

Aszinkron sín:

Minden eseményt egy előző esemény okoz!

Nincs órajel, **WAIT**, van viszont:

MSYN# (kérés - Master SYNchronization),
SSYN# (kész - Slave SYNchronization).

Ugyanazon a sínen gyors és lassú mester - szolga pár is lehet.

Máté: Architektúrák 4. előadás 1

Aszinkron sín működése (3.39. ábra)
 Akkor indulhat újabb tranzakció, ha **SSYN#** negált.

Ugyanazon a sínen gyors és lassú mester - szolga pár is lehet.

Máté: Architektúrák 4. előadás 2

Teljes kézfogás (full handshake):

Akkor indulhat, ha **SSYN#** negált!

- Mester: kívánságok beállítása, majd **MSYN#**, vár,
- Szolga: látja **MSYN#**-t: dolgozik, majd **SSYN#**, vár,
- Mester: látja **SSYN#** -t (a szolga kész), dolgozik, ha kell, majd negálja **MSYN#** -t,
- Szolga: látja **MSYN#** negálását, negálja **SSYN#** -t.

Máté: Architektúrák 4. előadás 3

Sínütemezés (kiosztás)

Ha egyszerre többen is igénylik a sítnt (**CPU, I/O vezérlő**), akkor a **sínütemező** (bus arbiter) dönt.

Általában **I/O** elsőbbséget kap (cikluslopás).

Máté: Architektúrák 4. előadás 4

Sínütemezés (kiosztás – bus arbitration)

- **Centralizált (3.40. (a) ábra):** (margaréta) láncolás (daisy chaining), egy vagy többszintű lehet.

3.40 (a) ábra. Egy centralizált egyszintű sínütemező láncolt engedélyezéssel

Az engedélyező jel lehet, hogy továbbterjed a B/K eszközök láncán, lehet, hogy nem.

B/K eszközök

Máté: Architektúrák 4. előadás **3.40a.swf** 5

Ha van **kérés** és a **sín szabad: sín foglalási engedély**.

3.40.(b) ábra. Egy centralizált kétszintű sínütemező láncolt engedélyezéssel

Ütemező

B/K eszközök

Néha további vezeték van az engedély fogadásának jelzésére (újabb sín kérés kezdődhet a sín használata közben).

Máté: Architektúrák 4. előadás **3.40b.swf** 6

- decentralizált**
- pl. 16 prioritású: 16 eszközhöz 16 kérés vonal, minden eszköz minden kérés vonalat figyel, tudja, hogy a saját kérése volt-e a legmagasabb prioritású.
- **3.41. ábra:** ha nem **foglalt** és **be**, akkor kérheti a sít (ki negálása, **foglalt** beállítása).

3.41. ábra. Decentralizált sinitűvezés

Máté: Architektúrák 4. előadás **3.41.swf** 7

PCI sítűvezés (kiosztás): request, grant.

3.54. ábra. A PCI sítű centrális utesztőt használ

Máté: Architektúrák 4. előadás 8

Sítű műveletek

Az eddigiek **közönséges sítű műveletek** voltak.

Blokkos átvitel (3.42. ábra): A kezdő memória címen kívül az adatsítűre kell tenni a mozgatandó adatok számát. Esetleges várakozó ciklusok után ciklusonként egy adat mozgatása történik.

Megszakítás kezelés: később tárgyaljuk részletesen.

Több processzoros rendszerekben:
olvasás – módosítás – visszaírás ciklus: szemafor.

Máté: Architektúrák 5. előadás 9

Példák sítűre

Az első **IBM PC (3.37. ábra)** 62 vonalas (vezeték, line), 20 címnek, 8 adatnak + **DMA**, megszakítás ...

PC/AT szinkron sítű (3.51. ábra): további 36 vezeték (címnek összesen 24, adatnak 16, ...).

Microchannel (IBM OS/2 gépekhez), szabadalmak

ISA (Industry Standard Architecture) lényegében 8.33 MHz-es **PC/AT** sítű (sávszélesség: 16.7 MB/s).

EISA (Extended ISA) 32 bitesre bővített **ISA** (sávszélesség: 33.3 MB/s).

Színes TV-hez 135 MB/s sávszélesség kellene (1024*768 pixel, 3 bájt*2, 30 kép/sec).
 lemez → memória → képernyő

Máté: Architektúrák 5. előadás 10

PCI (Peripheral Component Interconnect): 32 bites adat átvitel (33,3 MHz, sávszélesség: 133 MB/s) szabadon felhasználható licenz. Multiplexelt cím- és adatkivezetések.

Új változatai: 64 bites adat, 66 MHz, 528 MB/s.

Problémák:

- a memóriához lassú,
- nem kompatibilis az **ISA** bővítőkártyákkal.

Megoldás (3.52. vagy 2.30. ábra): több sítű
 Belső sítű, **PCI híd**, **PCI sítű**, **ISA híd**, **ISA sítű**.

Máté: Architektúrák 5. előadás 11

2.30. ábra. Egy tipikus PC PCI, SCSI és ISA sítűvel 2000 körül

Máté: Architektúrák 5. előadás **Nem kell** 12

Általános soros sín (USB)**Universal Serial Bus**

Igény: bármikor könnyen, azonos módon lehessen sokféle perifériát kapcsolni a géphez, akár a gép működése közben, hardver ismeretek nélkül:

- ne kelljen kikapcsolni a gépet,
- ne kelljen szétszedni a gépet,
- ne kelljen újra boot-olni,
- ne kelljen áramellátásról gondoskodni,
- ...

Plug 'n Play (csatlakoztasd és működik) perifériák.

Máté: Architektúrák

5. előadás

13

USB (Universal Serial Bus - általános soros sín):

Négy vezeték: adatok (2), tápfeszültség (1), föld (1).

USB 1.0 **1,5 Mbps** (billentyűzet, egér,...)

USB 1.1 **12 Mbps** (nyomtató, fényképezőgép,...)

USB 2.0 **480 Mbps** (DVD lejátszó,...)

A központi elosztó (**root hub**)

1 ms-onként üzenetekkel (**frame, 3.54. ábra**) kommunikál az eszközökkel.

A frissen csatlakoztatott eszköz címe 0.

Ha a központi elosztó tudja fogadni az eszközt, akkor egyedi címet (1-127) ad neki (**konfigurálja**).

Máté: Architektúrák

5. előadás

14

Frame – keret

Egy vagy több csomagból áll.

Az egyes csomagok haladhatnak a központból az eszközök felé vagy fordítva. A haladási irány egy kereten belül is változhat.

Az első csomag mindig **SOF**:

Start Of Frame – keret kezdet, szinkronizálja az eszközöket.

Máté: Architektúrák

5. előadás

15

A keret lehet

- **Control** – vezérlő:
 - Eszköz konfigurálás,
 - Parancs,
 - Állapot lekérdezés.
- **Isochronous** – izoszinkron: valós idejű eszközök használják, pl. telefon. Hiba esetén nem kell ismételni az üzenetet.
- **Bulk** – csoportos: nagy tömegű adat átvitelére szolgál.
- **Interrupt** – megszakítás: Az **USB** nem támogatja a megszakítást, helyette pl. 50 ms-enként lekérdezhető az eszköz állapota.

Máté: Architektúrák

5. előadás

16

A csomag lehet

- **Token** – parancs (központ küldi az eszköznek):
 - **SOF**.
 - **IN** – be: adatokat kér az eszköztől.
Az **IN** parancsban meg lehet adni, melyik eszköztől milyen adatokra van szükség.
 - **OUT** – ki: adatok fogadására kéri az eszközt.
 - **SETUP** – beállítás: konfigurálja az eszközt.

Máté: Architektúrák

5. előadás

17

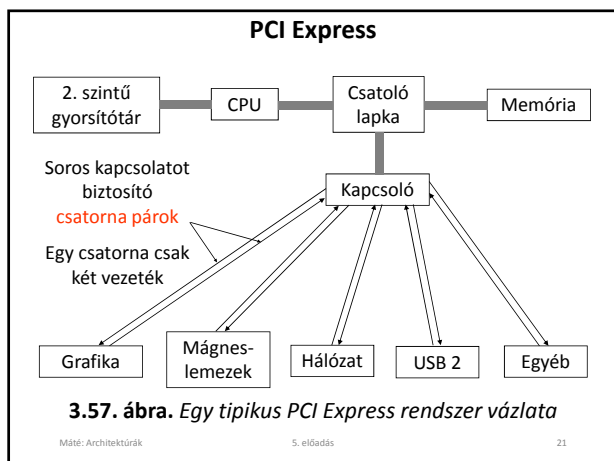
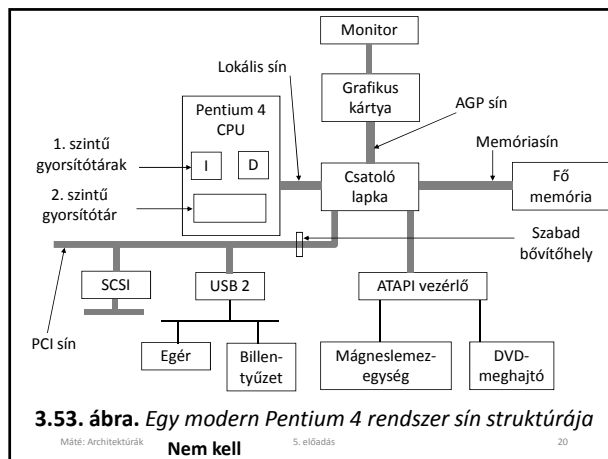
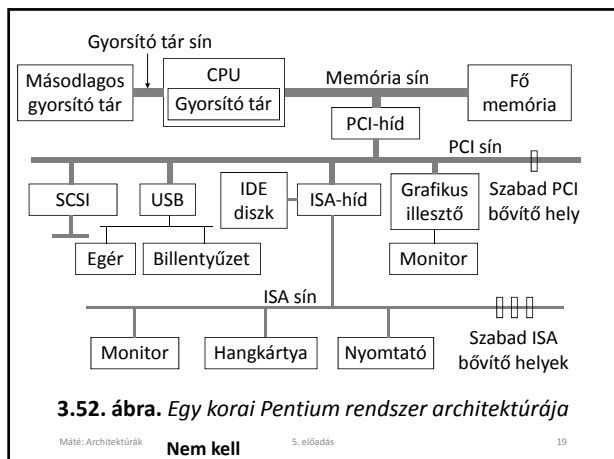
A csomag lehet (folytatás)

- **Data** – adat: 64 bájt információ mozgatása akármelyik irányban. A **Data** csomag részei:
 - **SYN**: 8 bit szinkronizáció,
 - **PID**: a csomag típusa (8 bites),
 - **PAYLOAD**: hasznos adat,
 - **CRC**: Cyclic Redundancy Code – ciklikus redundancia kód (16 bit az adatátvitel helyességének ellenőrzésére).
- **Handshake** – kézfogás:
 - **ACK**: az előző adatcsomagot hibátlanul vettem,
 - **NAK**: **CRC** hibát észleltem,
 - **STALL**: kérem, várjon, el vagyok foglalva.
- **Special** – speciális.

Máté: Architektúrák

5. előadás

18

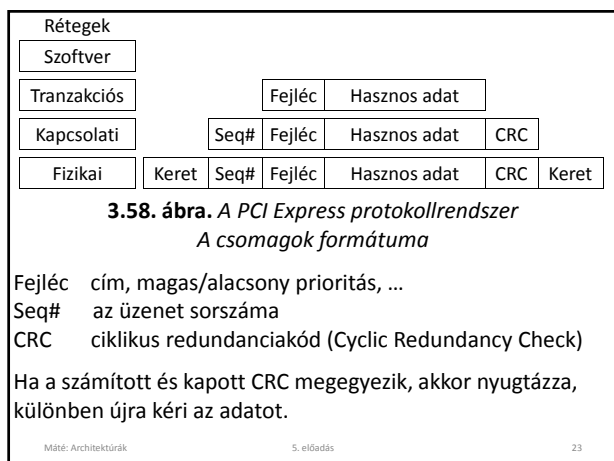


Hagyományos sín	PCI Express
Több leágazású sín	Központosított kapcsoló
Széles, párhuzamos sín	Keskeny, közvetlen soros kapcsolat
Bonyolult mester – szolga kapcsolat	Kicsi, csomagkapcsolt hálózat
	CRC kód: nagyobb megbízhatóság
	A csatlakozó kábel > 50 cm lehet
	Az eszköz kapcsolat is lehet
	Meleg csatlakoztatási lehetőség
	Kisebb csatlakozók: kisebb gép

- Nem kell nagy bővítő kártyával csatlakozni a sínhez
- A winchester a monitorba is kerülhet

Egy csatorna hasznos sávszélessége minimum 2 Gbps, de bíznak benne, hogy hamarosan 10 Gbps

Máté: Architektúrák Nem kell 5. előadás 22



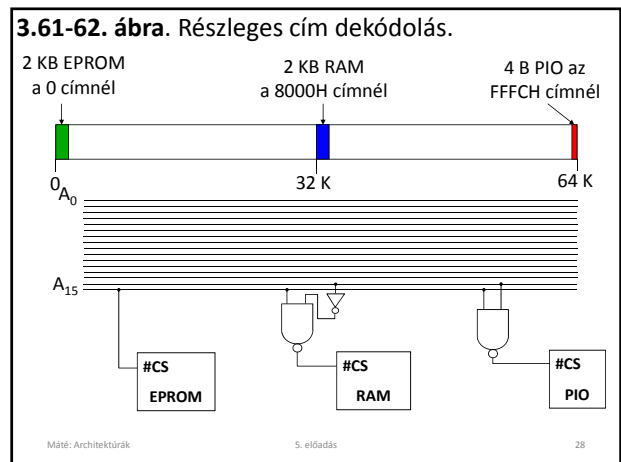
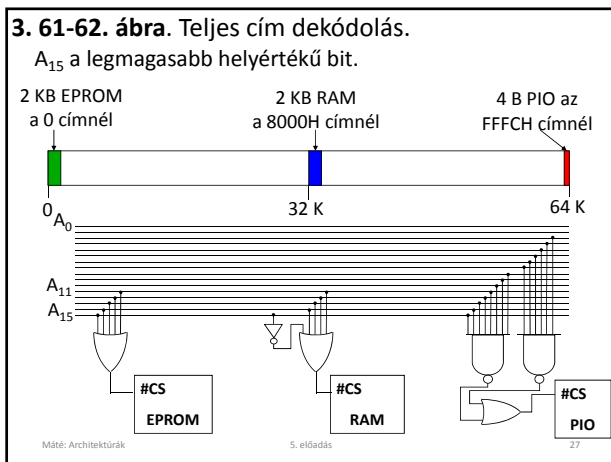
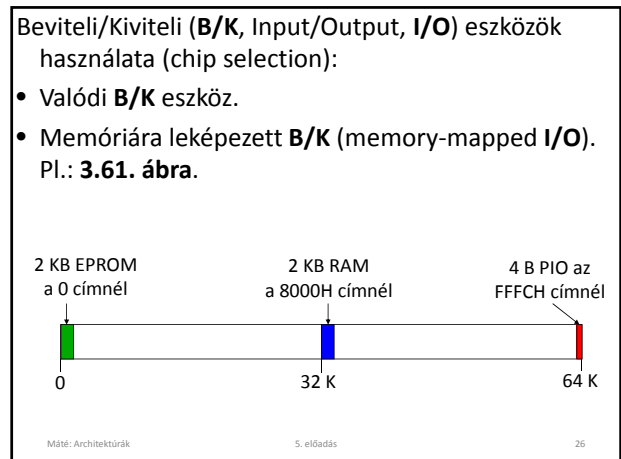
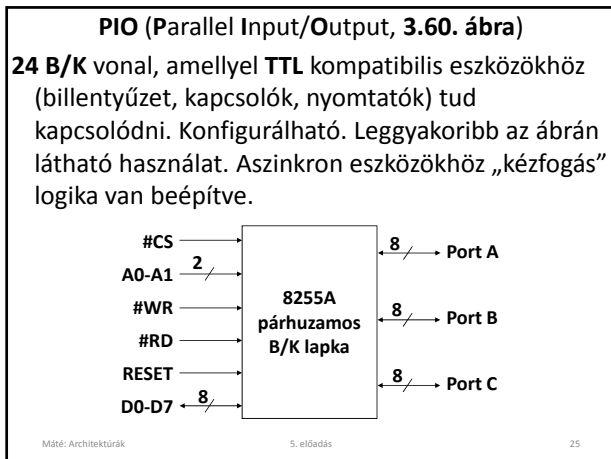
Input, output, interfész, I/O lapkák

UART (Universal Asynchronous Receiver and Transmitter) egy bájtot tud olvasni az adatsínről, és aztán sorosan továbbítja az eszközhöz (vagy fordítva), programmal konfigurálható (belső regiszterének beállításával):

- 5-8 bit szélesség,
- sebesség (50-19.200 bps),
- paritás ellenőrzés (páros, páratlan, nincs).

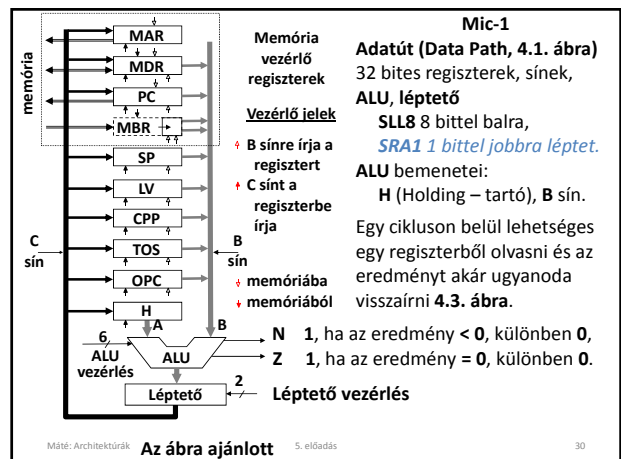
USART (Universal Synchronous Asynchronous Receiver and Transmitter): szinkron és aszinkron módon is tud működni.

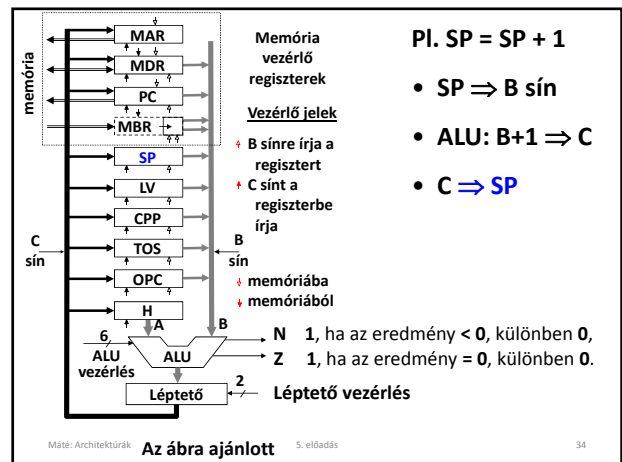
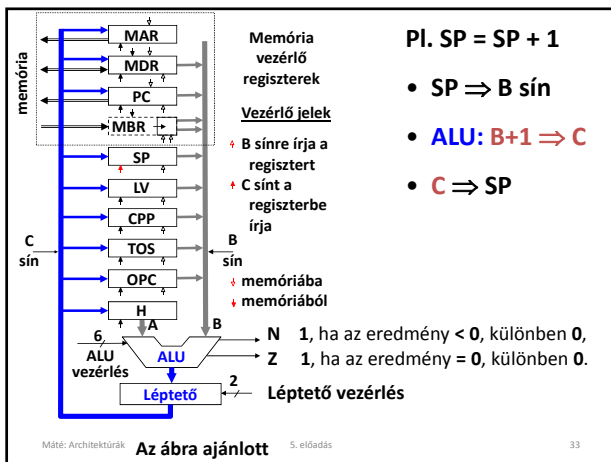
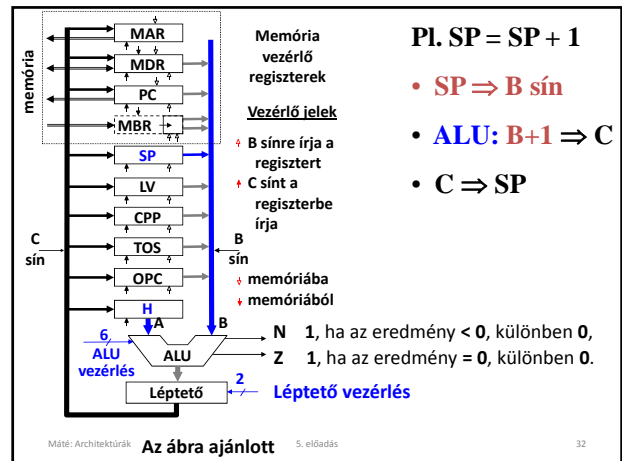
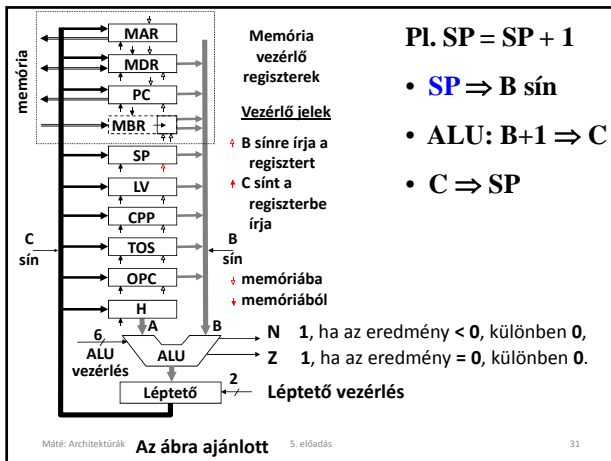
Máté: Architektúrák Nem kell 5. előadás 24



Mikroarchitektúra szint
 Feladata az ISA (Instruction Set Architecture – gépi utasítás szint) megvalósítása.
 Nincs rá általánosan elfogadott, egységes elv.
 A ISA-szintű utasítások „függvények”, ezeket egy főprogram hívja meg végtelen ciklusban.
 A függvények a mikroarchitektúra szintjén valósulnak meg (mikroprogram).
 A mikroprogram változói (a regiszterek) definiálják a számítógép állapotát, pl.: PC (Program Counter, utasításszámláló).

Máté: Architektúrák 5. előadás 29





Memóriaműveletek (4.1. ábra)

- **Szócímezés:** 32 bites adat írása, olvasása.
 szó cím = 4 * (bájt cím),
 a túlcsonduló bitek elvesznek
 MAR (Memory Address Register)
 MDR (Memory Data Register)
- **Bájt címezés:** gépi szintű utasítás bájt olvasás.
 PC (Program Counter): bájt cím,
 MBR (Memory Byte Register): bájt.

MBR kétfajta értelmezése (két vezérlőjel):

- **MBR:** MBR előjel kiterjesztéssel kerül a B síne,
- **MBRU:** MBR előjel nélküli kiterjesztéssel kerül a B síne.

Máté: Architektúrák 5. előadás 35

Az adatút vezérlése (4.1. ábra)

9 jel: a B síne írás a regiszterekből,
 8 jel: 6 az ALU és 2 a léptető vezérlésére,
 9 jel: a C sínről regiszterbe írás,
 3 jel: a memória eléréshez
 2 jel: szó íráshoz/olvasáshoz
 1 jel: bájt olvasáshoz.
 Összesen 29 jel szükséges

A B síne csak egy regiszter írhat egyszerre, ezért 9 helyett elég 4 jel, összesen 24 vezérlő jelre van szükség.

Máté: Architektúrák Az ábra ajánlott 5. előadás 36

Mikroutasítások

24 bit: az adatút vezérléséhez
 9 bit: a következő utasítás címének megadásához,
 3 bit: a következő utasítás kiválasztásának módjára.
 Ez adja a 36 bites mikroutasítást: **4.5. ábra**.

9	3	8	9	3	4
NEXT ADDRESS	JMPC JAMIN JAMIZ	SLL8 SRA1	F0 F1 ENA ENB INVA INC	H OPC TOS LV SP PC MDR MAR	WRITE READ FETCH
Addr	JAM	ALU	C	Mem	B sín

0 = MDR	1 = PC	2 = MBR	3 = MBRU	4 = SP
5 = LV	6 = CPP	7 = TOS	8 = OPC	9-15 semmi

Máté: Architektúrák Az ábra ajánlott 6. előadás 37

SP ⇒ B sín; ALU ⇒ C; C ⇒ SP

4.1 ábra. Mikroarchitektúra-példa adatútja

Máté: Architektúrák Az ábra ajánlott 5. előadás 38

mic1 1.swf

Memória ciklus

A memória ciklus az adatút végén kezdődik (a C sínről a regiszterek feltöltése után), ezért ha a memória címet módosította ez a mikroutasítás, akkor a memória cím a módosított **MAR** ill. **PC** regiszter értéke lesz.

Az olvasás eredménye csak két ciklussal később használható az **ALU**-ban, mert **MDR** ill. **MBR** csak a következő adatút ciklus vége felé töltődik fel a memóriából, addig **MDR** ill. **MBR** régi értéke érhető el.

Máté: Architektúrák 5. előadás 39

Memória ciklus: MAR = LV; rd; TOS = MDR

4.1 ábra. Mikroarchitektúra-példa adatútja

Máté: Architektúrák Az ábra ajánlott 5. előadás 40

mic1 1.swf

MAR = SP; rd Nem megengedett! MDR a memóriából és a C sínről egyszerre kapna értéket!
MDR = LV

4.1 ábra. Mikroarchitektúra-példa adatútja

Máté: Architektúrák Az ábra ajánlott 5. előadás 41

mic1 1.swf

Feladatok

Mi a sín időzítés, és miért fontos?
 Hogy működik a szinkron/aszinkron sín?
 Mire szolgál a mester/szolga szinkronizáció?
 Mi a teljes kézfogás?
 Milyen sín ütemezőket ismer?
 Hogy működik a centralizált sín ütemező?
 Hogy működik a decentralizált sín ütemező?
 Milyen sín műveleteket ismer?
 Miért előnyös a blokkos átvitel?

Máté: Architektúrák 4. előadás 42

Feladatok

Milyen sít ismer?
 Ismertesse az **ISA** sít!
 Ismertesse az **EISA** sít!
 Ismertesse az **PCI** sít!
 Hogy használható több sít egy gépen belül?
 Miért volt fontos az **ISA** sít megtartása fejlettebb sít alkalmazása esetén?
 Hogy történik a **PCI** sít ütemezése?
 Miért használnak külön sít a memória eléréséhez?
 Hogy illeszkedik a **PCI** sít a memória sít-höz?

Máté: Architektúrák

5. előadás

43

Feladatok

Milyen igények kielégítésére szolgál az általános soros sít (**USB**)?
 Milyen vezetékeket tartalmaz az **USB** sít, és miért?
 Miért kényelmes az **USB** sít használata?
 Mi a központi elosztó (root hub), és hogy működik?
 Mi történik eszköz **USB** porthoz csatlakoztatásakor?
 Mit jelent egy eszköz konfigurálása?
 Mi a keret (frame), és milyen kereteket ismer?
 Mi a csomag, és milyen csomagokat ismer?
 Milyen irányban haladhatnak a csomagok?

Máté: Architektúrák

5. előadás

44

Feladatok

Mire szolgál a **SOF** csomag?
 Mire szolgál az **IN/OUT** csomag?
 Mire szolgál az **ACK/NAK** csomag?
 Mi a **CRC**?
 Mire szolgál a **PCI Express**?
 Hogy kapcsolódik a **PCI Express** a **CPU**-hoz?
 Hogy kapcsolódik a **PCI Express** az eszközökhöz?
 Milyen előnyei vannak a **PCI Express**-nek a sít-nel szemben?
 Milyen rétegei vannak a **PCI Express** protokollnak?

Máté: Architektúrák

5. előadás

45

Feladatok

Mi az **UART**?
 Mire szolgál az **UART**?
 Paritás ellenőrzés szempontjából hogy működhet az **UART**?
 Mi az **USART**?
 Milyen digitális áramkörök segítségével tudja megvalósítani a párhuzamos-soros átalakítást?
 Mit jelent a **CS#** jelölés egy chip lábán?
 Mit jelent a **PIO** rövidítés?
 Milyen lábai vannak **PIO** lapkának?
 Hány regisztere van **PIO** lapkának?

Máté: Architektúrák

5. előadás

46

Feladatok

Mi biztosítja, hogy kapcsolók állapotának lekérdezésére és lámpák működtetésére is használható a **PIO** lapka?
 Mit jelent a valódi/memóriára leképezett **I/O**?
 Melyiknek mi az előnye, hátránya?
 Mit nevezünk teljes/részleges cím dekódolásnak?
 Melyiknek mi az előnye, hátránya?
 Mit nevezünk port-nak?

Máté: Architektúrák

5. előadás

47

Feladatok

Mi a mikroarchitektúra szint feladata?
 Hogy valósítja meg a feladatát?
 Mi az adatút?
 Milyen sítjei vannak a Mic-1 -nek?
 Mely regiszterek csatlakoznak az **A** sít-höz?
 Mely regiszterek csatlakoznak a **B** sít-höz?
 Mely regiszterek csatlakoznak a **C** sít-höz?
 Milyen jelek szükségesek az adatút vezérléséhez?
 Mire szolgál az **SP** regiszter két vezérlő bemenete?
 Mire szolgál az **MBR** regiszter két vezérlő bemenete?

Máté: Architektúrák

5. előadás

48

Az előadáshoz kapcsolódó**Fontosabb témák**

A CPU részei, feladatai, adatút
A CISC és a RISC kialakulása
Utasítás és processzor szintű párhuzamosítás
Adat típusok, karakter kódolás
CPU, Sínek. Sín protokoll. Mester – szolga. Sín vezérlő,
vevő, adóvevő. Sáv szélesség, sín szélesség.
Sokszorozott sín.
Sín időzítés: szinkron, aszinkron sín,
teljes kézfogás.
Sín ütemezés.

Máté: Architektúrák

4. előadás

49

Az előadáshoz kapcsolódó**Fontosabb témák**

Sín műveletek.
Általános soros sín. PCI express
Parallel Input/Output (PIO), valódi és memóriára
leképezett I/O eszköz. Teljes és részleges cím
dekódolás
Mikroarchitektúra szint. Mikroutasítások: Mic-1. A
Mic-1 működése. Memória műveletek. ALU és az
adatút vezérlése.

Máté: Architektúrák

5. előadás

50