

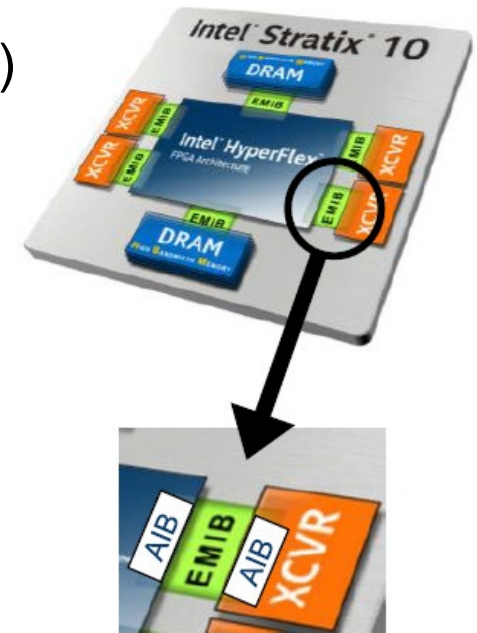
Áramkörön belüli rendszerek

SoC

System on a Chip

Buszrendszerek

- Off-chip buszok
 - “Tradicionális” buszok: ISA, PCI, NXP LocalBus
 - Pont-pont: PCIe, Intel QPI, AMD Hypertransport, AMD InfinityFabric
 - Chip-ek közötti adatbuszok (Intel EMIB / AIB)
- On-chip buszok
 - ARM AMBA: AHB, AXI, ACE
 - IBM CoreConnect: PLB, OPB
 - Intel (Altera) Avalon
 - OpenCores Wishbone
 - Accellera Open Core Protocol



Áramkörön belüli rendszerek

SoC / SoPC

System on a (Programmable) Chip

Alapfogalmak (1)

- Master (Initiator, Requirer)
 - Busz ciklust kezdeményezni tudó egység
- Slave (Target, Completer)
 - Busz ciklusban csak válaszolni tudó egység
- Busz ciklus fázisai
 - Arbitráció: melyik master kapja meg a buszvezérlés jogát
 - Cím: Slave egység kiválasztása
 - Adat: tényleges adatátvitel

Alapfogalmak (2)

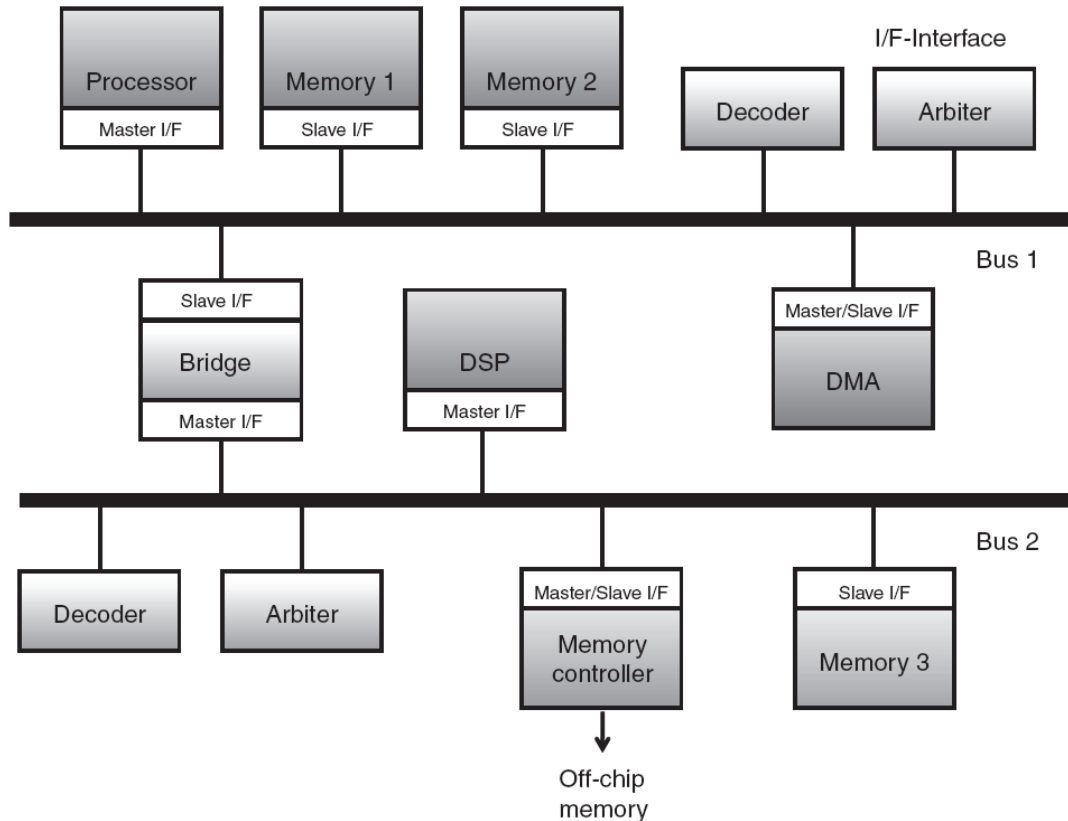


FIGURE 2.1

Example of an SoC with a bus-based communication architecture

SoC rendszer tervezési lépései

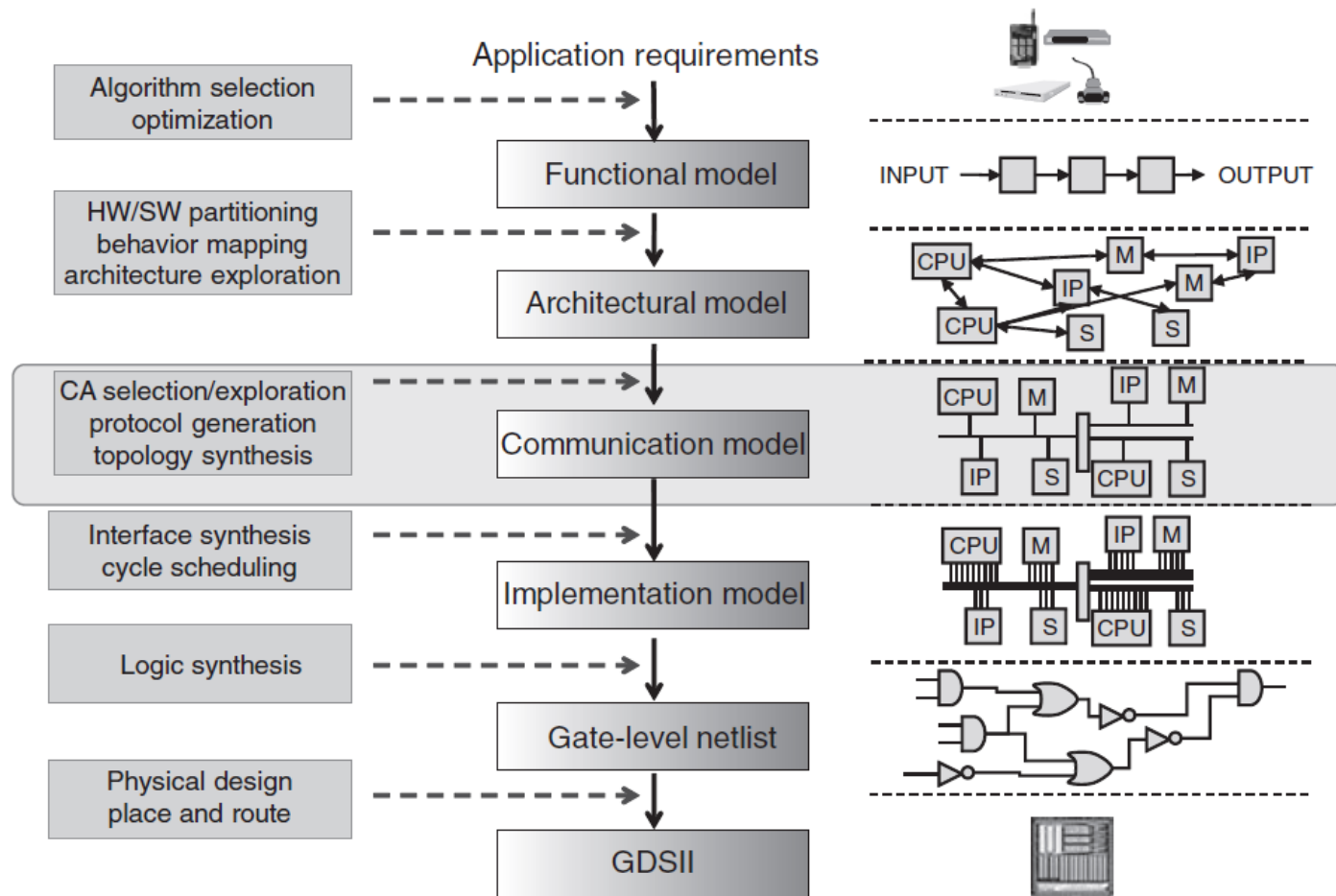


FIGURE 1.3

Ideal ESL design flow for MPSoCs (the on-chip communication architecture design phase is highlighted)

Követelmények növekedése

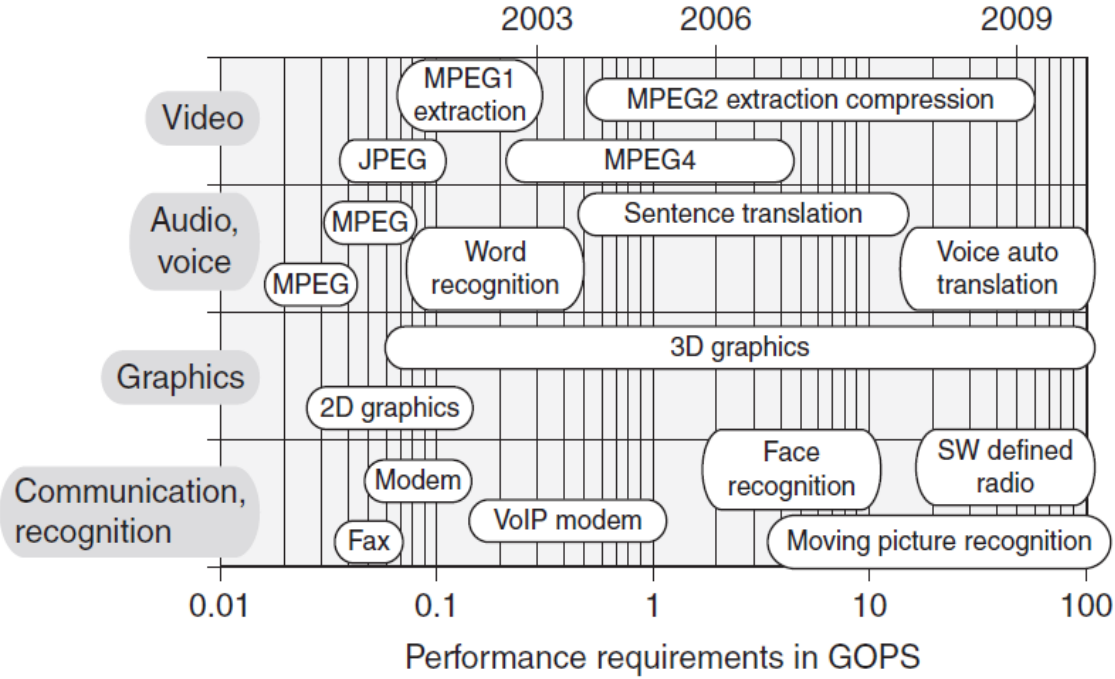


FIGURE 1.5

Increasing performance requirements for emerging applications [3,4]

Fizikai struktúra 1.

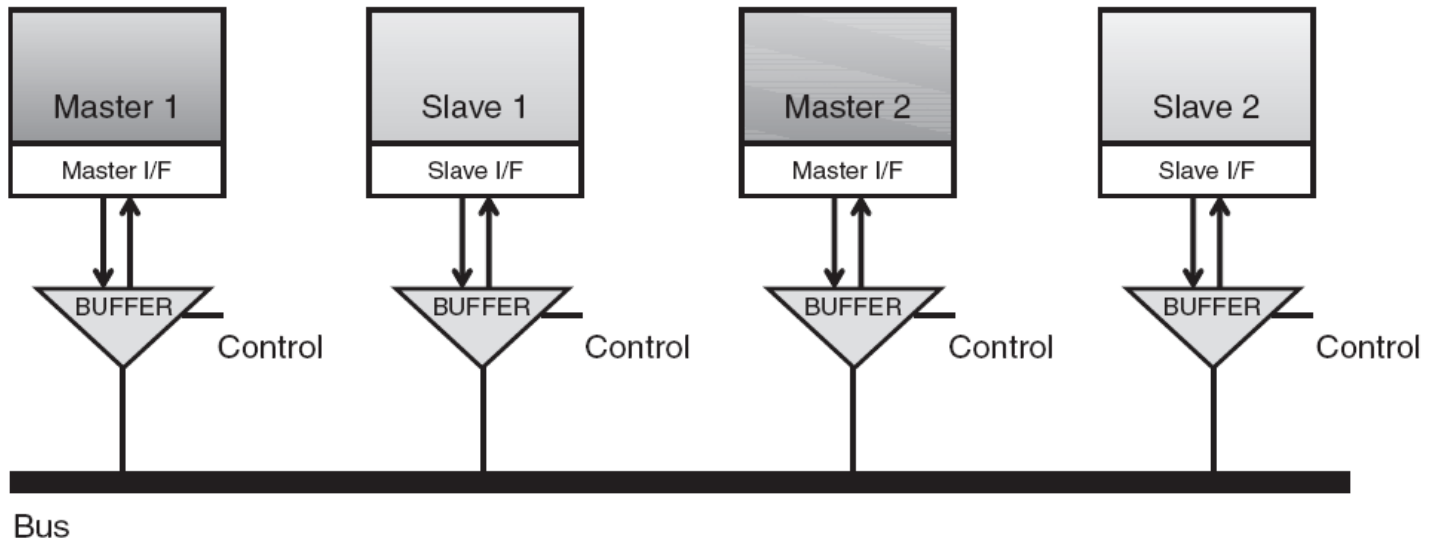


FIGURE 2.3(a)

Shared bus implementation alternatives: tri-state buffer based bidirectional signals

Fizikai struktúra 2.

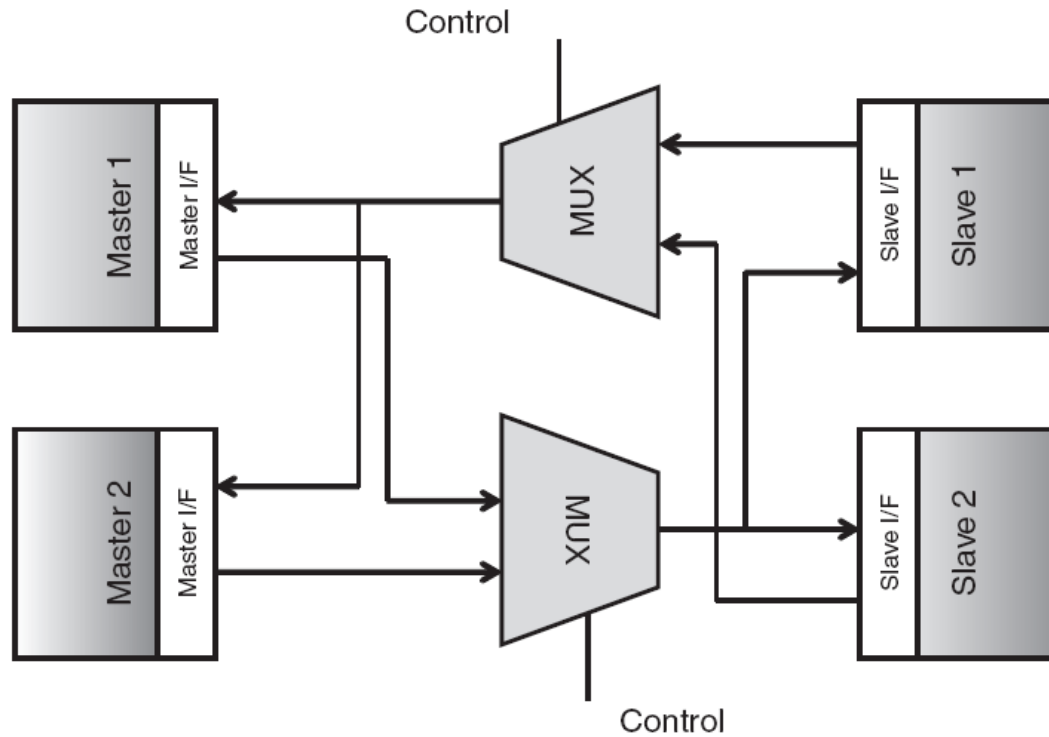


FIGURE 2.3(b)

MUX-based

Fizikai struktúra 3.

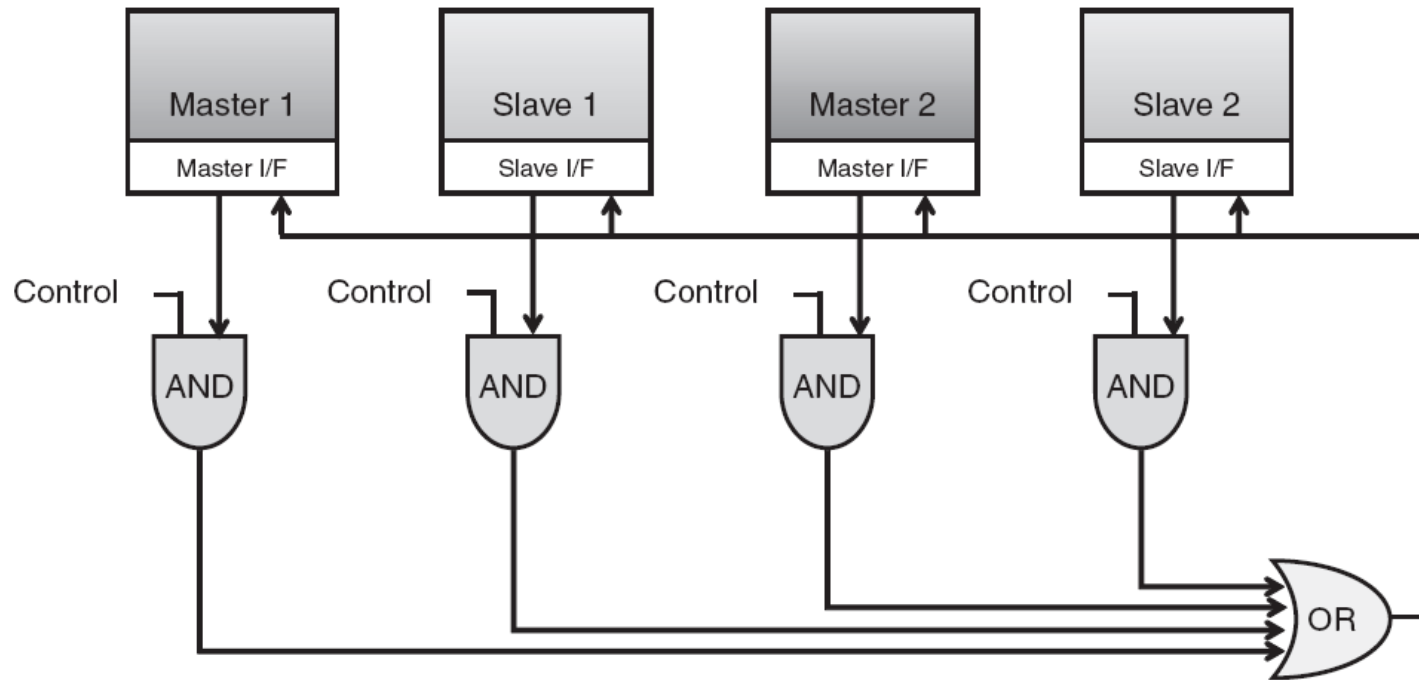


FIGURE 2.3(c)

AND-OR bus

Központi dekódolás

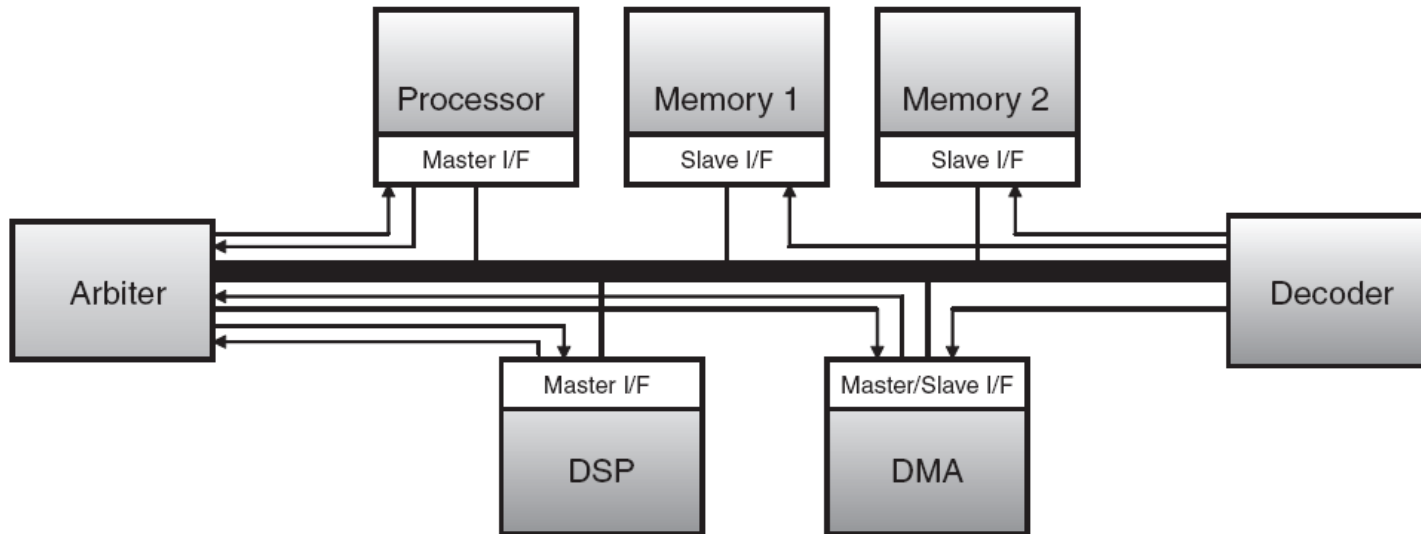


FIGURE 2.5(a)

Different implementation strategies for decoder and arbiter: centralized

Elosztott dekódolás

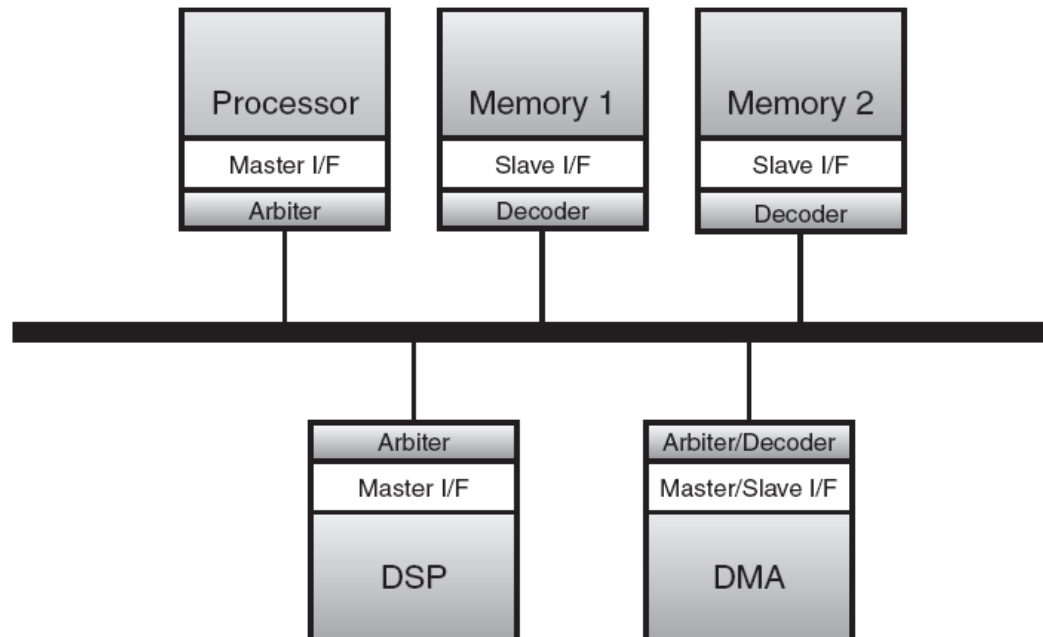


FIGURE 2.5(b)

Distributed

Szinkron-aszinkron átviteli mód

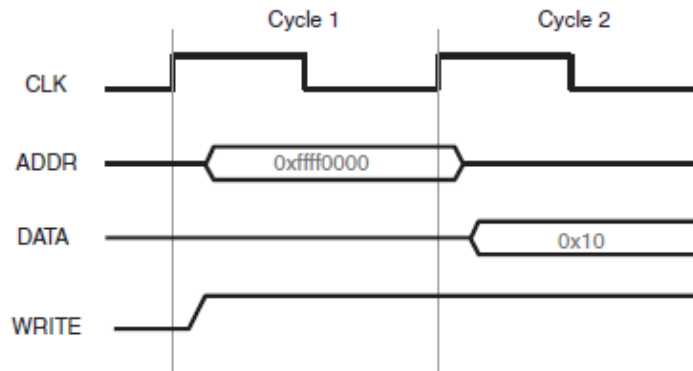


FIGURE 2.4(a)

Clocking strategies for buses: synchronous bus

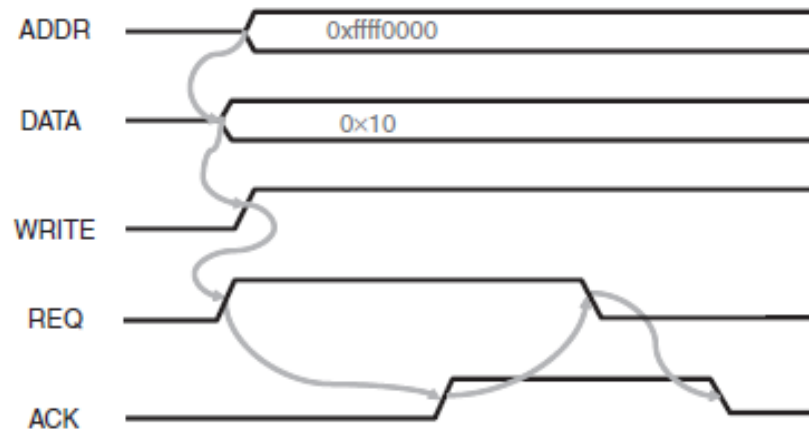


FIGURE 2.4(b)

Asynchronous bus

Adatátviteli módok - Egyszerű

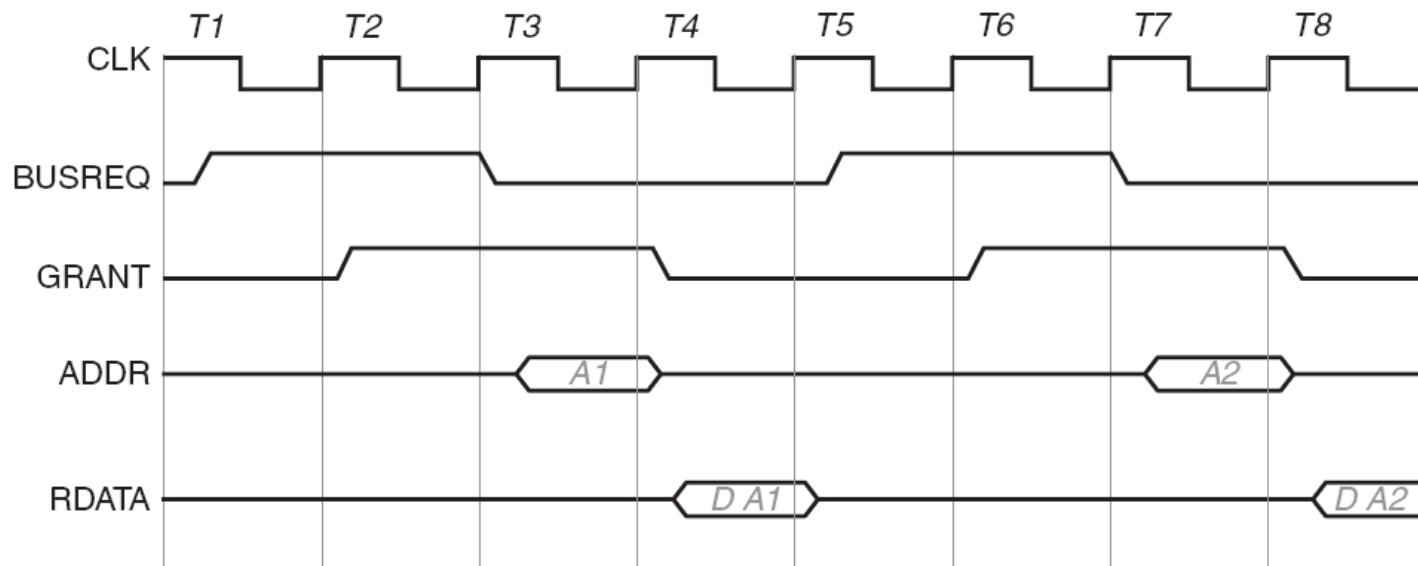


FIGURE 2.6

Single non-pipelined data transfer mode

Adatátviteli módok - Pipeline

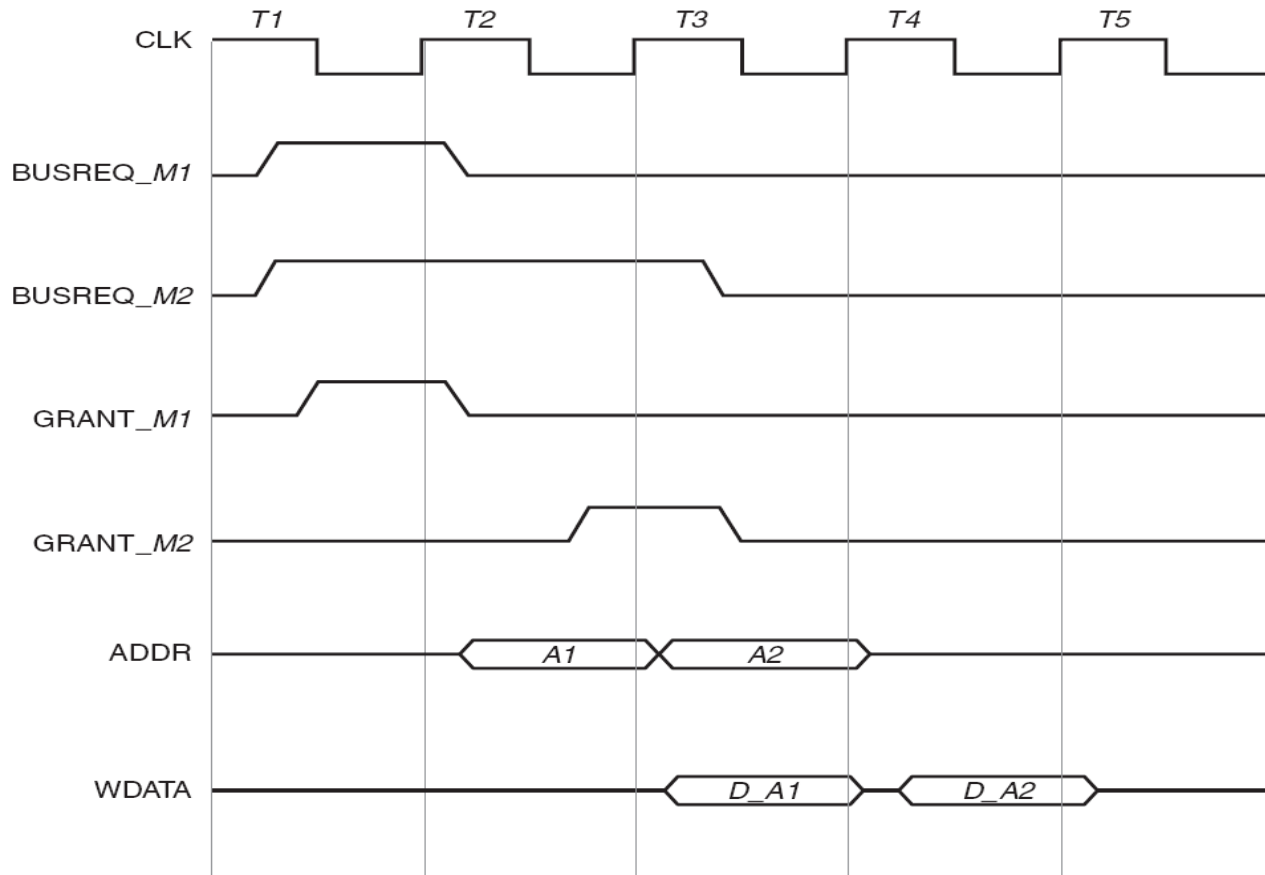


FIGURE 2.7

Pipelined data transfer mode

Adatátviteli módok - Burst

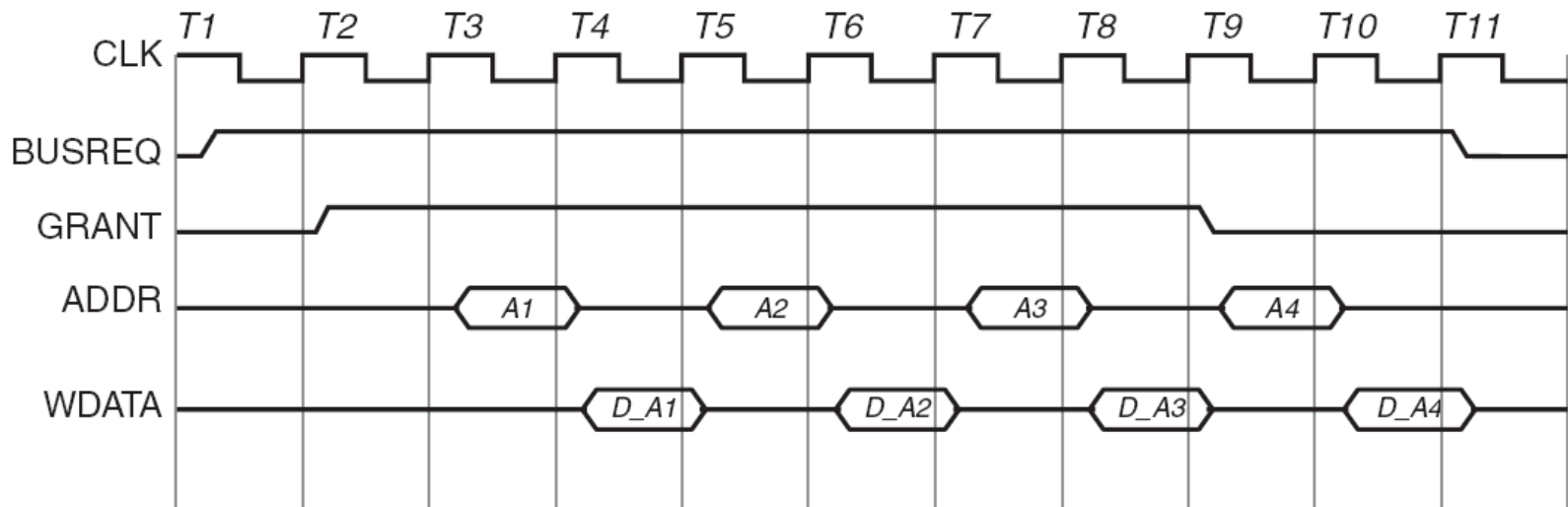


FIGURE 2.8(a)

Example of master writing four data items in burst transfer mode: non-pipelined burst transfer mode

Adatátviteli módok – Pipelined Burst

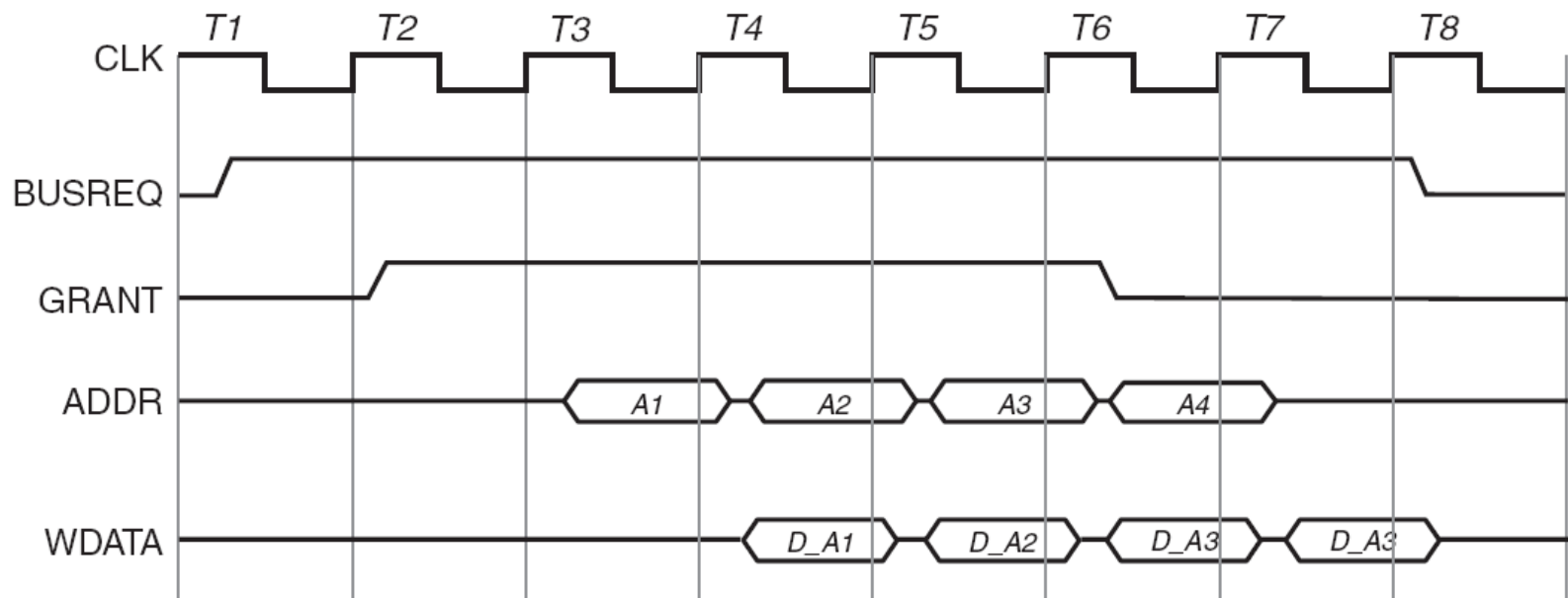


FIGURE 2.8(b)

Pipelined burst transfer mode

Busz topológiák - Egyszerű

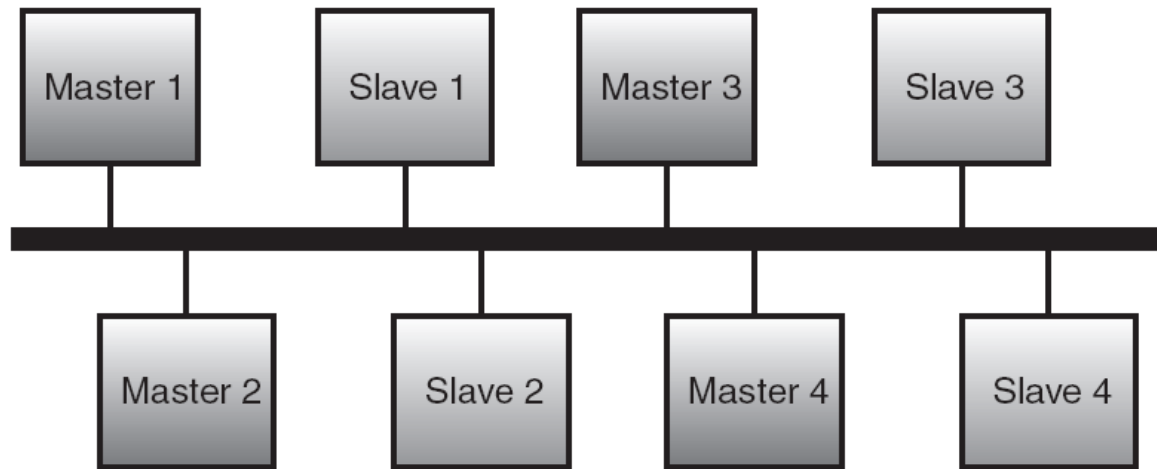


FIGURE 2.9(a)

Different bus-based communication architecture topology structures: single bus

Busz topológiák - Hierarchikus

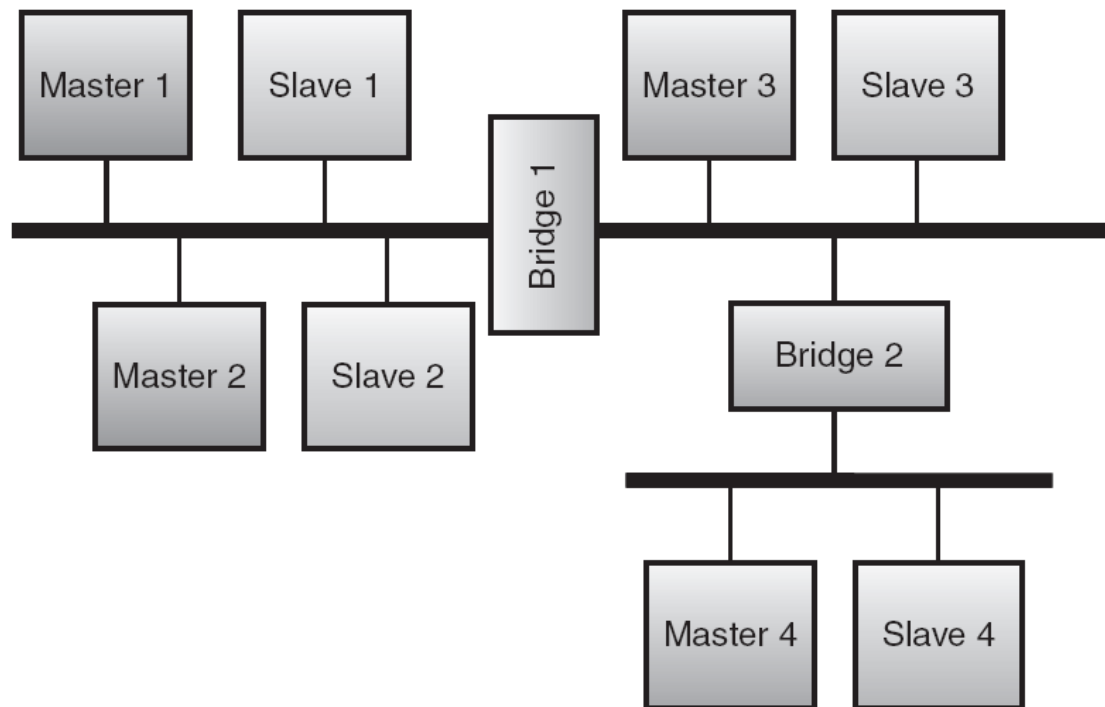


FIGURE 2.9(b)

Hierarchical bus

Busz topológiák - Megosztott

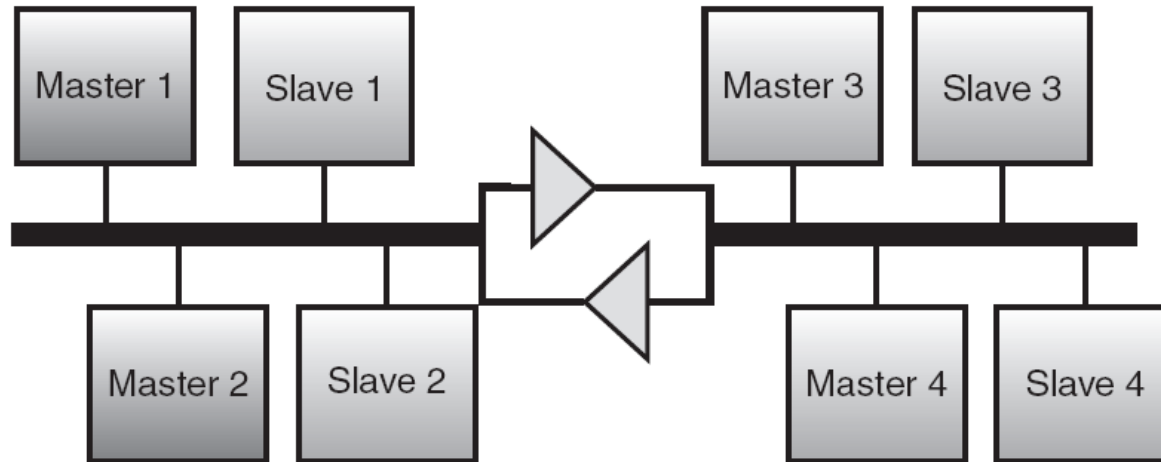


FIGURE 2.9(c)

Split bus

Busz topológiák – Teljes pont-pont

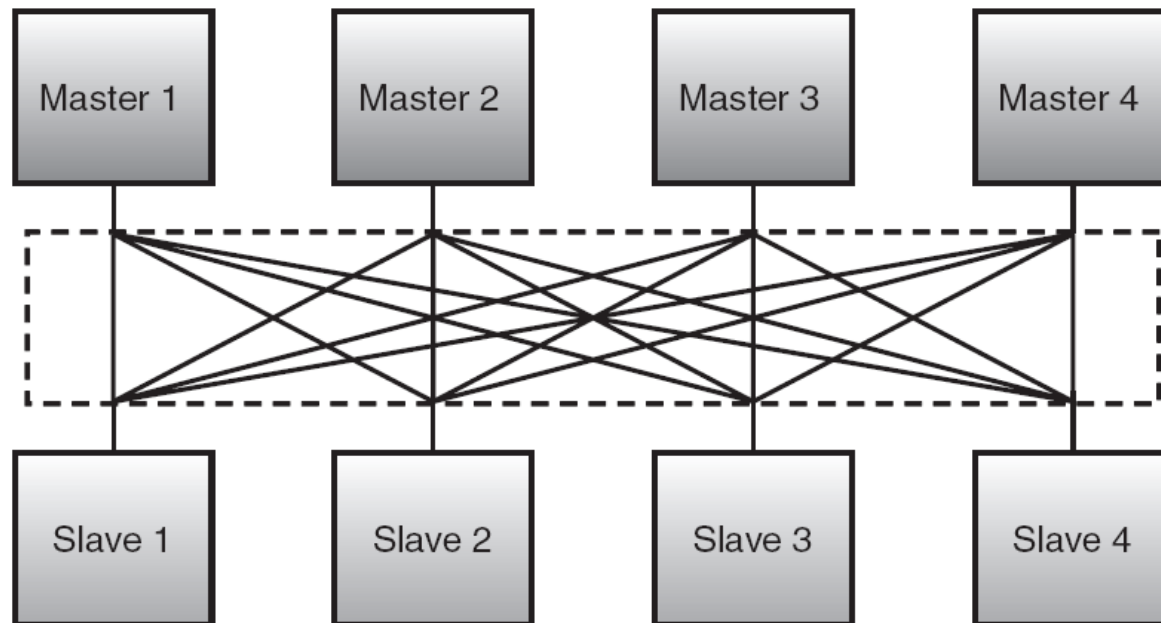


FIGURE 2.9(d)

Full bus crossbar (or point-to-point bus)

Busz topológiák – Részleges

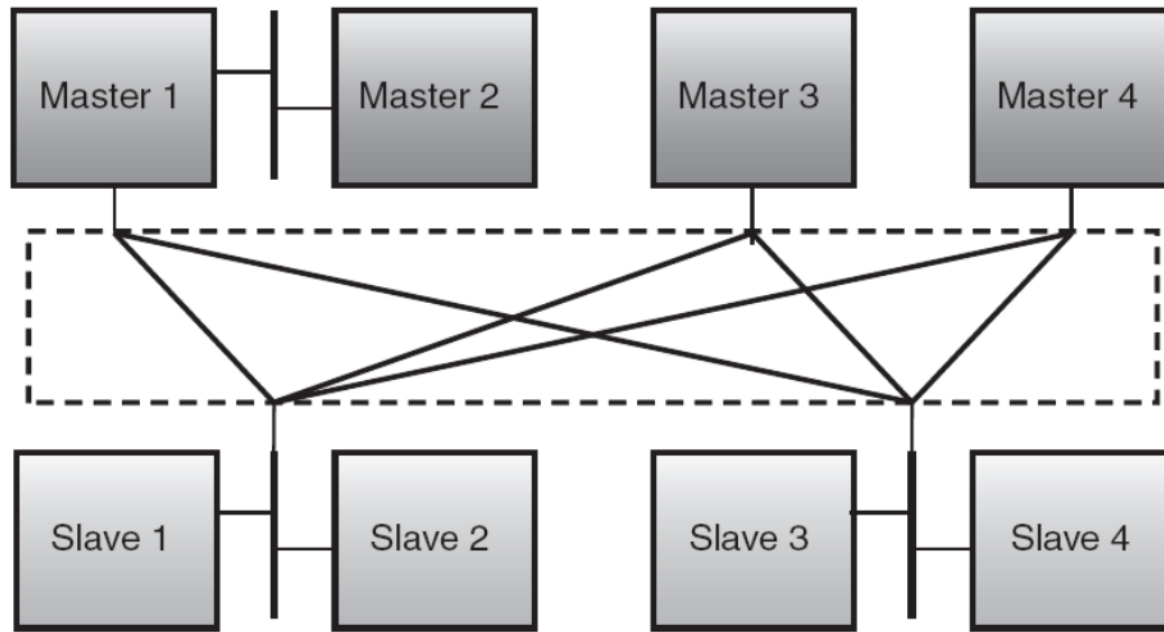


FIGURE 2.9(e)

Partial bus crossbar

Busztopológiák - Ring

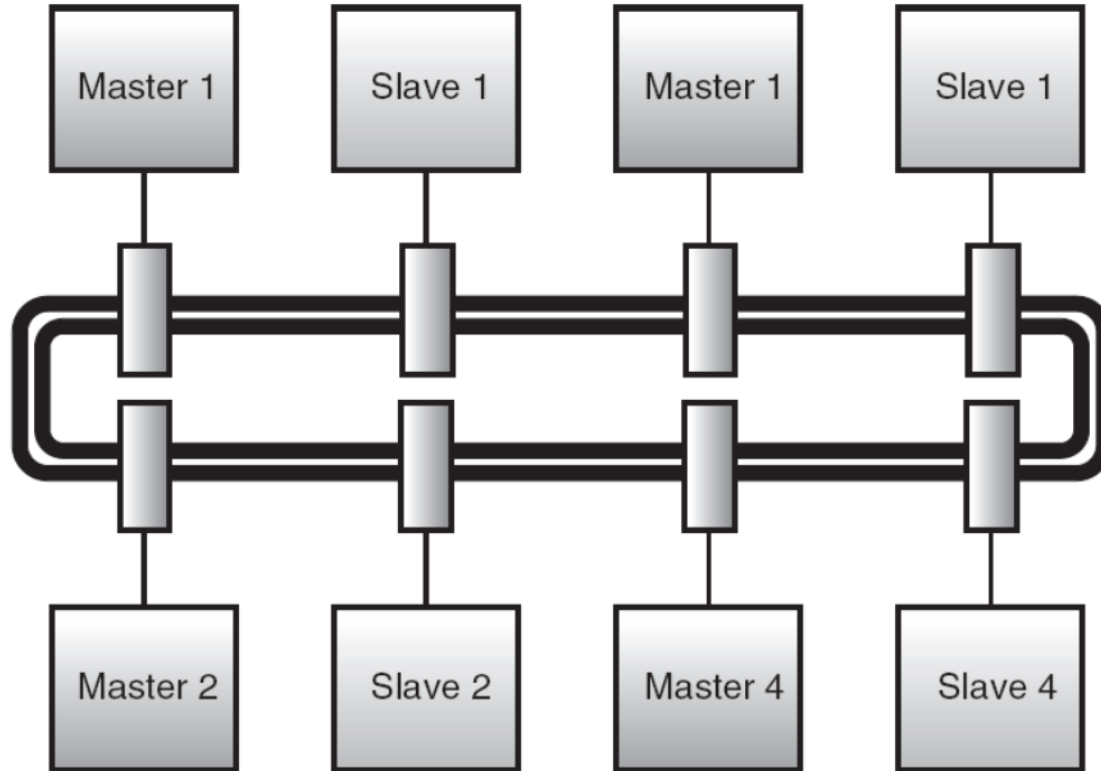


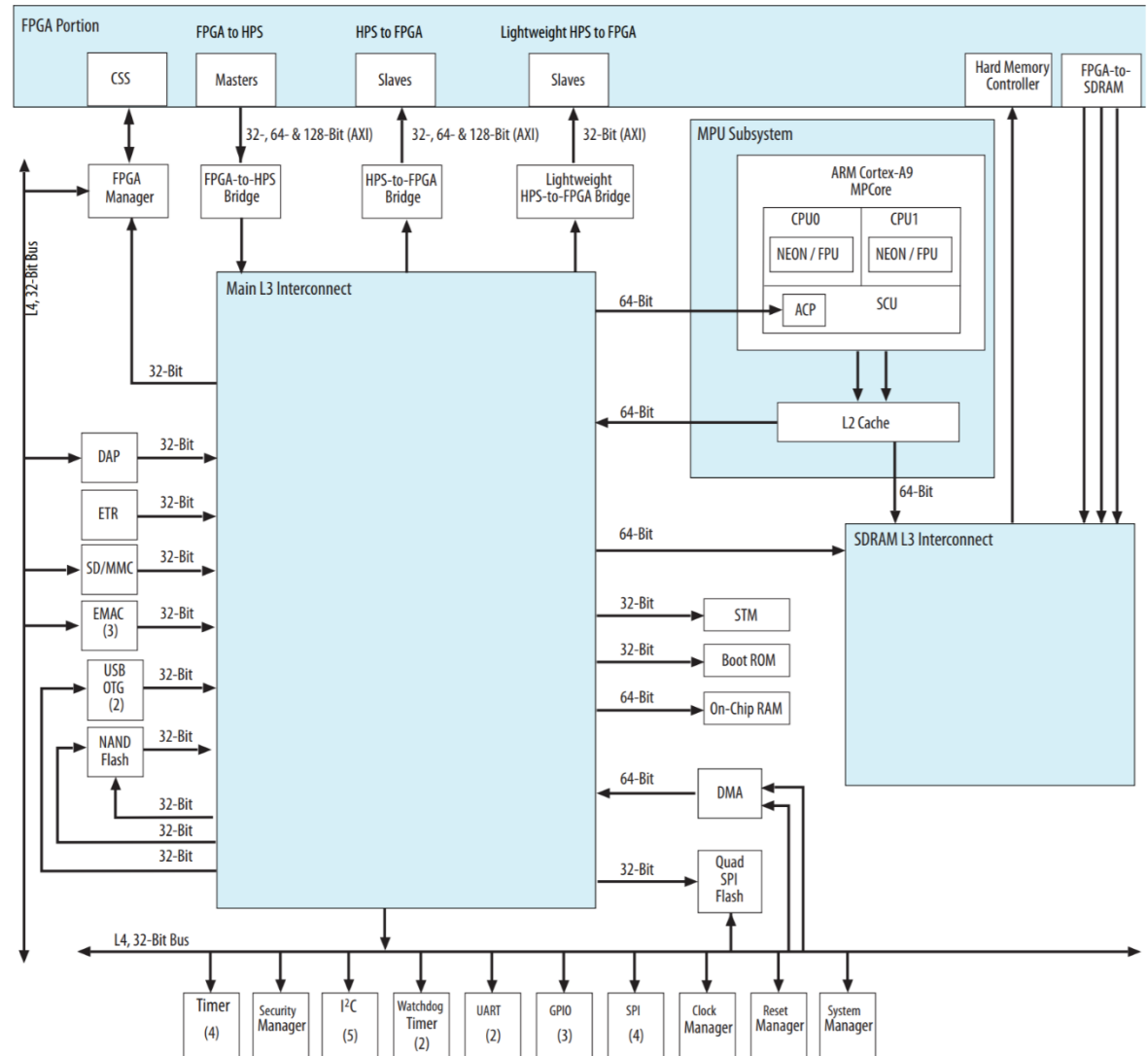
FIGURE 2.9(f)

Ring bus

SoC/SoPC rendszer példák

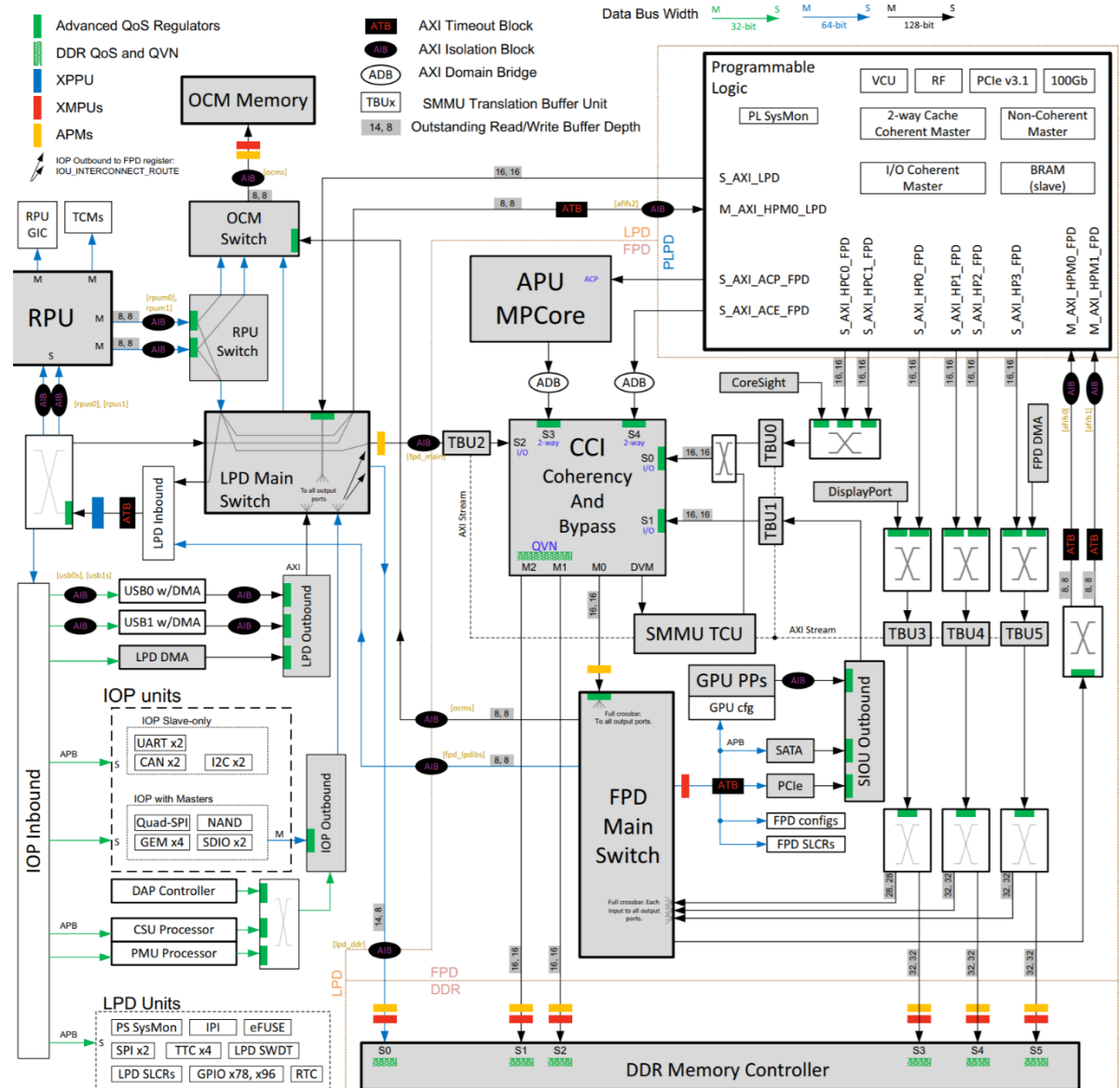
Intel Arria 10 SoC

- Arteris FlexNoC Interconnect

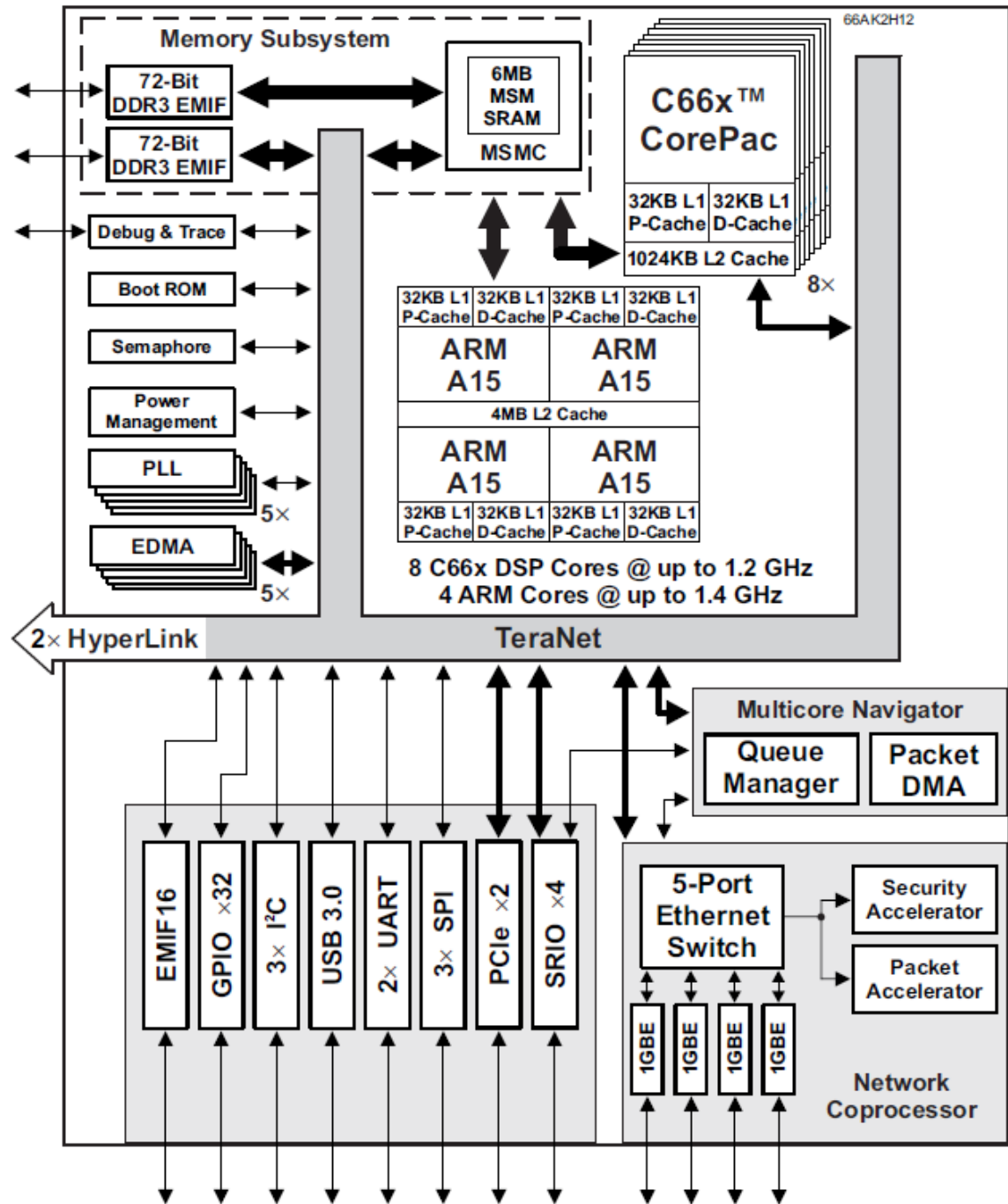


Xilinx Ultrascale MPSoC

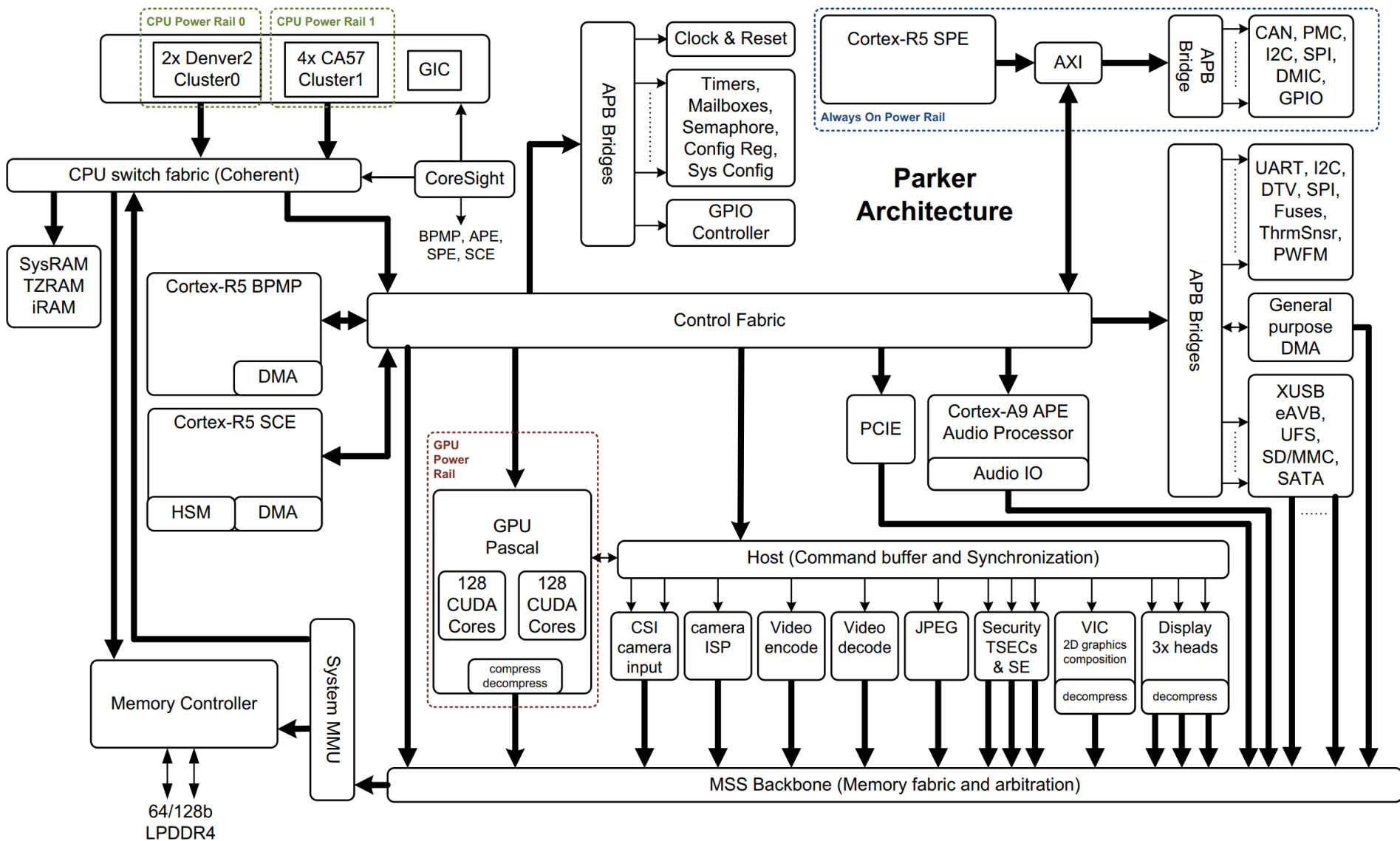
- ARM CCI-4 Interconnect



TI Key Stone II



NVIDIA Parker (Tegra TX2)



ARM AMBA buszrendszer

- Hosszú idejű evolúciós fejlődés
 - AHB/ASB/APB, AXI/AXIS, ACE, ATB, CHI
- Teljeskörű megoldás, bármely igényre
 - Egyszerű perifériabusz – többprocesszoros cache koherens kapcsolat
- Valódi „ipari” szabvány, széles körű használat
- Megfelelő tervezői, verifikációs és tesztelési támogatás

ARM AMBA 2.0 busrendszer

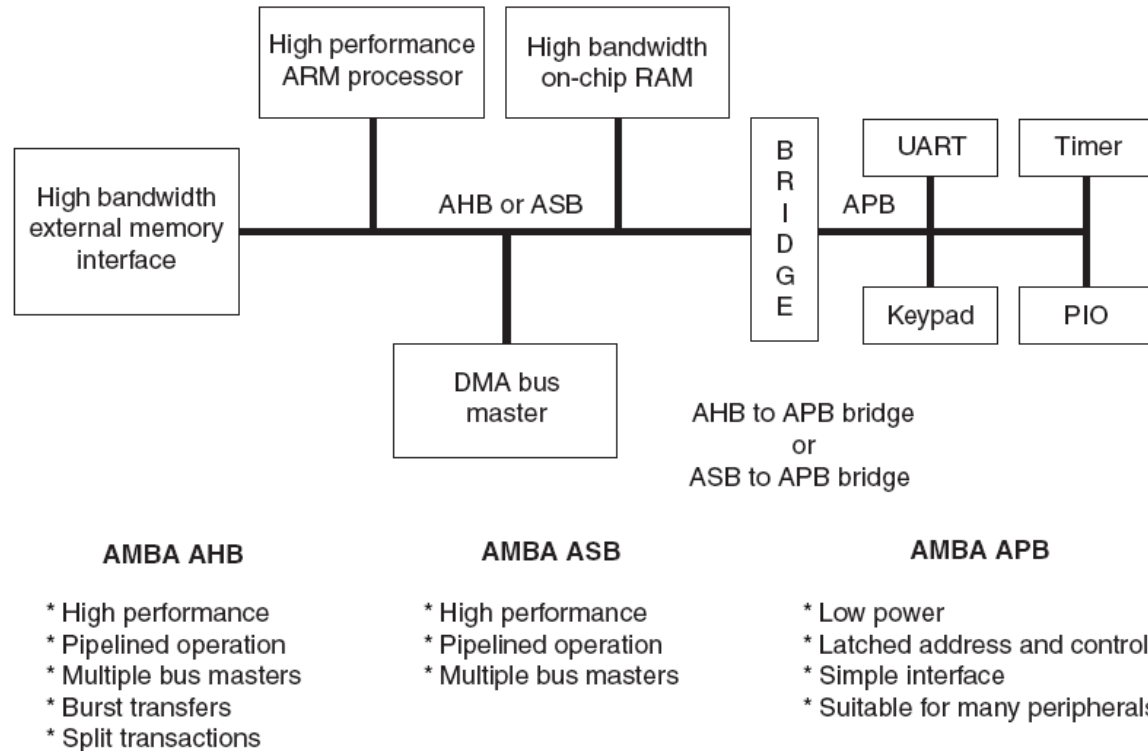


FIGURE 3.1

A typical AMBA 2.0 system [1]

Source: ARM Inc.

AHB topológia

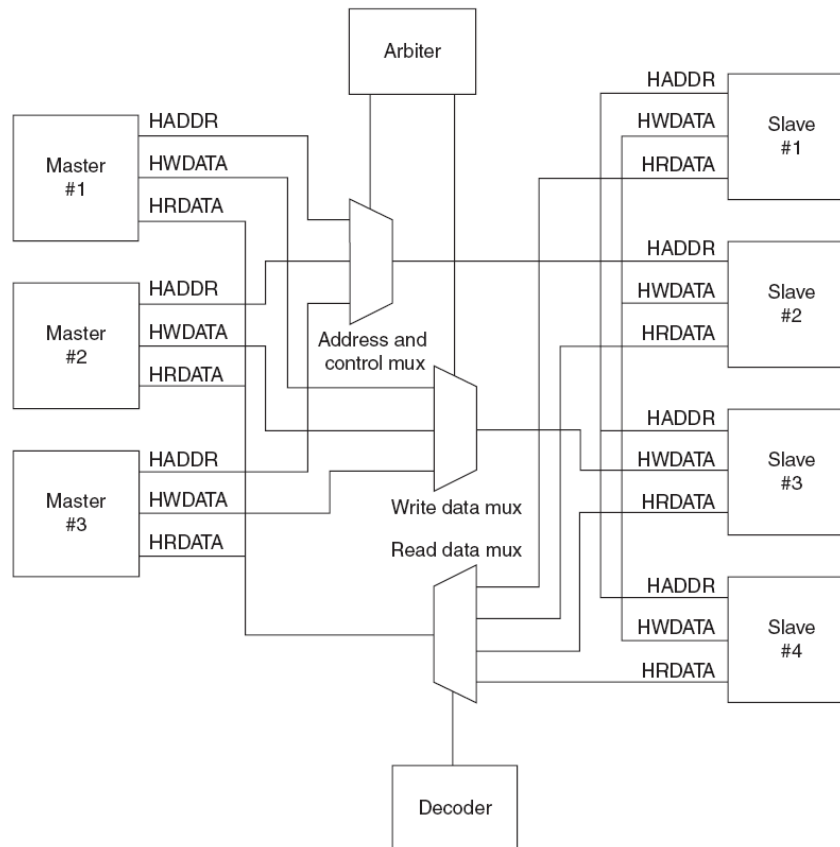


FIGURE 3.2

AHB multiplexer interconnection scheme [1]

Source: ARM Inc.

AHB bus adatátvitel

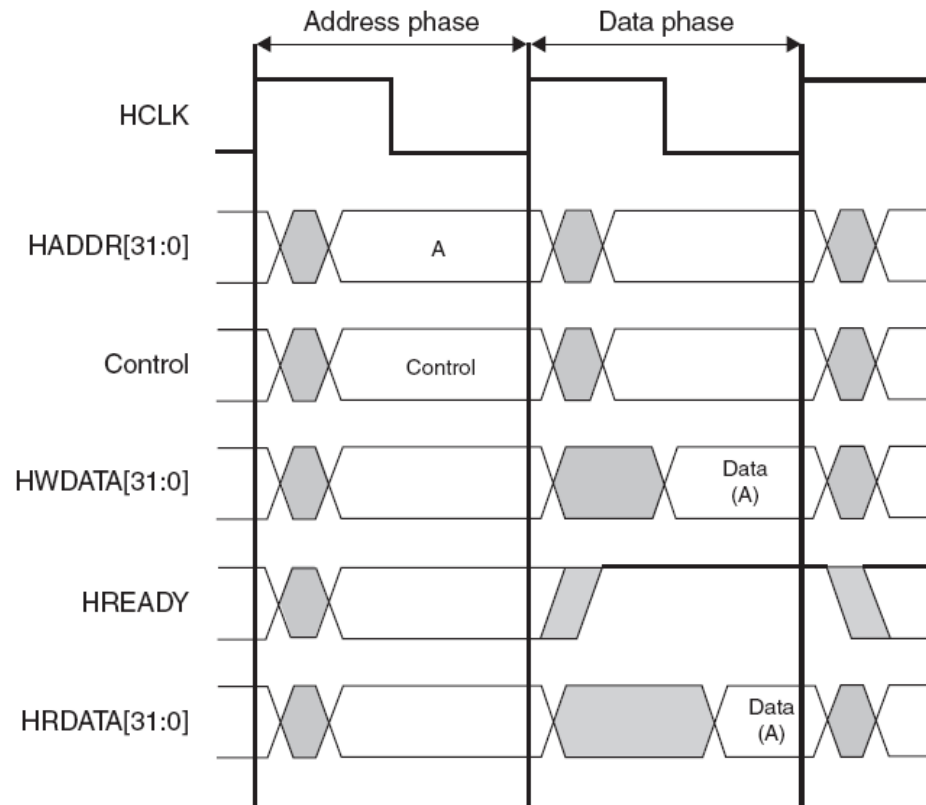


FIGURE 3.3

Basic data transfer on AHB bus [1]

Source: ARM Inc.

AHB bus adatátvitel várakozással

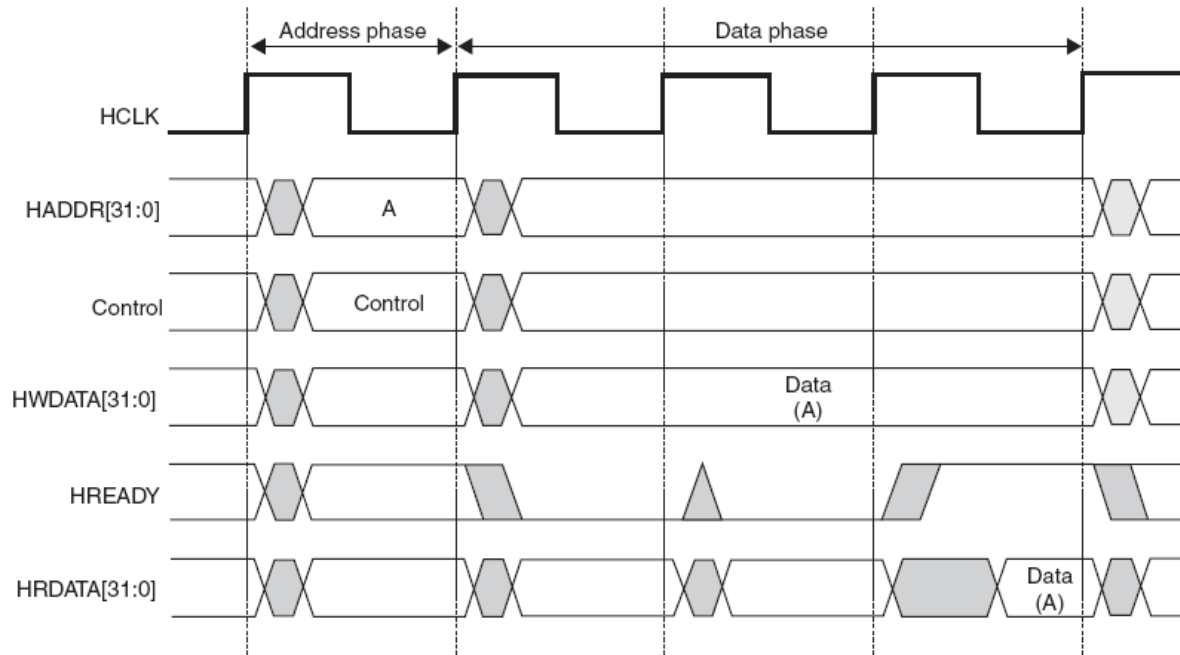


FIGURE 3.4

Basic data transfer on AHB bus with slave wait states [1]

Source: ARM Inc.

AHB busz adatátvitel- pipeline

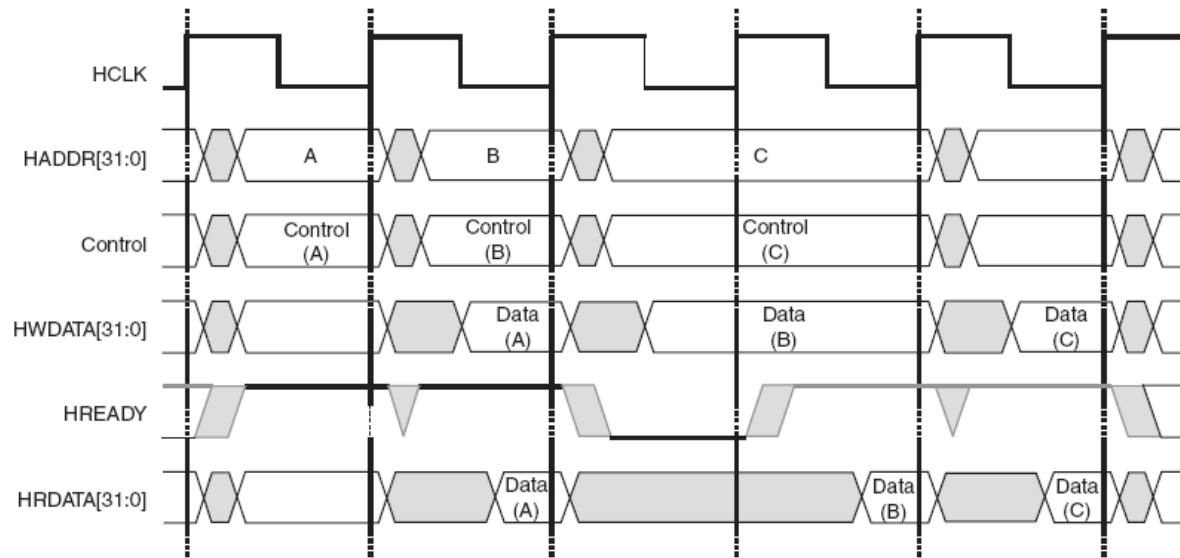


FIGURE 3.5

Pipelined data transfers on AHB bus [1]

Source: ARM Inc.

AHB vezérlőjelek (1)

- HSIZE: adatméret
 - \leq busz szélesség

HSIZE[2]	HSIZE[1]	HSIZE[0]	Size (bits)	Description
0	0	0	8	Byte
0	0	1	16	Halfword
0	1	0	32	Word
0	1	1	64	Doubleword
1	0	0	128	4-word line
1	0	1	256	8-word line
1	1	0	512	-
1	1	1	1024	-

AHB vezérlőjelek (2)

- HTRANS[1:0]
 - IDLE
 - Nincs átvitel
 - BUSY
 - Master várakoztatás
 - NONSEQ
 - Egyszerű átvitel
 - Burst első átvitele
 - SEQ
 - Burst
- HBURST[2:0]
 - SINGLE
 - INCR
 - WRAP4
 - INCR4
 - WRAP8
 - INCR8
 - WRAP16
 - INCR16
 - Wrap: $H\text{SIZE} \times \text{wrap}$ méret byte határon
 - 1Kbyte address limit

AHB ciklusok

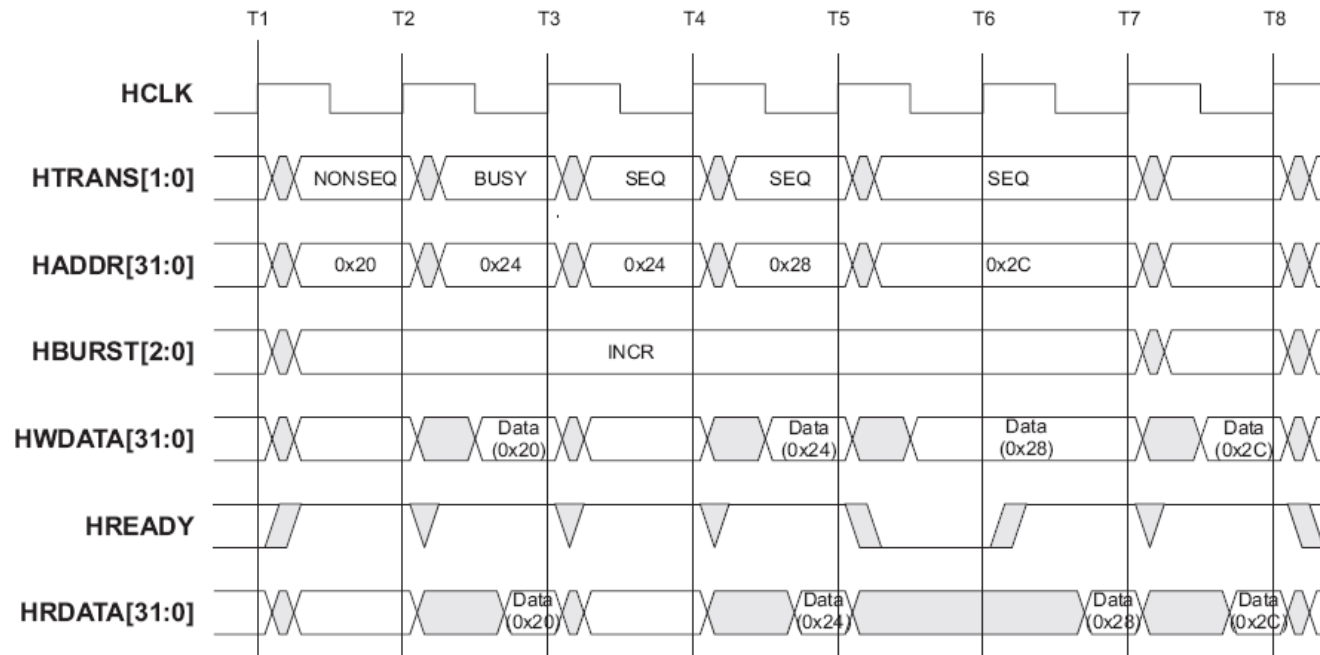


Figure 3-6 Transfer type examples

AHB INCR8 burst

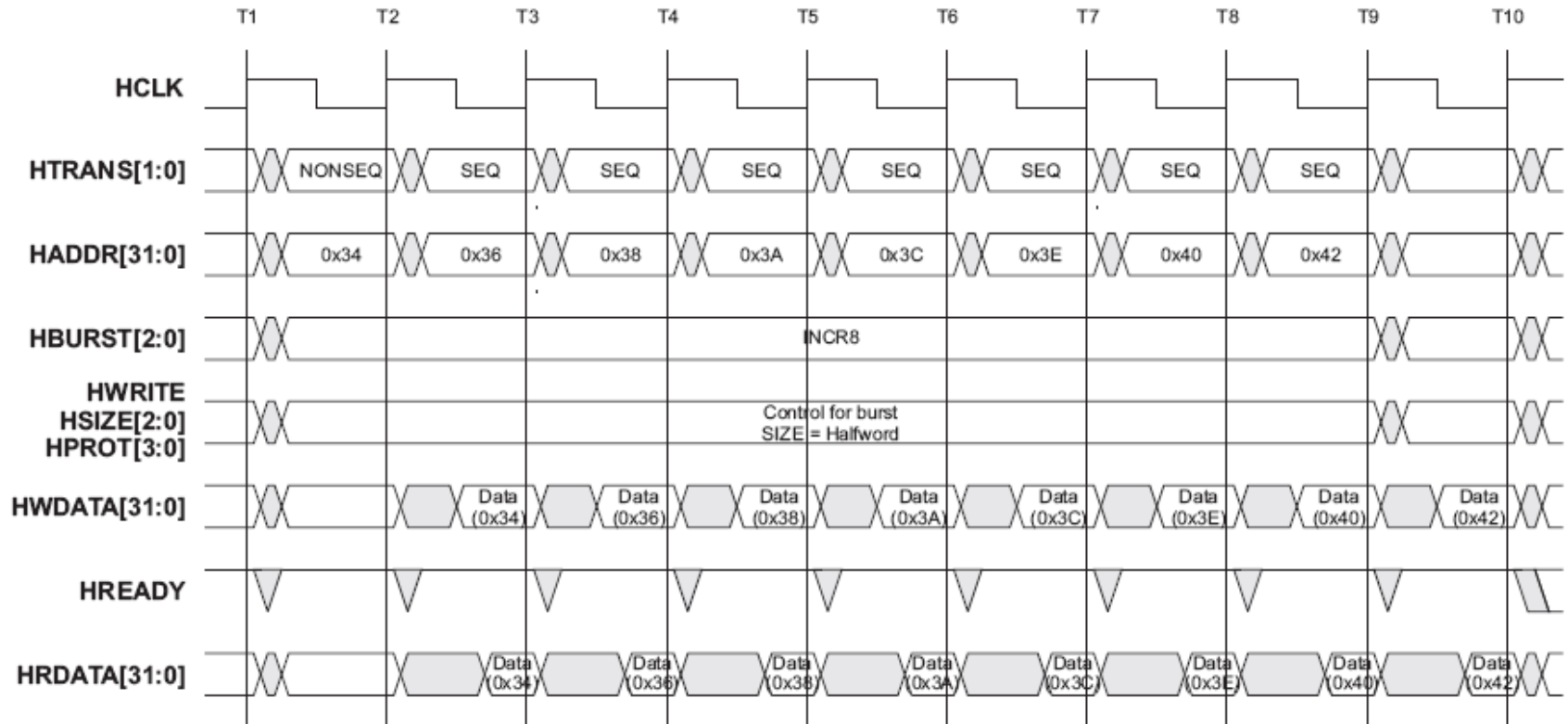


Figure 3-10 Eight-beat incrementing burst

AHB WRAP8 burst

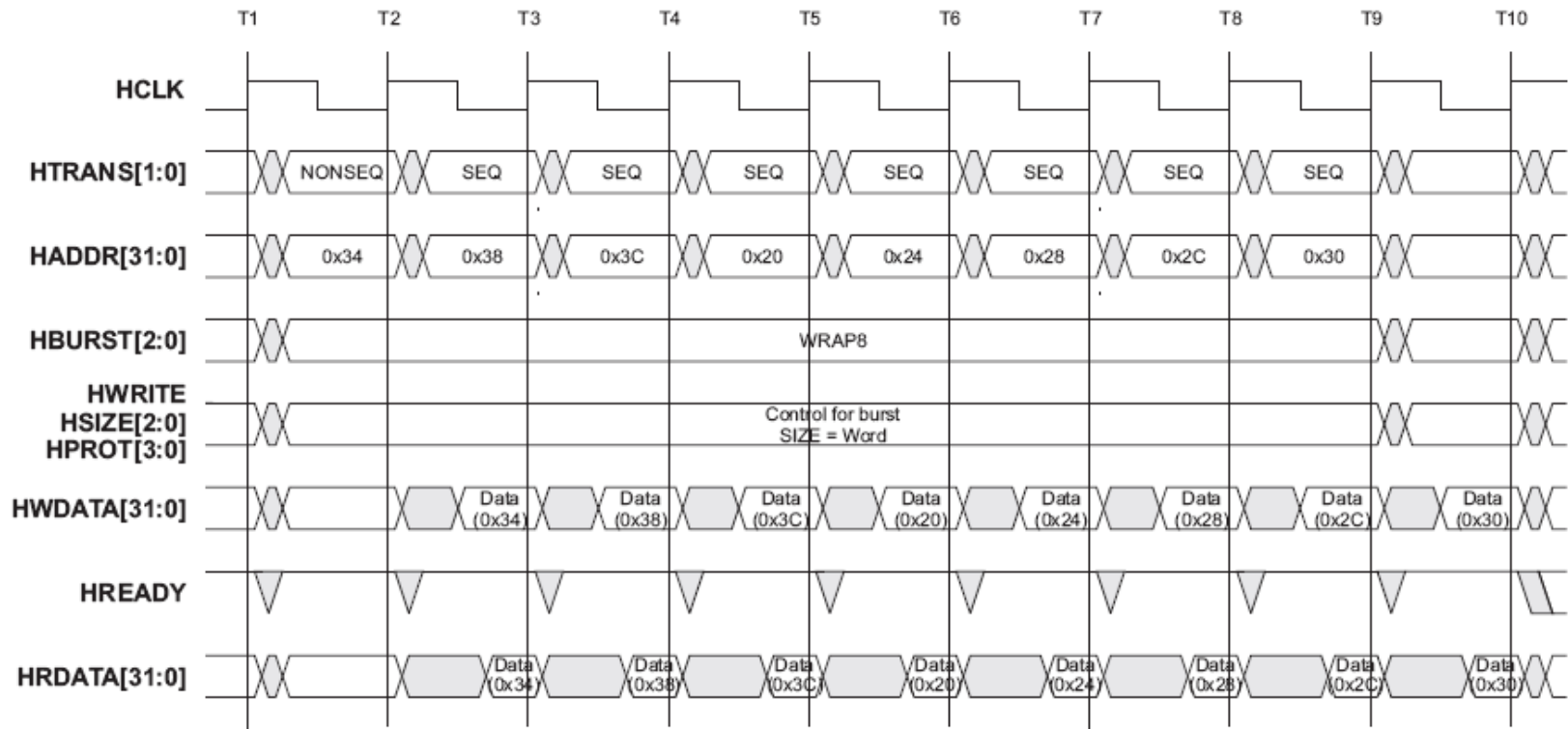


Figure 3-9 Eight-beat wrapping burst

AHB WRAP4 burst példa

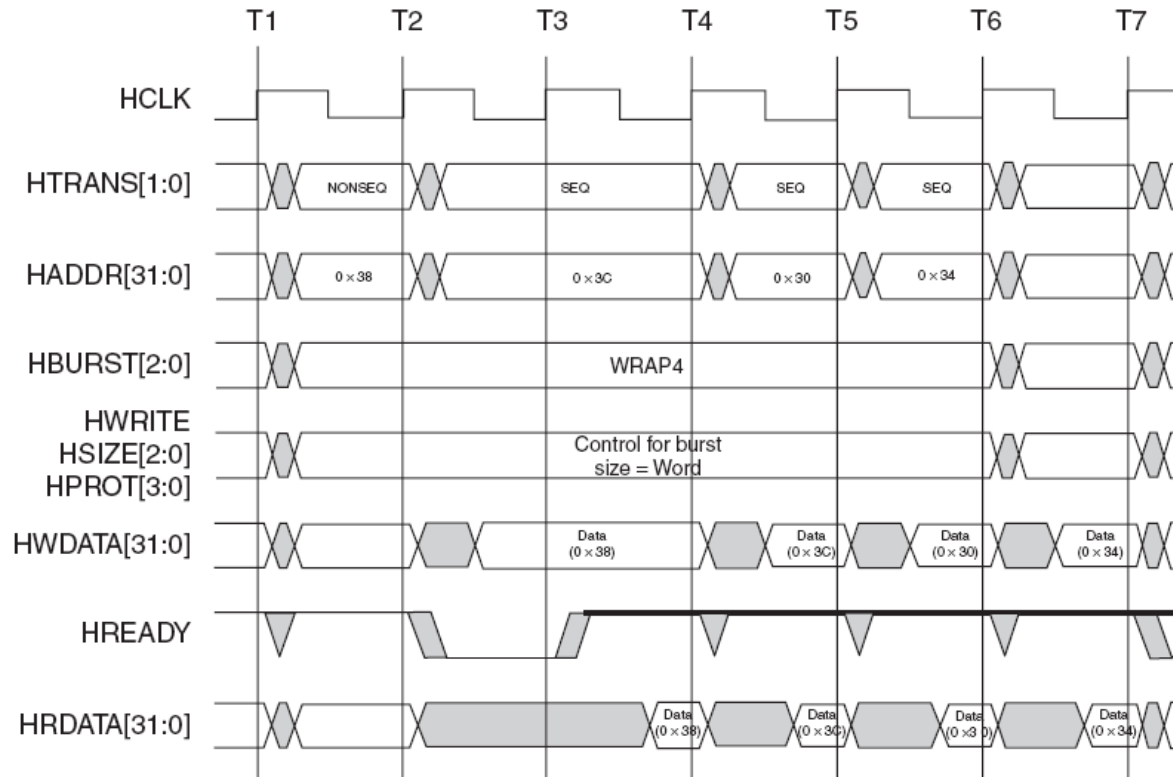


FIGURE 3.9

Example of a burst transfer—a wrapping burst of length 4 on the AHB bus [1]

Source: ARM Inc.

AHB nem-fix burst

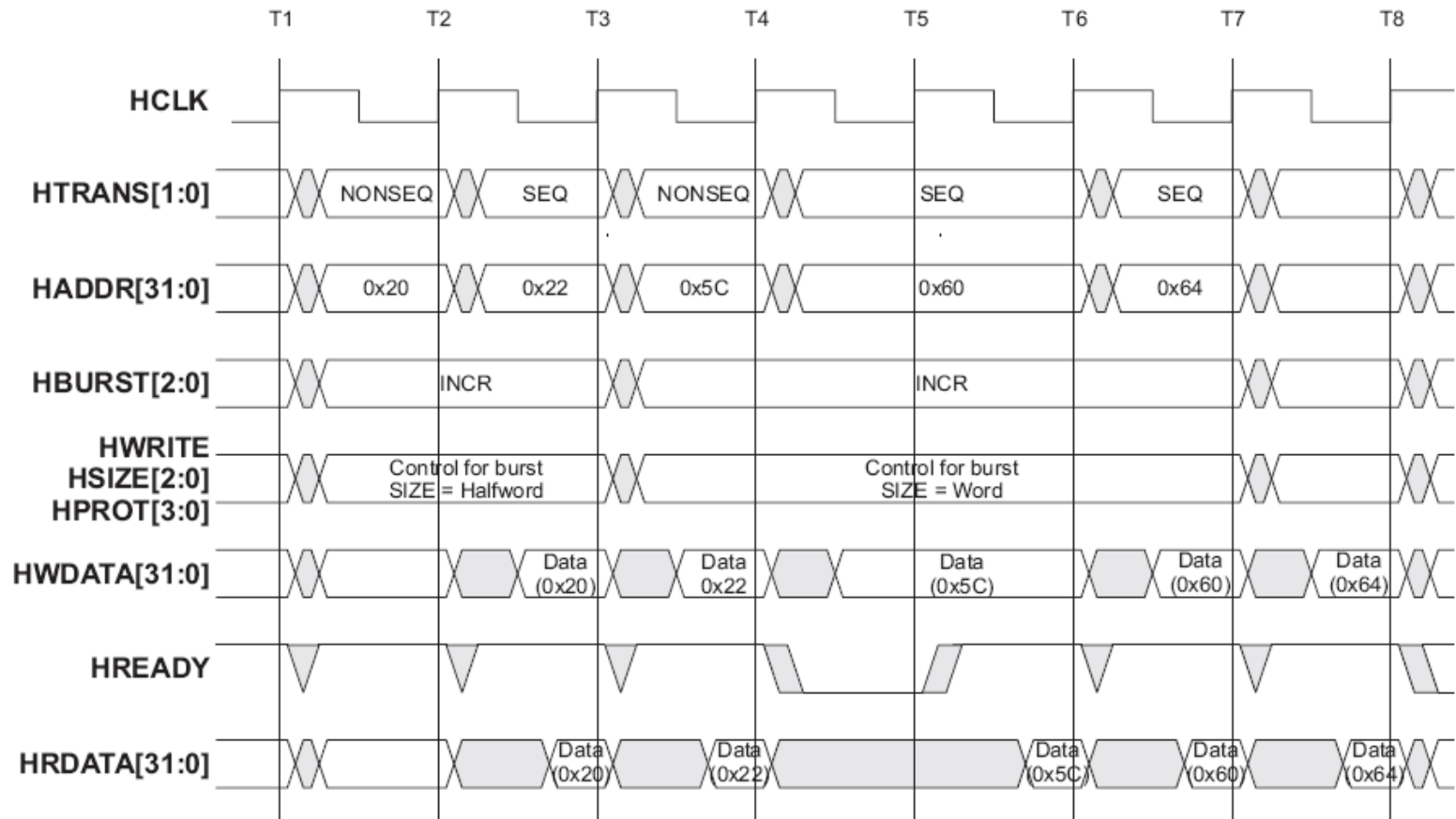


Figure 3-11 Undefined-length bursts

AHB burst szabályok

- A burst mindig egy 1K bájt tartományon belül marad! (MASTER felelőssége!)
- Adatmennyiség = Adatméret * Ütemek száma
- INCR használható SINGLE helyett
- Minden átvitel méret szerinti címhatárra illesztett
- Korai BURST lezárás (NONSEQ vagy IDLE)

AHB split adatátvitel

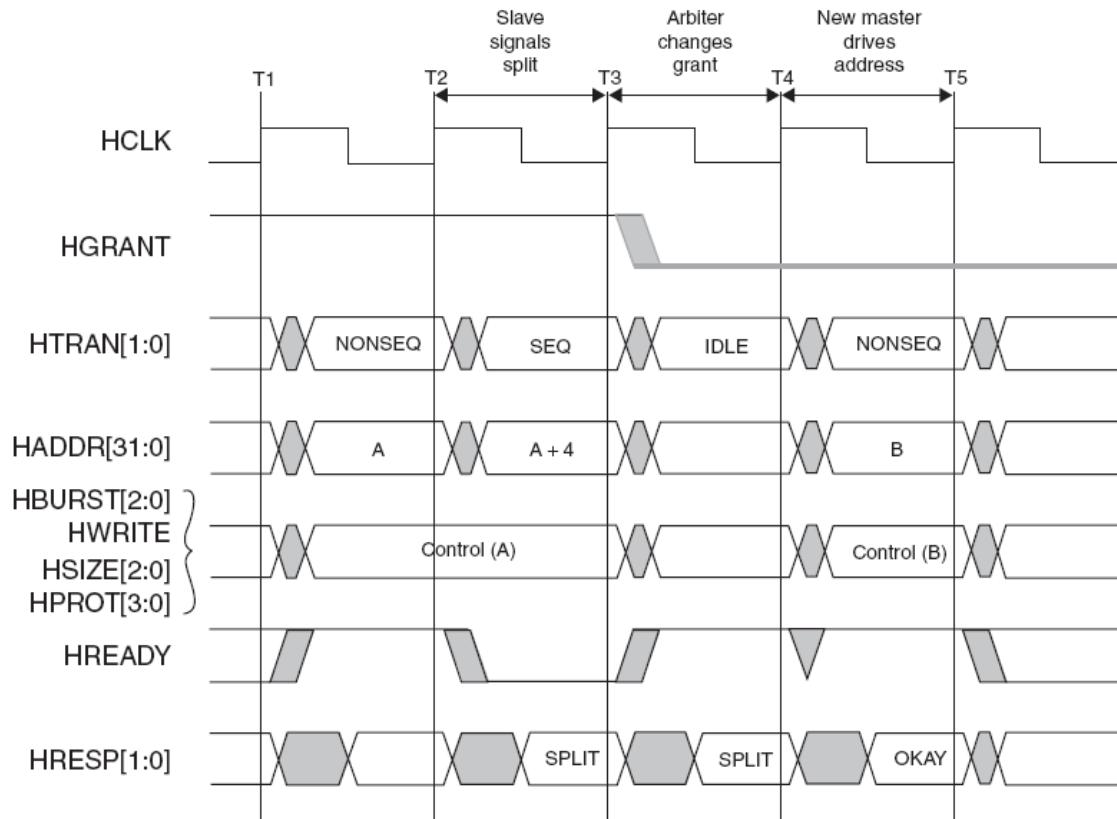


FIGURE 3.10

SPLIT transfer on the AHB bus [1]

Source: ARM Inc.

AHB arbitráció

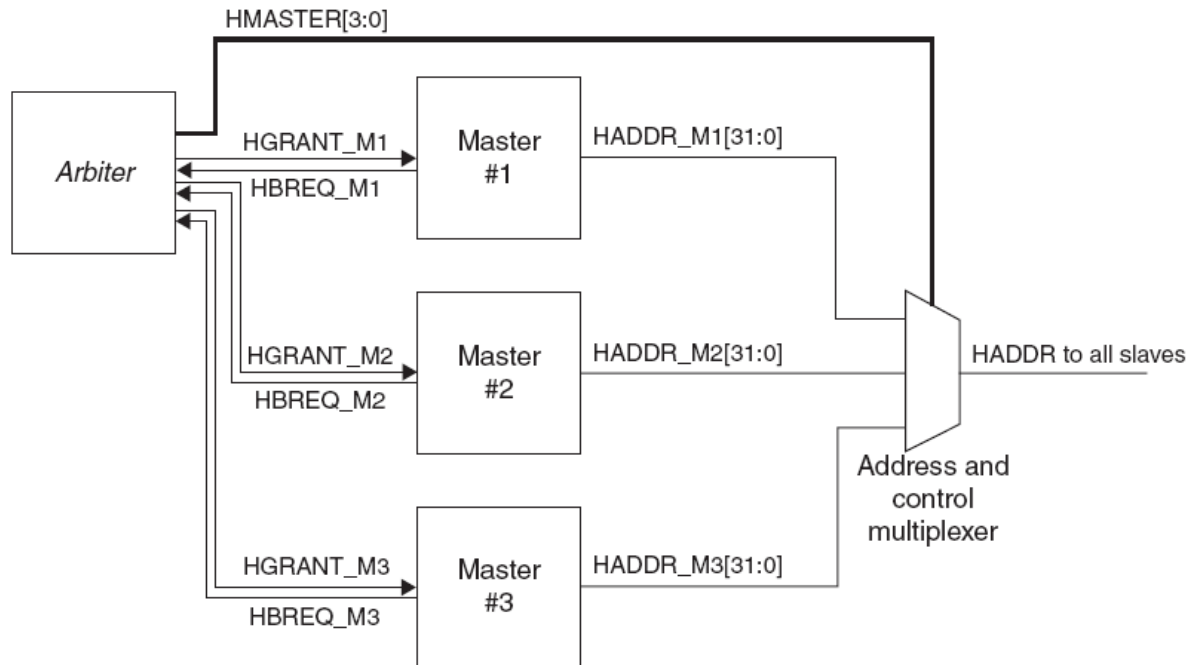


FIGURE 3.6

Arbitration on AHB bus [1]

Source: ARM Inc.

AHB arbitráció folyamata

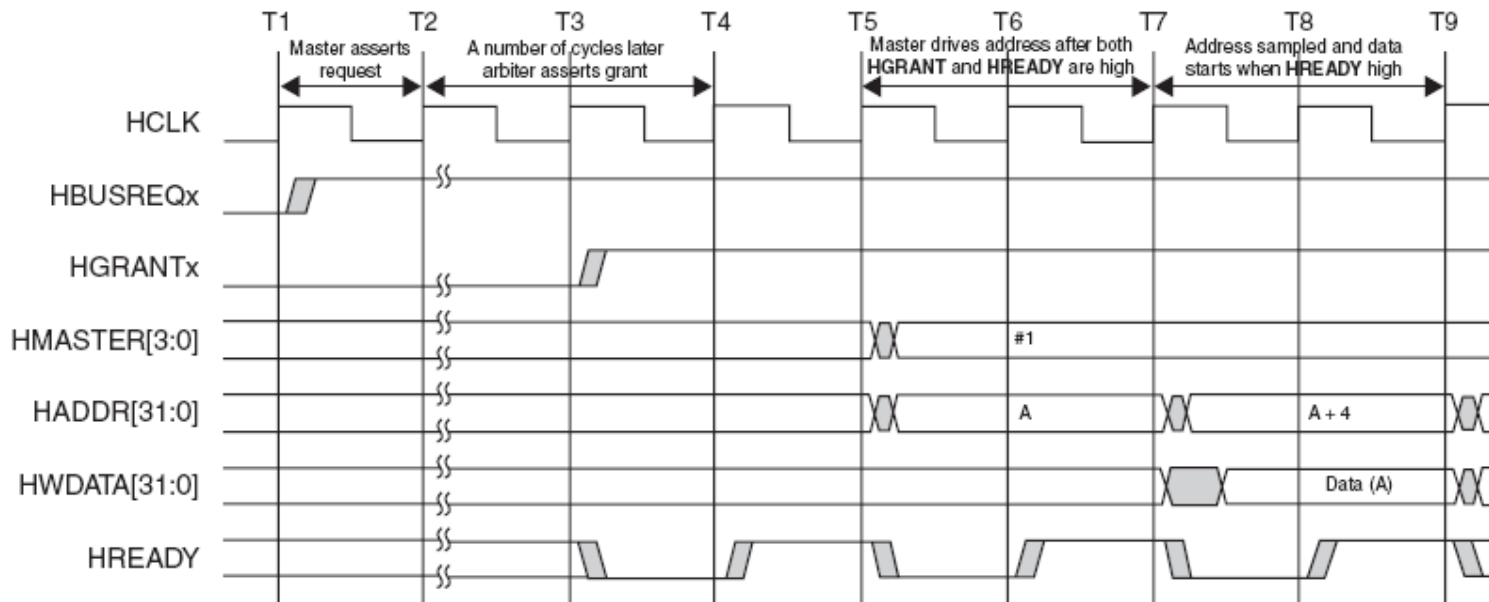


FIGURE 3.7

Cost of arbitration on AHB bus [1]

Source: ARM Inc.

Komplex AHB rendszerek

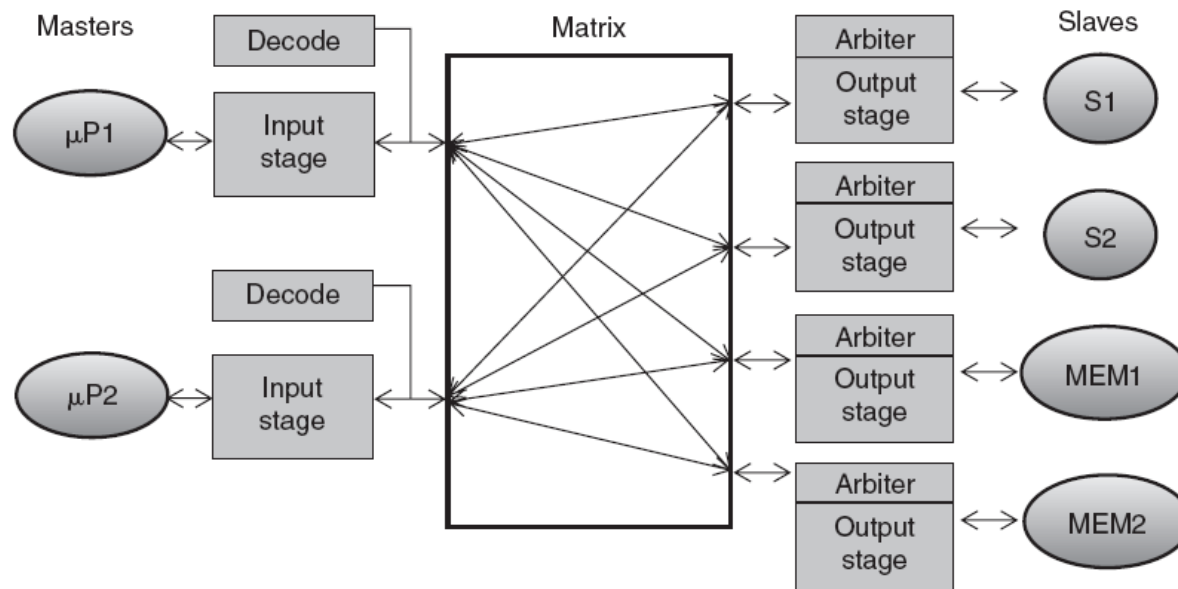


FIGURE 3.11(a)

An example of a 2 master, 4 slave AHB: full bus matrix topology

Komplex AHB rendszer 2.

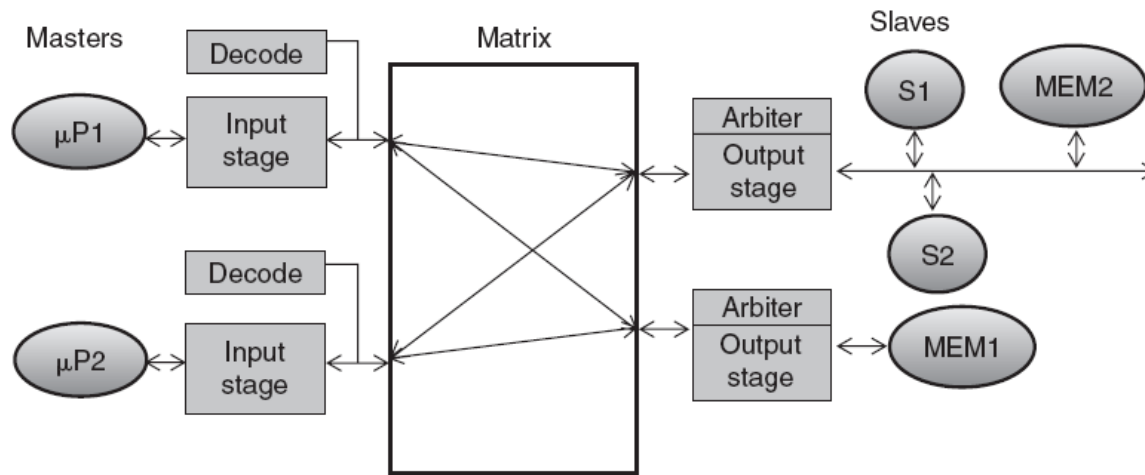


FIGURE 3.11(b)

Partial bus matrix topology

APB busz

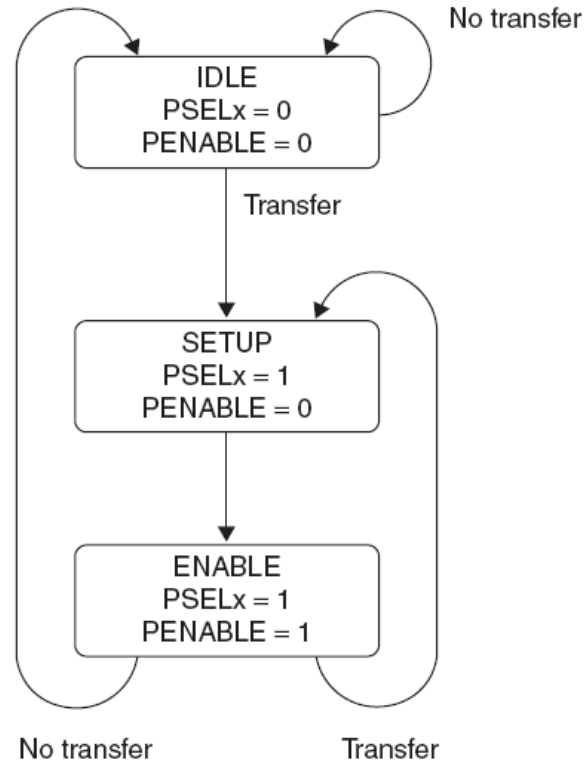


FIGURE 3.12

State diagram representing activity of the APB bus [1]

Source: ARM Inc.

APB olvasás buszciklus

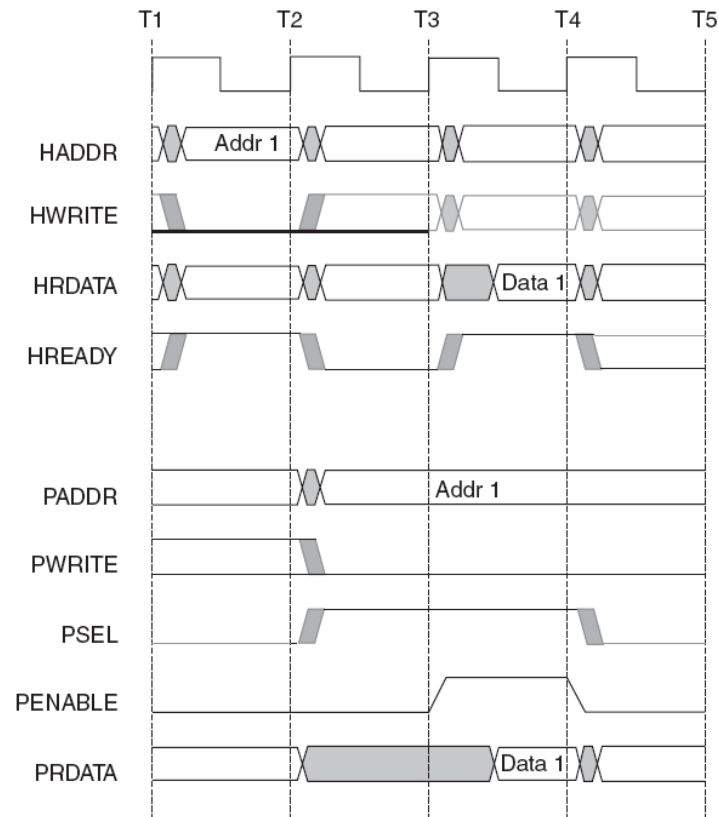


FIGURE 3.13

Read data request from the AHB to the APB bus [1]

Source: ARM Inc.

APB írás buszciklus

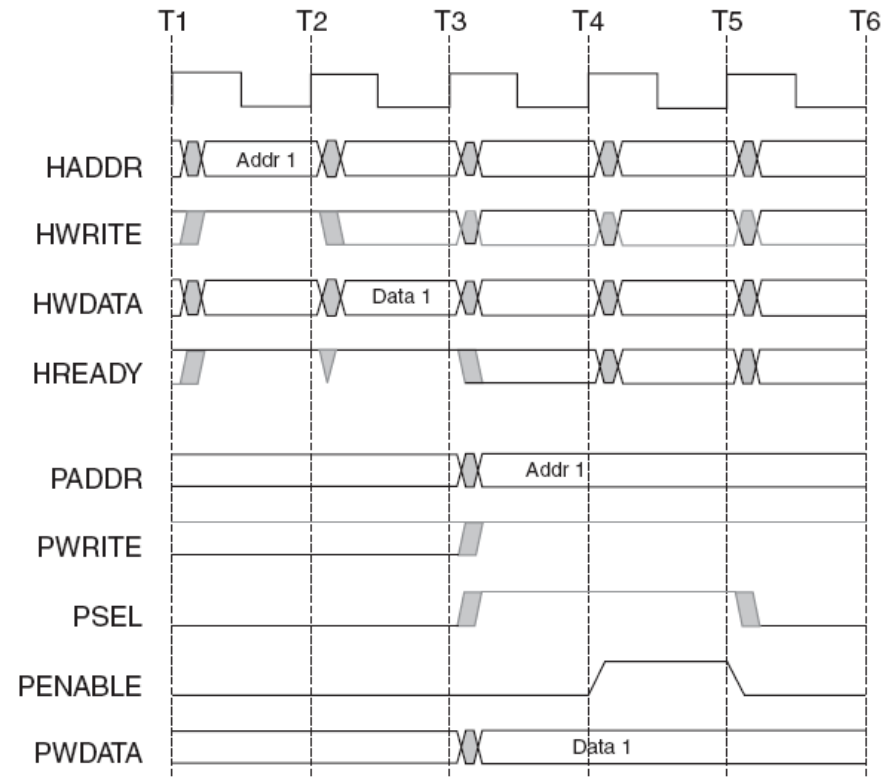
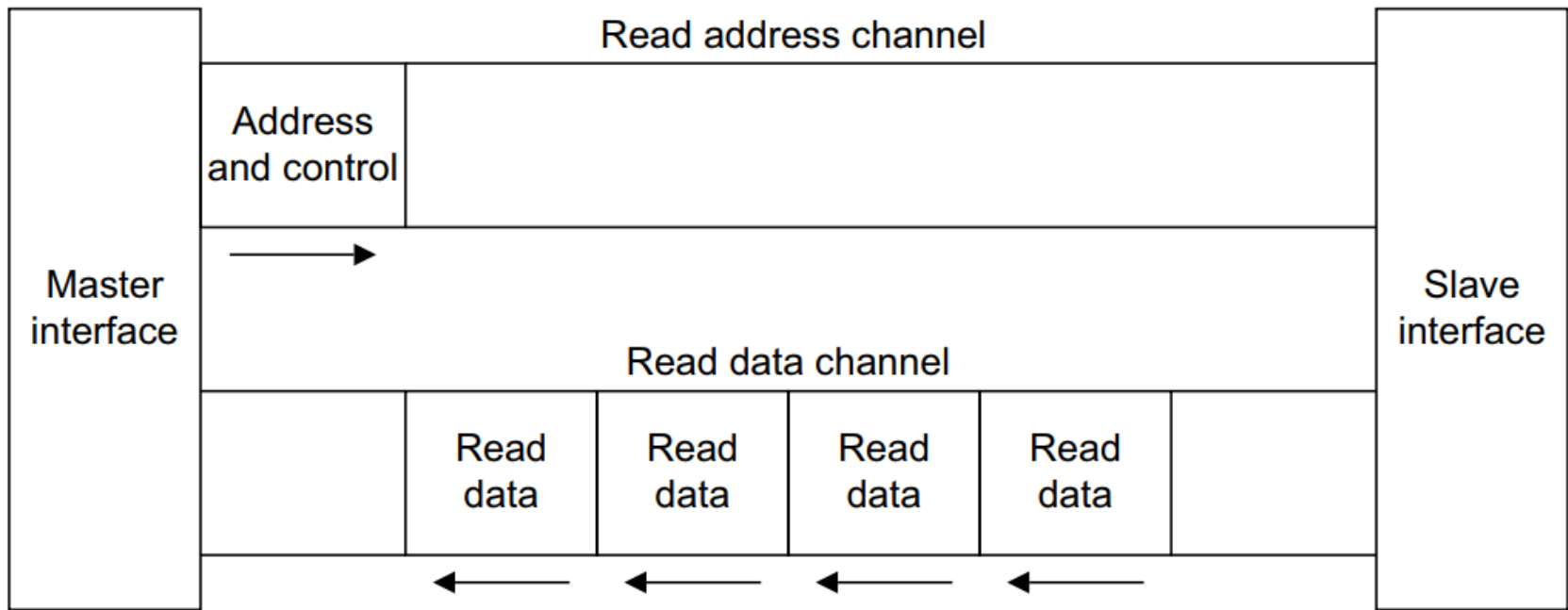


FIGURE 3.14

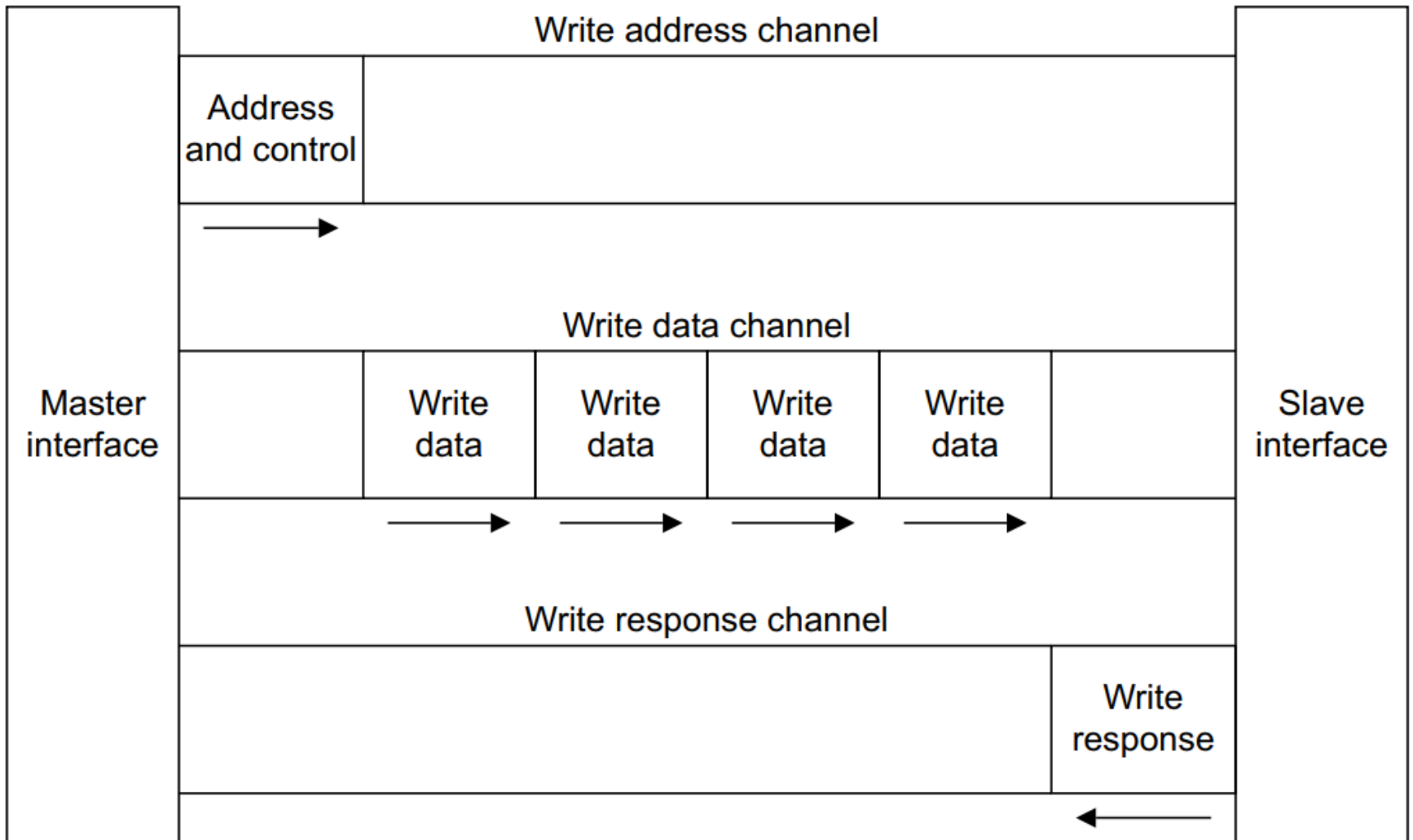
Write data request from the AHB to the APB bus [1]

Source: ARM Inc.

AMBA 3 – AXI Read Channels



AMBA 3 – AXI Write Channels



AXI – AHB

Table 3.1 Contrasting features of AXI and AHB

AMBA 3.0 AXI	AMBA 2.0 AHB
Channel-based specification, with five separate channels for read address, read data, write address, write data, and write response enabling flexibility in implementation.	Explicit bus-based specification, with single shared address bus and separate read and write data buses.
Burst mode requires transmitting address of only first data item on the bus.	Requires transmitting address of every data item transmitted on the bus.
OO transaction completion provides native support for multiple, outstanding transactions.	Simpler SPLIT transaction scheme provides limited and rudimentary outstanding transaction completion.
Fixed burst mode for memory mapped I/O peripherals.	No fixed burst mode.
Exclusive data access (semaphore operation) support.	No exclusive access support.
Advanced security and cache hint support.	Simple protection and cache hint support.
Register slice support for timing isolation.	No inherent support for timing isolation.
Native low-power clock control interface.	No low-power interface.
Default bus matrix topology support.	Default hierarchical bus topology support.

AXI tulajdonságok



FIGURE 3.16(a)

Burst addressing modes for AMBA 2.0 AHB bus

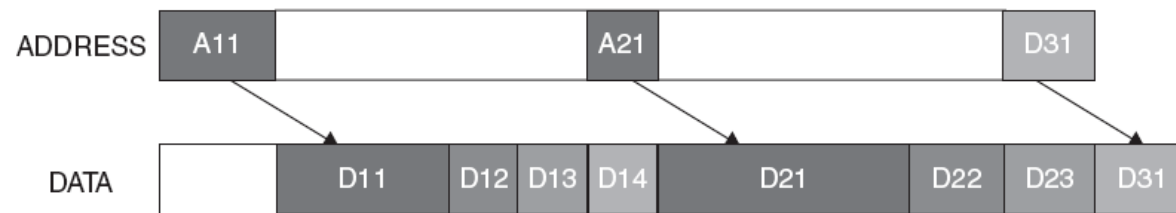


FIGURE 3.16(b)

AMBA 3.0 AXI bus

AXI tulajdonságok

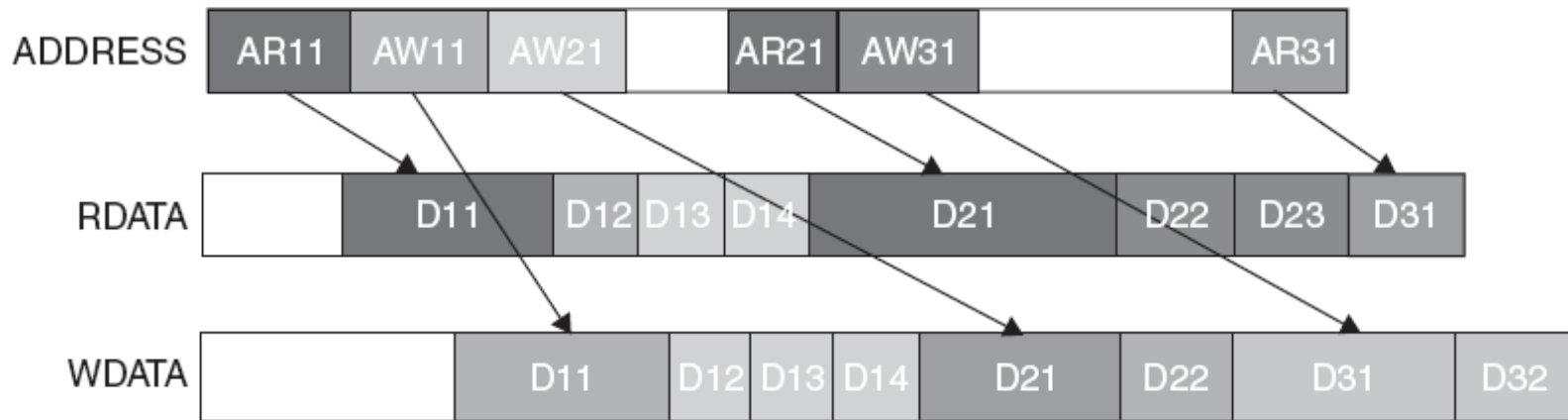


FIGURE 3.17

Better utilization of data buses in AXI

AXI tulajdonságok

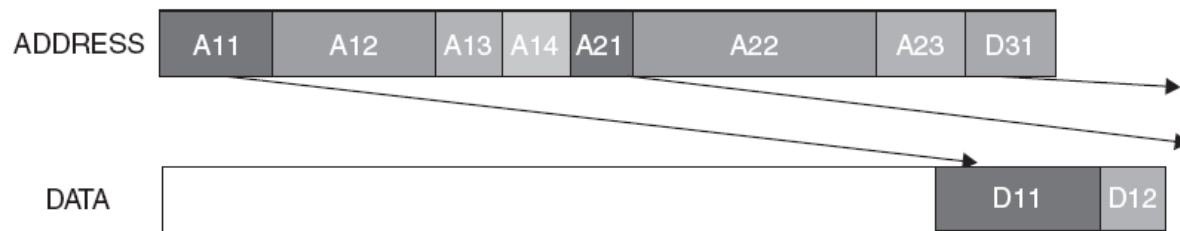


FIGURE 3.18(a)

Transaction sequence for access to a slow slave on AHB bus

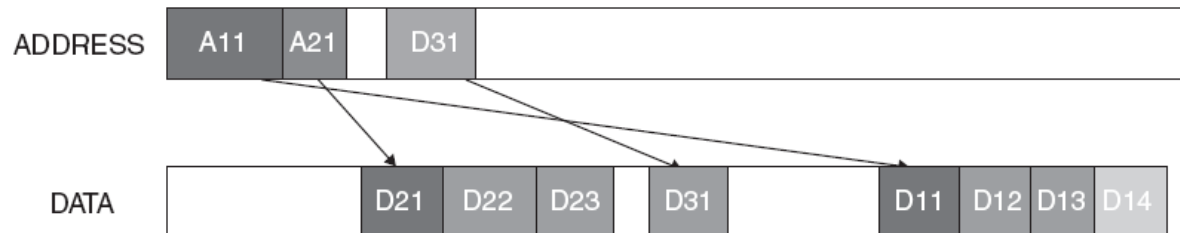


FIGURE 3.18(b)

AXI bus

IBM CoreConnect

- Általános célú SoC buszrendszer
- Három szintű
 - PLB Processor Local Bus
 - OPB On-Chip Peripheral Bus
 - DCR Device Control Register (bus)
- Szolgáltatásai, komplexitása az AMBA-hoz hasonló (2018 nem tárgyaljuk)

IBM CoreConnect

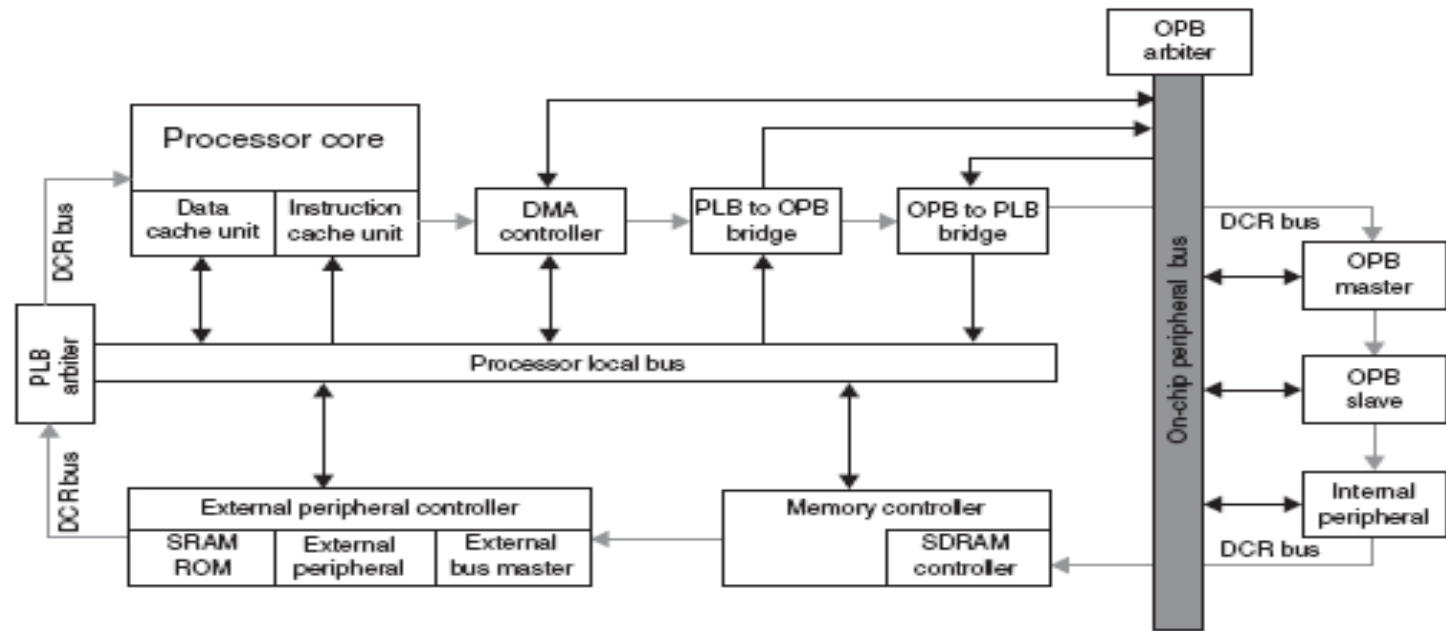


FIGURE 3.19

An example of CoreConnect-based SoC design [3]

Reprint Courtesy of International Business Machines Corporation copyright (2001) © International Business Machines Corporation

IBM CoreConnect – Xilinx EDK

- Xilinx EDK rendszerhez ajánlott buszrendszer
- Választás oka:
 - Önmagában sem rossz, de..
 - Virtex-IIPro sorozat: Beépített Power PC 405 PLB interfésszel kiegészítve
 - Sok létező IP elérhető (nem feltétlenül FPGA-ra, de könnyen portolhatók)
- Bemutató a Xilinx EDK fóliákról

Xilinx EDK – ARM AXI

- A 7-sorozat óta új stratégia
- A váltás oka:
 - A Zynq eszközben dual ARM9-PS
 - Az AXI rendszer szélesebb körben elterjedt
 - Sokka több újonnan fejlesztett és már létező IP elérhető (nem feltétlenül FPGA-ra, de könnyen portolhatók)
- Jó lépés, nekünk kedvező

További SoC buszok

- OpenCores WishBone
 - Egyszintű, nagysebességű szinkron kapcsolat
 - Közepes, vagy kis teljesítményű rendszerekhez
 - Nincs átlapolt működés, nincs megszakított átvitel
 - Multi master, 64 bit címzés, 8-64 bit adat
 - Egyedi vagy blokkos átvitelek
 - RMW ciklus a szemafor típusú műveletekhez
 - Ninc fix topológia, illetve többféle rendszer kialakítható

OpenCores Wishbone Pont-pont és adatfolyam

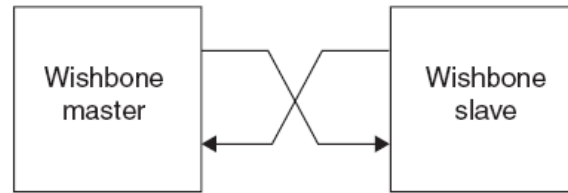


FIGURE 3.30(a)

Different topologies of the Wishbone bus architecture: point-to-point

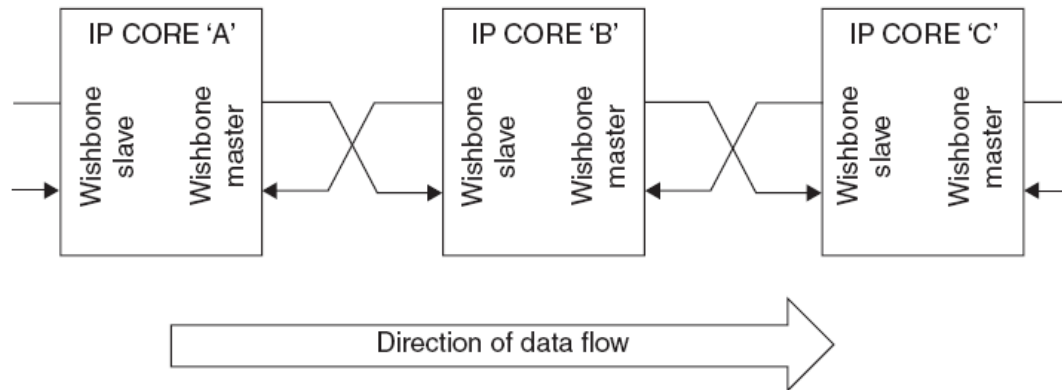


FIGURE 3.30(b)

Data flow

OpenCores Wishbone normál busz

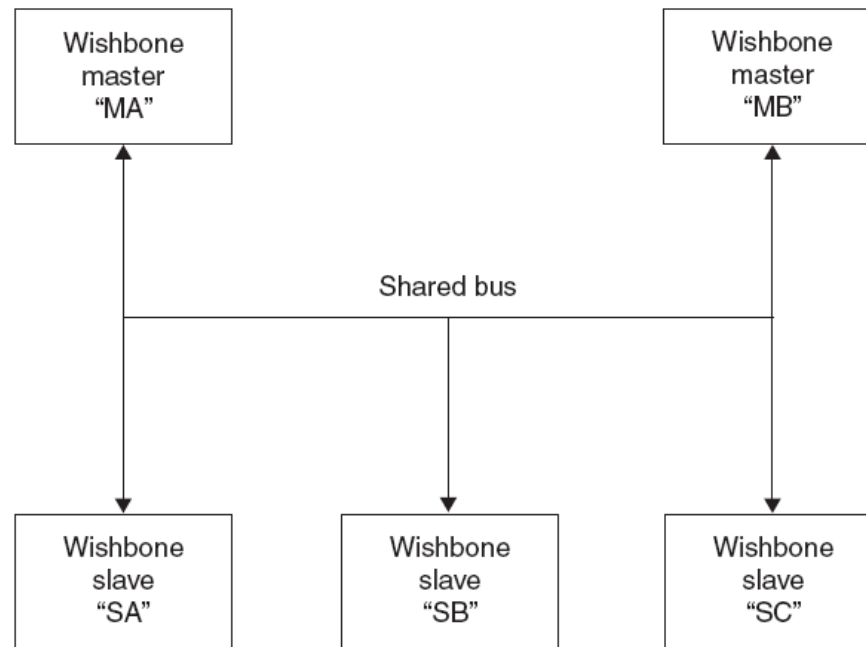


FIGURE 3.30(c)

Shared bus

OpenCores Wishbone hálózat

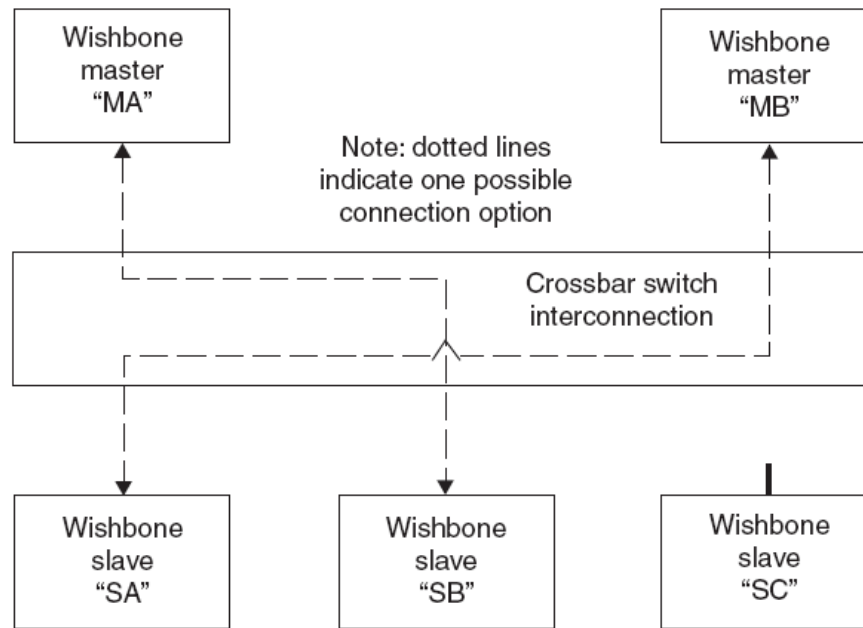


FIGURE 3.30(d)

Full or partial crossbar [6]

OpenCores Wishbone problémák

- Egyszerű, könnyen használható
- Flexibilis, az igényekhez igazítható
- Speciális tulajdonságok kezelése: TAGi jelek
- Nincs előírás az arbitrációra, a hibajelzésre

OpenCores Wishbone alapkapcsolat

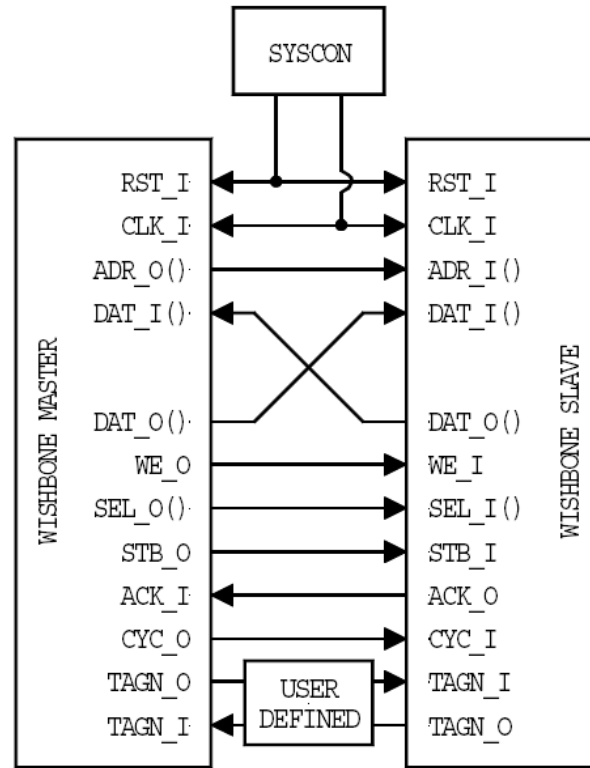


Figure 1-2. Standard connection for timing diagrams.

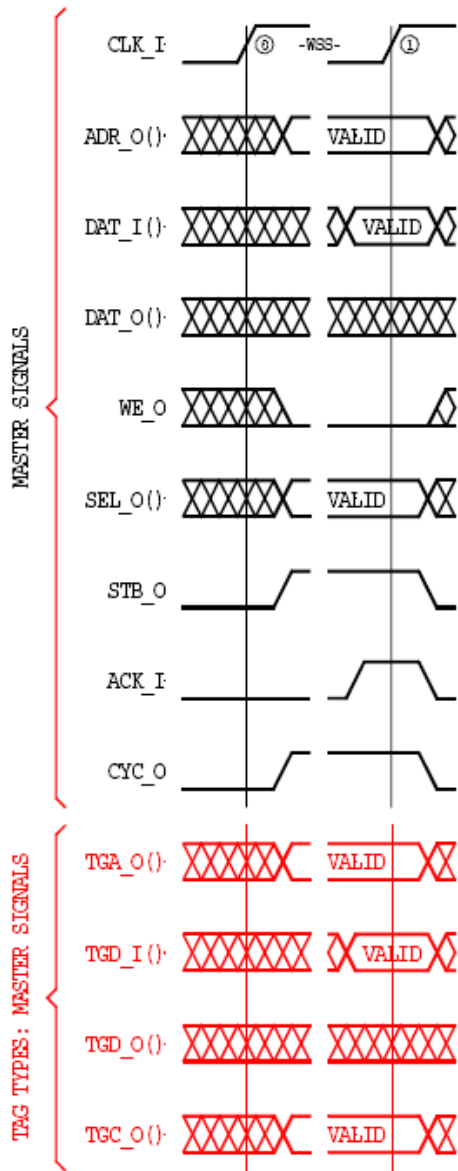


Figure 3-3. SINGLE READ cycle.

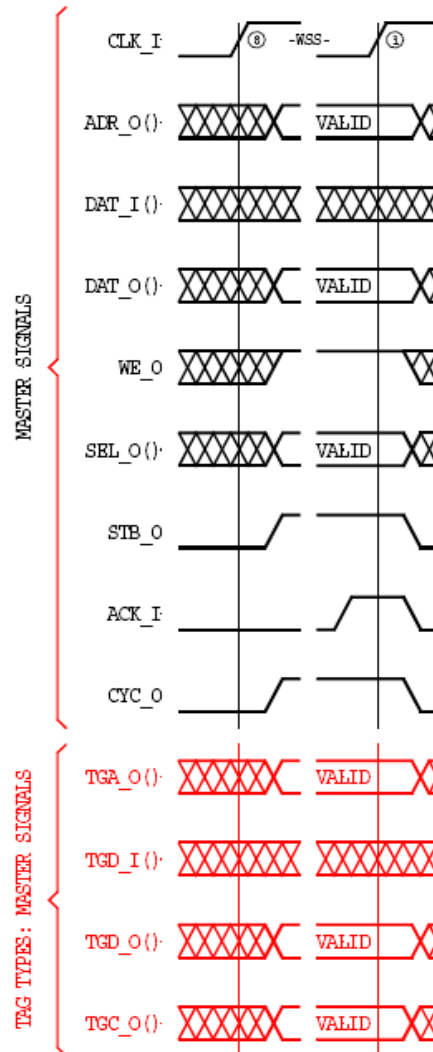


Figure 3-4. SINGLE WRITE cycle.

Wishbone busciklusok

Wishbone blokkos olvasás

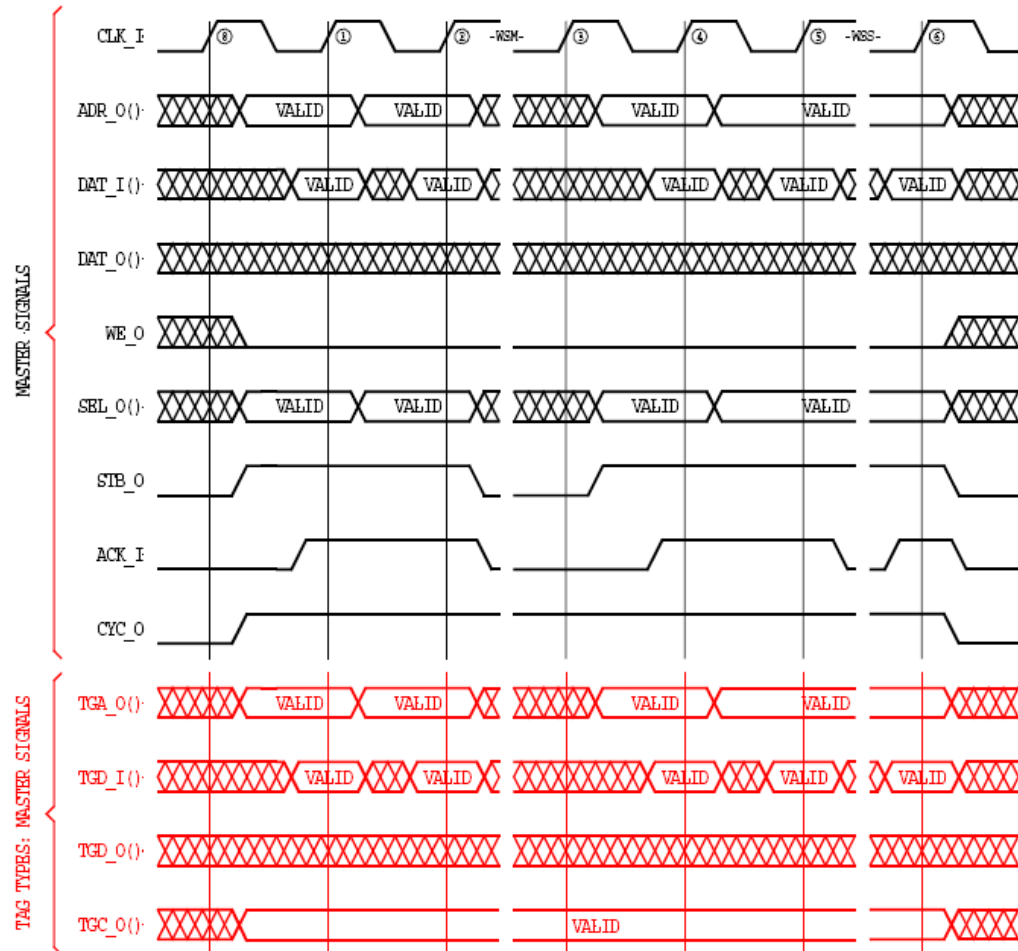


Figure 3-6. BLOCK READ cycle.

Wishbone blokkos írás

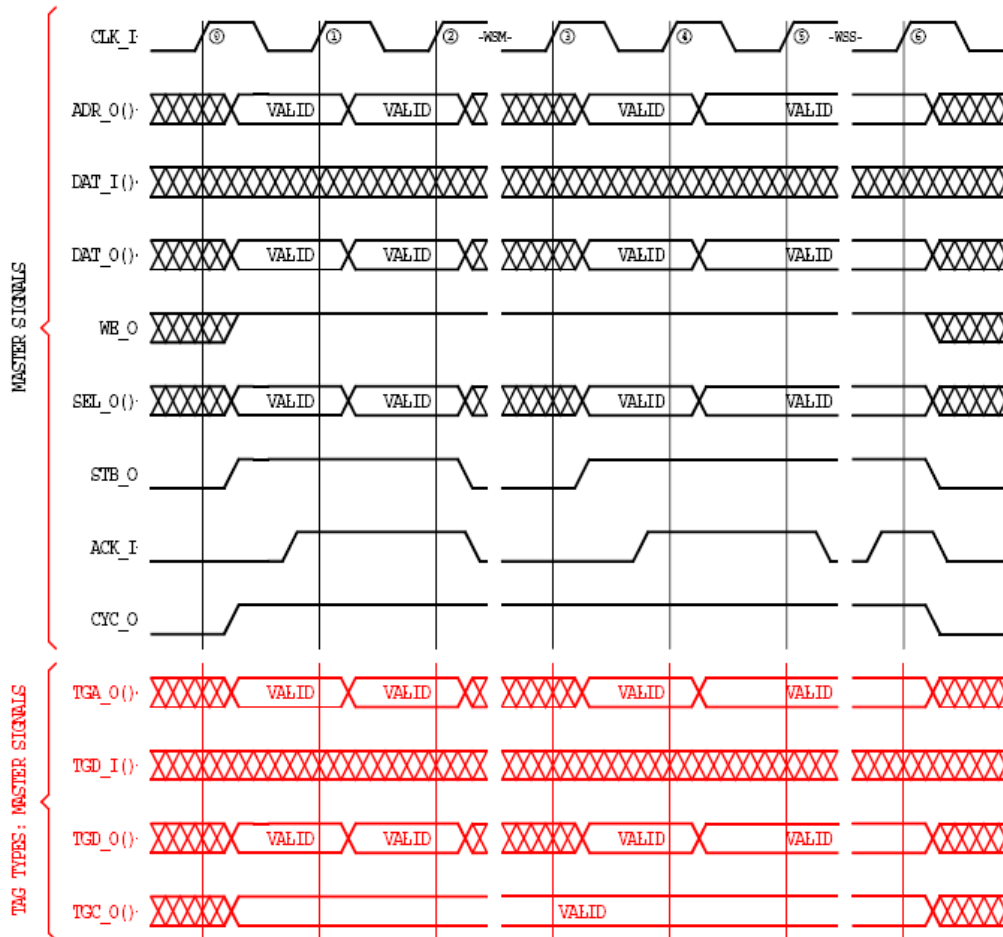


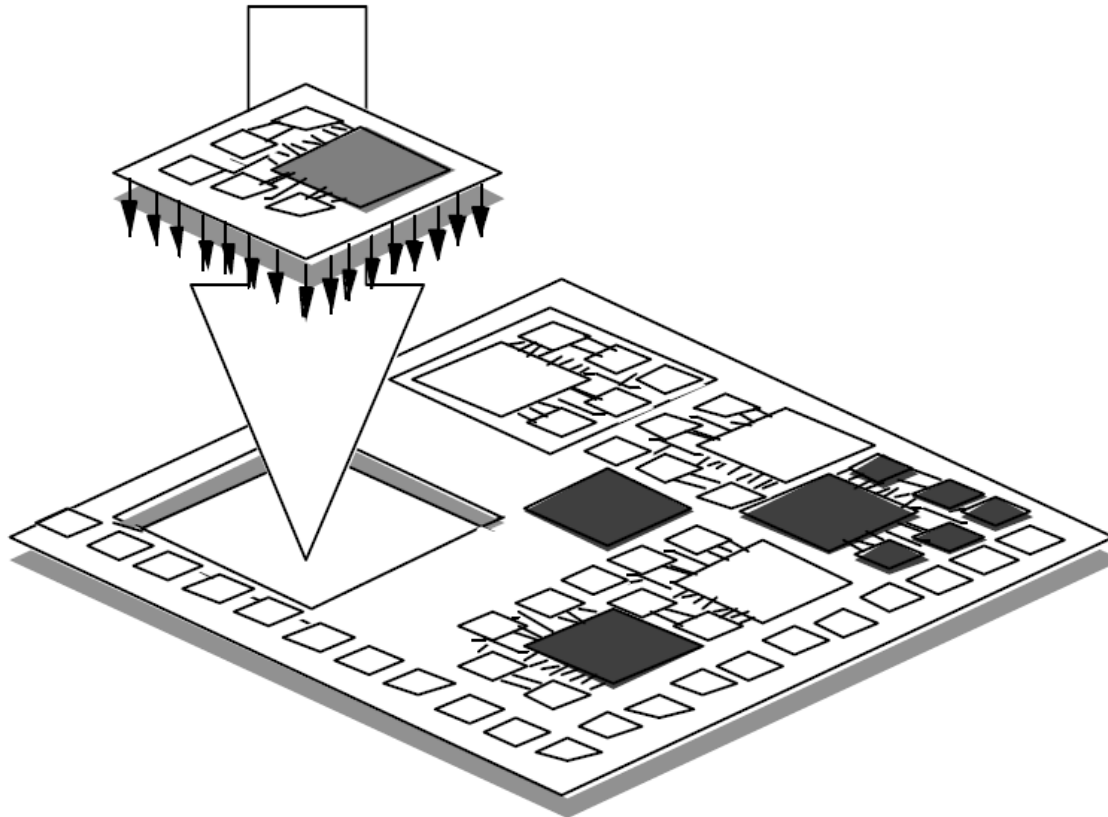
Figure 3-7. BLOCK WRITE cycle.

Socket alapú kommunikáció

- Busz alapú kommunikáció
 - Busz definíció, busz interfész
 - Busz-komponens interfész
- Jelentős befolyás a interfész kialakításában
- Nem teljesen hordozható
 - Buszonként egyedi kialakítást igényel
 - Egyedi tervezés, ellenőrzés, verifikáció

A probléma

- Rendszer építés különböző gyártóktól származó IP blokkok felhasználásával
- A kulcsszó az adapter réteg



Socket alapú SoC rendszer

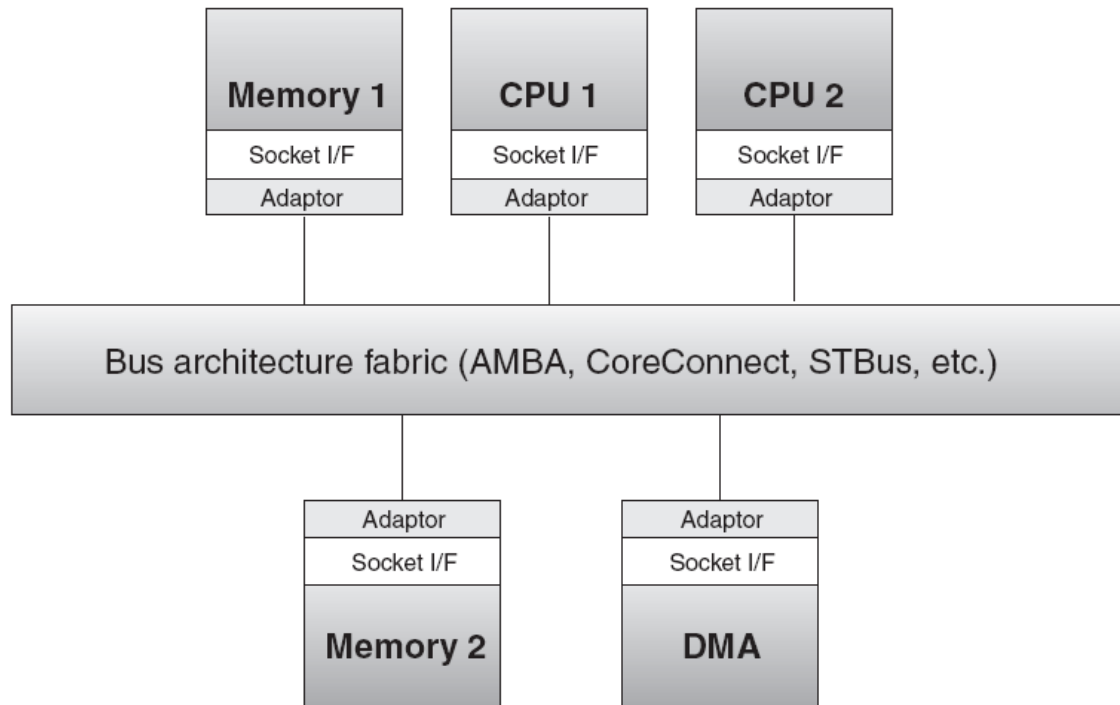


FIGURE 3.33

Example of system implemented with socket-based interface standards

Socket alapú kommunikáció

- OCP Open Core Protocol interfész
 - Szinkron pont-pont kapcsolat (illesztő réteg)
 - Busz független
 - Konfigurálható adatfolyam jelek (cím, adat, vezérlés)
 - Konfigurálható kiegészítő jelek
 - Pipeline és burst átviteli lehetőség
 - Többszörös szálak kialakíthatósága

OCP blokkvázlat

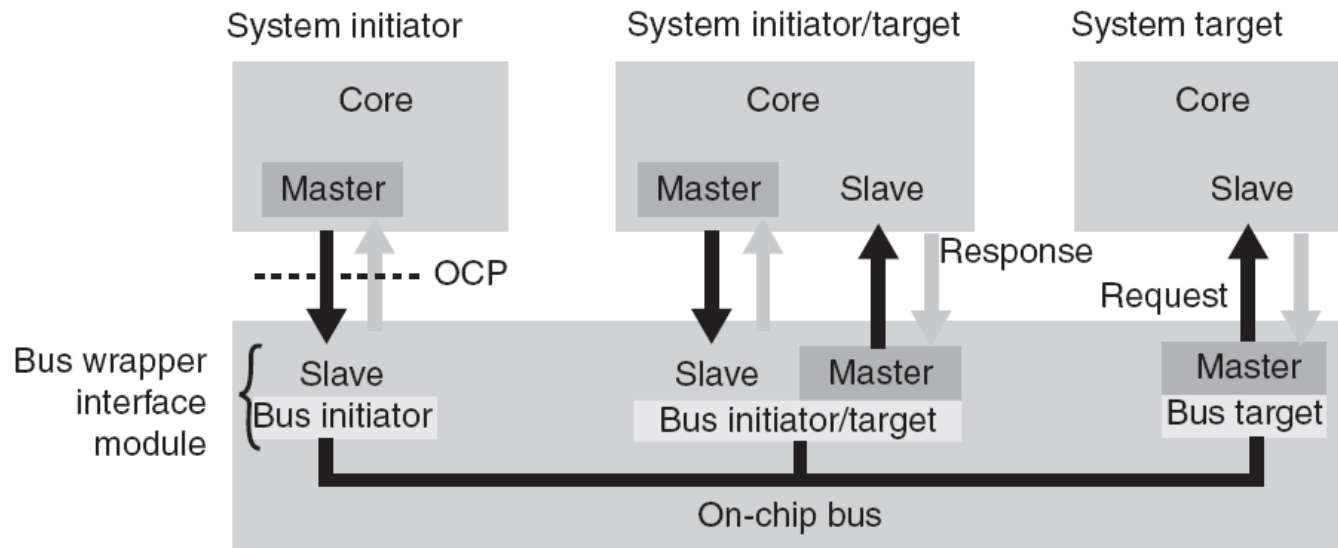


FIGURE 3.34

Example of system implemented with OCP socket-based interface standard [15]

Source: OCP-IP

OCP jelek

- Adatfolyam csoport
 - Alapvető átviteli jelek
 - Óra, cím, adat (Read, Write), típus, szinkronizáció
 - Típus: Read, Write, Idle, Broadcast, Exclusive read, Linked read, Non-posted write, Conditional write
 - Szinkronizáció
 - Egyszerű kiegészítések adatbájt, paritás, FIFO jelzések, cache kezelés,
 - Burst kiegészítések fix/tetszőleges, címkezelés(wrapping/incrementing/streaming)
 - Packing/nonpacking, eltérő szélességű interfészekre
 - Tag/ID kiegészítések
 - Eltérő sorrendű végrehajtások kezelésére
 - Szál kiegészítések
 - Master és slave egységek számára a szálak azonosítására

OCP protokoll hierarchia

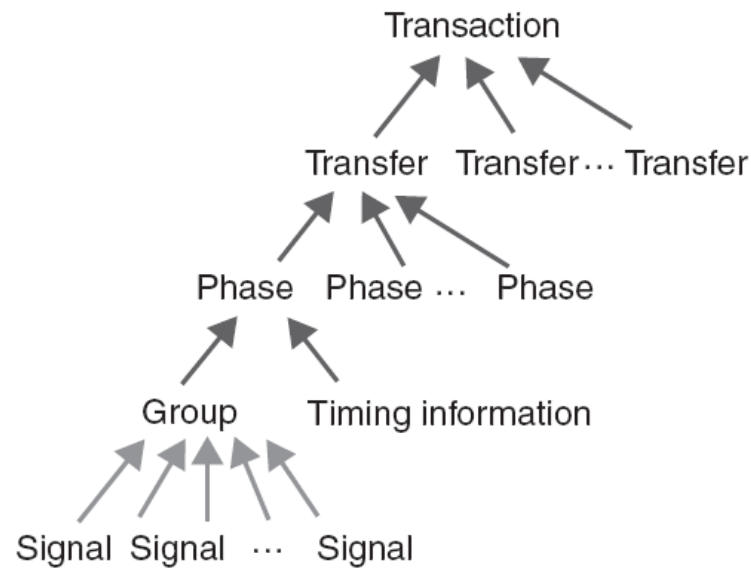


FIGURE 3.35

Hierarchy of elements that compose the OCP [15]

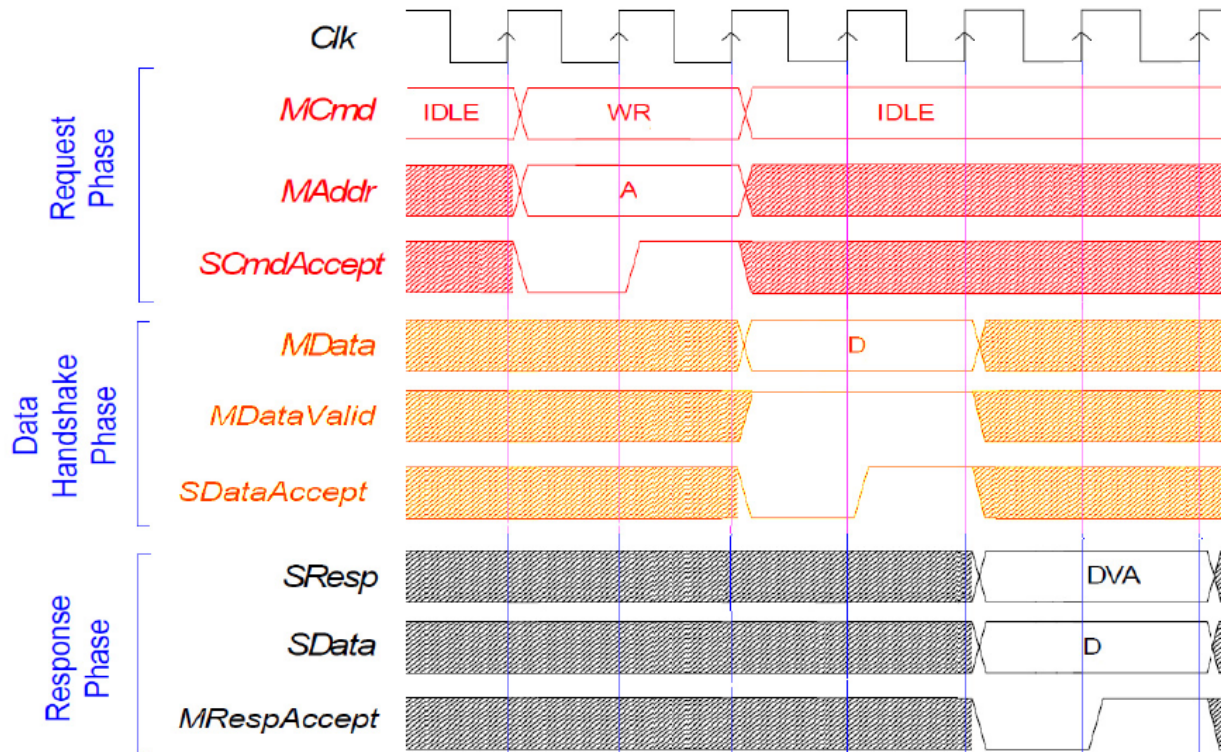
Source: OCP-IP

OCP hierarchia

- Jelek csoportja aktivitás alapján
 - Pl. adatátviteli jelek: kérés, válasz, szinkronizáció
- Fázisok, hasonlóan: kérés, válasz, szinkronizáció
- Átvitel: több fázisból áll, de mindig
 - Kérés szinkronizáció {válasz}
- Tranzakció: néhány, összetartozó átvitel együttese (adott címek, adatok és sorrend)

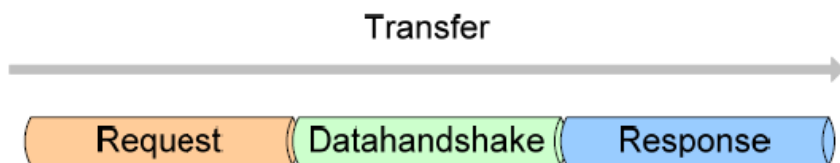
OCP átvitelek

- Egy teljes OCP átvitel

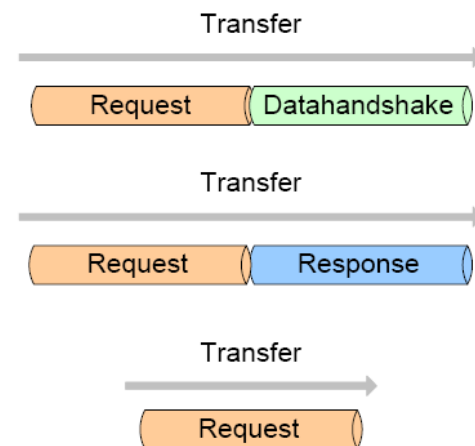


OCP átvitel típusok

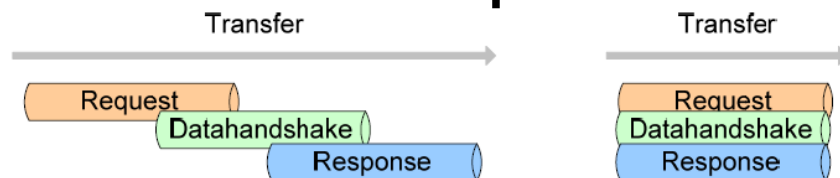
Normál



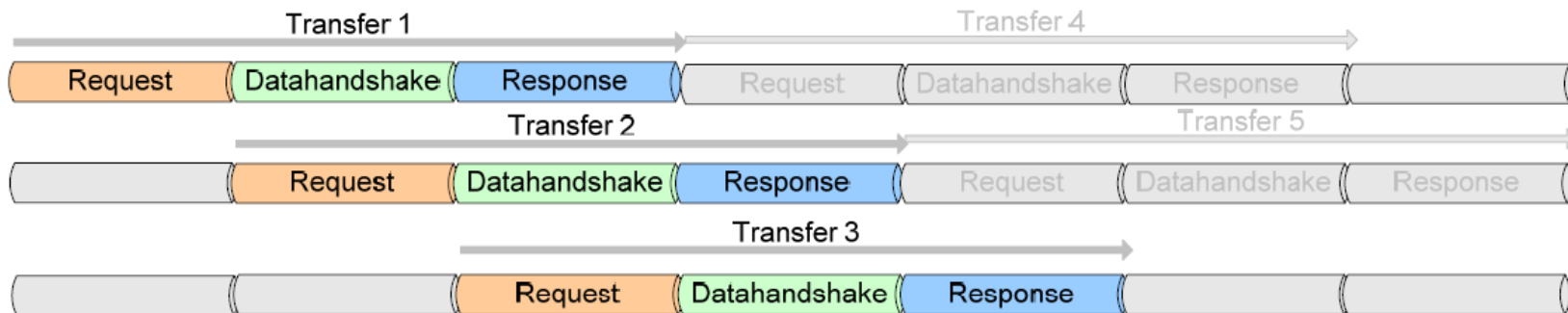
Hiányos



Átlapolt



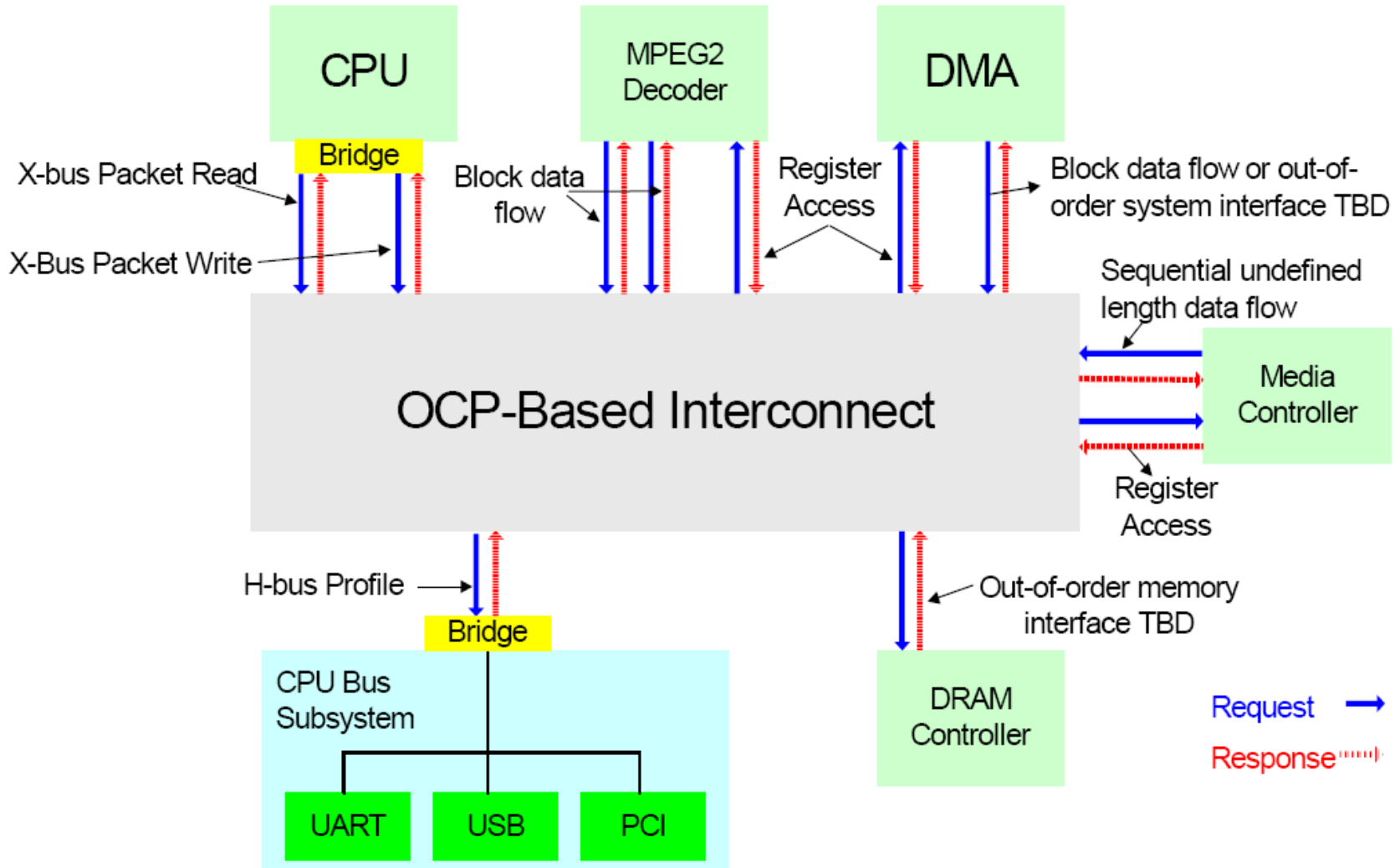
Pipe-line



OCP profilok

- Az OCP egyik erőssége a kialakított szabványos kapcsolatok lehetőségében van
- Előnyök:
 - Csökken az inkompatibilitás veszélye
 - Egyszerűsített illesztő rétegek a más típusú rendszerekhez
 - Könnyebb tesztkörnyezet fejlesztés
- Kétfajta alkalmazási cél
 - Új IP magok használjanak szabványos, natív profilokat egymás között
 - Külső eszközök közötti hídkapcsolatokhoz pedig egyedi profilok fejleszthetők

OCP profilok



OCP profilok

- Natív OCP profilok
 - Regiszter hozzáférés profil
 - Blokkos adatfolyam profil
 - Szekvenciális tetszőleges adatfolyam profil
- Híd profilok
 - Szimpla H-busz profil
 - X-busz packet írás profil
 - X-busz packet olvasás profil
- Biztonsági profil

Híd profilok

- A cél az elterjedt, szabványos buszokhoz szükséges kapcsolat realizálása
- Master vagy slave funkció is lehetséges
- Két típus:
 - H-bus (talán az AMBA AHB-re utal...)
 - Normál egyedi és blokkos átvitelek
 - X-bus (talán az AMBA AXI-re utal...)
 - X-bus Packet Write
 - X-bus Packet Read

X-bus profilok (WRITE, READ külön)

- Kifejezettem olyan mastereknek, amelyek egy-egy kérésre több fázisban szeretnének adatot írni vagy olvasni (de csak egy irányban)
- Csomag alapú kapcsolatok felépítésére szolgál
- Az adat szervezés, érvényesség ellenőrzés a master feladata

VSIA Virtual Component Interface

- Hasonló célú, mint az OCP
- Pont-pont, szinkron, socket alapú protokoll
- Három verzió:
 - PVCI: Periféria VCI
 - BVCI: Basic VCI
 - AVCI: Advanced VCI

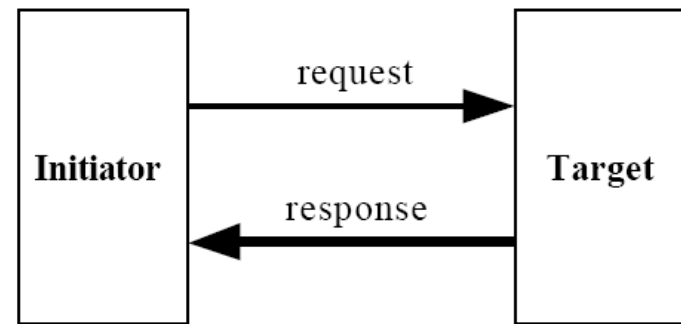


Figure 1: VCI is a Point-to-Point Connection

A VCI használata OCB esetén

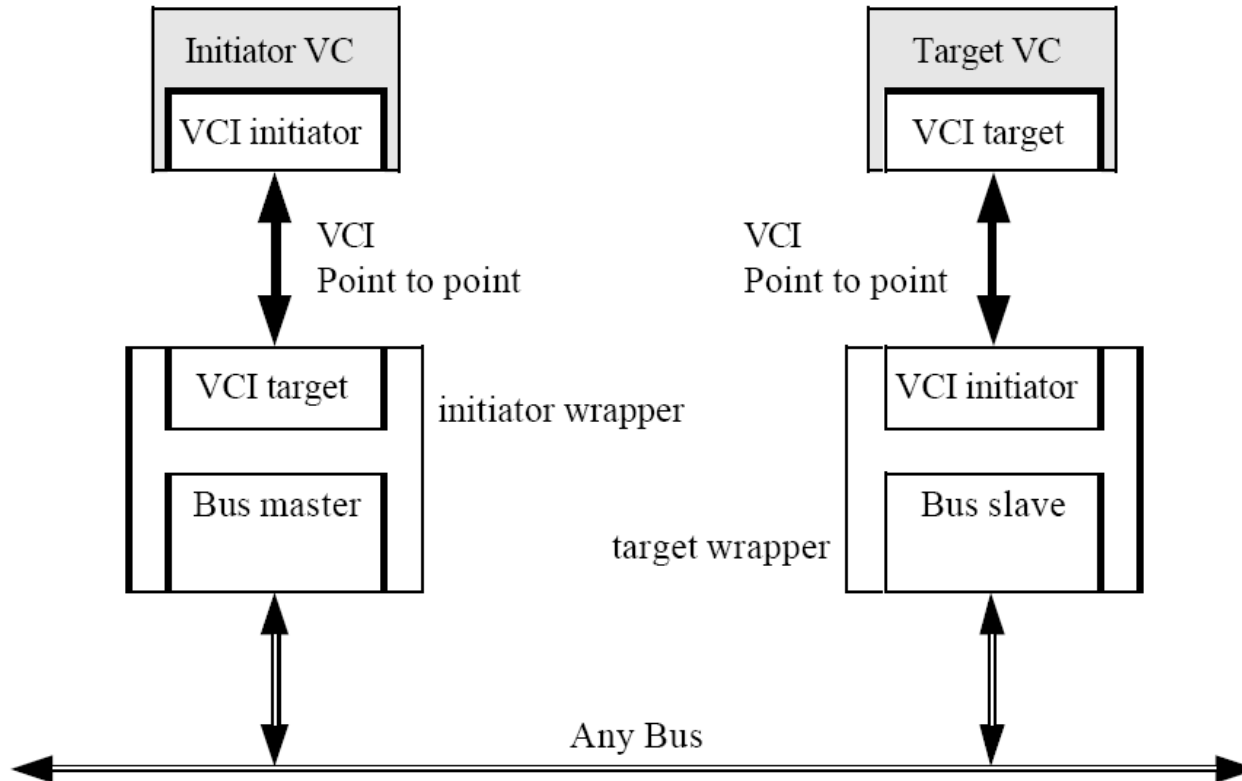
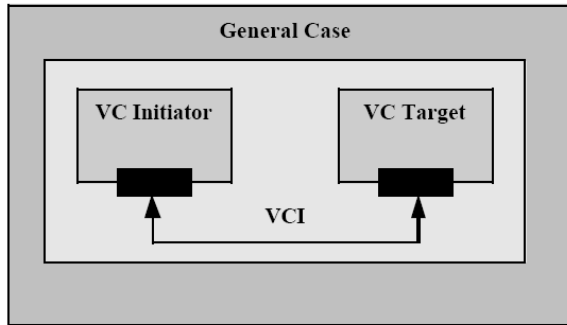


Figure 2: Two VCI Connections Used to Realize a Bus Connection

Tervezési tippek



Wrapped Bus Example

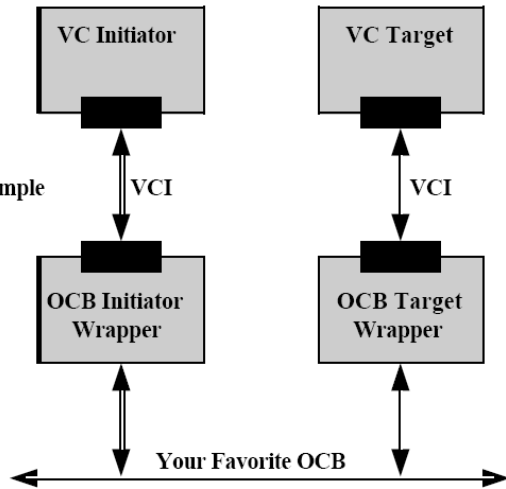


Figure 43: VCI Block Diagram

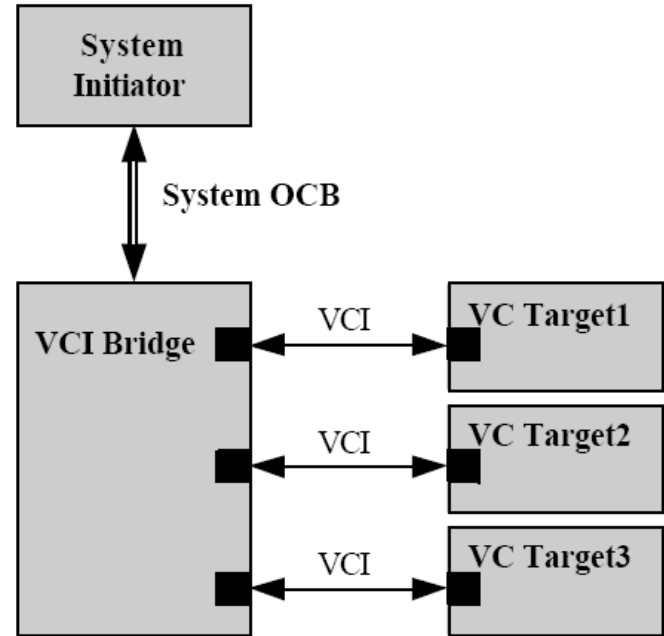


Figure 44: VCI with Star Topology

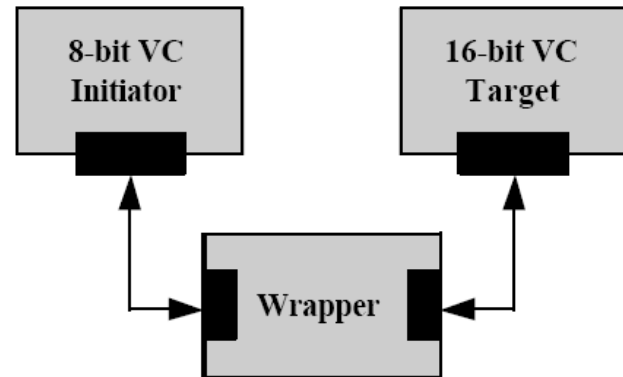


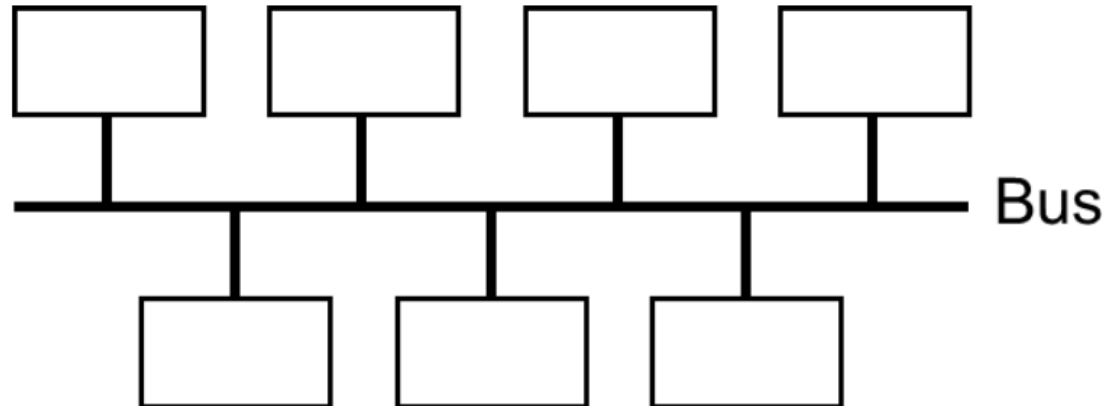
Figure 45: Interconnecting Different-Size VCI Components

Összegzés

- Az áramkörön belüli és áramkörön kívüli buszok jellemzői, követelményei eltérőek
- A komplex rendszerek összetett, nem feltétlenül egyetlen (akár hierarchikus) megosztott buszt kívánnak
- Nő a pont-pont kapcsolatok jelentősége
- Léteznek elterjedt busz szabványok, illetve a buszok feletti, socket alapú interfész megoldások is

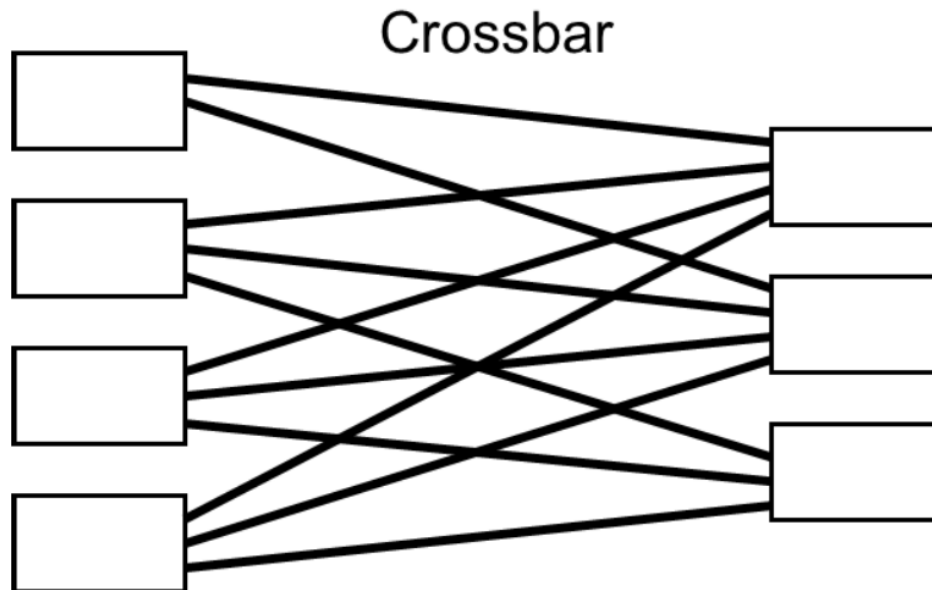
NoC rendszerek

- A busz kapcsolat gyakran szűk erőforrás
 - Funkcionális egységek száma
 - Master/slave arány
 - Regionális kapcsolatok speciális igényei
 - Szegmentált memória



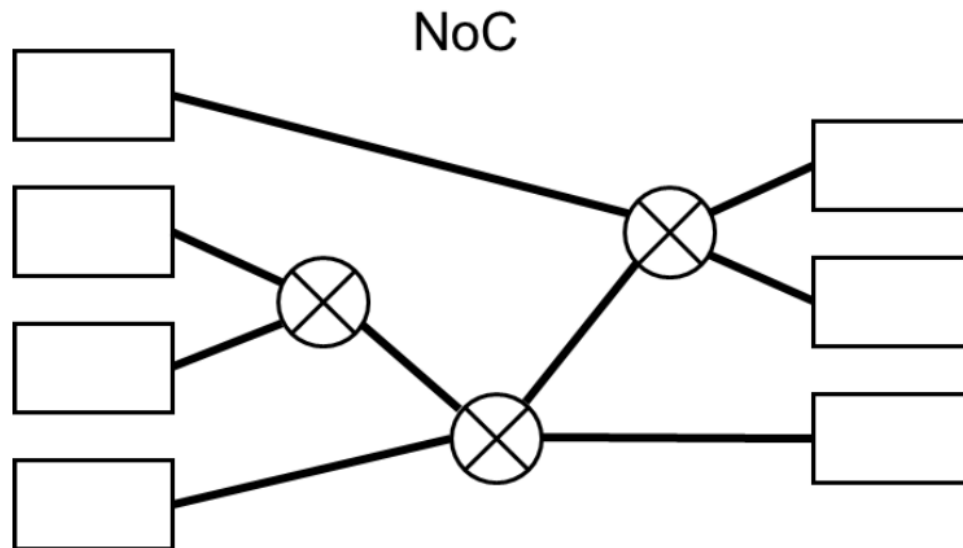
NoC rendszerek

- Áramkörön belül vezetékezés költsége kisebb
- Pont-pont kapcsolatok, egyedi adatátvitel
 - Sok esetben túlzó, nem hatékony



NoC rendszerek

- Kommunikációs igények analízálása
- Átviteli igények felmérése
- Megfelelő kapcsolatok kialakítása
- Közvetlen-/Csomagkapcsolt működési mód



NoC általános jellemzők

- NoC típus:
 - Heterogén: Különböző egységek (pl. SoC)
 - Homogén: Multi-/ Many Core architektúrák
- Topológia típusa
- Csomópontok száma
- Kapcsolatok/interfészek száma egy csomópontban
- Interfészek szélessége, átviteli bitek száma

NoC topológiák

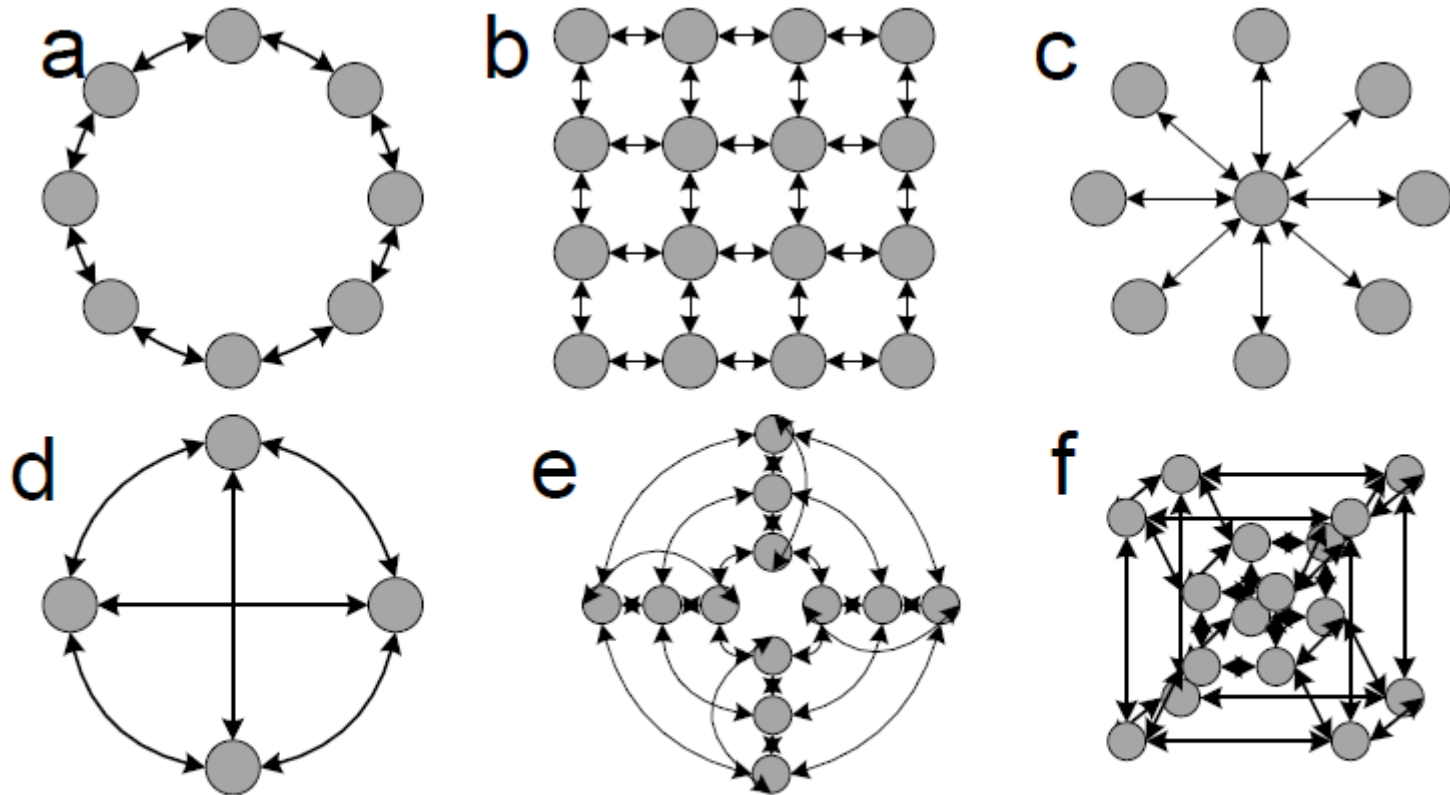
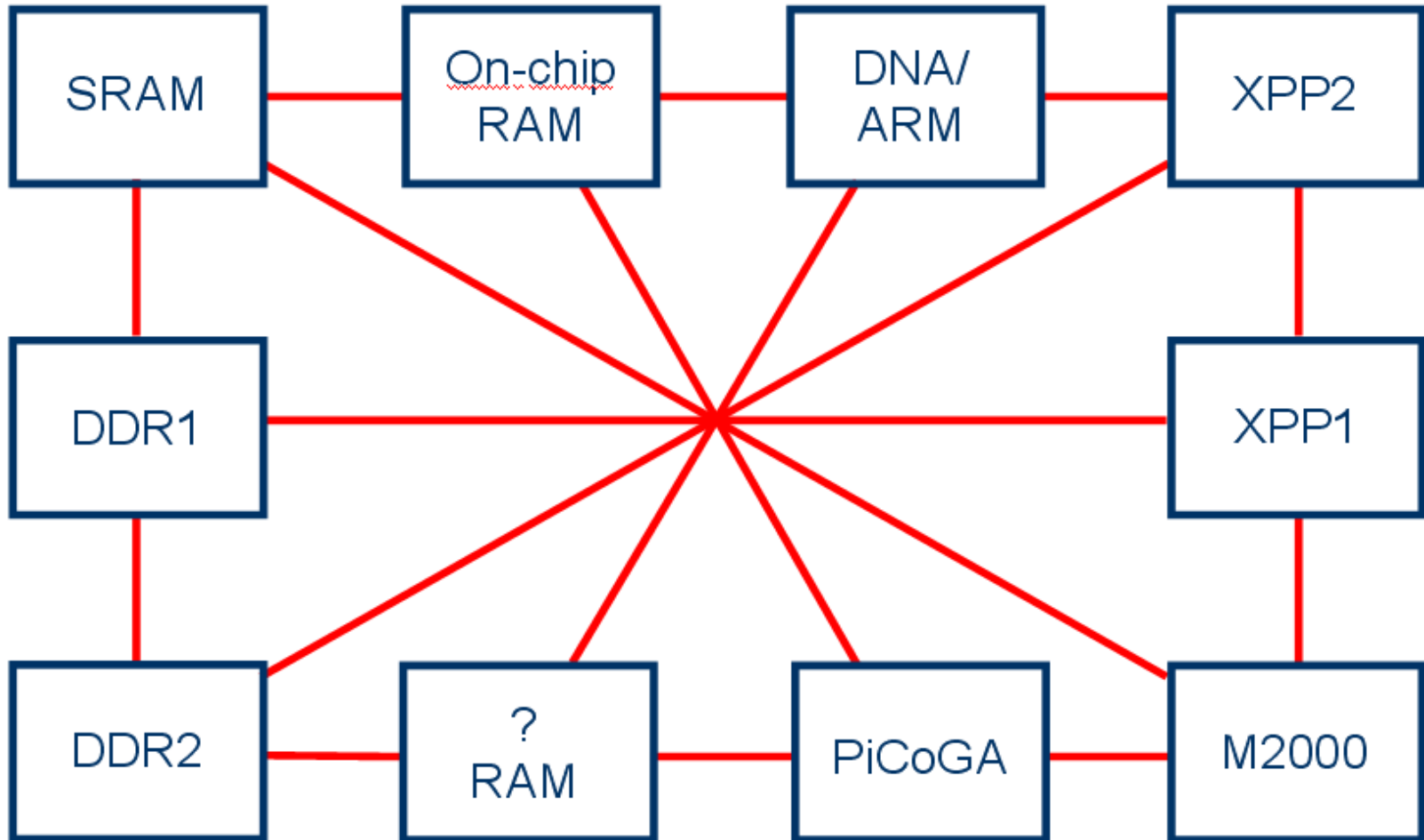


Figure 2. NoC topologies: (a) ring, (b) mesh, (c) star, (d) fully, (e) torus, and (f) hypercube

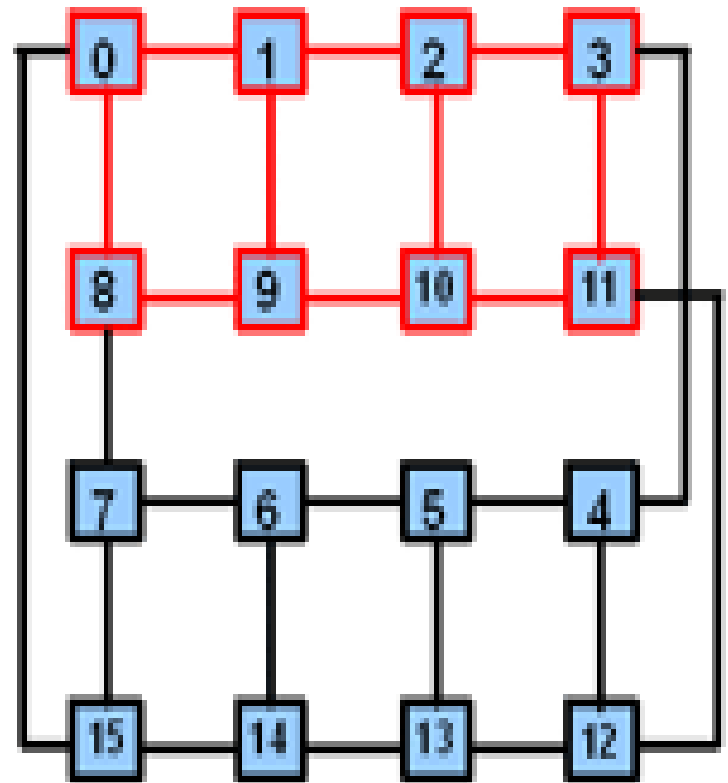
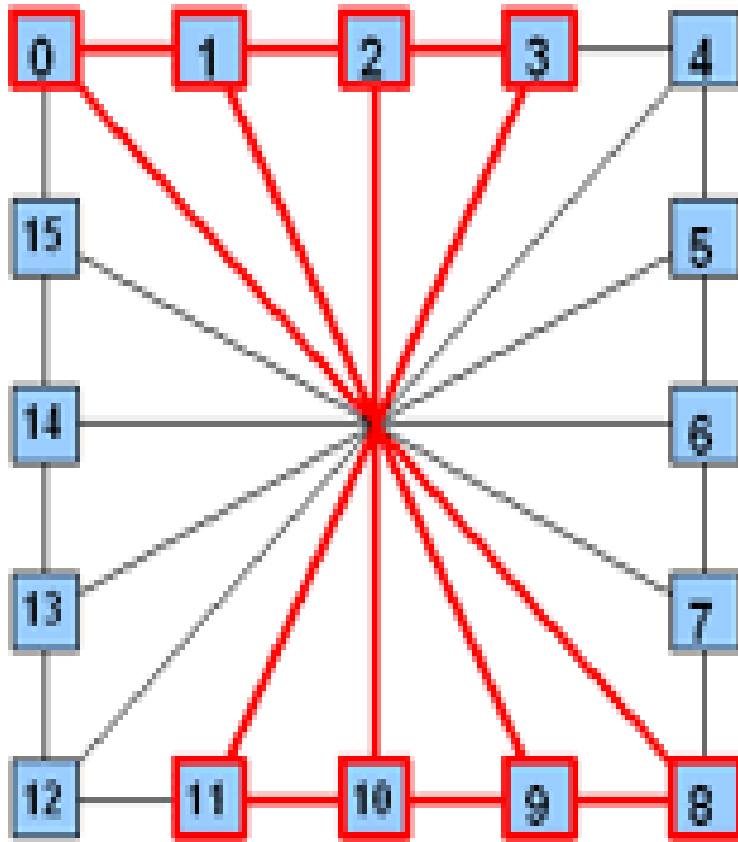
NoC topológia jellemzők

- Kommunikációs távolság
- Adatátvitel/Üzenetküldés ideje, sebessége
- Konkurrencia, párhuzamosság
- Adatfolyam kiegyenlítettség, hub csomópontok
- Huzalozási komplexitás, realizálhatóság

Példa: STNoC részlet

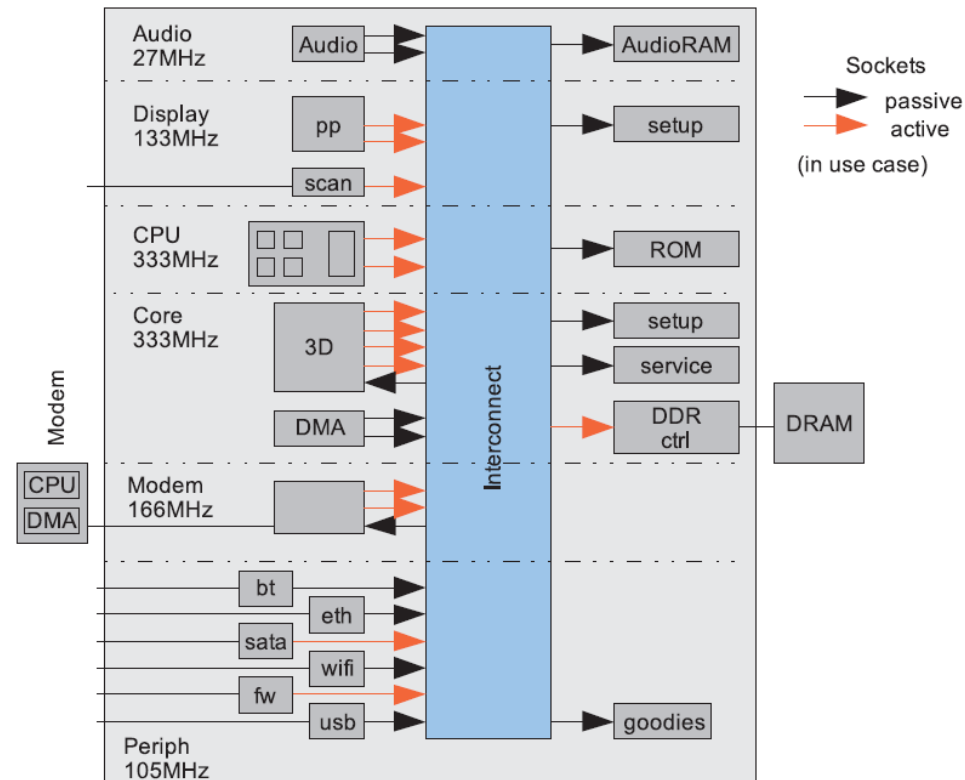


Példa: Az STNoC előnye



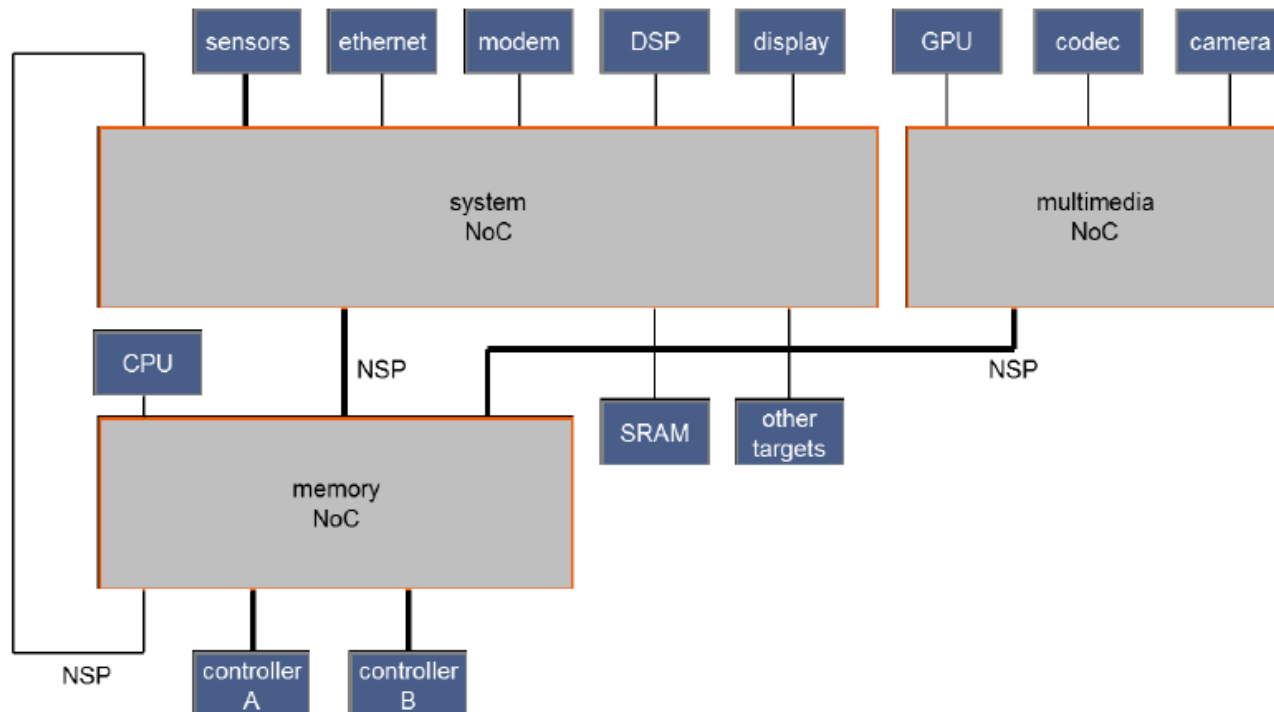
Heterogén NoC rendszerek

- Jellemzően SoC megoldások, a korábbi buszrendszerek kiváltására
- Jelentősen eltérő kommunikációs igények
- CPU, memória, DMA
- Speciális funkciók
- Perifériák
- Multimédia elemek



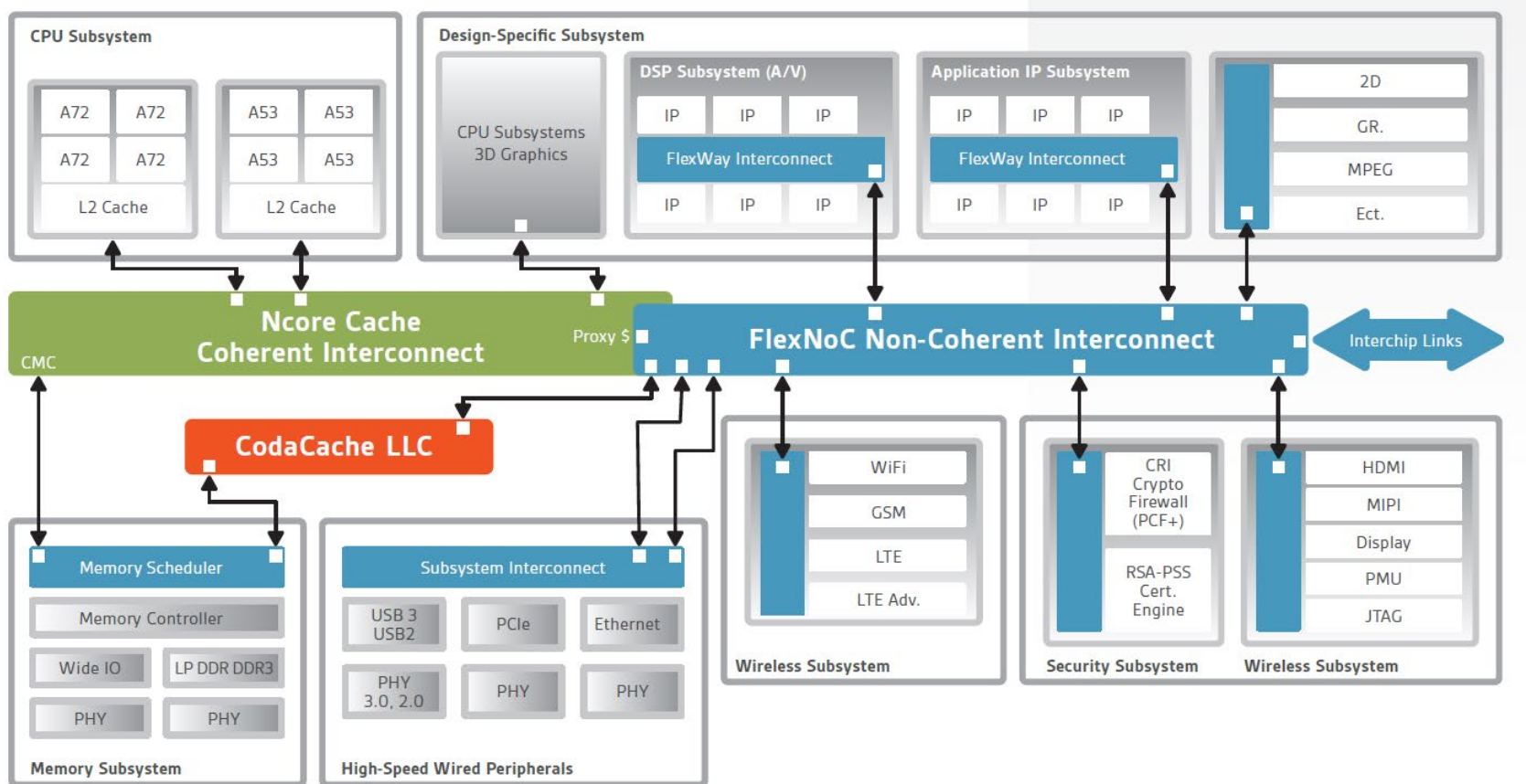
Heterogén NoC rendszerek

- Hierarchikusan felépített megoldások
 - NoC: egyedi tranzakciók, igény szerinti felépítés
 - NSP: NoC Socket Protocol a részhálózatok között



Heterogén NoC rendszerek

- Koherens és nem-koherens tartományok
- Protokoll függetlenség (AMBA, OCP, stb.)



Homogén NoC rendszerek

- Jellemzően sokprocesszoros rendszerek
- Multi-/Many Core SoC,(MPSoC)
- >10, akár 100 vagy 1000 processzorra
- Mesh elrendezés, két típusú funkcióval:
 - Feldolgozó elem
 - Kommunikációs elem
- A hálózatonban minden típuscsaládon belül az egységekre azonos igények, szerep, pozíció, használati mód, funkcionalitás

Homogén NoC rendszerek

- Az elrendezés lehet mesh, tórus, hiperkocka
- Rögzített,
- Az átvitel lehet közvetlen vagy csomagkapcsolt
- A feldolgozó elem
 - Általános CPU
 - CPU csoport (2-4-8...)
 - Speciális funkció
- Történelem: Transputer
 - Meiko Computing Surface

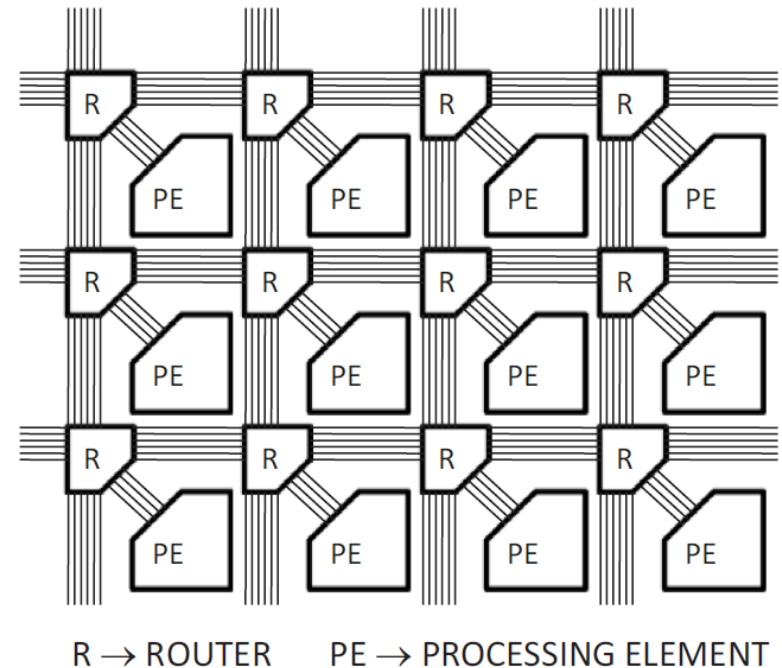


Figure 1.1: Mesh based network on-chip topology.

Homogén NoC rendszerek

- Adatátviteli megoldások
 - Közvetlen, áramkörkapcsolt átvitel
 - Hálózati, csomagkapcsolt átvitel
- Közvetlen:
 - Az átvitel előtt a forrás -> cél útvonalat ki kell alakítani (routing) és le kell foglalni (kizárás)
 - Az átvitel alatt a forrás-> cél közvetlenül kommunikál, szinkronizációs kapcsolatban van
 - A lefoglalt utak miatt blokkolás lép fel, távoli kapcsolatok jelentős korlátozást eredményeznek

Homogén NoC rendszerek

- Csomagkapcsolt átvitel
 - Nincs közvetlen kapcsolat, csak szomszédos
 - PE->R, R->R, R->R, ... R->R, R->R, R->PE
- Az üzenetet kisebb egységekre bontva küldi
 - Packet: Head, Body, Tail

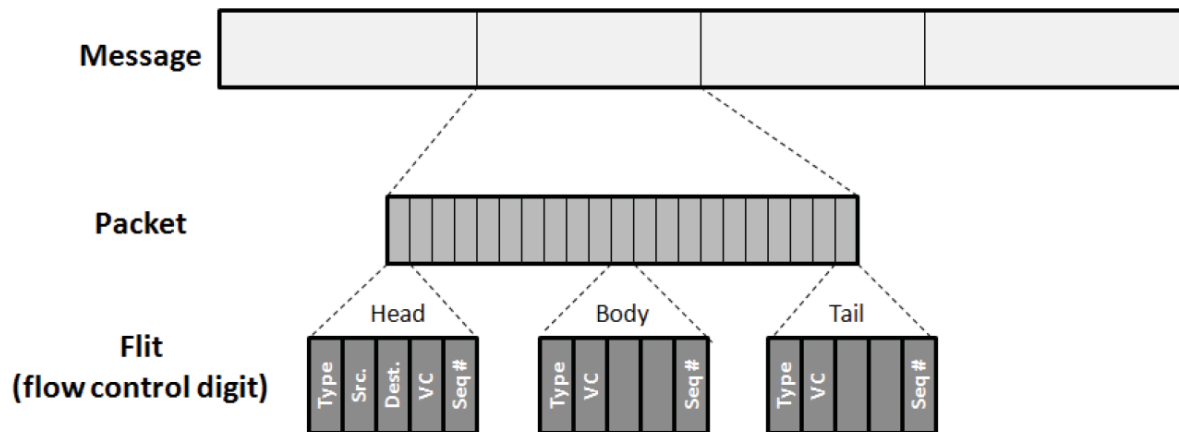


Figure 2.4: Data abstraction in NoCs.

Homogén NoC rendszerek

- Csomagkapcsolt átvitel megoldásai
 - Store/Forward: Csomagátvitel egy ütemben
 - Wormhole: Flit szintű ütemezés, rögzített úttal
 - Cut-through: Az előző kettő keveréke

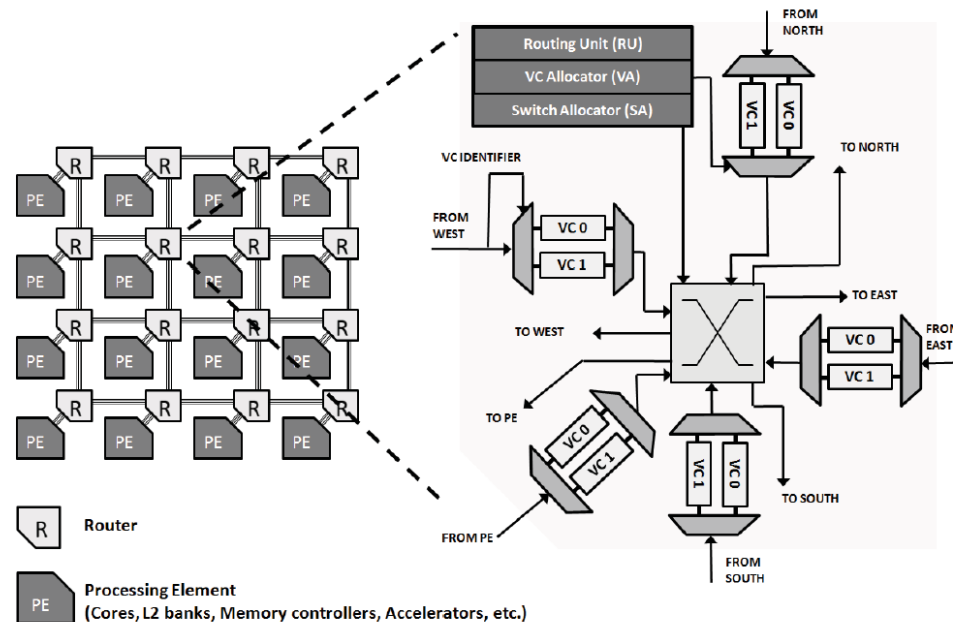
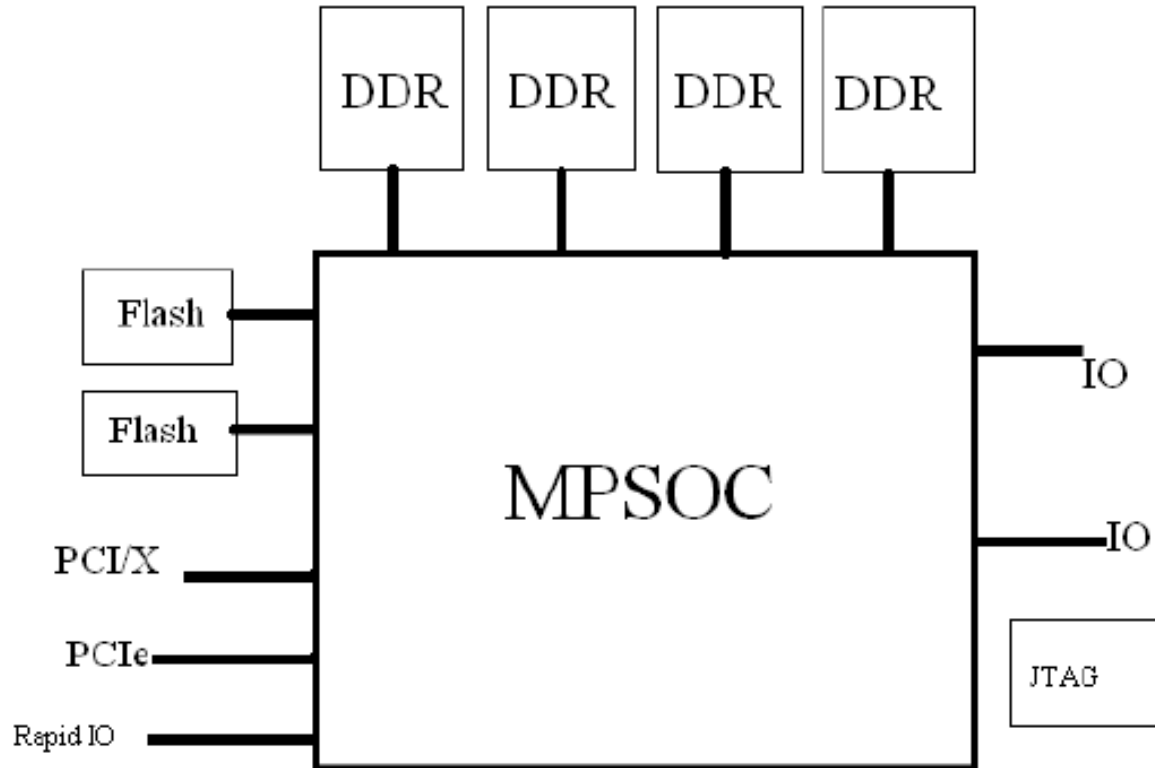


Figure 2.5: A typical NoC router design.

Példa: MPSoC rendszer

- 24 processzoros MPSoC OCP interfésszel
- Egy igazi NoC rendszer, szép felépítéssel
- Cél: A DDR2 memória elérés optimalizálása
 - External DDR2-Constrained NOC-Based 24-processors MPSoC Design and Implementation on Single FPGA (Zhoukun Wand, Omar Hammami, ENSTA ParisTech)
- Xilinx Virtex-4 FX FPGA
- MicroBlaze 32 bites Softcore processzorok
- 4 db DDR2 külső memóriamodul
- Arteris OCN kommunikációs hálózat

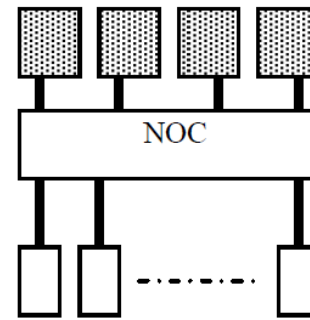
Követelmények



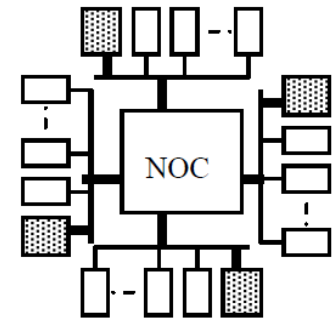
MPSOC Architecture Block Diagram

Rendszerfelépítési alternatívák

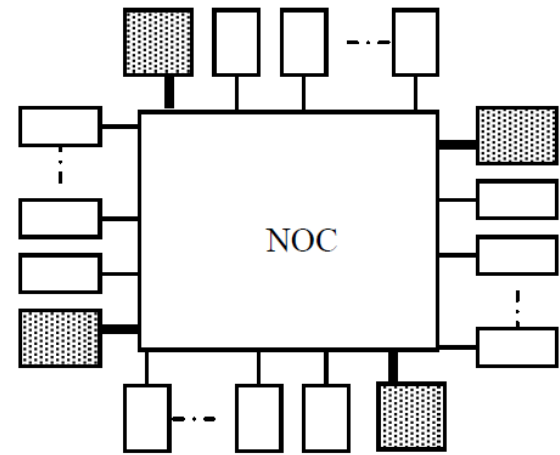
- Mi adja a legkedvezőbb kialakítást a 4 db külső memória és a 32 db CPU között?
- Teljesen globális
- Lokális + kapcsolatok
- Részben lokális + esetleg modulok közötti egyedi kapcsolatok



(a)



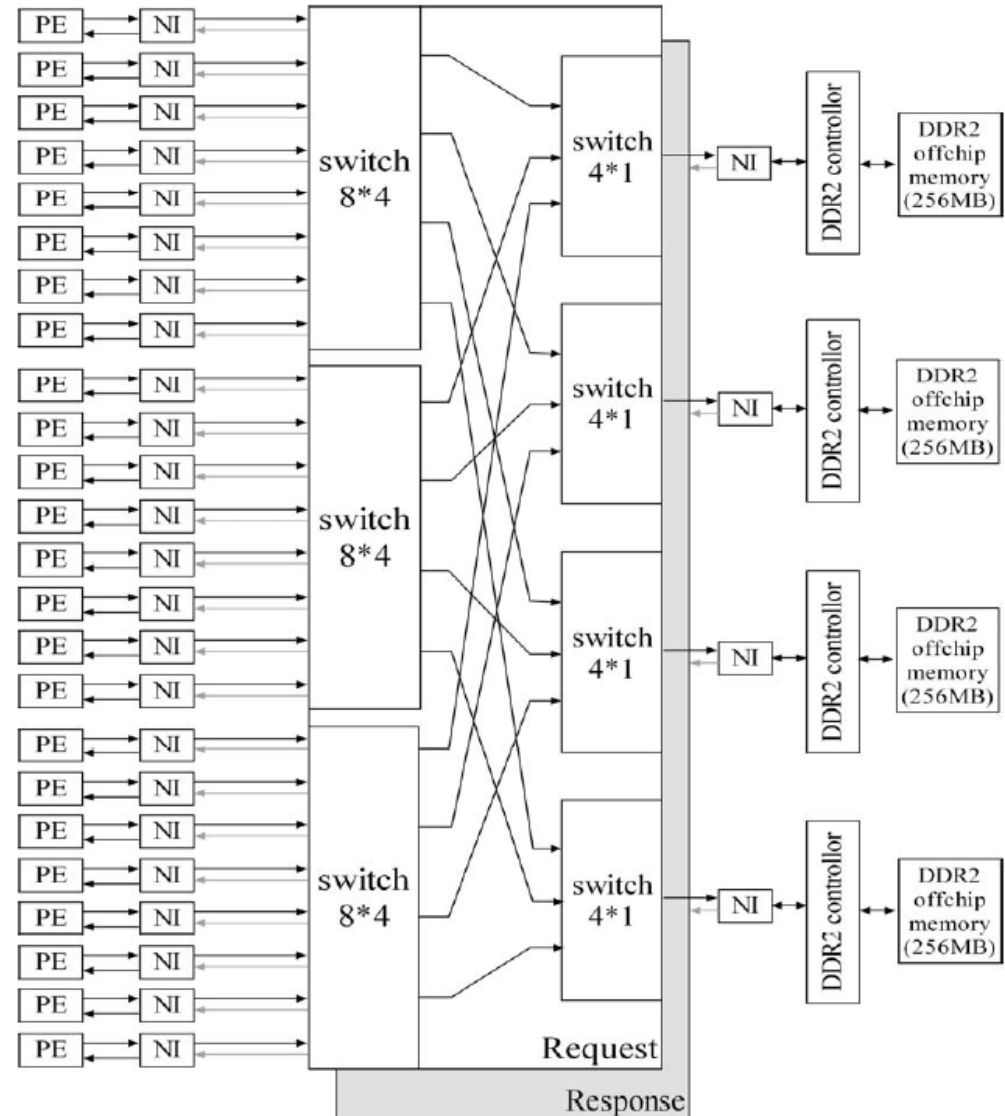
(b)



(c)

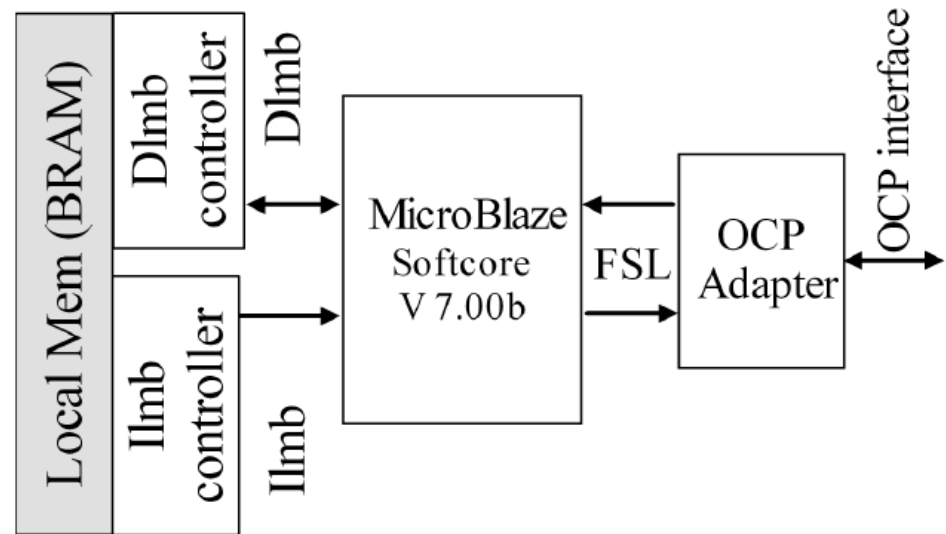
Globális kommunikációs hálózat

- Teljesen szimmetrikus, csomagkapcsolt hálózat
 - PE = processzor
 - NI = hálózati interfész
 - SW = switch-ek
 - DDR2 controller = memória vezérlők



Processzorok

- Xilinx MicroBlaze 32 bites softcore
 - Full CPU Core kiépítés (FPU, HWM, HWD, BS)
 - Egyenként 32KiB BlokkRAM lokális memória
 - Szétválasztott Program és Adat elérés
 - FSL interfész:
„regiszter alapú
FIFO adatfolyam”
 - OCP protokol

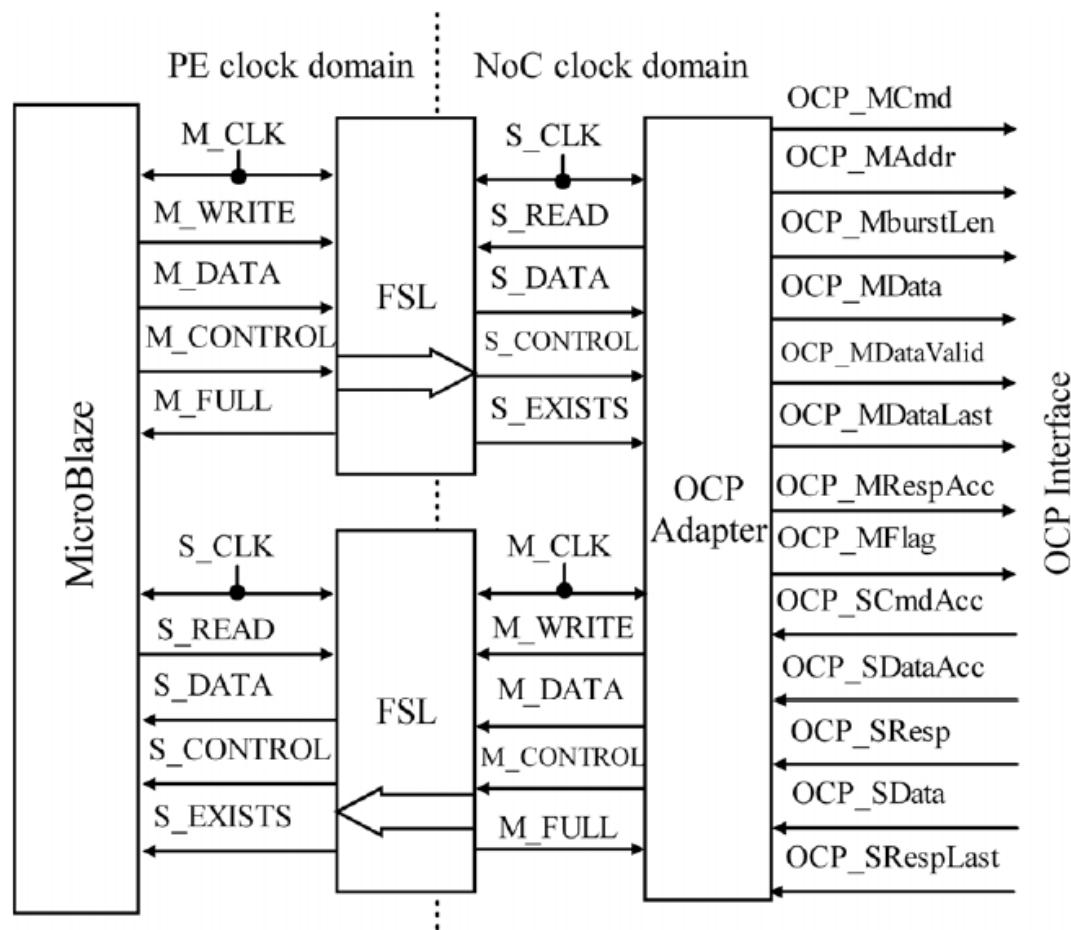


OCP alapú On Chip Network

- Arteris NoC Compiler
- OCP-to-NTTP Network Interface Units (NI)
 - Master és slave hálózati interfészek
- NTTP Noc Transaction and Transport Protocol
 - 3 rétegű kommunikációs mechanizmus
 - Tranzakció, átvitel, fizikai jelek
- GALS rendszerkialakítás
 - Globálisan aszinkron, lokálisan szinkron működés
 - FSL FIFO órajeltartományok közötti átmenethez

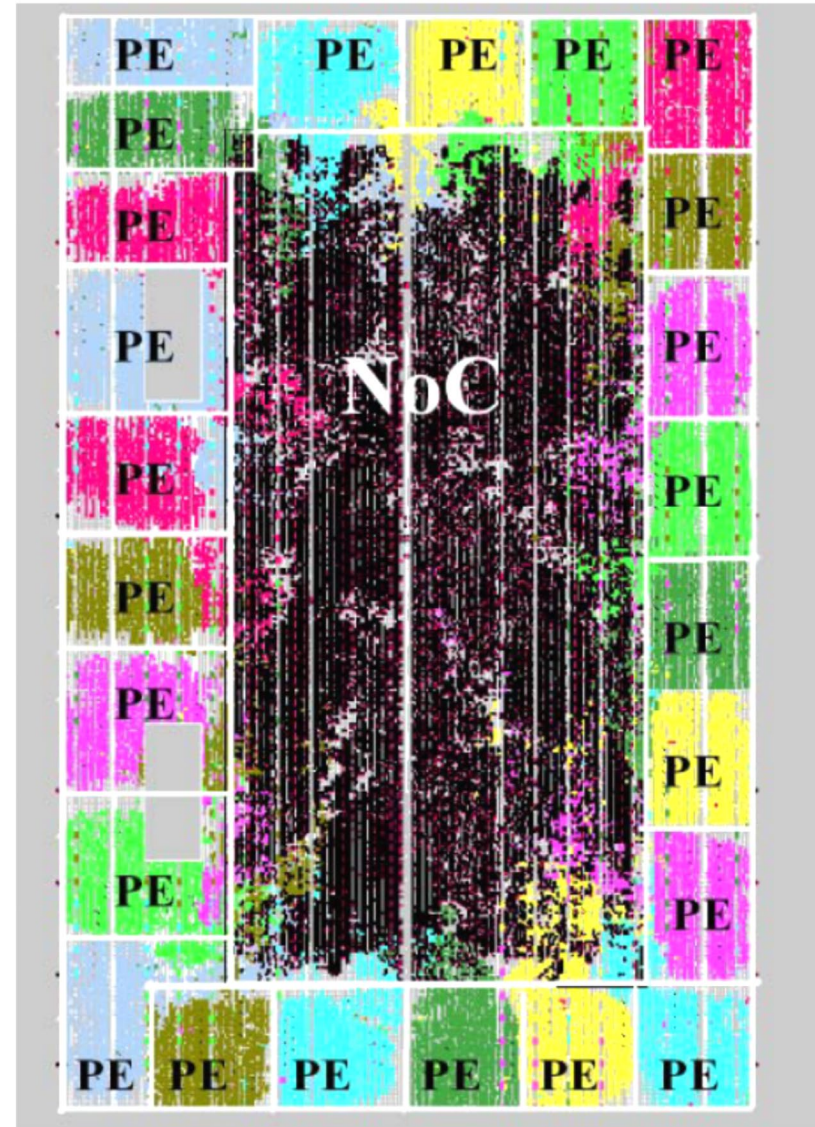
OCP interfész kialakítása

- FSL FIFO jelek ↔ OCP átviteli jelek



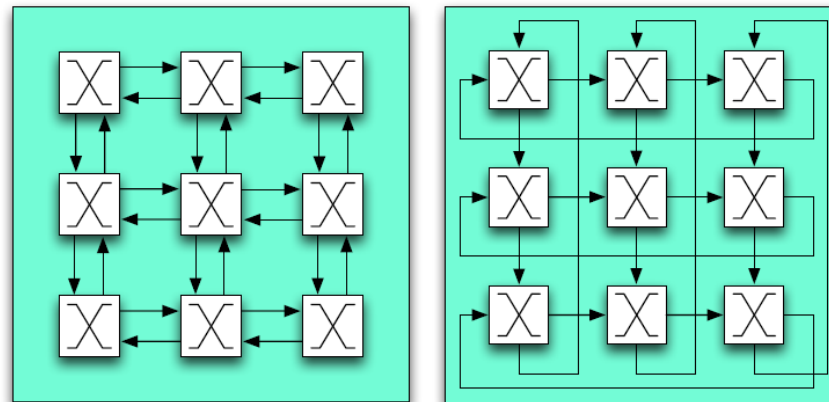
Eredmények

- Xilinx V4FX-140 FPGA
 - RAMB16 69%
384/552
 - DSP 37%
72/192
 - Logika 87%
55266/63168



FPGA alapú NoC: Hoplite

- Optimalizált NoC architektúra
- Many Core rendszer > 100 PE
- 2D struktúra, tórus elrendezéssel



(a) 2D Bidirectional Mesh

(b) 2D Directional Torus

Fig. 2: Topology of NoCs considered in this study – A Bidirectional Mesh and a Directional Torus.

FPGA alapú NoC: Hoplite

- Irányított adatáramlás előnyei
- Egyszerűsített router HW, kevesebb erőforrás: huzalozás, bufferek, logika
- Kisebb késleltetés, közvetlen adatkapcsolat

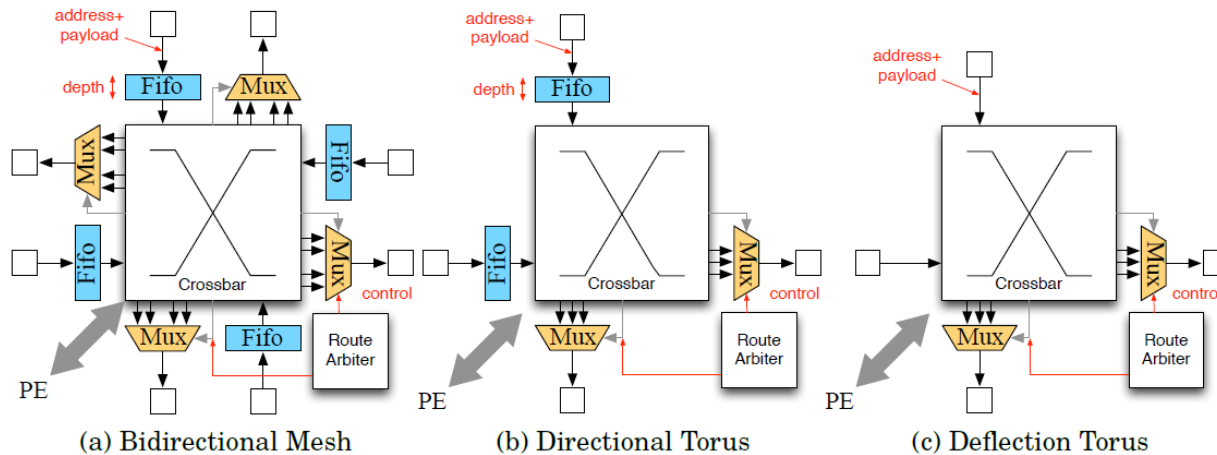
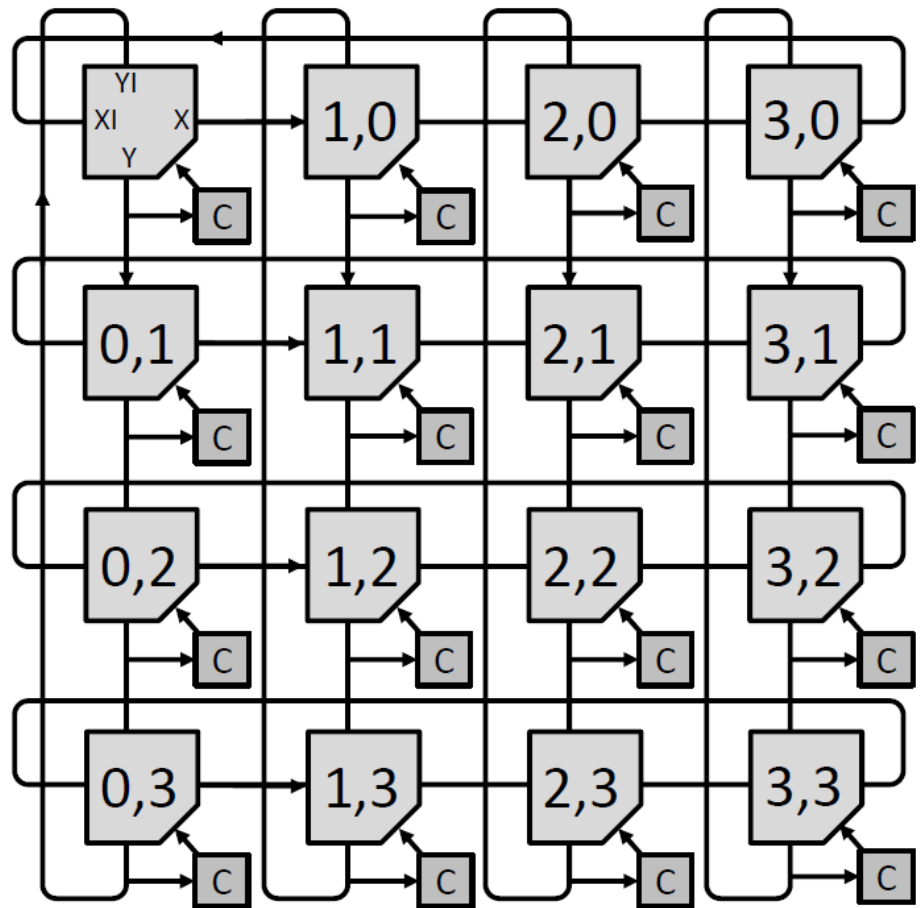


Fig. 3: Switch Organizations evaluated in this study. Bidirectional Buffered Mesh and Directional Buffered Torus needs FIFO buffers, and wires in each dimension, while Directional torus eliminates buffers as well.

FPGA alapú NoC: Hoplite

- 2D Directional Torus
- Egyszerűsített, (csökkentett) huzalozhatóság
- Deflected (átirányított) adat továbbítás
- Rugalmas konfliktus feloldás
- 30% gyorsulás



FPGA alapú NoC: Hoplite

- Tetszőleges adatútszélesség, akár 1024 bit
- Sok szabad erőforrás a PE blokkoknak
 - Logikai, memória, lokális huzalozás

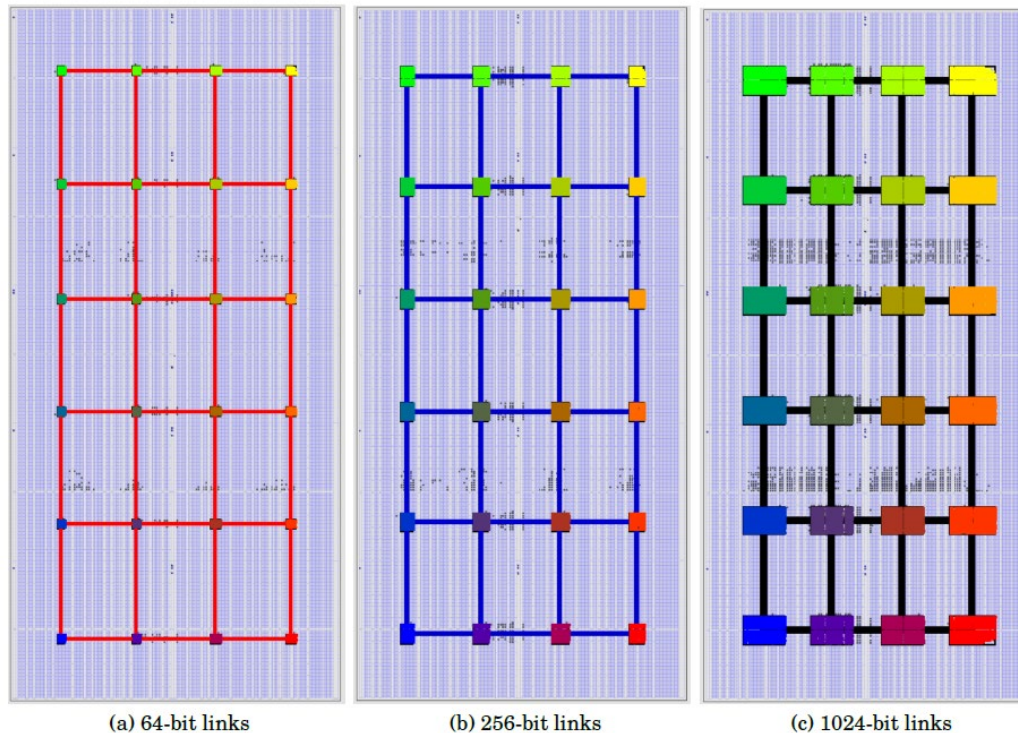


Fig. 7: Example 6×4 Hoplite NoCs with 64b, 256b, 1024b links, not folded, with one X and two Y sets of wraparound link pipeline registers, Xilinx KU040-2 target

GRVI cluster as PE

- Optimalizált RISC-V RV32I CPU megoldás
- Minden PE tartalmaz 8 CPU magot
- Ezeknek közös lokális adatmemóriájuk van
- Két-két CPU közös utasítás RAM-ot használ
- Lehetséges speciális HW gyorsító funkciót integrálni
- Közös interfész a Hoplite hálózathoz

GRVI cluster as PE

- PE kialakítása

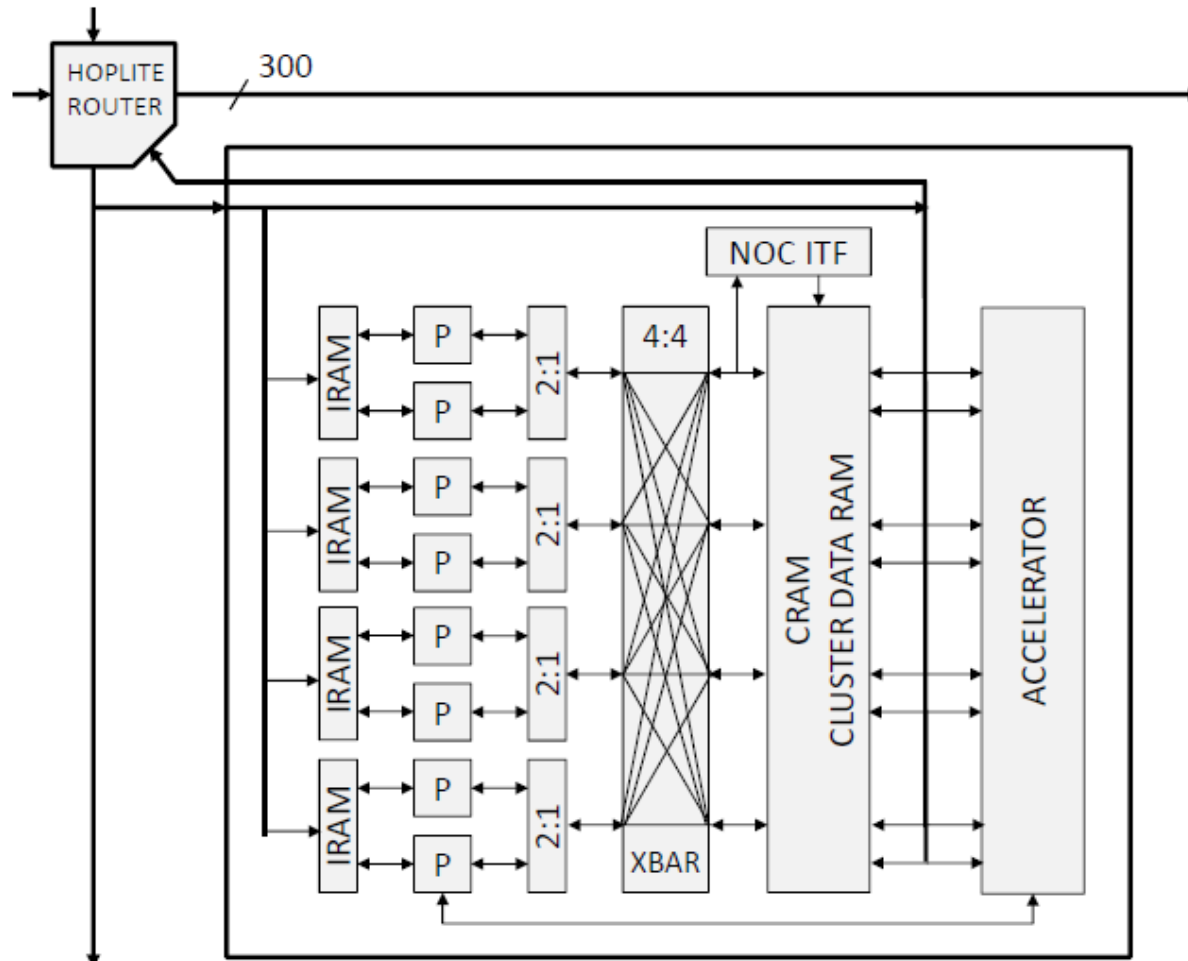
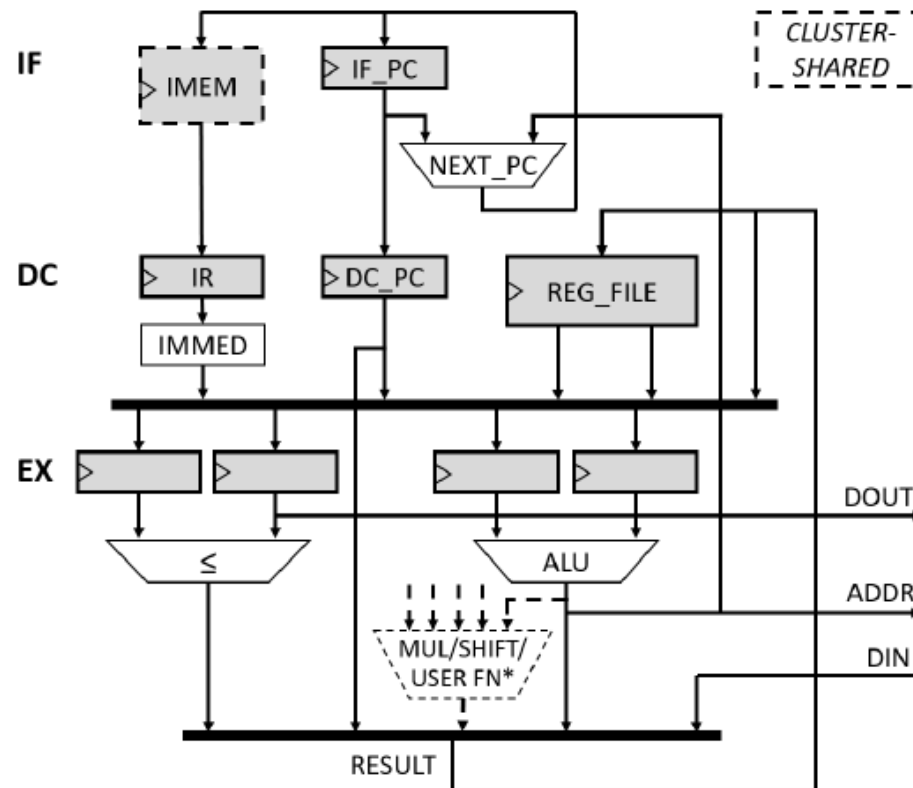


Fig. 8: GRVI cluster tile: one 300b Hoplite router, 8 GRVI PEs, RAMs, accelerator(s)

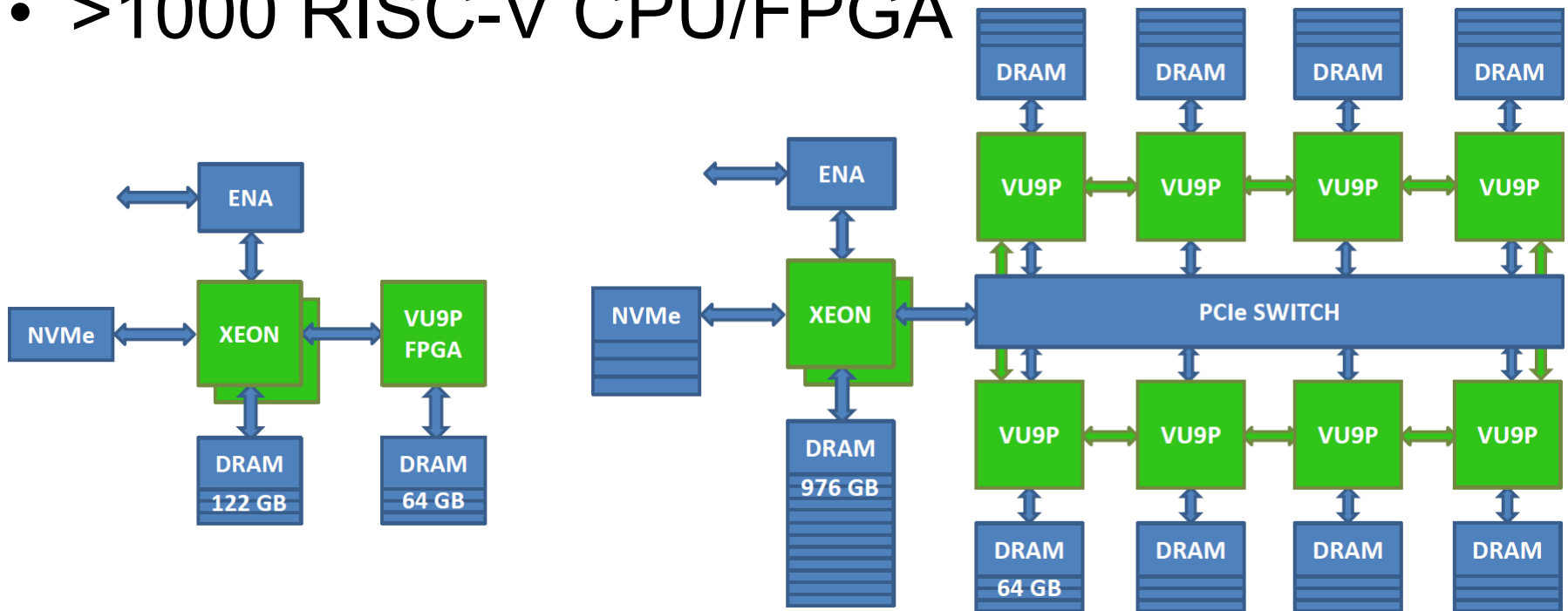
GRVI cluster as PE

- A RISC-V egyszerűsített CPU felépítése
- IMEM és speciális funkciók közösítettek



Példa implementáció 1

- Felhő környezetben, Amazon AWS EC2 F1
- Konfigurálható egységek
- F1.2 1 VU9P FPGA F1.16 8 VU9P FPGA
- >1000 RISC-V CPU/FPGA



Példa implementáció 2

- HotChips 31 2GRVI-Phalanx Poster
- JanGray, Gray Research LLC, <http://fpga.org>
- 64 bites RISC-V RV64I CPU-k