Decodificare memorie — Document detaliat

Acest document conține *toate* tabelele, deducțiile, ecuațiile logice pas-cu-pas, tabele adevăr, hărți Karnaugh (conceptuale), implementări practice și note hardware pentru decodificarea memoriei într-un sistem 8086 cu adresare fizică pe 20 biți (00000h–FFFFFh).

---

1. Cerințe și premize

- Adrese pe 20 biți (A19..A0).
- Vector de reset al 8086 (FFFF0h) trebuie să fie în memorie ROM/EPROM.
- Vom folosi o hartă rezonabilă (compatibilă cu multe proiecte didactice):
- **SRAM**: 64 KB la intervalul `00000h` – `0FFFFh` (A19..A16 = 0000)
- **EPROM**: 128 KB la intervalul `E0000h` – `FFFFFh` (A19..A17 = 111, A16 = don't-care pentru bloc de 128 KB)

  Dacă aveți alte cerințe, modificați doar valorile din tabele — ecuațiile și metoda rămân aceleași.

---

2. Tabele cu intervale (cerință explicită: adrese de început și final trebuie incluse)

2.1 SRAM (64 KB)

- Start: `00000h` (0000 0000 0000 0000 0000b)
- End: `0FFFFh` (0000 1111 1111 1111 1111b)
- Dimensiune: `0x10000` = 65536 bytes (64 KB)
- Note: intervalul acoperă toate combinațiile A19..A16 = `0000`; offset intern = A15..A0.

2.2 EPROM (128 KB)

- Start: `E0000h` (1110 0000 0000 0000 0000b)
- End: `FFFFFh` (1111 1111 1111 1111 1111b)
- Dimensiune: `0x20000` = 131072 bytes (128 KB)
- Note: intervalul acoperă A19..A17 = `111`; A16 = 0 sau 1 (două blocuri consecutive de 64 KB). Offset intern = A15..A0 (pentru un chip de 64KB).

---

3. Tabele de deducție (stilul cerut)

Mai jos găsiți tabele în stilul slide-urilor pe care le-ați trimis — clare, cu X (don't care) și cu toate coloanele relevante.

3.1 Tabel funcționare decoder 3-to-8 (format îmbunătățit)

Acesta este tabelul în stilul exact al celui din imagine (ordine clară, valori frumos aliniate, X = don't care). Arată funcționarea pentru un decoder tipic **74HC138**: activ când `E3 = 1`, `/E2 = 0`, `/E1 = 0`. Ieșirile sunt **active-low**.

E3	/E2	/E1	A2	A1	A0	/Y7	/Y6	/Y5	/Y4	/Y3	/Y2	/Y1	/Y0
1	0	0	0	0	0	1	1	1	1	1	1	1	**0**
1	0	0	0	0	1	1	1	1	1	1	1	**0**	1
1	0	0	0	1	0	1	1	1	1	1	**0**	1	1
1	0	0	0	1	1	1	1	1	1	**0**	1	1	1
1	0	0	1	0	0	1	1	1	**0**	1	1	1	1
1	0	0	1	0	1	1	1	**0**	1	1	1	1	1
1	0	0	1	1	0	1	**0**	1	1	1	1	1	1
1	0	0	1	1	1	**0**	1	1	1	1	1	1	1
0	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	1	X	X	X	1	1	1	1	1	1	1	1

3.2 Tabel stare/control pentru circuitul 8251 (stil tablă)

Am inclus tabelul de stare pentru 8251 (conform slide-ului trimis), transpus în format tabelar clar:

/CS	/RD	/WR	C//D	Operație
1	X	X	X	Magistrala de date în a 3-a stare
0	1	1	X	Magistrala de date în a 3-a stare
0	0	1	1	Citire a octetului de stare
0	0	1	0	Citire a datei
0	1	0	1	Scriere a cuvintelor de comandă
0	1	0	0	Scriere a datei

4. Tabel complet de mapping A19..A16 → memorie (format asemănător slide-urilor)

Tabelul afișează toate cele 16 combinații ale bitilor A19..A16 și care circuit de memorie este activ. Coloanele CS indică semnalele active-high (H=1) și active-low (/CS=0) pentru claritate.

A19 A18 A17 A16	Nibbles hex	SRAM CS (H)	/ CS_SRAM	EPROM CS (H)	/ CS_EPROM	Observație
0 0 0 0	0x0	1	0	0	1	SRAM (00000h–0FFFFh)
0 0 0 1	0x1	0	1	0	1	N/A
0 0 1 0	0x2	0	1	0	1	N/A
0 0 1 1	0x3	0	1	0	1	N/A
0 1 0 0	0x4	0	1	0	1	N/A
0 1 0 1	0x5	0	1	0	1	N/A
0 1 1 0	0x6	0	1	0	1	N/A
0 1 1 1	0x7	0	1	0	1	N/A
1 0 0 0	0x8	0	1	0	1	N/A
1 0 0 1	0x9	0	1	0	1	N/A
1 0 1 0	0xA	0	1	0	1	N/A
1 0 1 1	0xB	0	1	0	1	N/A
1 1 0 0	0xC	0	1	0	1	N/A
1 1 0 1	0xD	0	1	0	1	N/A
1 1 1 0	0xE	0	1	1	0	EPROM (E0000h–EFFFFh)
1 1 1 1	0xF	0	1	1	0	EPROM (F0000h–FFFFFh)

5. Ecuații logice (forma finală, clară pentru implementare)

SRAM (00000h–0FFFFh) - `CS_SRAM (H) = ¬A19 & ¬A18 & ¬A17 & ¬A16` - `/CS_SRAM = A19 OR A18 OR A17 OR A16`

EPROM (E0000h–FFFFFh) - `CS_EPROM (H) = A19 & A18 & A17` (A16 don't care) - `/CS_EPROM = ¬A19 OR ¬A18 OR ¬A17` (sau folosiți `NAND(A19,A18,A17)` ca implementare hardware)

Selectare 2×27C512 (active-low) - `/CS_EPROM_global = NAND(A19,A18,A17)` - `/CE_0 = /CS_EPROM_global OR A16` (E0000h–EFFFFh) - `/CE_1 = /CS_EPROM_global OR ¬A16` (F0000h–FFFFFh)

Selectare 2×32KB RAM pentru 64KB (active-low) - `/CS_SRAM_global = A19 OR A18 OR A17 OR A16` (active-low) - `/CE_RAM0 = /CS_SRAM_global OR A15` (primul 32KB) - `/CE_RAM1 = /CS_SRAM_global OR ¬A15` (al doilea 32KB)

---

6. Note practice și recomandări finale (scurte)

- Verificați semnalele /CS ale componentelor folosite (majoritatea memorii au /CE sau /CS active-low).
- Folosiți decodoare NAND/NOR dedicate pentru semnale active-low; evitați OR-ing pasiv pe placă.
- Asigurați setup/hold timing pentru citire/scriere în coordonare cu CPU.

---

7. Ce pot face în continuare

- Populez tabelul decoder (secțiunea 3.1) cu valorile exacte ale ieșirilor /Y7.. /Y0 conform unui anumit IC (ex: 74HC138) — spuneți dacă folosiți 74HC138 sau alt decoder.
- Generez diagrame schematice (PNG/SVG) care arată conexiunile între A19..A0, decodoare și pinii /CE ai memoriilor.
- Export document în PDF/Word formatat.