



ESERCITAZIONE 4

Livello d'architettura dell'insieme di istruzioni

Decodifica dell'indirizzo



2

Livello d'architettura dell'insieme di istruzioni

Livello d'architettura dell'insieme di istruzioni (1)

1) Si scrivano in notazione polacca inversa le seguenti espressioni

a) $6*(4-3)$

b) $(7/3)/((1-4)*2)+1$

c) $(5*2+7)-4/2+1$

Livello d'architettura dell'insieme di istruzioni (1)

Soluzioni (1)

Notazione Tradizionale

2 + 3

Operando1 Operatore Operando2

Notazione Polacca Inversa

2 3 +

Operando1 Operando2 Operatore

Operazioni scritte da sinistra verso destro, considerando le priorità.

Riferimento su Tanenbaum: pag 355 e seguenti.

Livello d'architettura dell'insieme di istruzioni (1)

Soluzioni (2)

a) $6*(4-3)$

▶ Operando1= 6, Operatore = *, Operando2=(4-3)

▶ Op1 = 4, Op2 = 3, Operatore=-;

▶ Soluzione: 6 4 3 - *

Livello d'architettura dell'insieme di istruzioni (1)

Soluzioni (3)

b) $(7/3)/((1-4)*2)+1$

- ▶ Operando1 = $(7/3)/((1-4)*2)$, Operatore = +, Operando2 = 1
- ▶ Op11 = $(7/3)$, Op12 = $((1-4)*2)$, Operatore = /
- ▶ Op111 = 7, Op112 = 3, Operatore = /
- ▶ Op121 = $(1-4)$, Op122 = 2, Operatore = *
- ▶ Op1211 = 1, Op1212 = 4, Operatore = -
- ▶ Soluzione: $7\ 3\ /\ 1\ 4\ -\ 2\ *\ /\ 1\ +$

Livello d'architettura dell'insieme di istruzioni (1)

Soluzioni (4)

c) $(5*2+7)-4/2+1$

► Operando1 = $(5*2+7)-4/2$, Operatore = +, Operando2 = 1

► Op11 = $(5*2+7)$ Op12 = $4/2$, Operatore = -

► Op111 = $5*2$, Op112 = 7, Operatore = +

► Op121 = 4, Op122 = 2, Operatore = /

► Op1111 = 5, Op1112 = 2, Operatore = *

► Soluzione: $5 * 2 + 7 + 4 / 2 - 1 +$

Livello d'architettura dell'insieme di istruzioni (2)

2) Indicare il tipo di indirizzamento delle seguenti istruzioni

- ▶ **CMP R2,R4**
- ▶ **MOV R2,#2**

Livello d'architettura dell'insieme di istruzioni (2)

Soluzione (1)

Riferimento sul testo: pag 351 e seguenti

Modalità di indirizzamento

- ▶ **Indirizzamento immediato**
- ▶ **Indirizzamento diretto**
- ▶ **Indirizzamento a registro**
- ▶ **Indirizzamento a registro indiretto**
- ▶ **Indirizzamento indicizzato**
- ▶ **Indirizzamento indicizzato esteso**

operando stesso come campo
indirizzo completo come campo
registro come campo
registro con indirizzo come campo
registro più spiazzamento costante
indirizzo come somma valori registri

Livello d'architettura dell'insieme di istruzioni (2)

Soluzione (2)

- ▶ **CMP R2,R4**
 - ▶ Operando 1:
 - ▶ Indirizzamento a registro: il valore è indicato con il registro che lo contiene
 - ▶ Operando 2:
 - ▶ Indirizzamento a registro: il valore è indicato con il registro che lo contiene
- ▶ **MOV R2,#2**
 - ▶ Operando 1:
 - ▶ Indirizzamento a registro: il valore è indicato con il registro che lo contiene
 - ▶ Operando 2:
 - ▶ Indirizzamento immediato: il valore è indicato con il valore stesso

Livello d'architettura dell'insieme di istruzioni (3)

3) Scrivere le istruzioni Assembly utili per ottimizzare l'utilizzo della CPU per il calcolo della moltiplicazione tra il numero 19 e un valore n intero e non negativo. Verificare, ponendo $n=3$, che l'operazione proposta sia corretta.

Livello d'architettura dell'insieme di istruzioni (3)

Soluzione (1)

Istruzioni Principali

Trasferimenti	
MOV DST, SRC	Sposta SRC in DST
Aritmetica	
ADD DST, SRC	Somma SRC a DST
SUB DST, SRC	Sottrae SRC da DST
MUL SRC	Moltiplica EAX con SRC (no segno)
DIV SRC	Divide EDX:EAX per SRC (no segno)

Livello d'architettura dell'insieme di istruzioni (3)

Soluzione (2)

Istruzioni Principali

Scorrimento	
SHL / SHR DST, #	Shift logico verso s/d di SRC di # bit
Booleane	
AND DST, SRC	AND di SRC e DST, res in DST
OR DST, SRC	OR di SRC e DST, res in DST
XOR DST, SRC	OR esclusivo di SRC e DST, res in DST
NOT DST	Rimpiazza DST con complemento a 1.

Livello d'architettura dell'insieme di istruzioni (3)

Soluzione (3)

Riferimento sul libro: paragrafo 5.5.3, pag 367 e seguenti

La moltiplicazione tra un qualsiasi numero n e un numero come 2^k , può essere ottimizzato come uno shift del numero verso sinistra di k cifre.

Esempio:

- $3 \times 2 = ?$
- $3 = (00000011)_2$ (rappresentazione a 8 bit)
- $2 = 2^1$
- Shift 1 cifra a sx
- $3 \times 2 = (00000110)_2 = 6$

Livello d'architettura dell'insieme di istruzioni (3)

Soluzione (4)

- ▶ $19 = 16 + 2 + 1$
- ▶ $19 = 2^4 + 2^1 + 2^0$
- ▶ Applicare 3 shift e successivamente sommare

- ▶ $3 = (00000011)_2$
- ▶ Shift 1: 00110000
- ▶ Shift 2: 00000110
- ▶ Shift 3: 00000011

▶ Somma:
00110000 +
00000110 +
00000011 =
00111001

Livello d'architettura dell'insieme di istruzioni (3)

Soluzione (5)

- ▶ $(00111001)_2 = (57)_{10}$
- ▶ $19 \times 3 = 57$
- ▶ VERIFICATO

Livello d'architettura dell'insieme di istruzioni (3)

Soluzione (6)

Istruzioni pseudo Assembly: caricamento in memoria di n

- ▶ MOV R1 \$n
- ▶ MOV R2 \$n
- ▶ MOV R3 \$n

Livello d'architettura dell'insieme di istruzioni (3)

Soluzione (7)

Istruzioni pseudo Assembly: shifting a sinistra dei valori nei registri

- ▶ SHL R1 4
- ▶ SHL R2 1

Istruzioni pseudo Assembly: Somma dei valori

- ▶ ADD R1 R2
- ▶ ADD R1 R3

Livello d'architettura dell'insieme di istruzioni (3)

Soluzione (8)

Verifica

1) Caricamento

Registro	Valore
R1	00000011
R2	00000011
R3	00000011

2) Left Shift

Registro	Valore
R1	00110000
R2	00000110
R3	00000011

3) Somma I

Registro	Valore
R1	00110110
R2	00000110
R3	00000011

3) Somma II

Registro	Valore
R1	00111001
R2	00000110
R3	00000011

Decodifica dell'indirizzo

Decodifica dell'indirizzo(1)

1) Si supponga di avere un calcolatore monoprocesso con 16-bit di indirizzamento (A0÷A15), una EPROM di 2 KB × 8 byte per il programma, una RAM di 2 KB × 8 byte per i dati, una PIO tipo Intel 8255A con 24 porte e un registro di controllo.

Descrivere il circuito che abilita il chip di I/O in modalità Memory-Mapped I/O, se la PIO è posizionata a partire dall'indirizzo FFFCH della memoria.

Suggerimento: porre la EPROM all'indirizzo 0 dello spazio di indirizzamento e la RAM all'indirizzo 8000H.

Decodifica dell'indirizzo(1)

Soluzione (1)

Riferimento sul libro pag 215 e seguenti

- ▶ 16 bit di indirizzamento A0÷A15
- ▶ EPROM e RAM richiedono uno spazio degli indirizzi di 2KB
- ▶ PIO richiede 4 byte
- indirizzi 00000xxxxxxxxxxx sono destinati alla EPROM
- indirizzi 10000xxxxxxxxxxx sono destinati alla RAM
- indirizzi 11111111111111xx alla PIO
- ▶ Per ogni componente va attivata (con valore 0) la porta di controllo \overline{CS}

Decodifica dell'indirizzo(1)

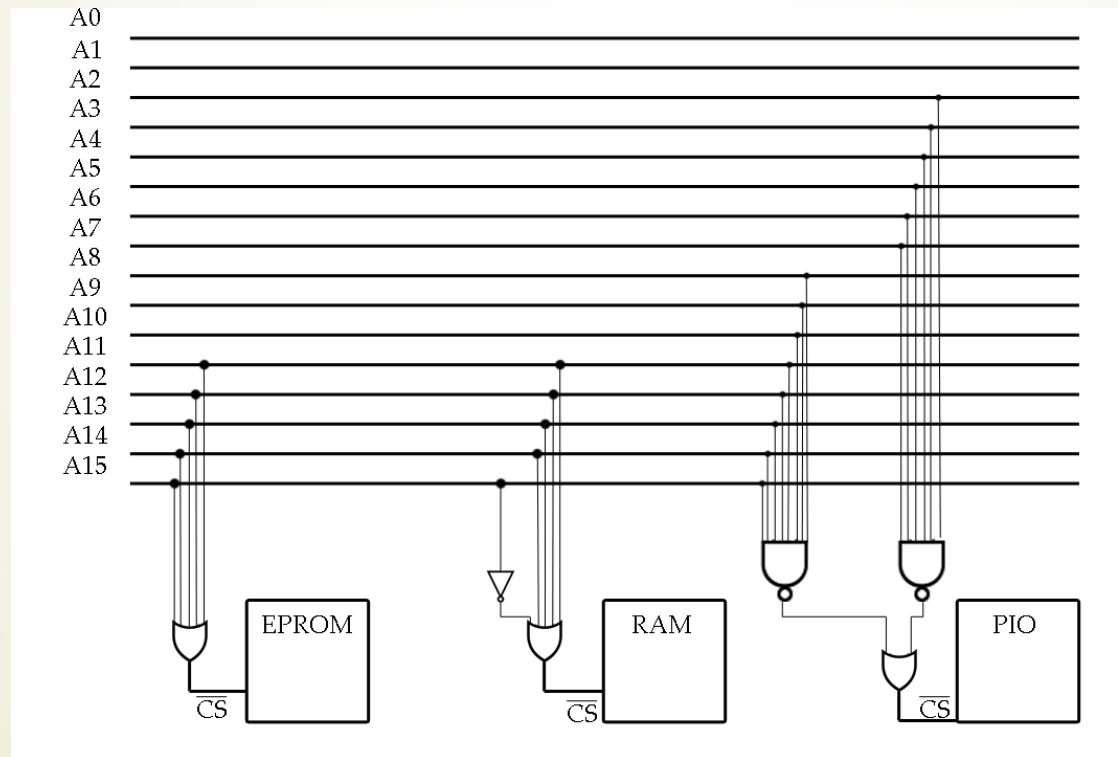
Soluzione (2)

- ▶ Attivare \overline{CS} della EPROM quando le linee A_{11} , A_{12} , A_{13} , A_{14} e A_{15} hanno segnale basso: porta OR.
- ▶ Attivare \overline{CS} della RAM quando le linee A_{11} , A_{12} , A_{13} , A_{14} hanno segnale basso, ma A_{15} lo ha alto: porta OR preceduta da un not per la sola A_{15} .
- ▶ Attivare \overline{CS} della PIO quando le linee A_2 , A_3 , A_4 , A_5 , A_6 , A_7 , A_8 , A_9 , A_{10} , A_{11} , A_{12} , A_{13} , A_{14} e A_{15} hanno segnale alto: due NAND () e () unite ad una porta OR.

Decodifica dell'indirizzo(1)

Soluzione (3)

Circuito



Decodifica dell'indirizzo(1)

Soluzione (4)

Soluzione alternativa

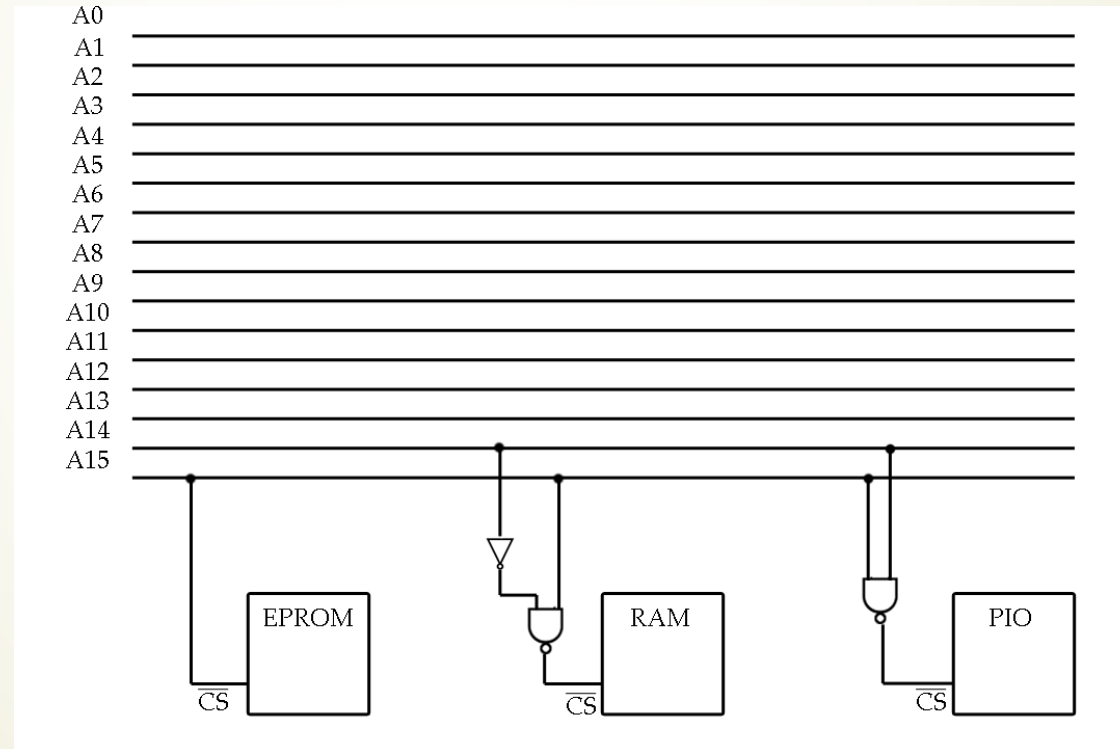
Notare:

- ▶ EPROM ha indirizzi che iniziano per 0 ($A_{15} = 0$)
- ▶ RAM ha indirizzi che iniziano per 10 ($A_{15} = 1, A_{14} = 0$)
- ▶ PIO ha indirizzi che iniziano per 11 ($A_{15} = 1, A_{14} = 1$)
- ▶ Collegare con lo stesso criterio visto in precedenza
- ▶ Attenzione: questa configurazione non permette di aggiungere ulteriori componenti al calcolatore.

Decodifica dell'indirizzo(1)

Soluzione (5)

Circuito



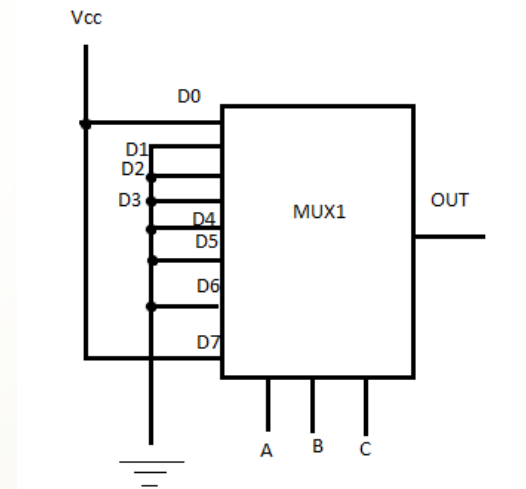
Approfondimento: multiplexer

Multiplexer: soluzione alternativa

Soluzione alternativa al problema già analizzato dell'uso dei multiplexer a 3 input di controllo per espressioni logiche a 4 variabili.

ESEMPIO: $ABC + \bar{A}\bar{B}\bar{C} + E$

CIRCUITO PER $ABC + \bar{A}\bar{B}\bar{C}$



Multiplexer: soluzione alternativa

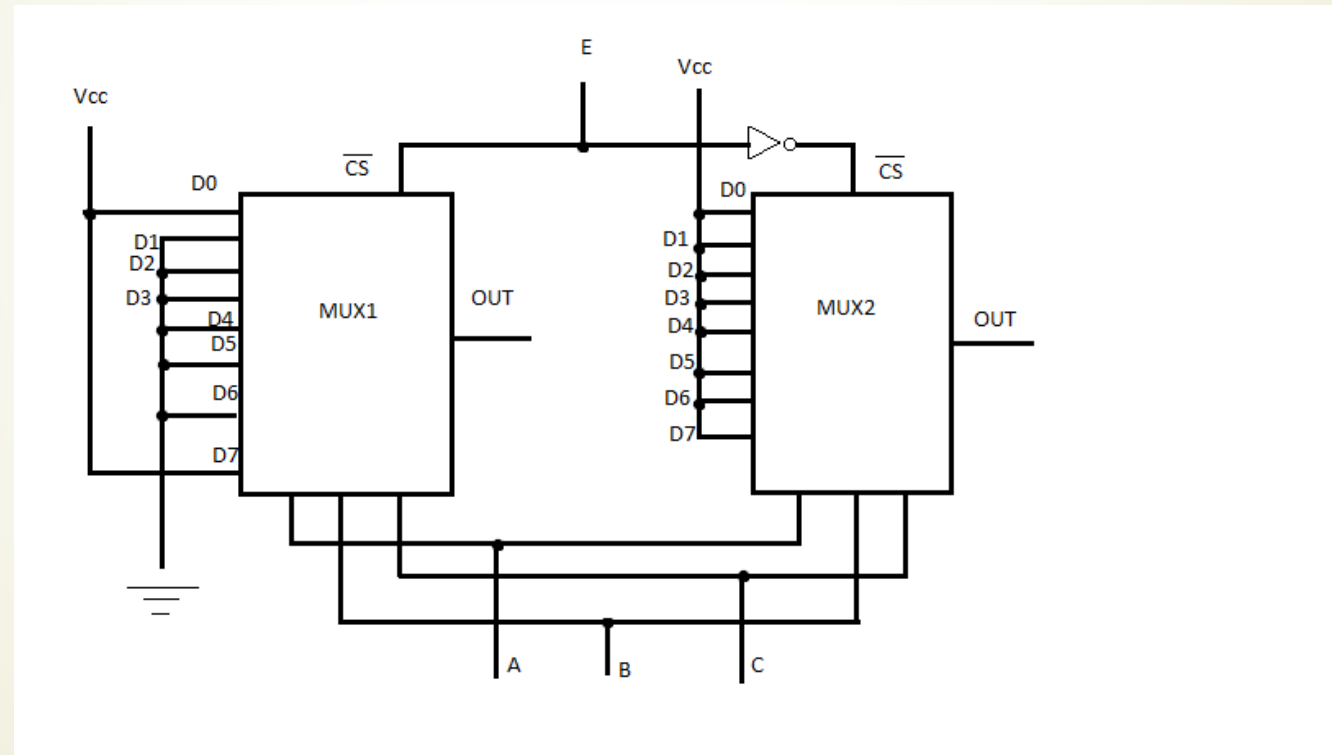
Idea: usare il bit di chip selection (che chiameremo \overline{CS}) che ci permette di selezionare il mux o meno.

Utilizzare più mux collegati alla variabile E (o, in alternativa, in forma negata) in modo tale che il risultato dipenda da esso.

Nel caso in analisi: il mux illustrato alla slide precedente dovrà attivarsi quando E è 0, mentre creeremo un secondo mux che restituirà 1 a vuoto che, invece, sarà selezionato nel caso in cui E sia 1.

Multiplexer: soluzione alternativa

Immagine dei due mux e del loro collegamento a E.



Multiplexer: soluzione alternativa

Il circuito mostrato in questa seconda figura implementa esattamente quanto detto finora:

- ▶ Se E ha valore 0, la porta CS del mux_1 riceve un segnale basso (0), attivando proprio il mux_1 . Da esso viene calcolato il valore della funzione logica, escludendo la variabile E . Questo valore viene mandato in output. Allo stesso tempo, il mux_2 non viene selezionato, (la sua porta logica CS riceve un segnale alto) e quindi non fa nulla.
- ▶ Se E ha valore 1, la porta CS del mux_2 riceve un segnale basso (0), che dunque viene attivato. Visto che è un circuito che restituisce sempre 1, farà uscire dal proprio output il valore 1, a prescindere dai valori presenti sui bit di controllo. Contemporaneamente mux_1 non viene selezionato, e quindi non farà nulla.

Multiplexer: soluzione alternativa

Applichiamo ora quanto descritto a un esempio più complesso, come l'esercizio 4 dell'esercitazione 3 (disponibile sul sito del corso).

$$Y = \bar{C}D + AB$$

Il mux selezionato da $D = 0$ dovrà realizzare solo la funzione logica AB , mentre quello selezionato da $D = 1$ dovrà realizzare la funzione logica $AB+C$.

Multiplexer: soluzione alternativa

- ▶ Nel mux₁ :
 - ▶ le porte D₆ e D₇ sono collegate alla massa (attivano gli AND del tipo 11x)
 - ▶ tutte le altre porte sono collegate a terra
- ▶ Nel mux₂ :
 - ▶ le porte D₀, D₂, D₄, D₆ sono collegate alla massa (implementano C, attivando tutti gli AND del tipo xx0)
 - ▶ la porta D₇ è collegata alla massa, poiché implementa AB (attiva l'unico ingresso non ancora attivato del tipo 110, cioè 111)
 - ▶ le altre porte sono collegate a terra.

