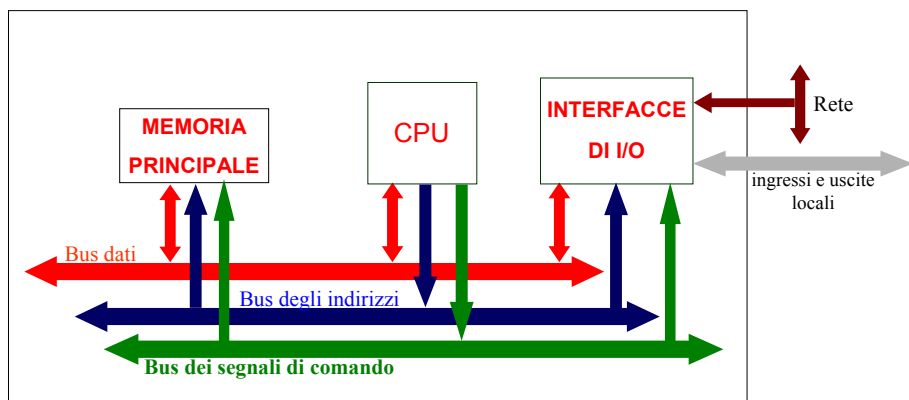


Calcolatori Elettronici L-A

Mapping di **chip di memoria e interfacce** negli spazi di indirizzamento di sistemi con **bus dati da 8 bit**

- Decodifica degli indirizzi
 - Decodifica completa
 - Decodifica semplificata
- Decoder e PAL

Richiamo: Sistema ad un Bus-master System bus



In questo schema a blocchi la CPU genera i segnali di indirizzo e di comando per la memoria e le interfacce

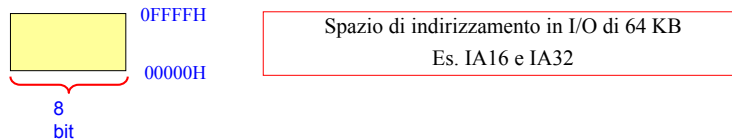
Richiamo: Spazio di indirizzamento in memoria



- Gli indirizzi si indicano solitamente in codice esadecimale
- la dimensione di uno spazio di indirizzamento si può esprimere in:
 - Kilobyte ($1\text{KB} = 2^{10}\text{ Byte} = 1024\text{ B}$)
 - Megabyte ($1\text{MB} = 1\text{K KB} = 2^{20}\text{ Byte} = 1.048.576\text{ B}$)
 - Gigabyte ($1\text{GB} = 1\text{K MB} = 2^{30}\text{ Byte} = 1.0\text{e}+9$)
- Esempi
 - il Pentium e il DLX hanno uno spazio di indirizzamento di 4 GB (indirizzi di 32 bit rappresentabili con 8 cifre esadecimali)
 - l'8086 ha uno spazio di indirizzamento di 1 MB (indirizzi di 20 bit rappresentabili con 5 cifre esadecimali)

Richiamo: - Interfacce di ingresso/uscita - Spazio di indirizzamento in I/O

- Le interfacce di I/O contengono registri su cui transitano i dati scambiati con il mondo esterno
- Così come i dispositivi di memoria, anche le interfacce di ingresso/uscita (**e precisamente i registri di transito dei dati e ogni altro registro indirizzabile dal software**) vanno mappate in uno spazio di indirizzamento
- Le interfacce di I/O possono essere mappate in uno spazio distinto da quello della memoria oppure nello stesso; in quest'ultimo caso si dice che l'I/O è mappato in memoria (**memory mapped I/O**)
- Lo spazio di indirizzamento in I/O è solitamente più piccolo dello spazio di indirizzamento in memoria; es: nelle architetture Intel IA16 e IA32 lo spazio di indirizzamento in I/O è di 64 KB



Richiamo: i cicli di bus

- Il trasferimento di un'informazione tra agenti del bus avviene con una sequenza di eventi detti nel loro insieme **ciclo di bus**
- in ogni ciclo di bus un agente master indirizza un agente slave
- Ad esempio, per leggere in memoria **all'indirizzo i** la CPU (master del bus) genera un ciclo di bus costituito dalla seguente sequenza di eventi:
 - la CPU mette sul **bus degli indirizzi** l'indirizzo **i** nonché l'indicazione del numero di byte che vuole trasferire a partire dall'indirizzo **i**
 - la CPU attiva **un segnale di comando** che identifica l'operazione desiderata (MRDC# per la lettura in memoria)
 - la CPU attende che la memoria indirizzata (slave) metta l'informazione desiderata sul **bus dati**
 - la CPU legge (cioè campiona) sul **bus dati** i segnali generati dalla memoria
- Questa sequenza di eventi si chiama **ciclo di lettura in memoria (memory read cycle)**

Richiamo: esempi di cicli di bus

- Altri esempi di cicli di bus sono i seguenti:
 - **ciclo di scrittura in memoria** (memory write cycle) caratterizzato dal segnale di comando **MWRC#**
 - **ciclo di lettura in I/O** (I/O read cycle) caratterizzato dal segnale di comando **IORDC#**
 - **ciclo di scrittura in I/O** (I/O write cycle) caratterizzato dal segnale di comando **IOWRC#**
 - Durante il corso studieremo in dettaglio questi e altri cicli di bus
- Ora ci addentreremo nei dettagli delle tecniche per allocare una risorsa esterna alla CPU nel Sistema a Microprocessore assegnandole un indirizzo (mapping di una periferica)

Condizione di visibilità di un dispositivo da parte del software

- Condizione necessaria affinché un dispositivo fisico (memoria, interfaccia, o altra entità) sia accessibile al software è:
 - il dispositivo deve essere mappato in uno spazio di indirizzamento
- Mappare in uno spazio di indirizzamento significa:
 - associare al dispositivo una finestra di indirizzi di quello spazio di indirizzamento
- Consideriamo il caso di un sistema con due spazi di indirizzamento (memoria ed input/output)
- La macchina di Von Neumann ha un solo bus degli indirizzi, quindi accede ad entrambi gli spazi di indirizzamento **con lo stesso bus** (un indirizzo alla volta)
- Si accede ai dispositivi mappati in uno spazio di indirizzamento con cicli di bus
- Lo spazio di indirizzamento indirizzato in un ciclo di bus viene discriminato in base a:
 - il segnale di comando (appartenente al bus dei comandi)
 - oppure
 - **un segnale di qualificazione che si affianca al bus degli indirizzi (p. es. M/IO*)**

Dimensione della finestra occupata da un dispositivo - esempi

- Un **dispositivo** accessibile attraverso il bus, sia esso un **dispositivo di memoria** (p. es. RAM, EPROM, FLASH), **una interfaccia** (p. es. un registro di uscita, un'interfaccia per leggere lo stato di interruttori, una porta parallela, una porta seriale, o un **agente master indirizzabile** (es. DMA controller, controllore di hard disk, etc.)), occupa in generale $n = 2^{*k}$ posizioni nello spazio di indirizzamento
- n rappresenta il numero di oggetti di 8 bit indirizzabili all'interno del **dispositivo** (es. numero di celle di memoria nelle RAM e nelle EPROM, il numero di registri in un register file, il numero di registri di transito, di stato o di comando nelle interfacce di I/O)
- k è fortemente variabile al variare del dispositivo:
 - In generale nei dispositivi di input/output (i. e. le interfacce) k è piccolo; in questo corso considereremo dispositivi di i/o in cui k è compreso tra 0 e 4
 - in generale nei dispositivi di memoria k è grande (p. es. per una RAM da 128 KB si ha $k = 17$)

Caratteristiche ai morsetti di un dispositivo indirizzabile su una finestra di $n = 2^k$ byte

- Qualunque dispositivo da 8 bit con all'interno $n = 2^k$ elementi indirizzabili separatamente ha al suo interno un **decoder di k variabili con ingresso di enable** che seleziona i singoli oggetti indirizzabili; di solito il dispositivo **si interfaccia al bus** attraverso i seguenti segnali:
 - indirizzi ($A[k-1..0]$): sono le k variabili decodificate dal decoder interno del disp. e individuano il byte a cui si intende accedere all'interno del chip (offset nel chip)
 - dati ($D[7..0]$): definiscono la porta bidirezionale che consente di scambiare dati con l'esterno
 - chip select (CS^*) detto anche "chip enable" (CE^*): è il segnale di enable del decoder; se CS^* non è attivo il chip non è indirizzato; a CS^* va portato un segnale che si attiva se l'indirizzo a cui si accede appartiene alla finestra dello spazio di indirizzamento in cui il chip è mappato
 - comandi:
 - Read Command (RD^*), detto anche Output Enable (OE^*): è il comando di lettura; quando RD^* e CS^* sono attivi, il dispositivo espone il contenuto della cella indirizzata
 - Write Command (WR^*): è il comando di scrittura; il nor di WR^* e CS^* è il "latch enable" ("clock") della cella indirizzata



Mapping di dispositivi da 8 bit in sistemi con bus dati da 8 bit - ipotesi di lavoro

- In questo corso consideriamo unicamente sistemi in cui la quantità di informazione associata a ogni indirizzo è un byte
- Supponiamo ora di considerare solo dispositivi da 8 bit, cioè dispositivi con porta dati da 8 bit; questo è in effetti il parallelismo di tutti i dispositivi visti fino ad ora (memorie, latch, registri edge-triggered, driver, transceiver..)
- Imponiamo infine l'ulteriore condizione che il parallelismo del bus dati sia di 8 bit
- In queste ipotesi l'assegnamento a un dispositivo di una finestra di indirizzi in uno spazio di indirizzamento avverrà in generale nel rispetto delle tre seguenti ulteriori condizioni restrittive:
 - la dimensione della finestra di indirizzi associata a un dispositivo è **una potenza di due**
 - la finestra è composta da **indirizzi contigui**
 - L'indirizzo iniziale della finestra è un **multiplo** della dimensione della finestra (**finestra allineata**)

Mapping allineato di dispositivi da 8 bit in sistemi con bus dati da 8 bit

Si consideri un dispositivo D di $n=2^k$ byte indirizzabili

- Si dice che D è mappato all'indirizzo A se gli indirizzi dei byte di D sono compresi tra A e A+n-1, cioè se A è l'indirizzo più basso tra tutti gli indirizzi associati a D
- Si dice che D è allineato se A è un multiplo di n, cioè se:
(indirizzo più basso di D) $\text{mod } n = 0$ (condizione di allineamento)
- Se D è allineato allora i k bit meno significativi di A sono uguali a zero

Esempi:

- Un dispositivo da due byte è allineato se è mappato a un indirizzo pari
- Un dispositivo da 8 byte è allineato se è mappato a un indirizzo il cui valore codificato in binario termina con 3 zeri
- Un dispositivo da 16 byte è allineato se il suo indirizzo iniziale in codice esadecimale ha la cifra meno significativa uguale a zero
- Un dispositivo da 64 KB è allineato se il suo indirizzo in codice esadecimale ha le quattro cifre meno significative uguali a zero

Come individuare univocamente una finestra allineata di 2^k byte in uno spazio di indirizzamento

- Supponiamo di mappare un dispositivo D di 2^k byte a un indirizzo A allineato di uno spazio di indirizzamento di 1 MB (bus di indirizzi di 20 bit):
- allora possiamo porre $A = \alpha \ll 0$ ove α è una configurazione binaria di $20 - k$ bit e gli indirizzi associati a D saranno compresi tra:

$$A_{\min} = A = \alpha \ll 0 \text{ e } A_{\max} = A + (2^k - 1) = \alpha \ll k$$

- dunque possiamo indicare l'indirizzo A_i dell' i -esimo byte di D come l'insieme di due campi concatenati:

$$A_i = \alpha \ll i \quad \text{dove } \alpha \in [0: (2^{20-k} - 1)] \quad \text{e} \quad i \in [0: (2^k - 1)]$$

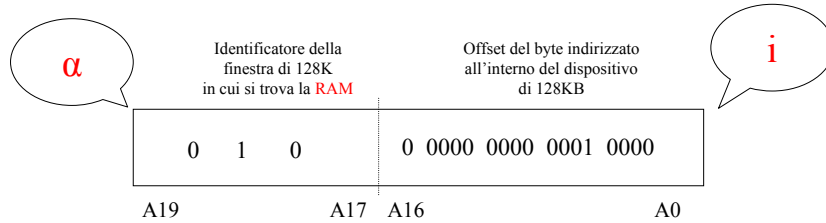
α individua tra le 2^{20-k} finestre allineate di 2^k byte presenti nello spazio di indirizzamento, quella su cui è mappato D

i individua l'offset nel chip del byte indirizzato

Campi in cui si suddivide l'indirizzo di dispositivi mappati in uno spazio di indirizzamento (esempio n. 1)

Indirizzamento di un byte di una RAM all'indirizzo 40010H in uno spazio di indirizzamento di un MB nell'ipotesi di disporre di un chip da 128 KB mappato all'indirizzo 40000H:

L'indirizzo viene suddiviso in due campi: il primo identifica la finestra di 128 KB in cui è mappata la RAM, il secondo identifica l'offset nella RAM

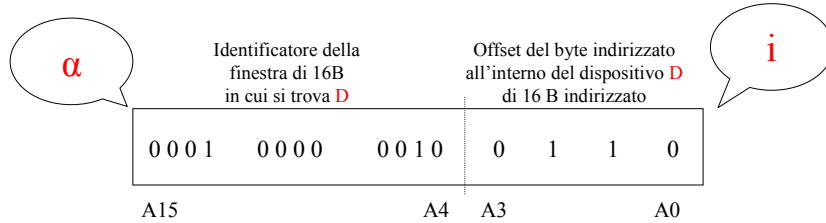


Si rappresenti graficamente lo spazio di indirizzamento e si indichi la finestra in cui è mappato il chip nonché la posizione del byte di indirizzo 40010H

Campi in cui si suddivide l'indirizzo di dispositivi mappati in uno spazio di indirizzamento (esempio n. 2)

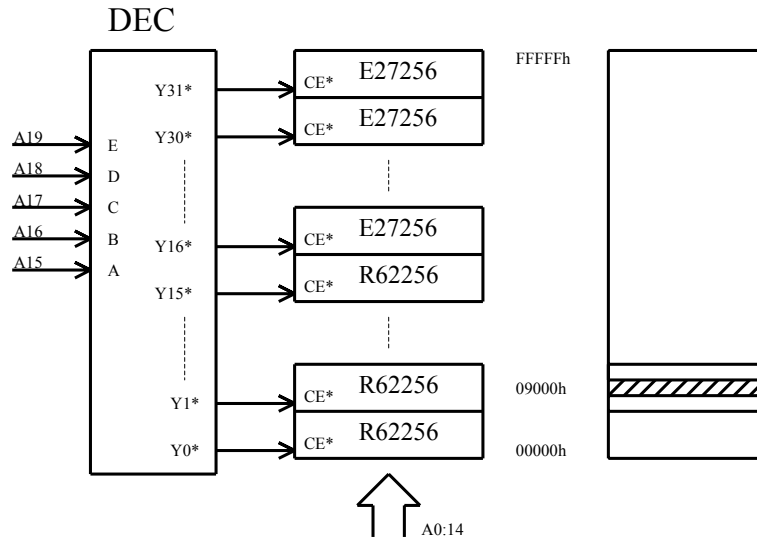
Indirizzamento di un byte all'indirizzo 1026H in un dispositivo di I/O **D** di 16 byte mappato all'indirizzo 1020H di uno spazio di indirizzamento di 64 KB

L'indirizzo viene suddiviso in due campi: il **primo** identifica la finestra di 16 B in cui è mappato il dispositivo, il **secondo** identifica l'offset nel dispositivo



Si rappresenti graficamente lo spazio di indirizzamento e si indichi la finestra in cui è mappato il chip nonché la posizione del registro interno di indirizzo 1026H

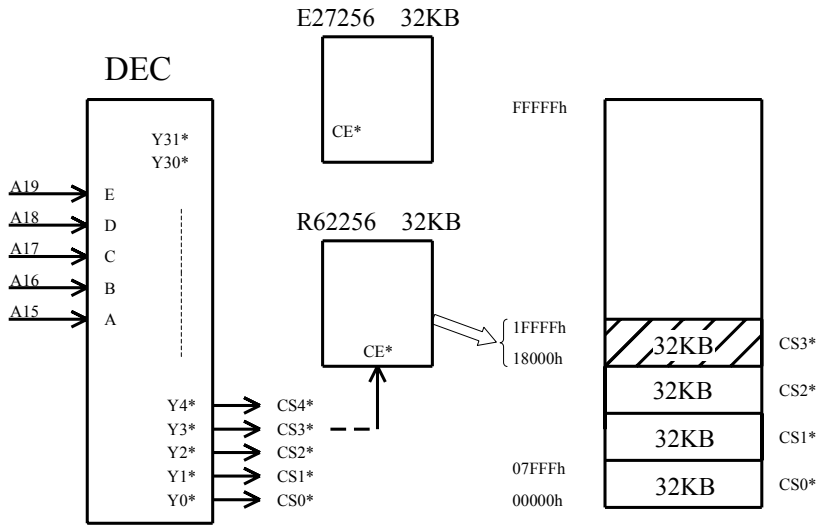
Mapping allineato: decodifica completa con decoder di tutto lo spazio fisico disponibile



Decodifica degli indirizzi in caso di mapping allineato primo caso: decodifica completa

- Consideriamo uno spazio di indirizzamento di 1 MB in cui sia mappato un dispositivo di 2^k byte
- Per individuare una cella di indirizzo $A_i = \alpha \#i$ possiamo decodificare tutti i 20 bit che compongono A_i
- Questa decodifica viene effettuata ricorrendo alla struttura dei decoder ad albero, con albero di due livelli
 - Il livello radice viene usato per decodificare α (che identifica la posizione in cui il chip è mappato); per decodificare α dobbiamo decodificare 20-k variabili
 - il secondo livello viene utilizzato per decodificare i (che identifica il byte all'interno del chip, serve un decoder di k variabili)
- Il **decoder di secondo livello si trova all'interno del chip**
- La decodifica di α è **a carico del progettista del sistema** che può utilizzare un decoder di 20-k variabili, oppure può limitarsi a realizzare il mintermine di 20-k variabili con cui si decodifica α
- Questo tipo di decodifica in cui vengono decodificati tutti i bit dell'indirizzo si chiama decodifica completa

Mapping allineato: decodifica completa con decoder di una parte dello spazio fisico disponibile



Mapping e Decodifiche 17

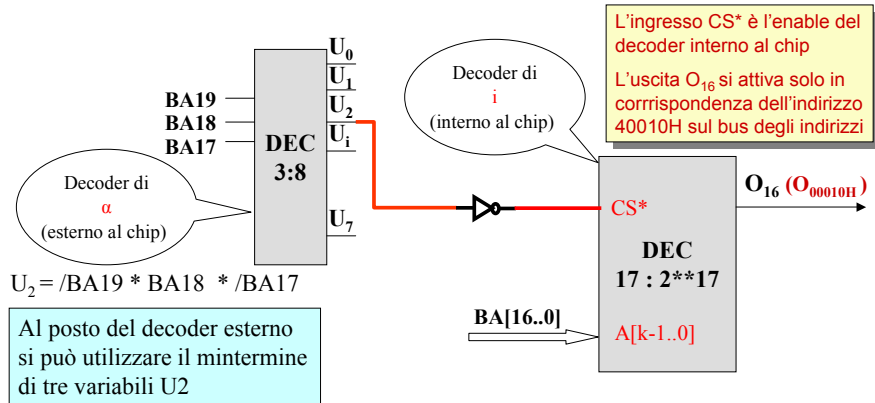
Esercizi

- Si disegni la cascata dei due decoder nell'ipotesi di esprimere l'indirizzo di una cella A_i con l'espressione $A_i = a_{##} i$, e di voler decodificare i seguenti indirizzi:
 - l'indirizzo 40010H all'interno di un chip di memoria da 128 KB in uno spazio di 1MB
 - l'indirizzo 0F005H all'interno di un dispositivo da 64 KB in uno spazio di 1MB
 - l'indirizzo 01026H all'interno di un dispositivo di I/O da 16 B in uno spazio di 64KB
 - l'indirizzo 02026H all'interno di un dispositivo di output da 1 B in uno spazio di 64KB
 - l'indirizzo 40010H all'interno di un chip di memoria da 1 MB in uno spazio di 1MB

Si noti che i decoder radice disegnati sono tutti diversi
Si noti inoltre che quasi tutte le uscite di ogni decoder non sono utilizzate

Soluzione del primo esercizio del lucido precedente

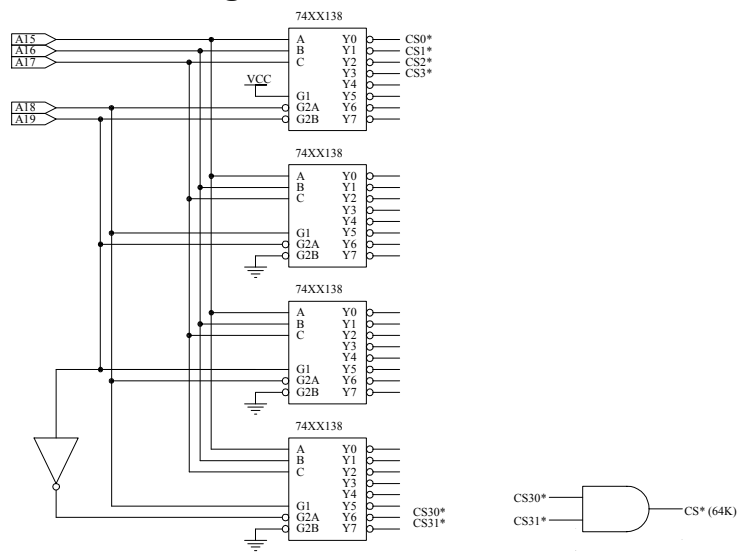
- Si disegni la cascata dei due decoder nell'ipotesi di esprimere l'indirizzo di una cella A_i con l'espressione $A_i = \alpha ## i$, e di voler decodificare il seguente indirizzo:
 - l'indirizzo 40010H all'interno di un chip di memoria da 128 KB in uno spazio di 1MB



Regola generale per interfacciare al bus degli indirizzi un dispositivo mappato in una determinata finestra α e determinarne l'espressione del CS* nel caso di decodifica completa

- In generale, se sono verificate le condizioni indicate nel lucido *ipotesi di lavoro a 8 bit*, l'interfacciamento al bus degli indirizzi di un dispositivo indirizzabile di 2^{*k} byte mappato in uno spazio di indirizzamento di 2^{*m} byte si ottiene come segue:
- I k bit meno significativi del bus degli indirizzi $BA[(k-1)..0]$ vengono ordinatamente interconnessi agli ingressi $A[(k-1)..0]$ del dispositivo, in modo da consentire al decoder interno del dispositivo di decodificare il campo "i" dell'indirizzo (si ricordi che ogni indirizzo può essere suddiviso nei due campi α e i , ove i è l'offset nel chip).
- I bit $BA[(m-1)..k]$ vengono utilizzati per formare il mintermine di $(m-k)$ variabili la cui configurazione individua la finestra " α " in cui il dispositivo indirizzato è mappato.
- Il mintermine viene negato e inviato all'ingresso CS* del dispositivo considerato.
- Possiamo realizzare con una PAL il mintermine (negato) associato ad " α "

Uso dei DECODER per la decodifica degli indirizzi e la generazione dei CS*



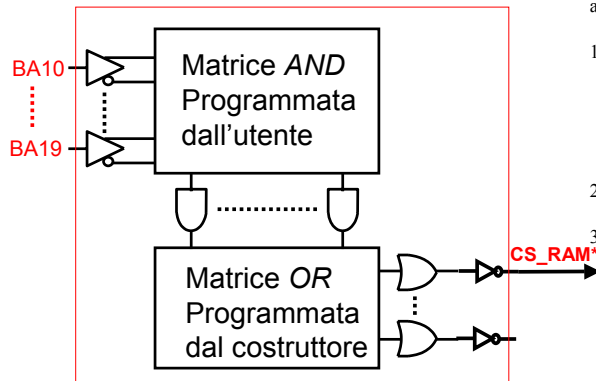
Mapping e Decodifiche 21

Uso delle PAL nella generazione dei CS*

- Le espressioni dei CS* dei dispositivi slave inseriti in un sistema a microprocessore possono essere realizzati con PAL combinatorie con uscita in logica negativa.
- Queste PAL sono identificate da un "part number" del tipo: **iLo**, ove **i** indica il numero degli ingressi, **o** indica il numero delle uscite e **L** indica che la PAL è combinatoria con uscita in logica negativa (p. es. La PAL 12L10 ha 12 ingressi e 10 uscite; ogni uscita è il negato della somma logica di due termini prodotto programmabili)
- Una PAL 10L8 può essere utilizzata per generare da 1 a 8 CS*, purchè per il CS* di ogni dispositivo sia sufficiente decodificare un numero massimo di 10 indirizzi.
- Quando si vuole programmare una PAL di tipo L, le espressioni delle uscite vanno scritte in logica positiva, dopodichè sui piedini di uscita ci troveremo il negato dell'espressione programmata.

Esempio di CS* realizzato con una PAL

Questa è la struttura di una PAL di tipo L



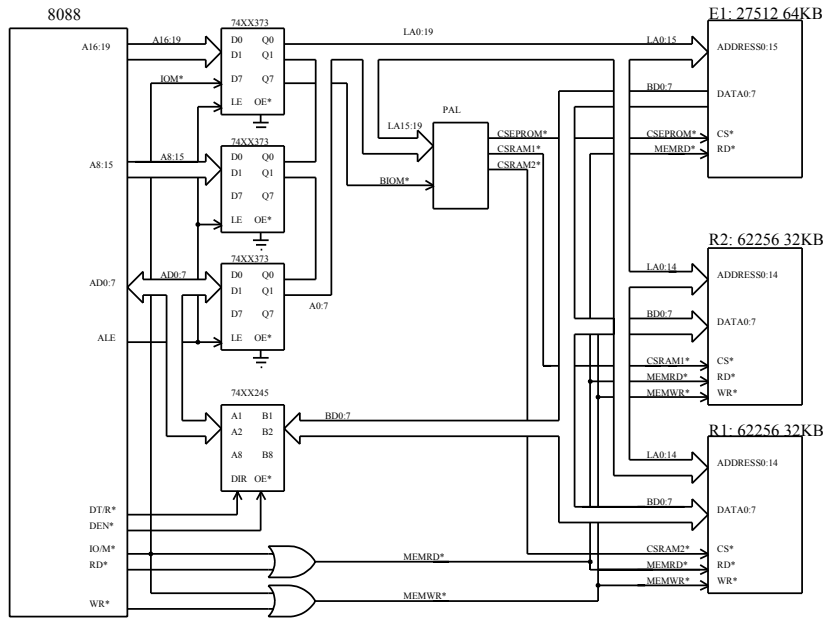
Supponiamo di voler generare il CS*
Di una ram da 128 KB mappata
all'indirizzo 00000H. Allora
procediamo come segue:

1. Inseriamo nel progetto una Pal (nell'esempio di fianco una ipotetica PAL 10L2) e colleghiamo agli ingressi i 10 bit più significativi del bus degli indirizzi
2. Assegnamo a una uscita della PAL il nome CS_RAM*
3. Scriviamo l'espressione di CS_RAM in logica positiva:
 $CS_RAM = \neg BA19 * \neg BA18 * \neg BA17$
(abbiamo scritto l'espressione della variabile a monte del negatore di uscita.....)

PAL:
Programmable
Array Logic

- Con le altre uscite della PAL si possono generare i CS* di altri dispositivi
- Cosa succede se a uno degli ingressi mandiamo il segnale M/IO*?

Schema completo µP, PAL e memorie

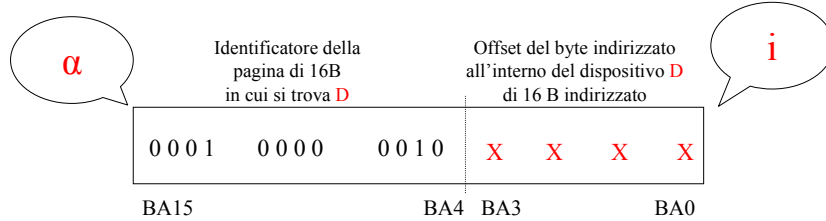


Mapping e Decodifiche 25

Introduzione alla decodifica parziale (o semplificata)

Riconsideriamo il dispositivo D di I/O di 16 byte mappato all'indirizzo 1020H di uno spazio di indirizzamento di 64 KB

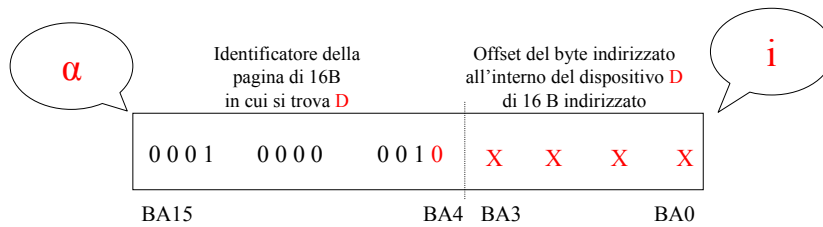
- I 16 indirizzi associati a questo dispositivo si ottengono facendo assumere ai 4 bit meno significativi dell'indirizzo tutte le 16 combinazioni di 0 e 1
- Se esprimiamo CS_D come mintermine (poi negato) di BA[15..4], allora D resta mappato esclusivamente nella finestra tra 1020H e 102FH
- Per qualunque configurazione di A[15..4] diversa da 0102H, CS_D non potrà attivarsi



Cosa succede se togliamo qualche bit di indirizzo nell'espressione di CS_D?
La decodifica degli indirizzi generati dal bus master non è più completa, ma diventa una decodifica "parziale". Qual è la conseguenza?

Decodifica parziale: un esempio

Supponiamo di togliere dall'espressione di CS_D il termine /BA4. Allora CS_D diventa funzione unicamente di BA[15..5] e non dipende più da BA4.
In queste condizioni, se una configurazione di BA[15..0] individua un byte di D, allora anche l'indirizzo ottenuto invertendo BA4 attiva CS_D. Dunque possiamo dire che la finestra in cui è mappato D viene duplicata nello spazio di indirizzamento: D ora è mappato non solo nella finestra tra 1020H e 102FH ma anche nella finestra tra 1030H e 103FH



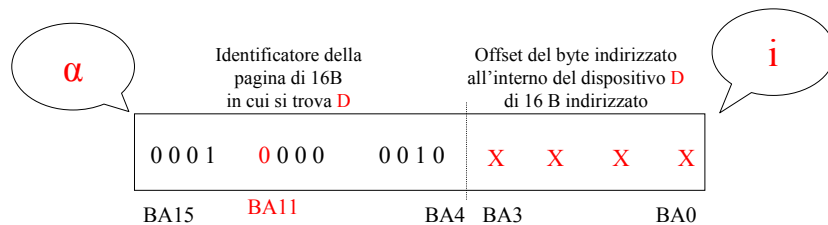
Decodifica parziale: generalizzazione

- Consideriamo un dispositivo di 2^k byte mappato in uno spazio di indirizzamento di 2^m byte. Con la decodifica completa il CS è un prodotto di $m-k$ termini.
- Se rinunciamo a decodificare un bit, il dispositivo risulta mappato in due finestre dello spazio di indirizzamento.
- Se rinunciamo a decodificare 2 bit, il dispositivo risulta mappato in 4 finestre dello spazio di indirizzamento. La porzione dello spazio di indirizzamento in cui è mappato il dispositivo viene quadruplicata rispetto al caso di decodifica completa.
- Se rinunciamo a decodificare j bit, il dispositivo risulta mappato in 2^j finestre dello spazio di indirizzamento. La porzione dello spazio di indirizzamento in cui è mappato il dispositivo viene moltiplicata per 2^j rispetto al caso di decodifica completa; diremo che l'immagine di D nello spazio di indirizzamento è replicata 2^j volte.

- L'effetto della decodifica parziale (detta anche semplificata o incompleta) è duplice:
 1. Da un lato si ottengono espressioni più semplici dei CS in quanto si riduce il numero di ingressi della rete che lo realizza
 2. Dall'altro il dispositivo viene ad occupare una porzione dello spazio di indirizzamento di dimensione multipla rispetto allo stretto necessario, per cui si ottiene un uso inefficiente della risorsa "spazio di indirizzamento"

Contiguità delle finestre replicate nello spazio di indirizzamento in caso di decodifica parziale

- In caso di decodifica semplificata la porzione dello spazio di indirizzamento in cui risulta mappato il dispositivo (di dimensione pari a $(2^{**j}) \cdot (2^{**k})$) è costituita da indirizzi adiacenti se i j bit non decodificati sono quelli meno significativi di α
- Se i bit non decodificati non sono i meno significativi di α , allora il dispositivo viene mappato in finestre non contigue
- Si verifichi l'affermazione precedente con degli esempi e si cerchi di individuare la distanza tra le finestre replicate in funzione della significatività dei bit di indirizzo non decodificati



Se, per esempio, il bit non decodificato è BA11, allora D risulta mappato nella finestra tra 1020H e 102FH e nella finestra tra 1820H e 182FH (finestre non contigue)