

**Architettura degli Elaboratori**  
**10 Dicembre 2001**

**Esercizio 1. (INF e TWM)**

a) Si converta in base 10 il seguente numero binario:

$$10010101 = 149_{(10)}$$

b) Si converta in base 2, base 8 e base 16 il seguente numero intero (decimale):

$$112 = 1110000_{(2)} \quad 160_{(8)} \quad 70_{(16)}$$

c) Si converta in base 10 il seguente numero in complemento a 2:

$$111111111001100 = -52_{(10)}$$

d) Si determinino le rappresentazioni con **16 bit** in complemento a 2 dei seguenti numeri:

$$215 = 0000000011010111_{(\text{compl. a } 2)}$$

$$-215 = 1111111100101001_{(\text{compl. a } 2)}$$

e) Si determinino le rappresentazioni con **il minor numero di bit** di:

$$10001110 + 01111011 = 100001001$$

$$101 * 101 = 11001$$

**Esercizio 2. (INF)**

Si progetti un dispositivo sequenziale sincrono che riceve in ingresso una linea seriale  $x$  sulla quale vengono trasmesse sequenze di bit a lunghezza variabile. Il dispositivo genera tre uscite  $r$ ,  $b_1$  e  $b_0$ :  $r$  vale normalmente 0 e  $b_1$  e  $b_0$  sono normalmente non specificate mentre, in corrispondenza del bit finale delle sequenze in ingresso,  $r$  vale 1 e  $b_1$  e  $b_0$  assumono valori secondo la tabella di conversione

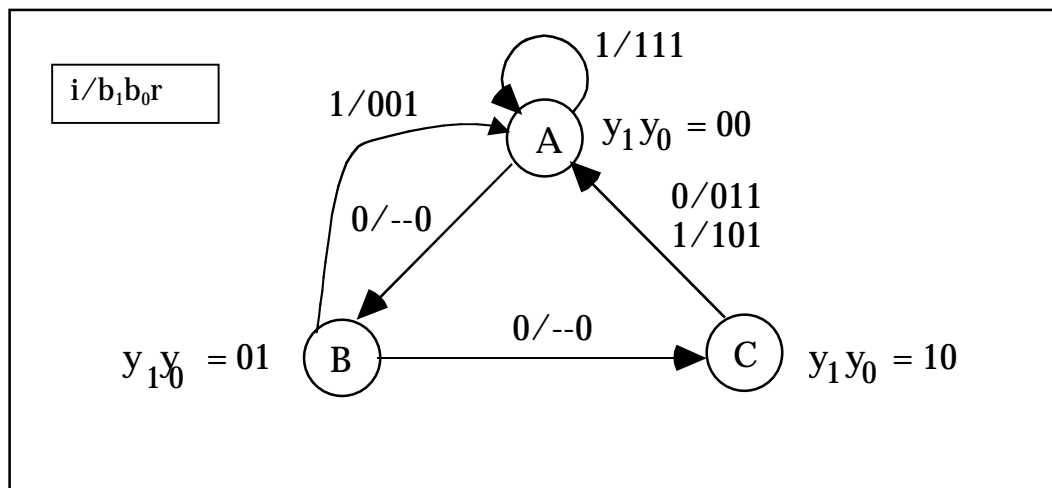
input		$b_1$	$b_0$
1	→	1	1
01	→	0	0
001	→	1	0
000	→	0	1

E. g.

$x =$	01	1	000	...
$b_1 =$	-0	1	--0	...
$b_0 =$	-0	1	--1	...
$r =$	01	1	001	...

a) Si disegni il diagramma degli stati del dispositivo sul retro di questo foglio.

b) Si sintetizzino le espressioni booleane di stato futuro e di uscita utilizzando le seguenti mappe di Karnaugh supponendo che i 4 valori delle variabili  $y_1 y_0$ : 00, 01, 10, 11 codifichino i 4 (possibili) stati A, B, C, D del dispositivo.



$y_1y_0 \backslash x$	0	1
00		1
01		1
11	--	--
10	1	1

$$r = x + y_1$$

$y_1y_0 \backslash x$	0	1
00	--	1
01	--	
11	--	--
10	1	

$$b_0 = \bar{x} + \bar{y}_1\bar{y}_0$$

$y_1y_0 \backslash x$	0	1
00	--	1
01	--	
11	--	--
10		1

$$b_1 = x\bar{y}_0$$

$y_1y_0 \backslash x$	0	1
00	1	
01		
11	--	--
10		

$$D_0 = \bar{y}_1\bar{y}_0\bar{x}$$

$y_1y_0 \backslash x$	0	1
00		
01	1	
11	--	--
10		

$$D_1 = \bar{x}y_0$$

**Esercizio 2. (TWM)**

- a) Scrivere un'espressione logica, nella forma somma di prodotti che realizzi la funzione F descritta dalla seguente tabella di verità

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

$$F = A'BC + ABC'$$

Nota: nel seguito l'apice indica la negazione logica ( $A' = \text{NOT } A$ ).

- b) Riscrivere nella forma somma di prodotti la seguente espressione logica:

$$(A + B)'(A' + C')C = A'B'C$$

- c) Semplificare la seguente espressione logica:

$$(A + B + B')(B + 0)C = BC$$

- d) Riscrivere la seguente espressione utilizzando solo connettivi nand:

$$(A B) + C = \text{NAND}(\text{NAND}(A, B), C)$$

- e) Assegnare dei valori alle variabili delle due seguenti espressioni in maniera tale che la prima valga 1 e la seconda 0.  $(AB) \quad (B C)$ :

$$A = 1 \quad B = 1 \quad C = 0$$

## Architettura degli Elaboratori

### 20 Marzo 2002

#### ASSEMBLER (11 punti)

Una subroutine Assembler 68000 deve determinare se una stringa  $S$  è palindroma. Diciamo che una stringa è palindroma quando leggendola a rovescio otteniamo la stessa stringa. Ad esempio 'pattap' e 'socos' sono palindrome mentre 'pippo' non lo è.

La subroutine accetta in ingresso l'indirizzo base di  $S$  mediante il registro A0 e restituisce nel byte meno significativo del registro D0 il valore FF, nel caso la stringa sia palindroma, 0 altrimenti. La stringa, secondo l'usuale convenzione, è terminata dal byte 0. Si assuma che la lunghezza della stringa sia almeno 2 e al massimo di 65535 caratteri.

Il listato (che comprende un esempio di chiamata) è già delineato qui di seguito, con opportuni commenti. Si completino direttamente su questo foglio **tutte e sole** le parti indicate da sottolineatura (N.B.: tutti i punti sono già esplicitamente forniti).

	lea	string,a0	esempio di chiamata alla subroutine
	bsr.s	palindrome	nel caso della stringa dichiarata in calce
	break		
palindrome	movem.l	a1/a0/d1,-(sp)	salvataggio registri
	movea.l	a0,a1	inizializza A1 per cercare la fine della stringa
look4end	tst.b	(a1)+	ho trovato il byte 0?
	bne.s	look4end	se no continuo a cercare
	subq.l	#1,a1	sposto A1 indietro di una posizione (torna sullo 0)
	move.l	a1,d0	in coppia con la prossima istruzione...
	sub.l	a0,d0	calcolo in D0 la lunghezza della stringa
	lsr.w	#1,d0	divido tale lunghezza (dato unsigned) per 2
	subq.w	#1,d0	preparo D0 come contatore per una DBcc
check	move.b	(a0)+,d1	in coppia con la prossima istruzione...
	cmp.b	-(a1),d1	confronto i caratteri estremi della stringa
	dbne	d0,check	continuo fino a che sono uguali o la stringa finisce
	beq.s	esci	se mi sono interrotto per fine stringa esco (con successo)
	clr.b	d0	altrimenti imposto il fallimento ed esco
esci	movem.l	(sp)+,a0/a1/d1	ripristino registri
	rts		
string dc.b	'abcdcba',0		

**ASSEMBLER (8 punti)**

Una subroutine Assembler 68000 deve ordinare un vettore  $V$  di  $n$  numeri interi senza segno a 16 bit. La subroutine accetta in ingresso l'indirizzo base  $V$  mediante il registro A0 e il numero  $n$  nella word meno significativa del registro D0 e non restituisce output.

Il metodo di ordinamento utilizzato è il seguente: partendo dall'inizio si confronta ogni elemento fino a  $n-1$  col successivo e li si scambia nel caso il primo sia maggiore del secondo (in questo modo si ottiene nell'ultimo posto il valore maggiore). Si itera questo procedimento fermandoci, rispettivamente ad  $n-2$ , ad  $n-3$ , ..., ad 1 e a quel punto il vettore è ordinato.

Il listato (che comprende un esempio di chiamata) è già delineato qui di seguito, con opportuni commenti. Si completino direttamente su questo foglio **tutte e sole** le parti indicate da sottolineatura (N.B.: tutti i punti sono già esplicitamente forniti).

	lea move.w bsr.s break	vett,a0 #5,d0 bubblesort	esempio di chiamata alla subroutine nel caso del vettore dichiarato in calce
bubblesort	movem.l subq.w	d0/d1/a1/d2,-(sp) #2,d0	salvataggio registri inizializzo D0 per $n-1$ iterazioni
outerloop	move.w movea.l	d0,d1 a0,a1	inizializzo D1 per ' $D0+1$ ' iterazioni inizializzo A1 per scandire $V$
innerloop	move.w cmp.w bls.s move.w move.w	(a1)+,d2 (a1),d2 noswap (a1),-2(a1) d2,(a1)	in coppia con la prossima istruzione... ... confronto un elemento col successivo se il primo è minore continuo la scansione altrimenti scambio i due elementi... ...usando un registro dati come temporaneo
noswap	dbra dbra movem.l rts	d1,innerloop d0,outerloop (sp)+,d0/d1/a1/d2	proseguo la scansione itero su un numero inferiore di elementi ripristino registri
vett	dc.w	8,7,6,5,4	

# Architettura degli Elaboratori

## 5 Giugno 2002

### ASSEMBLER (15 punti)

Una subroutine Assembler (Motorola 68000) accetta in ingresso:

1. l'indirizzo base di una matrice  $M0$  di dimensione  $mxm$  di numeri interi long con segno mediante il registro A2,
2. l'indirizzo base di una matrice  $M1$  di dimensione  $mxn$  di numeri interi long con segno mediante registro A3,
3. i due numeri interi  $n$  e  $m$  nelle word meno significative di D4 e D5.

Detta subroutine somma alla matrice  $M0$  la trasposta di  $M1$ .

Ad esempio con  $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 9 & 7 & 5 & 3 & 1 \\ 1 & 3 & 5 & 5 & 5 \end{pmatrix}$  e  $\begin{pmatrix} 1 & -1 & 0 \\ 1 & -1 & 0 \\ 1 & -1 & 0 \\ 1 & -1 & 0 \\ 1 & -1 & 0 \end{pmatrix}$  la subroutine modifica la prima in  $\begin{pmatrix} 2 & 3 & 4 & 5 & 6 \\ 8 & 6 & 4 & 2 & 0 \\ 1 & 3 & 5 & 5 & 5 \end{pmatrix}$ .

Il listato (che comprende un esempio di chiamata) è già delineato qui di seguito, con opportuni commenti. Si completino direttamente su questo foglio **tutte e sole** le parti indicate da sottolineatura (N.B.: tutti i punti sono già esplicitamente forniti).

```
lea      matrix0,a2
lea      matrix1,a3
move.w   #3,d4
move.w   #5,d5
bsr.s    matrixsum
break
```

matrixsum	movem.l	a2/a3/d4-d1,-(sp)	salvataggio registri
	move.w	d4,d6	in coppia con la prossima istruzione...
	asl.w	#2,d6	...d6 manterrà '4*m'
	subq.w	#1,d4	imposto d4 come contatore di ciclo esterno
	subq.w	#1,d5	predispongo d5 come valore iniziale del ciclo interno
outerloop	move.w	d5,d7	inizializzo d7 come contatore del ciclo interno
	clr.w	d0	inizializzo l'indice d0 per l'inizio della colonna
innerloop	move.l	(a3,d0.w),d1	in coppia con la prossima istruzione, sommo...
	add.l	d1,(a2)+	...l'elemento indicizzato da d0 a quello di a2
	add.w	d6,d0	sposto l'indice d0 al prossimo elemento della colonna
	dbra.w	d7,innerloop	itero sulla colonna
	addq.l	#4,a3	sposto a3 sul prossimo inizio di colonna
	dbra.w	d4,outerloop	itero sulla riga
exit	movem.l	(sp)+,a2/a3/d4-d1	ripristino registri
	rts		

matrix0	dc.l	1,2,3,4,5
	dc.l	9,7,5,3,1
	dc.l	1,2,5,5,5
matrix1	dc.l	1,-1,0
	dc.l	1,-1,0
	dc.l	1,-1,0
	dc.l	1,-1,0
	dc.l	1,-1,0

1. Cosa si intende per traccia di un disco magnetico?
2. Perché si esegue l'operazione di ricalibrazione di un disco magnetico?
3. In cosa si differenziano i dischi RAID dai dischi standard (SLED)?
4. Si descriva brevemente come sono distribuiti i dati nel RAID di livello 0?
5. Dal punto di vista del fisico, qual'è la principale differenza tra un CD-ROM and un CD-R (CD-Recordable)?
6. Cosa s'intende per arbitraggio di un bus?
7. Quali fattori determinano la larghezza di banda di un bus?
8. Cosa indica l'acronimo ISA?
9. Una memoria principale di 64MB con parole di 64 bit, quante parole allineate contiene? E quante non-allineate?
10. Cos'è l'indirizzamento diretto?
11. Cos'è il vettore dell'interrupt?
12. Si indichi un criterio di scelta, utilizzato nelle implementazioni della memoria virtuale, per determinare quali pagine rimuovere dalla memoria principale.
13. Quali vantaggi offre la segmentazione rispetto alla paginazione?
14. In una snooping cache, come viene eseguita un operazione di scrittura in memoria, nel caso questa generi una write miss?