész szint: a nyomtatott áramkörökön lévő analóg- vagy digitális integrált áramkörök, aktív- és passzív alkatrészek, csatlakozók szintje. A kártyán belüli információátadás és egyéb folyamatok is nyomtatott vezetékezésen történnek.

Honnan ered a PC-104 rendszer architektúra, mi volt a fő előnye, milyen lehetőségeket biztosított?

A PC-104 kifejezés egyaránt utal a PC-104 busz struktúrára és a PC-104 form-factorra. A PC-104 fő előnye, hogy a kártyákból felépülő beágyazott rendszerek számára a gyengébb mechanikai tulajdonságokkal bíró slot alapú kialakítás helyett egy sokkal robusztusabb, stack jellegű elrendezést definiál.

Sorolja fel a beágyazott rendszerekben használt buszok típusait a funkciójuk jellege szerint. Adjon egy-két konkrét példát, ahol lehetséges.

- Rendszer busz: pl. ISA, PCI, PCI Express
- Rendszer busz kiterjesztés: pl. PCMCIA, Cardbus
- Integrált áramkörök közötti kommunikáció: I2C, JTAG
- Nagy sebességű periféria buszok: USB, Firewire, Ethernet
- Kis sebességű periféria buszok: RS család, CAN

Ismertesse a párhuzamos rendszerbuszok részeit! Mit jelent a multiplexált buszfelépítés és hogyan történik ilyenkor az átvitel ütemezése? Mondjon egy példát egy ilyen típusú rendszerbuszra!

A rendszerbusz két vagy több kártyát összekötő kommunikációs vonalak összessége.

Részei:

- Adatbusz: az adatok és utasítások utaznak rajta, jellemzően kétirányú. A szélessége változó lehet, nagyobb sebességigény mellett jellemzően nagyobb a szélesség is.
- Cím busz: az adatot küldő és/vagy fogadó felet azonosító információkat szállít, jellemzően egyirányú (master -> slave). A szélessége a címtér méretét, a rendszer eszközkapacitását szabja meg.
- Vezérlési busz: a busz működéséhez elengedhetetlen vezérlési és időzítési információkat hordozza (pl. órajel, írás/olvasás vezérlés, megszakításkérések).
- Tápfeszültség busz: a buszra kapcsolódó eszközök tápellátását, energiaszétosztást végez.

Multiplexált busz: az adat- és címvezetékek fizikailag azonosak, a különféle információk időben felváltva használják a vonalakat. Ilyen busz például a PC-104 Plusz.

Mi jellemzi a master (Iniciator) ill. Slave (Target) funkciójú eszközöket az átvitel lebonyolítása szempontjából? Írja le a mechanizmust!

A master az, aki kezdeményezi a busz tranzakciót úgy, hogy parancsot (vezérlést) és címet küld a slave felé. A slave erre válaszol, majd írási- vagy olvasási kéréstől függően adatot fogad vagy adatot küld.

Mit jelent a programozott (lekérdezéses) és a megszakításos perifériakezelés? Mik az előnyök és hátrányok? Mi jellemzi a válaszidőket?

A lekérdezéses perifériakezelés azt jelenti, hogy a busz master a tranzakció kezdeményezése után aktívan várakozik a slave válaszára, folyamatosan figyel/lekérdezi annak állapotát. Előnye, hogy ekkor a lehető leggyorsabban megy végbe a tranzakció, hátránya viszont, hogy ekkor a processzor nem tud mással foglalkozni, így üresjáratban van.

A megszakításos perifériakezelés esetén a master a tranzakció kezdeményezése és a szükséges információk megadása után folytatja a munkáját, a slave választ nem várja meg, a slave készenlétéről a mastert megszakítás útján tájékoztatja. Előnye, hogy a processzor várakozás helyett "hasznosan töltheti az idejét", hátránya azonban, hogy a megszakításkérés kiszolgálása nem feltétlenül következik be azonnal, vagyis a tranzakció ez esetben hosszabb, mint a lekérdezéses esetben.

A PCI busz jellemzői

- Hierarchikus rendszerben 256 eszköz csatlakoztatására alkalmas (a hierarchiát a busz számozása adja).
- Párhuzamos adatátvitel jellemzi, 32- és 64 bites adatszélesség lehetséges, támogatja a blokkos átvitelt.
- Három címtartományt definiál: memória-, I/O- és konfigurációs tartományokat.
- Az adat- és címvonalak időben megosztottak.
- Központi arbitráció.
- Egyéb vonalak: interfész vezérlés, megszakítás, hiba vonalak stb.

A PCIe (PCI Express) busz jellemzői

- Pont-pont összeköttetés,
- nagy sebességű (2.5 Gb/s) soros kommunikáció,
- csomag alapú átvitel,
- LVDS jelek, 8B/10B kódolás,
- a sebesség tovább növelhető több vonal együttes használatával.

Periféria buszok/kapcsolatok kialakítási lehetőségei. Az RS típusú fizikai kapcsolatok jellemzői: RS232 / RS422 / RS485 (Szimm./aszimm., pont-pont, busz, sebesség, távolság)

RS-232:

- Kis sebességű (20 kbps),
- pont-pont összeköttetés,

- félduplex vagy duplex kommunikáció,
- aszinkron,
- aszimmetrikus jelvezetés,
- maximális távolság 15 méter.

RS-422:

- Nagyobb sebességű, mint az RS-232: 10 méterig 1 Mb/s, 100 méterig 100kb/s, 1 km-ig 10 kb/s ajánlott,
- kettőnél több pont összekötésére is alkalmas,
- szimmetrikus (differenciális) jelvezetés,
- az adatsebesség csökkentésével akár 1 km-ig növelhető maximális távolság.

RS-485:

- Tipikus átviteli sebesség 100 kb/s,
- kétvezetékes, multimaster-multislave típusú, soros busz,
- szimmetrikus (differenciális) jelvezetés,
- max. 1200 méter távolság.