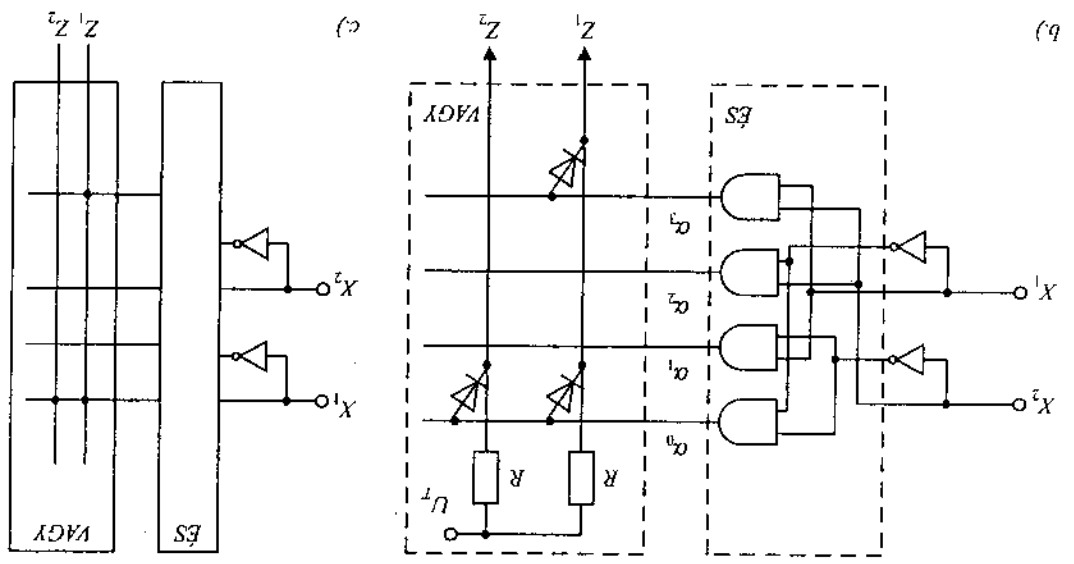
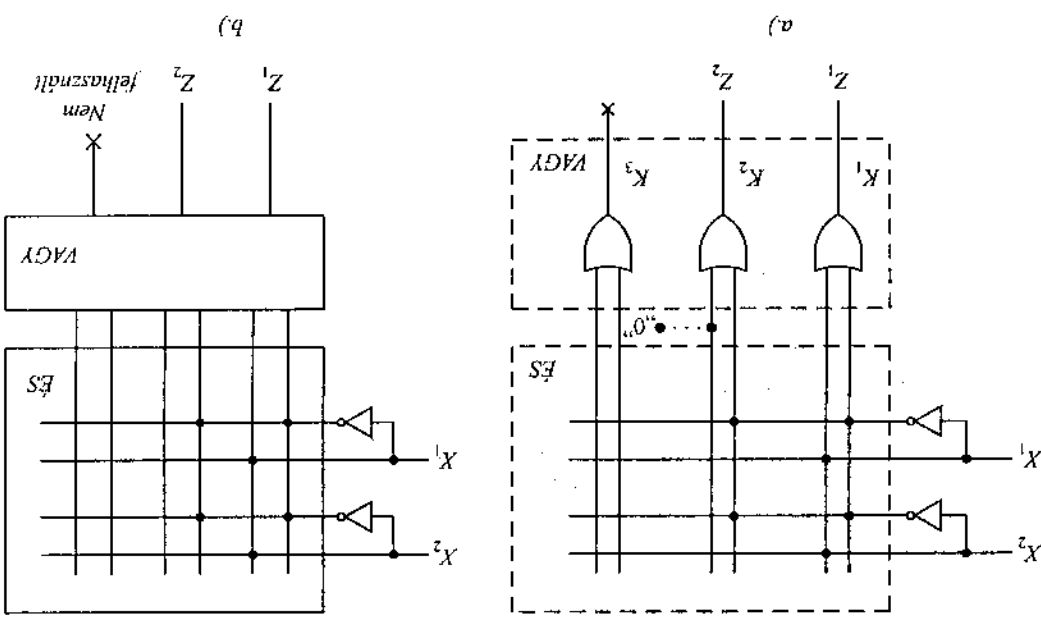
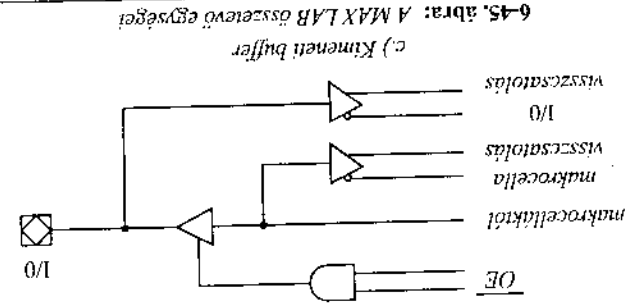
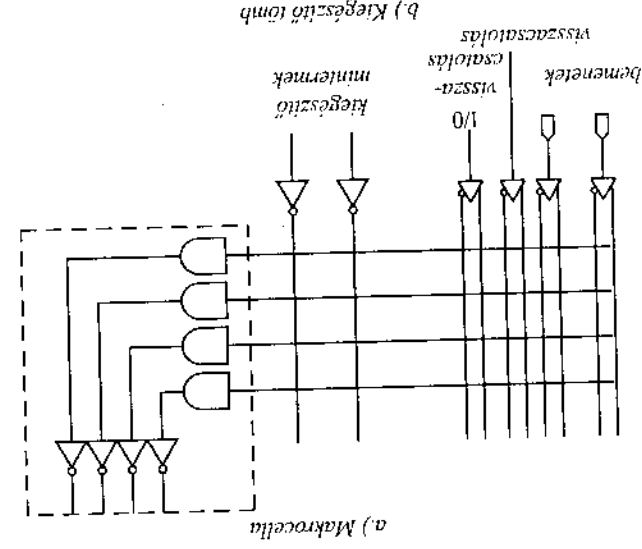
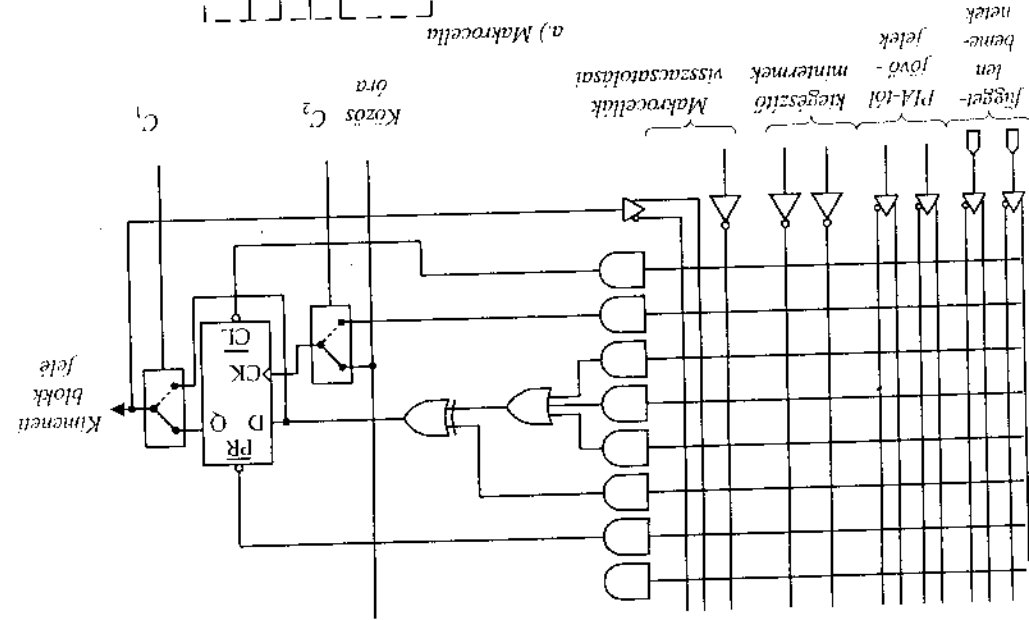


Kapcsolómatrix, mint logikai műveletvégző ROM



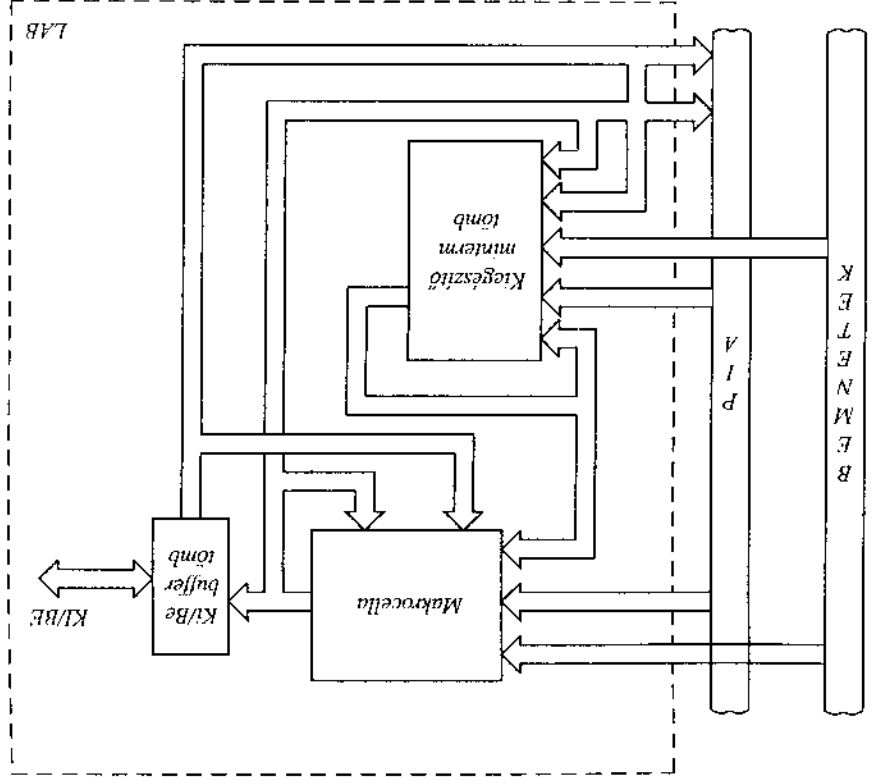
Kapcsolómatrix, mint logikai műveletvégző FAL





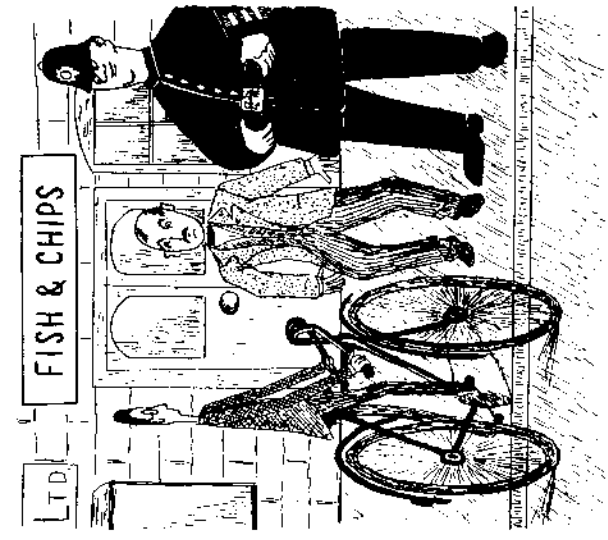
6-45. ábra: A MAX LAB összevő egységei

6-46. ábra: MAX LAB belső felépítése



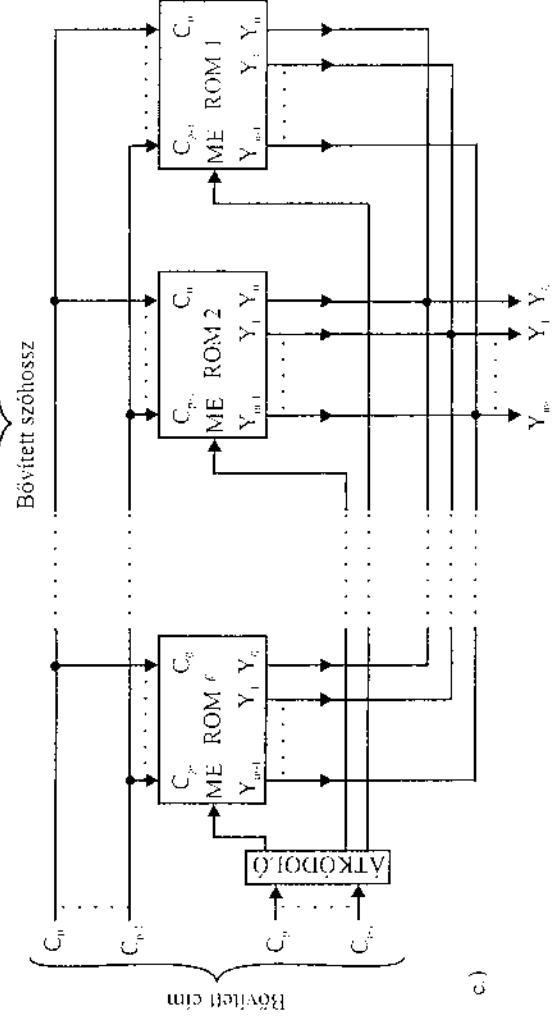
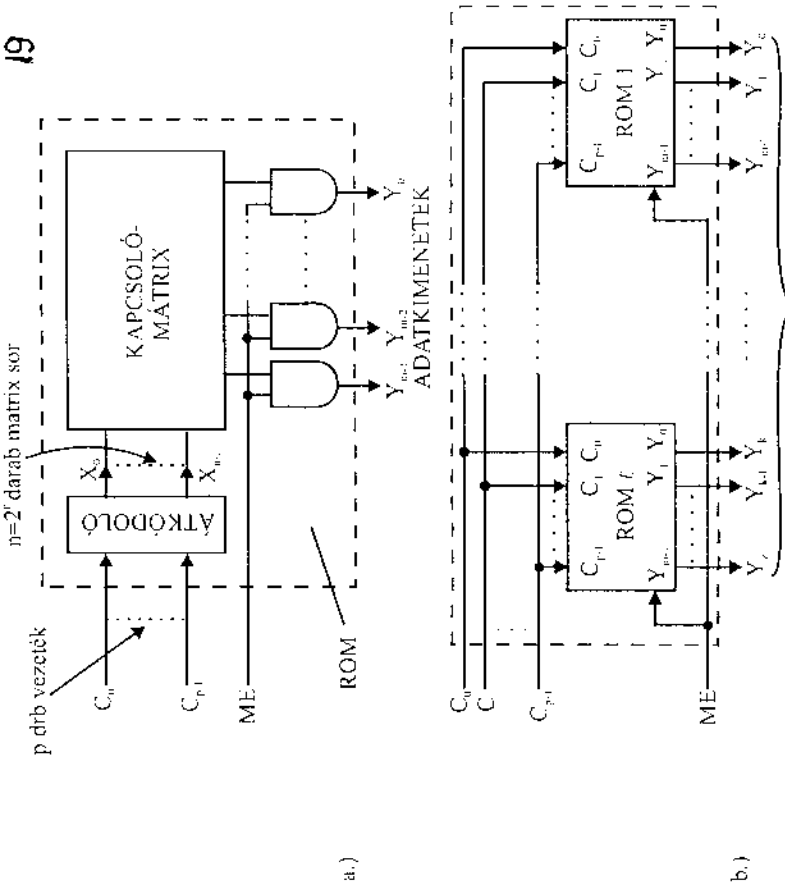
A makrocella kimenetére egy *kimeneti buffer* csatlakozhat, amennyiben nem belső feladatokat lát el (6-45c. ábra). Itt a kimenetek három állapotúak, a bemenetek fixen csatlakoznak a cellához. A felsorolt egységeket egy LAB tömb foglalja magába, mely valójában a MAX-struktúra alapeleme (6-46. ábra). A LAB-modul flexibilisen módosítható. A több LAB-ot tartalmazó MAX-struktúrát a PIA kapcsolja össze egyetlen rendszerre. Egy képzített MAX-struktúrát szemléltet a 6-47. ábra.

A bemutatott rendszer alkalmazása elképzelhetően szoftver határ nélkül, erre a „MAX+PLUS” programot fejlesztették ki, melynek segítségével gyakorlatilag minden típusú digitális hálózat megtervezhető. A kindulás történhet állapotgráfokból, igazságtáblázatokból, vagy logikai formulákból is.



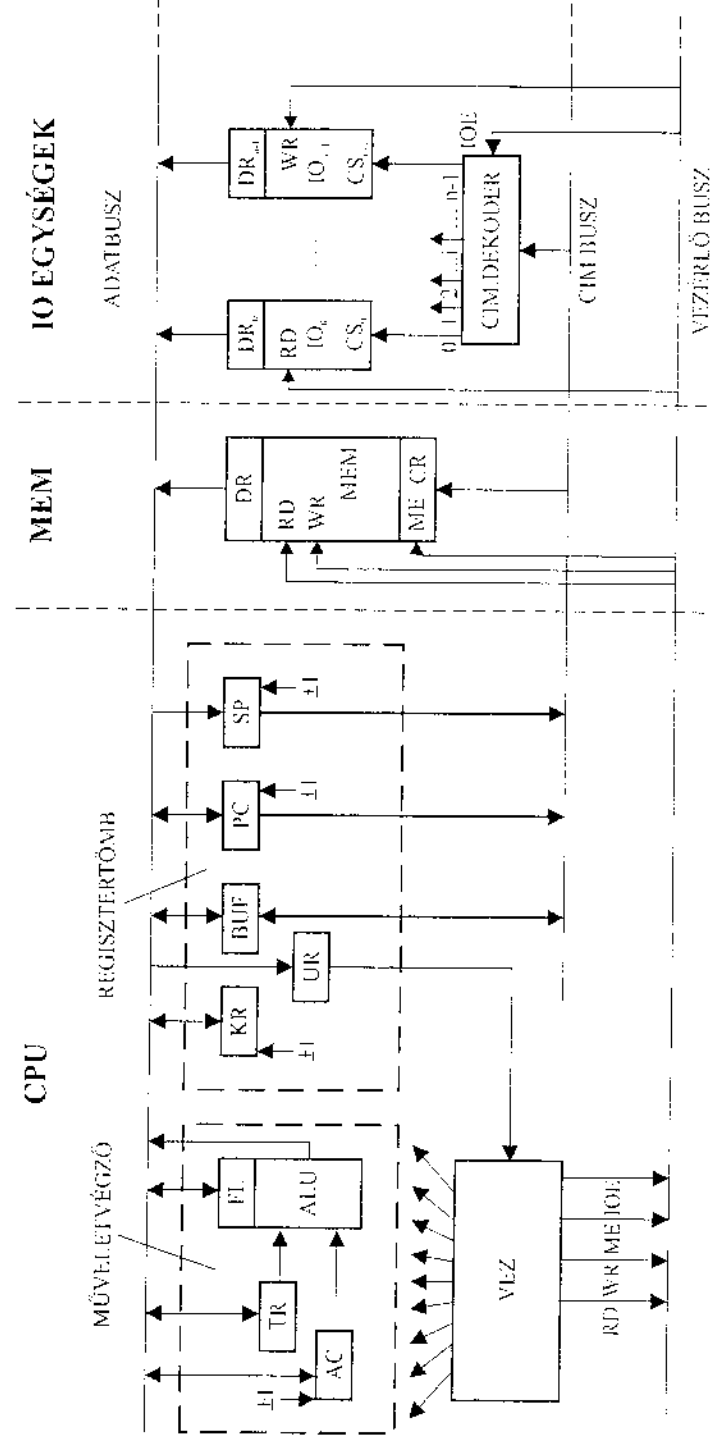
Hihetetlenül megrövidült

George Gamow:
Tompkins úr kalandjai a fizikával,
Gondolat, Budapest 1976.



Szóhossz-, illetve kapacitásbővítési elv szemléltetése
adott ROM - memória modul felhasználásával

- Funkcionális egységek:**
- Regiszterek
 - Kódátalakítók
 - Demultiplexerek
 - Multiplexerek
 - Számlálók
 - Aritmetikai műveletvégzők
 - Digitális komparátorok
 - Memóriák
 - D-A átalakítók
 - A-D átalakítók



Egy szoftver-hardver vegyes alkalmazásán alapulva szerkesztendő univerzális digitális rendszer blokkvázlata

CIKLUS	ELEMI HARDVER	MAGYARAZAT
--------	---------------	------------

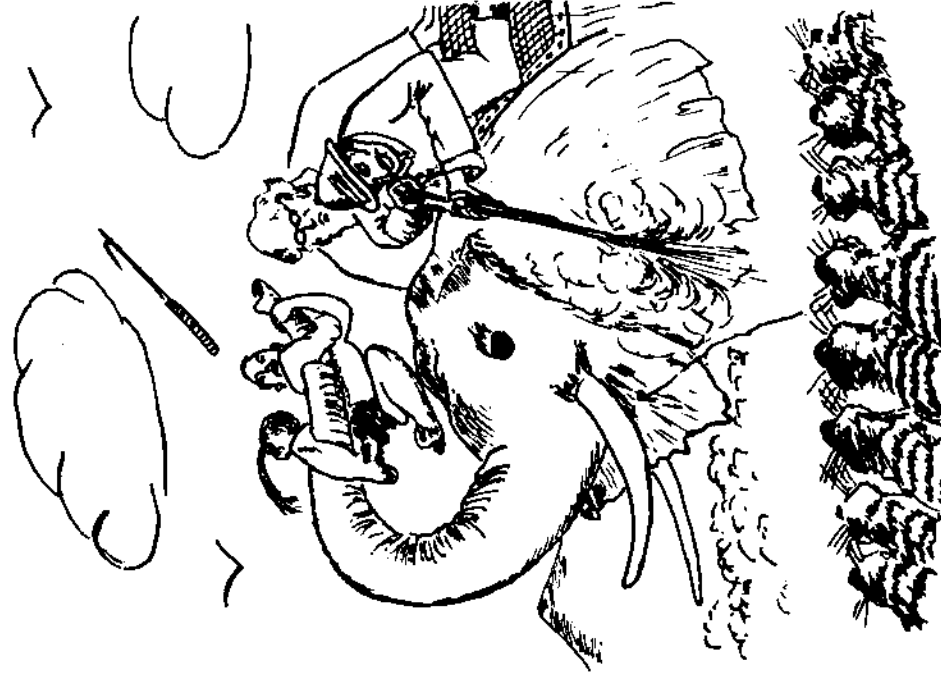
1.	(FETCH)	<p>CR ← PC</p> <p>Az MK programozó-beli címét átírjuk CR-be</p> <p>INR PC</p> <p>DR ← (CR)</p> <p>UR ← DR</p> <p>VEZ ← UR</p> <p>rekeszbeli CIM1 kiolvasásához (INR PC = PC + 1)</p> <p>Kiolvaszuk MK-1 a programmezőből</p> <p>Atvisszük MK-1 az UR-be</p> <p>Betöltyük MK-1 a VEZ-be, ahol a vezérlő komparálásal felismeri, hogy MK = LDA, ezután a gép már tudja, hogy milyen típusú utasítást hajt végre.</p> <p>Az eddig lezajlottakat UTASÍTÁS-LEHÍVÁSI CIKLUSnak, más néven: FETCH-nek nevezzük. Minden a FETCH-csel kezdődik.</p>
----	---------	--

2.	CR ← PC	<p>INR PC</p> <p>CIM1-et tartalmazó rekesz címét átírjuk CR-be</p> <p>PC-t előkészítjük a soron következő utasítás MK-jának ki-olvasásához</p> <p>Kiolvaszuk CIM1-et a programmezőből, mivel CIM1 mu-tatja meg OP1 helyét az adatmezőben, ezért</p> <p>BUF → DR</p> <p>DR-t a BUF-regiszter segítségével átölytjük CR-be, így előkészítetjük az OP1 kiolvasását</p>
3.	DR ← (CR)	<p>AC ← DR</p> <p>OP1 kiolvasása az adatmezőből</p> <p>OP1 átöltése AC-be és ezzel az AC → OP1 művelet befe-jeződött</p>

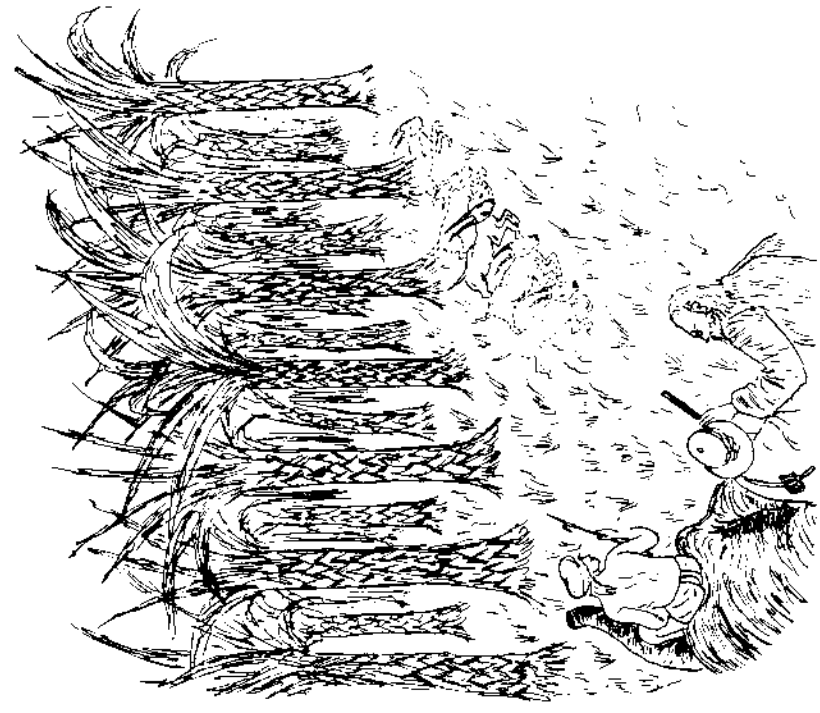
A- LDA CIM1 utasítás végrehajtásának részleteit mozzanatai a HM-egységeknél

CIKLUS	ELEMI HW	MAGYARAZAT
1.	FETCH	– Mindig egyforma
2.	:	– A 2. ciklus is megegyezik, csak itt CIM2 szerepel
3.	DR ← (CR)	<p>– OP2 kiolvasása az adatmezőből</p> <p>– OP2 átöltése TR-be</p> <p>– TR → DR</p> <p>– AC → AC+TR</p>

A- ADD CIM2 utasítás végrehajtásának részleteit



Egy nagy, bazontos tigriscorda támadta meg elefántjukat



Sir Richard Ióvesre készen állt, amikor a professzor megállította