

Soluzioni agli esercizi del Capitolo 4

Questo documento contiene le soluzioni ad un numero selezionato di esercizi del Capitolo 4 del libro “Calcolatori Elettronici - Architettura e organizzazione”, Mc-Graw Hill 2017.

Sarò grato a coloro che mi segnaleranno errori di qualunque genere. Sarò altrettanto grato a chi mi segnalerà errori nel libro.

Vorrei invitare coloro che avessero sviluppato soluzioni alternative a quelle da me proposte, o soluzioni a esercizi non compresi tra quelli qui trattati, a trasmettermele, in modo da migliorare i contenuti di questo sito.

Inviare le segnalazioni a: giacomo.bucci@unifi.it

Aggiornato il 18 aprile 2017

4.2 Le ragioni sono spiegate nella seconda parte del Paragrafo 10.3.5.

4.4 Si supponga di avere il numero esadecimale 1234. Stabiliamo che sulla macchina che segue lo schema little-endian il byte contenente 34, si trovi in posizione i , per cui per cui il byte contenente 12 sarà in posizione $i + 1$. La macchina little-endian trasmette prima il byte all'indirizzo i , che la macchina big-endian trasferisce in memoria alla posizione j . Successivamente viene trasmesso il byte in $i + 1$ che viene memorizzato in posizione $j + 1$. A questo punto sulla macchina big-endian il numero viene interpretato come 3412.

4.5 Cominciamo con la realizzazione della memoria richiesta (64 MB con parole di 16 bit) con la SRAM AS7C34096 (512 KB).

Nella versione con tempo di accesso di 12 ns, il costruttore indica un assorbimento di corrente $I_{cc} = 0,200$ A. Tenuto conto della tensione di alimentazione di $V_{cc} = 3,3$ V, La potenza assorbita da un singolo componente P_d è pari a:

$$P_d = V_{cc} \cdot I_{cc} = 0,2 \cdot 3,3 = 0,660 \text{ W}$$

Il numero di componenti richiesti è pari a:

$$n = 2^{26} / 2^{19} = 2^7 = 128$$

per cui la potenza totale dissipata è pari a:

$$P = 0,66 \cdot 128 = 84,48$$

W ovvero una corrente totale assorbita pari a

$$I_T = P / V_{cc} = 84,5 / 3,3 = 25,6 \text{ A} \quad (4.7)$$

Anche a prescindere da considerazioni di costo, una simile corrente sconsiglia l'uso della RAM statica. Va inoltre considerato che gli integrati sarebbero ben 128, occupando una superficie circa 4 volte quella occupata dai 32 integrati necessari per la stessa memoria in RAM dinamica (vedi sotto).

Vediamo ora cosa succede usando il componente AS4C1M16E5. Questa RAM dinamica ha una capacità di 2 MB e consuma 0,740 W. Per ottenere 64 MB occorrono 32 integrati e pertanto la potenza dissipata totale è

$$32 \cdot 0,74 = 23,68 \text{ W}$$

per un assorbimento di corrente pari a: $I_T = P_D/V_{cc} = 23,7/5 = 4,74 \text{ A}$ circa 5 volte più piccola della corrente assorbita dalla soluzione con SRAM, con circa un quarto di occupazione sulla scheda. (La tensione di alimentazione di questo integrato è 5 V.)

Dal confronto precedente risultano evidenti le motivazioni per le DRAM.

4.6 Anzitutto osserviamo che (a partire dal basso), se si vuole rendere semplice la logica di selezione, non sono immediatamente impiegabili gli integrati di 128 e 256 KB, in quanto la memoria deve iniziare dopo i primi 64 KB e l'impiego di tali integrati richiederebbe che nella logica di selezione intervenissero anche le più significative linee di indirizzamento interno all'integrato. Pertanto, conviene disporre, come primo dal basso, un integrato da 64 KB, il quale termina a un confine di 128 KB. In tal modo di seguito può essere impiegato un integrato da 128 KB, la cui ultima posizione risulterà a un confine di 256 K. Quindi, a seguire, possono essere impiegati i due integrati da 256 KB e, di seguito, il restante integrato da 128 KB e il restante da 64 KB.

Si ha quindi la disposizione di Figura 4.1. Agli integrati 1 e 6 vengono portate, sui

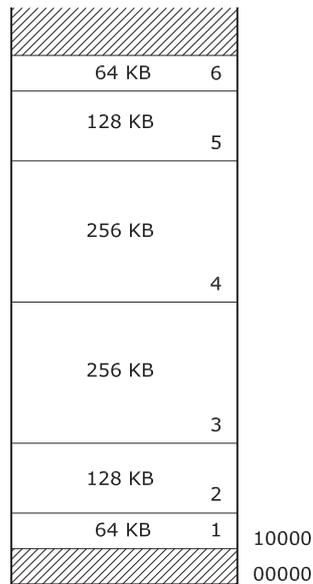


Figura 4.1 Disposizione degli integrati per l'Esercizio 4.6.

pin di indirizzo, le linee $A_{15}-A_0$ del bus indirizzi; agli integrati 2 e 5 le linee $A_{16}-A_0$; agli integrati 3 e 4 le linee $A_{17}-A_0$. I *chip select* sono dati dalle seguenti espressioni.

$$\begin{aligned} CS_1 &= \overline{A_{21}} \cdot \overline{A_{20}} \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \\ CS_2 &= \overline{A_{21}} \cdot \overline{A_{20}} \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot A_{17} \\ CS_3 &= \overline{A_{21}} \cdot \overline{A_{20}} \cdot A_{19} \cdot \overline{A_{18}} \\ CS_4 &= \overline{A_{21}} \cdot \overline{A_{20}} \cdot A_{19} \cdot A_{18} \\ CS_5 &= \overline{A_{21}} \cdot \overline{A_{20}} \cdot A_{19} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \\ CS_6 &= \overline{A_{21}} \cdot \overline{A_{20}} \cdot A_{19} \cdot \overline{A_{18}} \cdot A_{17} \cdot \overline{A_{16}} \end{aligned}$$

4.7 Il blocco di 576 KB può essere realizzato con 2 integrati da 256 K più uno da 64 K. Questo blocco inizia alla posizione 100000h e si espande fino alla posizione 18FFFFh. Infatti, la dimensione del blocco vale $80000h + 10000h = 90000h$, in quanto 80000h è la misura del blocco da 512 K, mentre 10000h è la misura del blocco da 64 K. Dunque il blocco in questione parte dall'indirizzo 100000h e termina all'indirizzo 18FFFFh.

Con i restanti integrati si copre uno spazio di 128 K (pari a 20000h), più 3×64 K (pari a $3 \times 10000h$), Per un totale di 320 K (50000h). Dunque il secondo blocco parte dall'indirizzo 200000h e termina all'indirizzo 24FFFFh.

In Figura 4.2 si dà la rappresentazione della suddivisione risultante dello spazio degli indirizzi.

- I numeri associati alle aree coperte individuano i corrispondenti integrati. La ripartizione delle aree occupate corrisponde alla dimensione dei componenti.
- Sono stati individuati tre aree B0, B1 e B2 corrispondenti a una suddivisione in blocchi da 1 MB. Per ragioni di spazio l'area B0 è stata rappresentata in forma ridotta, mentre dell'area B2 si rappresenta solo la parte di interesse. All'interno di B1 e B2 (per la parte di interesse) la figura è in scala. Le aree di B0, B1 e B2 non coperte dai nostri integrati sono tratteggiate;
- Sulla destra vengono riportati i numeri di blocco che corrisponderebbero alle differenti suddivisioni, se esse fossero tutte di quella misura. Più precisamente:
 - se la memoria fosse suddivisa in blocchi da 256 KB, i due integrati da 256 KB (1 e 2) giacerebbero sui blocchi 4 e 5 (numerati in forma assoluta rispetto allo 0), corrispondenti al blocco 0 e 1 (numerati in forma relativa rispetto alla base di B1);
 - se la memoria fosse suddivisa in blocchi da 128 KB, l'integrato da 128KB (4) giacerebbe sul blocco numero 10h (assoluto), ovvero in posizione 0 rispetto al blocco B2;
 - se la memoria fosse suddivisa in blocchi da 64 KB, l'integrato (3) che segue i due da 256 KB giacerebbe sul blocco numero 18h (assoluto), mentre gli ultimi tre integrati da 64 KB (5, 6 e 7), impiegati alla fine della nostra memoria, giacerebbero sui blocchi 22h, 23h e 24h (assoluti), ovvero in posizione 3, 4 e 5 relativamente al blocco B2.

L'individuazione dei numeri di blocco viene utile per la determinazione della codifica dei selettori degli integrati. In Figura 4.2

Nella Tabella 4.1 viene riportata la struttura degli indirizzi con l'indicazione delle linee che intervengono nella determinazione dei CS (*chip select*). I segni “-” individuano i campi di indirizzamento entro gli integrati. Agli integrati da 256 KB vengono portate le linee $A_{17} - A_0$; all'integrato da 128 KB le linee $A_{16} - A_0$, agli integrati da 64 KB le linee $A_{15} - A_0$.

Verifica i CS dei 64 KB

Vale la pena fare alcune osservazioni sulla struttura degli indirizzi. Le linee di indirizzo sono 23. Dunque lo spazio degli indirizzi è di 2^{23} posizioni, pari a 8 M. In altre parole ci sono 8 blocchi da 1 MB, i quali sono individualmente selezionati attraverso i 3 bit più significativi dell'indirizzo. Se ora consideriamo, per esempio il primo blocco da 256 KB (l'elemento 1), i bit $A_{22} - A_{18}$ danno il corrispondente numero di blocco da 256 K posizioni nello spazio degli indirizzi. A sua volta questo numero può essere interpretato come formato di due campi: il campo $A_{22} - A_{20}$, rappresenta il numero di blocco B1 (da 1 MB); il campo $A_{19} - A_{18}$ rappresenta il numero di blocco da 256 K entro il blocco B1.

Allo stesso modo, con riferimento al blocco 6 (da 64 KB), il campo $A_{22} - A_{16}$ rappresenta il numero d'ordine del blocco (cioè 23) nello spazio degli indirizzi suddiviso in

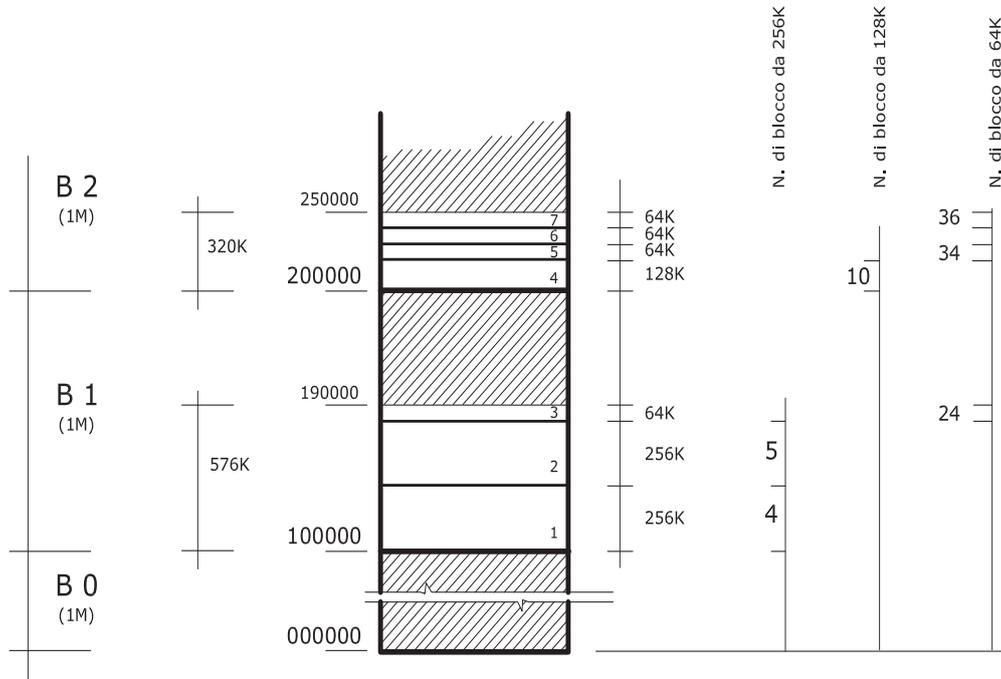


Figura 4.2 Disposizione degli integrati per l'Esercizio 4.7.

	A ₂₂	A ₂₁	A ₂₀	A ₁₉	A ₁₈	A ₁₇	A ₁₆	A ₁₅	A ₁₄	...	A ₁	A ₀
CS ₁	0	0	1	0	0	—	—	—	—	...	—	—
CS ₂	0	0	1	0	1	—	—	—	—	...	—	—
CS ₃	0	0	1	1	0	0	0	—	—	...	—	—
CS ₄	0	1	0	0	0	0	—	—	—	...	—	—
CS ₅	0	1	0	0	0	1	0	—	—	...	—	—
CS ₆	0	1	0	0	0	1	1	—	—	...	—	—
CS ₇	0	1	0	0	1	0	0	—	—	...	—	—

Tabella 4.1 Struttura degli indirizzi per l'Esercizio 105.

porzioni di 64 K; alternativamente il campo A₂₂– A₁₆ può essere riguardato come formato da due sottocampi, di cui A₂₂– A₂₀ contiene il numero del blocco B2 (terzo megabyte nello spazio degli indirizzi), mentre A₁₉– A₁₆ dà il numero d'ordine (3) del blocco da 64 MB entro B1.

La Tabella 4.1 consente di individuare i chip select dei corrispondenti integrati:

$$\begin{aligned}
 CS_1 &= \overline{A_{22}} \cdot \overline{A_{21}} \cdot A_{20} \cdot \overline{A_{19}} \cdot \overline{A_{18}} \\
 CS_2 &= \overline{A_{22}} \cdot \overline{A_{21}} \cdot A_{20} \cdot \overline{A_{19}} \cdot \overline{A_{18}} \\
 CS_3 &= \overline{A_{22}} \cdot \overline{A_{21}} \cdot A_{20} \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot \overline{A_{16}}
 \end{aligned}$$

$$\begin{aligned} CS_4 &= \overline{A_{22}} \cdot A_{21} \cdot \overline{A_{20}} \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \\ CS_5 &= \overline{A_{22}} \cdot A_{21} \cdot \overline{A_{20}} \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot A_{17} \cdot \overline{A_{16}} \\ CS_6 &= \overline{A_{22}} \cdot A_{21} \cdot \overline{A_{20}} \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot A_{17} \cdot A_{16} \\ CS_7 &= \overline{A_{22}} \cdot A_{21} \cdot \overline{A_{20}} \cdot \overline{A_{19}} \cdot A_{18} \cdot \overline{A_{17}} \cdot \overline{A_{16}} \end{aligned}$$

Per completezza in Figura 4.3 mostriamo uno schema che è una via di mezzo tra lo schema logico e lo schema realizzativo. Per passare a quest'ultimo occorrerebbe riportare i componenti con tutti i loro piedini e, per quanto riguarda le porte stabilire i collegamenti tra con i piedini degli integrati che le contengono in modo da rispettare lo schema di figura.

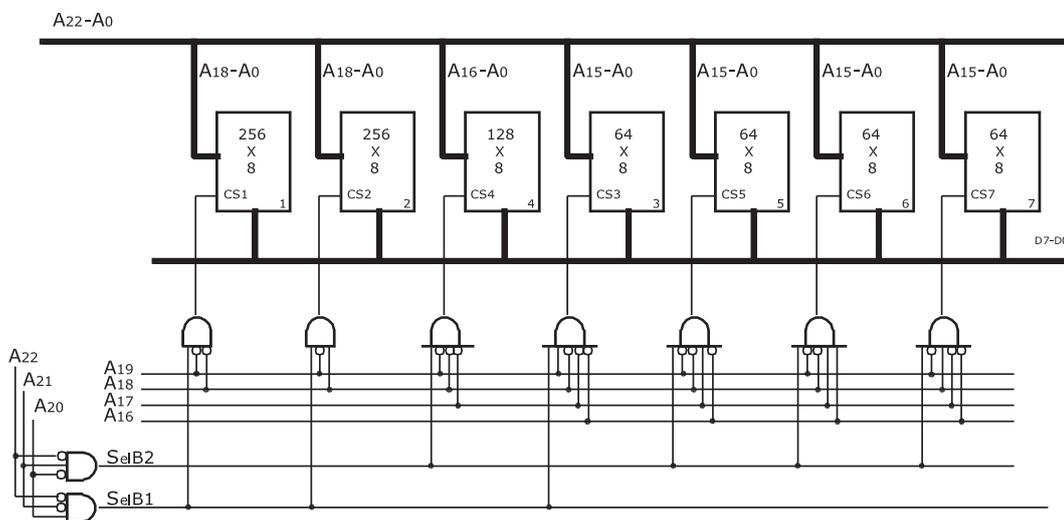


Figura 4.3 Schema complessivo della memoria richiesta per l'esercizio 105.

4.8 Osserviamo anzitutto che il parallelismo di parola non ha alcuna influenza, in quanto per ipotesi gli indirizzi sono assegnati alle parole. Il problema da risolvere è schematizzato in Figura 4.4: si tratta di impiegare un integrato per coprire un blocco che non inizia a un multiplo della sua dimensione.

Nella pratica è bene evitare di dislocare integrati in posizioni diversa dai confini naturali per le complicazioni che ciò comporta (anche se questo è possibile, come mostrato qui di seguito).

Per semplicità facciamo l'ipotesi che le linee di indirizzo siano 24, in modo che lo spazio complessivo sia composto da 16 blocchi da 1 M; ovvero da 4 blocchi da 4 M.

L'integrato dovrà essere selezionato quando viene presentato un indirizzo nell'intervallo 200000 – 5FFFFFF. Ciò impone che vengano riconosciuti sia gli indirizzi nel campo 200000 – 3FFFFFF che quelli nel campo 4000000 – 5FFFFFF. Agli integrati vanno le linee $A_{21} - A_0$, mentre le due restanti linee di indirizzo servono per la selezione.

Per esprimere il CS del nostro integrato bisogna però immaginare la memoria come suddivisa in blocchi di 2 M, in modo che esso risulti dall'OR della selezione del blocco (di 2M) numero 1 o numero 2. In altre parole:

$$CS = CS_L + CS_H = \overline{A_{23}} \cdot \overline{A_{22}} \cdot A_{21} + \overline{A_{23}} \cdot A_{22} \cdot \overline{A_{21}}$$

dove $CS_L = \overline{A_{23}} \cdot \overline{A_{22}} \cdot A_{21}$ seleziona il campo 200000–3FFFFFF, mentre $CS_H = \overline{A_{23}} \cdot A_{22} \cdot \overline{A_{21}}$ seleziona il campo 4000000–5FFFFFF.

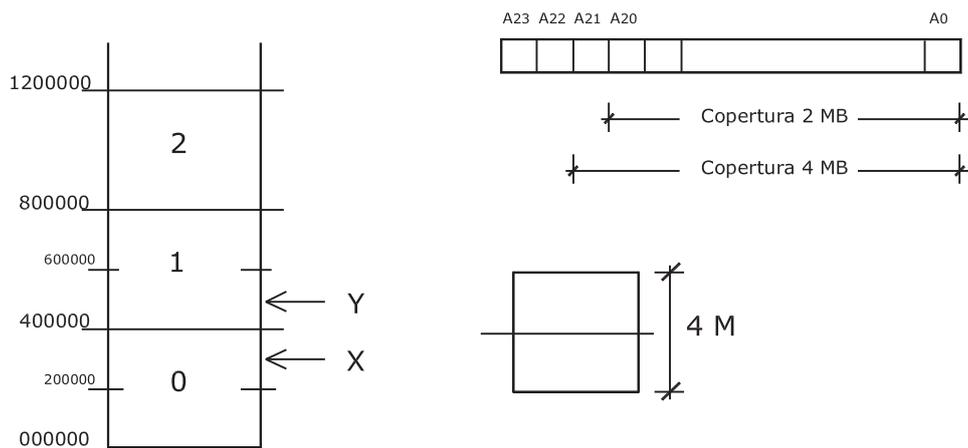


Figura 4.4 Il problema del blocco da allocare ad un indirizzo di partenza non multiplo della propria dimensione

Indichiamo con $I_{21} - I_0$ i piedini dell'integrato. A $I_{20} - I_0$ vengono collegati ordinatamente $A_{20} - A_0$. Mentre per quanto si riferisce a I_{21} valgono queste considerazioni:

- Quando si ha un indirizzo come X, ovvero con $A_{21} = 1$ è asserito CS_L . In tal caso I_{21} deve essere 0, in quanto l'indirizzo in questione è nella parte bassa dei 4 M di interesse;
- Quando si ha un indirizzo come Y (ovvero $A_{21} = 0$) è asserito CS_H . In tal caso I_{21} deve essere 1, in quanto l'indirizzo in questione è nella parte alta dei 4 M di interesse;

In conclusione si ha

$$I_{21} = CS_L \overline{A_{21}} + CS_H \overline{A_{21}} = (CS_L + CS_H) \overline{A_{21}}$$

Si osservi che l'integrato è selezionato solo quando sono asseriti o CS_L o CS_H . Dunque

$$I_{21} = \overline{A_{21}}$$

In altre parole basta collegare direttamente $\overline{A_{21}}$ a I_{21} . Come illustrato in Figura 4.5.

Un'importante osservazione

Con riferimento alla Figura 4.5, vale la pena domandarsi cosa accadrebbe se si portasse direttamente A_{21} al piedino I_{21} dell'integrato, senza cioè interporre il negatore.

Nel caso di memoria di lettura/scrittura ...non accadrebbe niente! Infatti, tutto si ridurrebbe a una rilocazione interna effettuata automaticamente dalla logica propria dell'integrato. Tutto accadrebbe come se, entro l'integrato, i 2 MB inferiori fossero rilocati nella parte superiore e viceversa per i 2 MB superiori, come schematizzato in Figura 4.6.

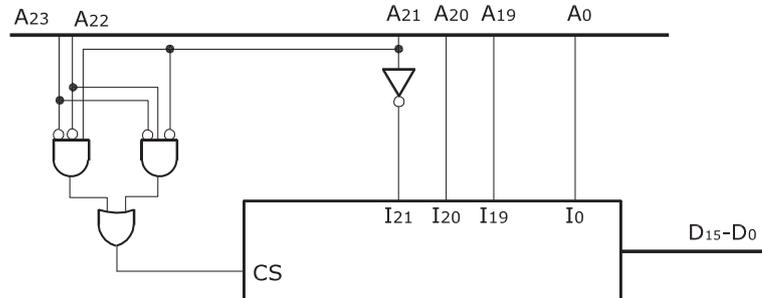


Figura 4.5 Collegamenti dell'integrato (Esercizio 107).

Si faccia caso al fatto che quella di Figura 4.6 è solo una plausibile schematizzazione di ciò che accade internamente all'integrato. Dall'esterno non si percepisce niente: una lettura/scrittura indirizza univocamente una posizione di memoria. Dove essa si trovi non ha alcuna rilevanza nell'impiego della memoria stessa.

Del resto l'integrato viene selezionato quando si ha un campo di indirizzi compresi nel campo 200000–5FFFFFFF, mentre ai piedini di ingresso $I_{21} - I_0$ vengono presentate le linee di $A_{21} - A_0$, il cui contenuto, quando CS è asserito, rappresenta un indirizzo all'interno di quel campo. Dove si trovi la cella indirizzata è inessenziale, l'importante è che ci sia una relazione biunivoca tra indirizzi e celle.

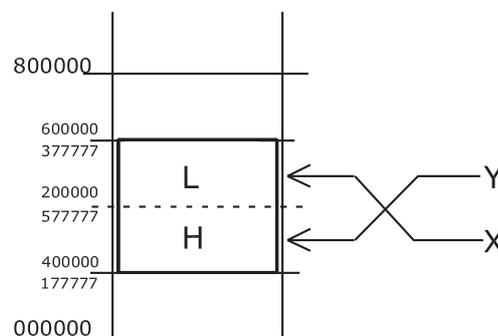


Figura 4.6 Schematizzazione di come sarebbe "suddiviso internamente" l'integrato se si collegasse direttamente A_{21} al piedino I_{21} dell'integrato.

Conclusion La conclusione è che se si ha un integrato con i piedini di indirizzo, corrispondente a uno spazio di $I = 2^i$, esso può essere sistemato a qualunque indirizzo di partenza (ovviamente multiplo di 2), purché:

- si colleghino all'integrato le linee di indirizzo $A_{i-1} - A_0$;
- si costruisca una logica di selezione che abiliti l'integrato solo e soltanto quando l'indirizzo ricade nel campo che si intende ricoprire. La logica di selezione impiega non solo

le linee $A_m - A_i$ ma anche possibili linee meno significative facenti parte dell'indirizzamento interno dell'integrato, come la A_{21} del nostro esempio.

Il fatto che la cella effettivamente indirizzata non si trovi all'interno della struttura dell'integrato in posizione corrispondente a quella che appare nello spazio di memoria non ha alcuna rilevanza.

Si può anche osservare che:

- Se nella costruzione della memoria gli integrati vengono sistemati a “confini naturali” (multipli della dimensione dell'integrato), la selezione degli integrati stessi si effettua decodificando solo i bit che decodificano il numero di blocco a confini naturali. Inoltre l'ordine di indirizzamento interno all'integrato corrisponde all'ordine (funzionale) percepito all'esterno.
- Se nella costruzione della memoria gli integrati vengono sistemati a “confini non naturali”, nella selezione degli integrati stessi intervengono anche linee di indirizzo facenti parte dell'indirizzamento interno dell'integrato. Il numero di queste dipende da dove è posizionato il confine: una linea se il confine è a metà del confine naturale, due linee se è a un quarto, tre se è a un ottavo e così via. L'ordine di indirizzamento interno all'integrato **non** corrisponde all'ordine (funzionale) percepito all'esterno.

Il caso della ROM Diverso è il caso di memoria ROM (o meglio EPROM). Supponiamo che quella in questione sia una memoria EPROM, programmata utilizzando un dispositivo esterno (cioè con l'integrato fuori dalla sede su cui verrà montato durante il normale funzionamento). E' da presumere che il dispositivo di programmazione veda il blocco da 4 MB all'indirizzo convenzionale 0. Dunque, l'integrato verrebbe scritto in modo naturale. Se ora lo si andasse a montare secondo lo schema di Figura 4.5, ma senza il negatore su A_{21} , si troverebbe che ciò che ci si aspetta nella parte bassa viene visto come se si trovasse nella parte alta e viceversa. Per esempio, un'istruzione di salto, che portasse a una posizione che nello spazio di indirizzi è nella metà superiore, avrebbe l'effetto di portare alla posizione dislocata in modo corrispondente nella metà inferiore. Per avere sovrapposizione dei due spazi occorrerebbe che il dispositivo di programmazione “truccasse” gli indirizzi di scrittura in modo da avere la divisione nelle due parti discusse in precedenza.

4.9 Si osservi che si richiede che un blocco di 2 MByte sia dislocato a partire da un confine corrispondente a 1 MByte come illustrato a sinistra in Figura 4.7. La parte destra mostra lo spazio degli indirizzi riferiti alle fasce verticali. Si vede che gli integrati da 512 KB non cadono su un confine naturale.

Poiché la soluzione ottimale è quella che impiega gli integrati di maggior dimensione, usiamo questi anche a costo di una codifica più complicata (si faccia riferimento all'esercizio 107). In sintesi la soluzione è questa:

- Due integrati da 512 KB (0 e 1); essi richiedono 19 linee di indirizzo. Ad essi si portano le linee $A_{20} - A_2$ e vengono selezionati in questo modo (si noti che A_{20} entra nella decodifica):

$$CS_0 = BE_0(\overline{A_{21}}A_{20} + A_{21}\overline{A_{20}}) ;$$

$$CS_1 = BE_1(A_{21}A_{20} + \overline{A_{21}}\overline{A_{20}}).$$
- Due integrati da 256 KB (2 e 5) (256 KB). Ad essi si portano le linee $A_{19} - A_2$ e vengono selezionati in questo modo:

$$CS_2 = BE_2\overline{A_{21}}A_{20} ;$$

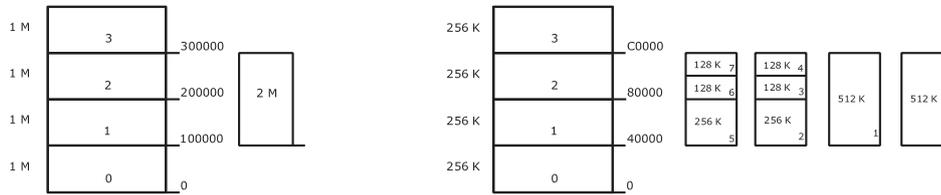


Figura 4.7 Schematizzazione del problema dell'Esercizio 4.9. A sinistra viene la divisione della memoria nello spazio effettivo degli indirizzi, a destra viene schematizzata la copertura con integrati di diverso taglio delle 4 fasce.

$$CS_5 = BE_3 \overline{A_{21}} A_{20};$$

- Quattro integrati da 128 KB (3, 4, 6 e 7). Ad essi si portano le linee $A_{18} - A_2$ e vengono selezionati in questo modo:

$$CS_3 = BE_2 A_{21} \overline{A_{20}} \overline{A_{19}};$$

$$CS_4 = BE_2 A_{21} \overline{A_{20}} A_{19};$$

$$CS_6 = BE_3 A_{21} \overline{A_{20}} \overline{A_{19}};$$

$$CS_7 = BE_3 A_{21} \overline{A_{20}} A_{19}.$$

4.10 La soluzione di minor costo è quella che utilizza l'integrato da 128 kbyte kbyte, i tre da 64 kbyte e tanti integrati da 32 kbyte quanti ne servono per arrivare a coprire lo spazio di memoria richiesto.

Essendo per 8086, la memoria richiesta deve essere divisa in due "colonne" da 256 kbyte, selezionate tramite \overline{BHE} e A_0 . Una, tra le molte possibili soluzioni, è quella di Figura 4.8.

Per quanto si riferisce alla decodifica degli indirizzi, si osserva anzitutto che gli integrati relativi al byte meno significativo devono essere selezionati da A_0 (basso), mentre quello relativo al bit più significativo vengono selezionati da BHE (pure attivo basso). Inoltre, poiché lo spazio degli indirizzi coperto corrisponde al mezzo megabyte inferiore, su tutta la memoria vale $A_{19} = 0$. Indicando con CS_i i *chip select* degli integrati, si ha:

$$CS_5 = BHE \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}}$$

$$CS_6 = BHE \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot A_{17}$$

$$CS_7 = BHE \cdot \overline{A_{19}} \cdot A_{18} \cdot \overline{A_{17}}$$

$$CS_8 = BHE \cdot \overline{A_{19}} \cdot A_{18} \cdot A_{17} \cdot \overline{A_{16}}$$

$$CS_9 = BHE \cdot \overline{A_{19}} \cdot A_{18} \cdot A_{17} \cdot A_{16}$$

$$CS_0 = \overline{A_0} \cdot \overline{A_{19}} \cdot \overline{A_{18}}$$

$$CS_1 = \overline{A_0} \cdot \overline{A_{19}} \cdot A_{18} \cdot \overline{A_{17}} \cdot \overline{A_{16}}$$

$$CS_2 = \overline{A_0} \cdot \overline{A_{19}} \cdot A_{18} \cdot A_{17} \cdot \overline{A_{16}}$$

$$CS_3 = \overline{A_0} \cdot \overline{A_{19}} \cdot A_{18} \cdot A_{17} \cdot A_{16}$$

$$CS_4 = \overline{A_0} \cdot A_{19} \cdot A_{18} \cdot A_{17} \cdot A_{16}$$

Dalle precedenti è facile ricavare le forme corrispondenti allo schema di Figura 4.9. Ad esempio:

$$\begin{aligned} \overline{CS_5} &= \overline{BHE \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}}} &= \overline{BHE} + \overline{\overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}}} \\ \overline{CS_3} &= \overline{\overline{A_0} \cdot \overline{A_{19}} \cdot A_{18} \cdot A_{17} \cdot \overline{A_{16}}} &= \overline{A_0} + \overline{\overline{A_{19}} \cdot A_{18} \cdot A_{17} \cdot \overline{A_{16}}} \end{aligned}$$

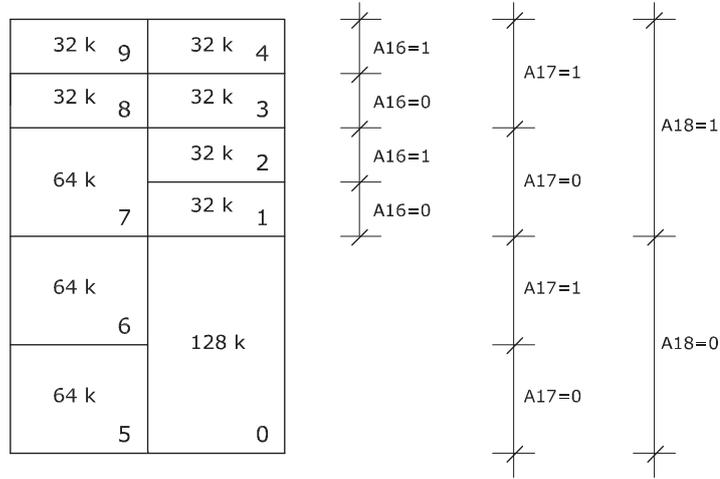


Figura 4.8 Una possibile disposizione degli integrati nello spazio di memoria dell'Esercizio 4.10. Vengono indicati i valori dei bit di indirizzo corrispondenti alle differenti posizioni. Poiché lo spazio degli indirizzi coperto corrisponde al mezzo megabyte inferiore, su tutta la memoria vale $A_{19} = 0$.

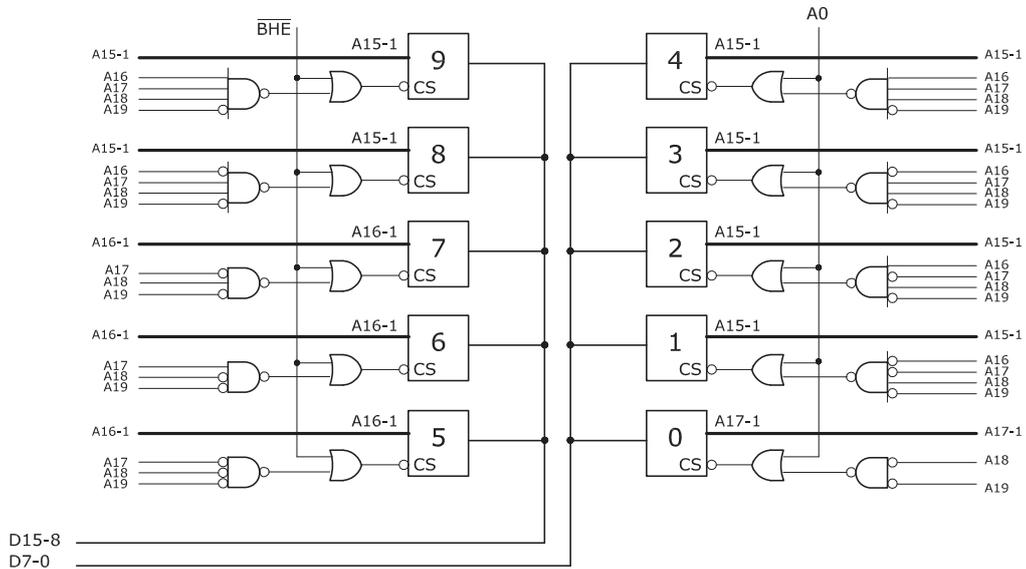


Figura 4.9 Collegamento dei singoli integrati e rete di decodifica degli indirizzi (Esercizio 108). Gli integrati sono numerati come in Figura 4.8.

4.11 In questo caso si tratta di disporre di 8 colonne di 64Kb. Su tutti gli integrati deve essere $A_{19} = A_{18} = A_{17} = 1$ e $A_{16} = 0$. Ai due integrati da 64 Kb si portano A_{15-0} ; agli integrati di 32 Kb si portano A_{14-0} , mentre A_{15} entra nel chip select; agli integrati di 16 Kb si portano A_{13-0} , mentre A_{15} e A_{14} entrano pure nel chip select; agli integrati di 8 Kb si portano A_{12-0} , mentre A_{15} , A_{14} e A_{13} entrano nel chip select. In conclusione, con i numeri assegnati alle posizioni di Figura 4.10 si hanno i chip select seguenti:

$$\begin{aligned}
 CS_0 &= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \\
 CS_1 &= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \cdot \overline{A_{15}} \\
 CS_3 &= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \\
 CS_5 &= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \\
 CS_7 &= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \\
 CS_9 &= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \\
 CS_{11} &= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \\
 CS_{13} &= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \\
 CS_2 &= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \cdot A_{15} \\
 CS_4 &= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \cdot \overline{A_{15}} \cdot A_{14} \\
 CS_6 &= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \\
 CS_8 &= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot A_{13} \\
 CS_{10} &= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot A_{13} \\
 CS_{12} &= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot A_{13} \\
 CS_{14} &= \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot A_{13}
 \end{aligned}$$

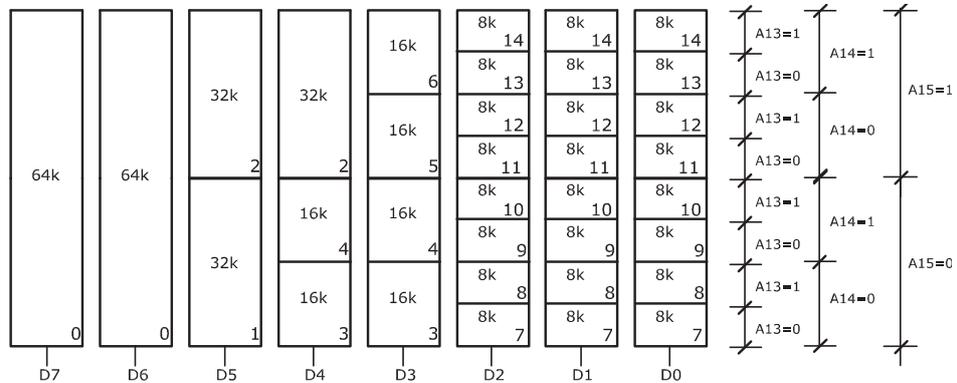


Figura 4.10 Disposizione degli integrati per il caso dell'Esercizio 4.11.

4.12 Una soluzione ottima è quella di Figura 4.11. Ai piedini di indirizzo degli integrati da 64 KB si portano le linee $A_{16} - A_1$, a quelli da 32 KB le linee $A_{15} - A_1$.

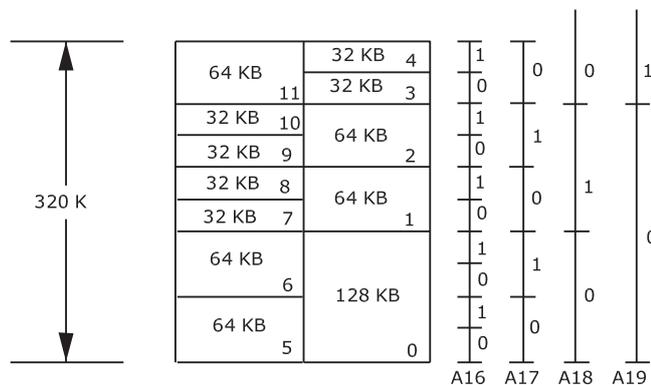


Figura 4.11 Rete dell'Esercizio 4.12.

Ne derivano le seguenti espressioni per i chip select:

$$\begin{array}{ll}
CS_5 = BHE \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} & CS_0 = \overline{A_0} \cdot \overline{A_{19}} \cdot \overline{A_{18}} \\
CS_6 = BHE \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot A_{17} & CS_1 = \overline{A_0} \cdot \overline{A_{19}} \cdot A_{18} \cdot \overline{A_{17}} \\
CS_7 = BHE \cdot \overline{A_{19}} \cdot A_{18} \cdot \overline{A_{17}} \cdot \overline{A_{16}} & CS_2 = \overline{A_0} \cdot \overline{A_{19}} \cdot A_{18} \cdot A_{17} \\
\vdots & \\
CS_{11} = BHE \cdot A_{19} \cdot \overline{A_{18}} \cdot \overline{A_{17}} & CS_3 = \overline{A_0} \cdot A_{19} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot \overline{A_{16}} \\
& CS_4 = \overline{A_0} \cdot A_{19} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16}
\end{array}$$

Per quanto riguarda la complessità che si avrebbe facendo partire la memoria dall'indirizzo 1000h, si osservi che tale indirizzo è quello del byte successivo ai primi 4 KB. Pertanto, con il parallelismo di 2 byte, le due fasce corrispondenti partirebbero dall'indirizzo 800h (metà di 1000h) corrispondente a 2 KB. In altre parole resterebbe scoperto il primo blocco di 2 KB dello spazio degli indirizzi di ciascuna fascia. Con gli integrati a disposizione la selezione risulterebbe assolutamente complicata.

Per rendersene conto supponiamo di disporre gli integrati come in Figura 4.11, ma, ovviamente, a partire dall'indirizzo 800h e consideriamo l'integrato da 64 KB corrispondente all'integrato 5 di Figura 4.11. Tale integrato andrebbe a coprire i 31 blocchi da 2 KB, aventi numero d'ordine da 1 a 31 che compongono il primi 64 K dello spazio degli indirizzi (della fascia di sinistra), lasciando scoperto il blocco di 2 KB di ordine 0; inoltre andrebbe a coprire il blocco di 2KB avente numero d'ordine assoluto 32, ovvero numero d'ordine 0 nei secondi 64 KB dello spazio degli indirizzi di fascia. Ai piedini di indirizzo dell'integrato verrebbero portate ancora le linee $A_{16} - A_1$, mentre la sua selezione sarebbe ottenuta come OR della selezione di 32 blocchi da 2 KB

$$CS_5 = CS_{5,1} + CS_{5,2} + \dots + CS_{5,31} + CS_{5,32}$$

rappresentando $CS_{5,i}$ la selezione del generico blocco di 2KB entro i 64KB ricoperti dall'integrato. Ovvero

$$CS_{5,1} = BHE \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot \overline{A_{16}} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot A_{12}$$

$$CS_{5,2} = BHE \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot \overline{A_{16}} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot A_{13} \cdot \overline{A_{12}}$$

$$CS_{5,2} = BHE \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot \overline{A_{16}} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot A_{13} \cdot A_{12}$$

$$CS_{5,31} = BHE \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot \overline{A_{17}} \cdot A_{16} \cdot A_{15} \cdot A_{14} \cdot A_{13} \cdot A_{12}$$

$$CS_{5,32} = BHE \cdot \overline{A_{19}} \cdot \overline{A_{18}} \cdot A_{17} \cdot \overline{A_{16}} \cdot \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}}$$

Per l'integrato da 128 KB (l'integrato 0 in Figura 4.11) l'OR sarebbe di 64 termini. È del tutto evidente che si tratta di problema malposto almeno dal punto di vista pratico.

4.13 Occorre anzitutto osservare che il blocco di memoria di 512 KB che inizia dalla posizione A00000 determina la struttura degli indirizzi di Tabella 4.2. Infatti: A00000 è l'indirizzo di partenza e poiché il blocco deve essere di 512 KB (cioè 80000h B), gli indirizzi da coprire vanno da A00000 a $A00000+80000-1=A00000+7FFFF=A7FFFF$.

	A_{31}	...	A_{24}	A_{23}	A_{22}	A_{21}	A_{20}	A_{19}	A_{18}	...	A_2	A_1	A_0
Ind partenza = A00000	0	...	0	1	0	1	0	0	0	...	0	0	0
Ind finale = A7FFFF	0	...	0	1	0	1	0	1	1	...	1	1	1

Tabella 4.2 Struttura degli indirizzi per l'Esercizio 4.13.

A_0 e A_1 non sono disponibili in quanto essi sono sostituiti dai 4 segnali $BE_0 \dots BE_3$. Ciascuna fascia è, ovviamente di 128 KB, dunque l'indirizzamento entro la fascia richiede 17 bit, ovvero i bit $A_{18}-A_2$. I bit $A_{31} \dots A_{19}$ vengono usati per la selezione del blocco di 512 KB, attraverso la linea SEL:

$$SEL = \overline{A_{31}} \cdot \overline{A_{30}} \cdot \dots \cdot \overline{A_{24}} \cdot A_{23} \cdot \overline{A_{22}} \cdot A_{21} \cdot \overline{A_{20}} \cdot \overline{A_{19}}$$

Con gli integrati a nostra disposizione la soluzione ottima deve utilizzare quello da 128 KB e i tre da 64 KB. In Figura 4.12 viene presentata una possibile allocazione nello spazio degli indirizzi.

Posto: $SEL_0 = SEL \cdot BE_0$, $SEL_1 = SEL \cdot BE_1$, $SEL_2 = SEL \cdot BE_2$, $SEL_3 = SEL \cdot BE_3$, ne derivano le seguenti espressioni per i chip select:

$$\begin{aligned} CS_0 &= SEL_0 \\ CS_{10} &= SEL_1 \cdot \overline{A_{18}} & CS_{11} &= SEL_1 \cdot A_{18} \\ CS_{20} &= SEL_2 \cdot \overline{A_{18}} & CS_{21} &= SEL_2 \cdot A_{18} \cdot \overline{A_{17}} & CS_{22} &= SEL_2 \cdot A_{18} \cdot A_{17} \\ CS_{30} &= SEL_3 \cdot \overline{A_{18}} \cdot \overline{A_{17}} & CS_{31} &= SEL_3 \cdot A_{18} \cdot \overline{A_{17}} & CS_{32} &= SEL_3 \cdot A_{18} \cdot A_{17} \\ CS_{33} &= SEL_3 \cdot A_{18} \cdot A_{17} \end{aligned}$$

Si consideri ora la sistemazione di Figura 4.13. Apparentemente non ci sono differenze, anzi dallo schema sembra che la soluzione di Figura 4.13 sia più complessa perché si deve portare A_{17} al decodificatore della colonna 1, dove prima non serviva. Se però si scrivono le espressioni per i CS si ottiene:

$$\begin{aligned} CS_0 &= SEL_0 \\ CS_{10} &= SEL_1 \cdot \overline{A_{18}} & CS_{11} &= SEL_1 \cdot A_{18} \cdot \overline{A_{17}} & CS_{12} &= SEL_1 \cdot A_{18} \cdot A_{17} \\ CS_{20} &= SEL_2 \cdot \overline{A_{18}} & CS_{21} &= SEL_2 \cdot A_{18} \cdot \overline{A_{17}} & CS_{22} &= SEL_2 \cdot A_{18} \cdot A_{17} \\ CS_{30} &= SEL_3 \cdot \overline{A_{18}} & CS_{31} &= SEL_3 \cdot A_{18} \cdot \overline{A_{17}} & CS_{32} &= SEL_3 \cdot A_{18} \cdot A_{17} \end{aligned}$$

In sostanza si è ottenuta una decodifica degli indirizzi molto più regolare; inoltre diversamente dal caso precedente non si deve mai decodificare le due combinazioni $\overline{A_{18}} \cdot \overline{A_{17}}$ e $\overline{A_{18}} \cdot A_{17}$; in sostanza la rete di decodifica diventa più leggera.

Ovviamente quanto sopra si applica anche agli esercizi precedentemente svolti. Il lettore è invitato a riorganizzare la disposizione degli integrati delle figure 4.10 e 4.11.

4.14 La soluzione è immediata in quanto la capacità totale è data dal prodotto dei termini riportati nel testo:

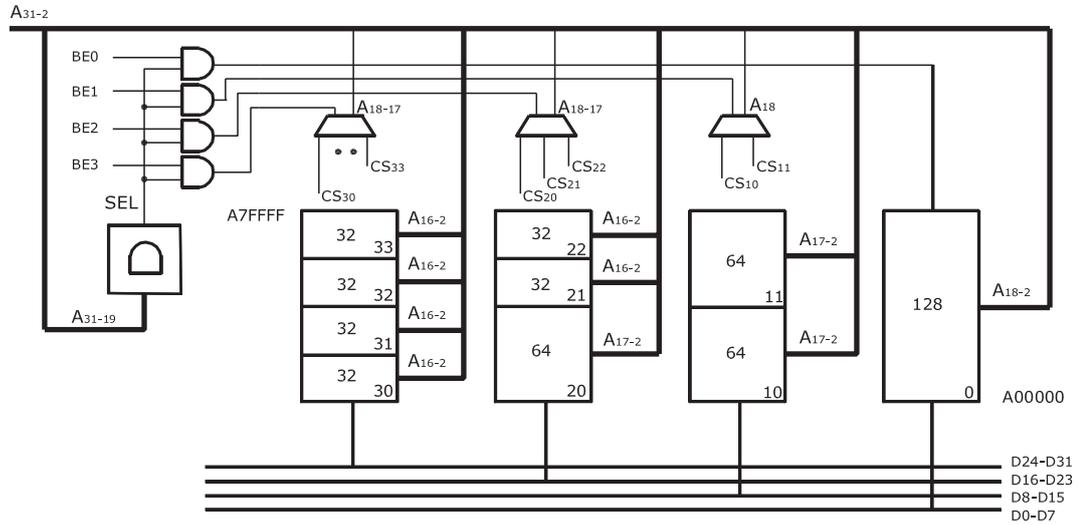


Figura 4.12 Schema di memoria dell'Esercizio 4.13.

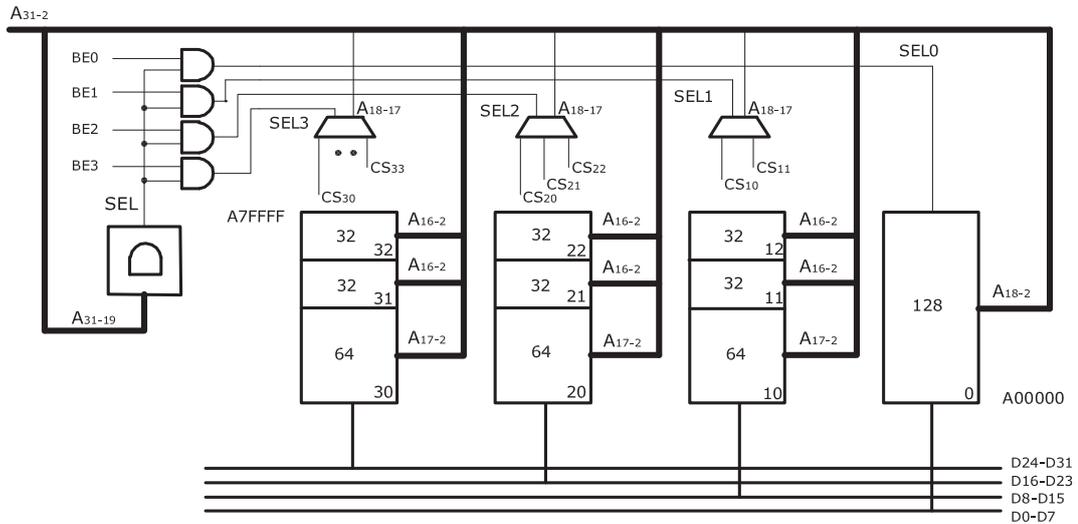


Figura 4.13 Modifica della dislocazione degli integrati (rispetto alla figura 4.12).

$$\text{Capacità Totale} = 4096 \times 8 \times 1024 \times 512 = 16384 \text{ MB.}$$

4.15 Il disco nell'esempio del Paragrafo 4.7.1 del testo ha una velocità di 5400 rpm, ovvero 90 giri al secondo. La velocità di trasferimento è data dal rapporto tra il numero di byte che si trovano su una traccia e il tempo richiesto a effettuare un giro (si trascurano gli effetti legati al passaggio da settore a settore sulla stessa traccia). Indicando con v la velocità di trasferimento in byte al secondo e con g il numero di giri al secondo, si ha:

$$v = n^\circ \text{ byte per traccia} \times n^\circ \text{ giri al secondo}$$

ovvero si ha che ogni traccia deve contenere:

$$\text{byte per traccia} = \frac{v}{g} = \frac{4 \text{ Mbyte/sec}}{90 \text{ giri/sec}} = 46603$$

Ogni settore ha una capacità di 512 byte, quindi, il numero di settori per traccia è di:

$$\frac{46603}{512} = 91 \text{ settori per traccia.}$$