Esame di **Architetture degli Elaboratori** - 01/07/2021

A.A. 2020-21

Cognome e Nome _____ Matricola _____

Soluzione

Per tutta la verifica, **N** sarà uguale alle cinque o sei cifre del numero della matricola dello studente, dapprima privata di eventuali lettere, e poi trascritta nel verso opposto aggiungendo infine zeri fino a raggiungere un numero di sei cifre.
Es.: se la matricola è 237424, allora **N = 424732**
se la matricola è 237400, allora **N = 473200**
se la matricola è I-37424, allora **N = 424730.**

1 - 3 pt. Osservando le 6 cifre di N, **N = $N_5N_4N_3N_2N_1N_0$** inteso essere in base 10, si converta in base **B=N_5** il numero **N**, mostrando i calcoli necessari.

Risposta: per esempio se N = 473200, allora B = 4 e infine, sfruttando la potenza di due della base in questo caso particolare,

$473200 = 01110011100001110000_2 = 1303201300_4$.

2 - 3 pt. Osservando le 6 cifre di N, **N = $N_5N_4N_3N_2N_1N_0$** inteso essere in base 10, si rappresentino le due cifre più significative **N_5N_4** di N in aritmetica eccesso 128 **dopo avere cambiato di segno il valore decimale che esse rappresentano.** Mostrare i calcoli necessari.

Risposta: nel caso $N_5 = 4$ e $N_4 = 7$ dobbiamo rappresentare il valore decimale -47 in aritmetica eccesso 128. Poiché il valore 0 è rappresentato in quella aritmetica come 128, da 128-47 = 81 otteniamo subito la rappresentazione in base 2 di $81 = 01010001_2$.

3 - 3 pt. Un chip di memoria ha una capacità di N Byte. Se ogni anno la densità di transistor nell'unità di memoria raddoppia, quale sarà la capacità del chip tre mesi dopo a parità di superficie? Mostrare i calcoli necessari.

Risposta: Poiché in un anno la capacità raddoppia, a ogni trimestre l'incremento di capacità è uguale alla radice della radice di 2, cioè alla radice quarta di 2, cioè $2^{1/4}$. L'incremento annuale in altre parole evolve nel corso dei trimestri secondo la progressione

... -> $2^{-1/4}$N -> $2^{0/4}$N = N -> $2^{1/4}$N -> $2^{1/2}$N -> $2^{3/4}$N -> $2^{4/4}$N = 2N -> ...
Quindi, la quantità di cui dev'essere scalata la capacità è uguale a $2^{1/4}$ = ??2 = 1.189.

4. Si consideri il numero **K = N_5N_4** formato dalle due cifre più significative di **N = $N_5N_4N_3N_2N_1N_0$**. Quante porte AND a tre ingressi si riescono a realizzare con **K** transistor? Se non vengono utilizzati tutti, quante porte NOT si possono realizzare con i transistor rimanenti? Si motivi il risultato.

Risposta: ricordando che una porta AND a 3 ingressi necessita di 3+1 = 4 transistor per essere realizzata, dall'esempio si trova immediatamente K = 42 che quindi permette di realizzare 42/4 = 10 porte AND a 3 ingressi. Infine, con i restanti 2 transistor si possono realizzare 2 porte NOT.

5 - 3 pt.
a) Realizzare lo schema di un circuito combinatorio a scelta che adopera **tutte** le porte logiche trovate all'esercizio precedente:

Per esempio realizziamo:
- 3 porte AND a 3 ingressi A_1, A_2, A_3 verso una porta AND a 3 ingressi da cui esce il segnale B_1
- 3 porte AND a 3 ingressi A_4, A_5, A_6 verso una porta AND a 3 ingressi da cui esce il segnale B_2
- 1 porta AND a 3 ingressi A_7, A_8, A_9 da cui esce il segnale B_3
- 1 porta AND che accetta B_1, B_2 e B_3 come ingressi da cui esce il segnale E.

Le porte NOT possono essere messe a piacere, per esempio a negare B_1 e B_3.

b) Scrivere l'espressione booleana corrispondente al circuito combinatorio appena realizzato

Risposta: anche se logicamente il circuito ha ben poco senso, la sua espressione è
$E = B'_1B_2B'_3 = ((A_1A_2A_3)(A_1A_2A_3)(A_1A_2A_3))'(A_4A_5A_6)(A_4A_5A_6)(A_4A_5A_6)(A_7A_8A_9)'$

6 - 3 pt. Lo studente consideri il proprio nome e cognome e, tra i due, scelga la parola che contiene il numero maggiore di caratteri distinti. Questi caratteri siano i simboli che un codice binario a **lunghezza variabile** deve codificare. Si mostrino le corrispondenti codifiche nel caso in cui la loro lunghezza sia complessivamente minima.

Risposta: ad esempio nel caso del nome e cognome del docente, il nome contiene i caratteri C,D,E,F,I,O,R i quali possono essere codificati con lunghezza variabile adoperando 3 bit:

C = 00
D = 010
E = 011
F = 100
I = 101
O = 110
R = 111

7 - 3 pt. Durante la trasmissione di una sequenza di caratteri adoperando il codice all'esercizio precedente si verifica un errore. Si dia un esempio, se esiste, di errore rilevabile e un esempio, se esiste, di errore non rilevabile dallo stesso codice.

Risposta: non è difficile rendersi conto che qualunque sequenza di bit che viene ricevuta ha senso e, quindi, nessun errore è rilevabile. Per esempio, 00111100101010101000111 10... viene decodificato come 00|111|100|101|010|101|00|011|10... .

8 - 3 pt. Nell'ambiente di programmazione Arduino esiste la funzione "pinmode(pin_number, modalità)".
a) Quali sono le 3 modalità (o **mode**) che si possono impostare con questa funzione? Risposta
Le tre modalità possibili sono: OUTPUT, INPUT, INPUT_PULLUP.

b) E' possibile applicare tutte le modalità di tale funzione a un ingresso **analogico** del microcontrollore? Motivare brevemente la risposta:
É possibile. Infatti, mentre il campionamento dei segnali si può applicare solo ai pin di ingresso analogico (a_0, ..., a_N), tali pin possono essere anche liberamente utilizzati come pin digitali standard. Ne consegue che tutte le modalità disponibili possono essere applicate anche a questi pin.

9 - 3 pt. Si consideri una memoria cache associativa a **K** vie, in cui **K = N_5** è la cifra più significativa di **N = $N_5N_4N_3N_2N_1N_0$**. La cache è regolata dalla seguente politica di accesso: il dato viene cercato nella prima tabella; se il campo TAG non corrisponde allora lo stesso dato viene cercato nella tabella successiva. Se il tempo di accesso a una tabella nella cache è di 10 ns, qual è il tempo medio di accesso alla memoria nel caso di *cache hit*? Mostrare i calcoli necessari:

RISPOSTA: Poichè non è possibile sapere in quale tabella è presente il dato, mediamente un cache hit richiederà (10 + 20 + ... + K*10) / K ns

10 - 3 pt. Un processore superscalare esegue la sequenza di istruzioni macchina corrispondente alla seguente riga di codice: if R1>R2 then R1=5 else R2=85.
Si spieghi se, e come questo processore può utilizzare utilmente l'esecuzione fuori ordine e i registri ombra al fine di eseguire più rapidamente la stessa riga di codice.

RISPOSTA: Poichè la valutazione della condizione R1>R2 è onerosa, il processore assegnerà appena possibile (eventualmente fuori ordine) le costanti 5 e 85 rispettivamente a due registri ombra, e solo dopo avere risolto la condizione copierà il primo registro ombra in R1 oppure il secondo registro ombra in R2.

11 [INF] - 3pt. Considerate le sei cifre $\mathbf{MN_4N_3N_2N_1N_0}$ che formano il numero $\mathbf{N = MN_4N_3N_2N_1N_0}$, convertire il numero $\mathbf{-N_4 -2 - 4^{-M}}$ ("meno N_4 meno 2 meno 4 alla -M") in codifica *floating point IEEE 754* a 32 bit. Mostrare i calcoli.

RISPOSTA: Il numero in questione è uguale a $-(N_4+2+4^{-M})$. Quindi, può essere convertito in un binario negativo contenente una parte intera uguale a N_4+2 e una parte frazionaria uguale a 4^{-M}, subito rappresentabile in binario come $0.0...01_2$ avendo posto M-1 zeri a destra del punto decimale. Il resto dell'esercizio segue la tradizionale procedura di conversione nella codifica *floating point* richiesta.

12 [INF] - 3 pt. È data la seguente mappa di Karnaugh

AB / CD	00	01	11	10
00	1	1	S	1
01	0	0	0	0
11	S	0	0	S
10	D	0	0	D

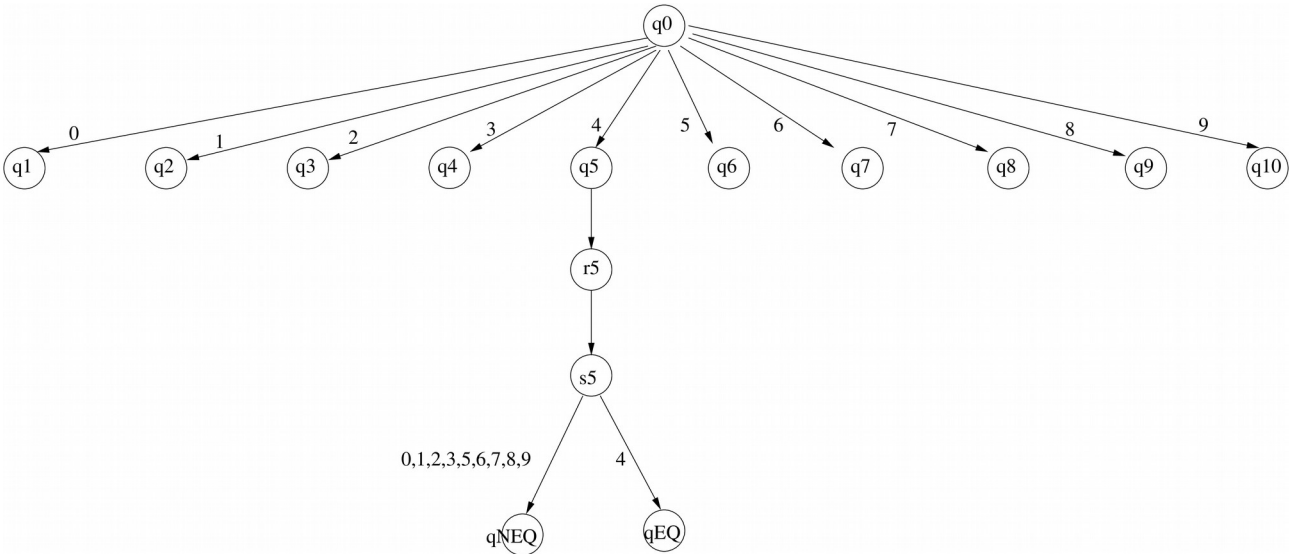
in cui S vale 0 se la cifra **più** significativa $\mathbf{N_5}$ in N è pari oppure 1 se è dispari, e D vale 0 se la cifra **meno** significativa $\mathbf{N_0}$ in N è pari oppure 1 se è dispari. Qual è l'espressione booleana dell'uscita E dal circuito combinatorio descritto da questa mappa?

RISPOSTA: in dipendenza dai valori assunti da S e D, avremo:
- 2 coperture se S=0 e D=0: E = A'C'D' + B'C'D'
- 2 coperture se S=1 e D=0: E = C'D' + B'CD
- 3 coperture se S=0 e D=1: E = A'C'D' + B'C'D' + B'CD'
- 2 coperture se S=1 e D=1: E = AB' + C'D'

Erano ovviamente accettate anche soluzioni basate sulla copertura degli zeri.

13 [INF] - 3 pt. Disegnare il grafo, oppure illustrare a parole il funzionamento di una macchina di Moore definita sull'insieme delle cifre decimali I={0,1,...,9}, in grado di riconoscere se la prima cifra $\mathbf{N_0}$ in ingresso alla macchina è **diversa o uguale** alla quarta cifra $\mathbf{N_3}$ in ingresso, in cui $\mathbf{N_0}$ è la cifra meno significativa in **N** e $\mathbf{N_1}$ è la cifra immediatamente alla sua sinistra: $\mathbf{N = N_5N_4N_3N_2N_1N_0}$. I nodi devono essere tutti etichettati in modo consistente. A questo punto si dica quali nodi la specifica macchina ha percorso durante il funzionamento.

RISPOSTA: La macchina al solito parte da uno stato iniziale q0, e di qui prosegue su uno stato scelto tra dieci possibili, chiamiamoli q1, ..., q10 a seconda del valore N_0 in ingresso compreso tra 0 e 9. A questo punto procede attraverso due successivi stati, chiamiamoli rN_1 e sN_2, fintantoche non si presenta la quarta cifra in ingresso. Sia N_0+4 l'etichetta numerica dello stato in cui la macchina è arrivata a seguito dell'ingresso N_0. Da questo stato, tracciando gli archi corrispondenti è semplice determinare che la macchina terminerà in in qEQ se N_0 è uguale a N_3, o infine in qNEQ se N_0 è diverso da N_3. A questo punto è immediato elencare i **cinque nodi** che la macchina ha percorso in dipendenza dai dati N_0 e N_3 pervenuti all'ingresso. Nella figura sotto è indicato con completezza il solo caso in cui N_0 = 4, per non ingombrare il disegno con un numero eccessivo di archi.



14 [INF] - 9 pt. Si supponga di disporre della seguente routine, la quale determina il valore massimo in un array di elementi positivi o nulli presente nella memoria (R0 punta inizialmente alla testa dell'array e R3 contiene il numero di elementi) e a quel punto lo sostituisce col valore -1:

```
subr:                           ; maximum search subroutine
        mov r2, #-1             ; start with -1 in r2
loop_s: ldr r1, [r0],#4         ; load array element in r1
        cmp r2, r1              ; if r2 < r1...
        movlt r2, r1            ; ...then copy r1 in r2
        subs r3, r3, #1         ; decrement index and set flag
        bne loop_s              ; repeat if index is nonzero
loop_s2:ldr r1, [r0,#-4]!       ; load array element in r1
        cmp r2, r1              ; if r1 is a largest value...
        moveq r1, #-1           ; ...then assign -1 to r1
        streq r1, [r0]          ; ...and -1 in largest element
        moveq pc,lr             ; ...and return
        b loop_s2               ; jump back and load next element
```

Appoggiandosi alla routine precedente, scrivere un programma in assembly per ARM il quale accede a un array di sei locazioni in memoria, inizializzate rispettivamente con le cifre $\mathbf{N_0\ N_1\ N_2\ N_3\ N_4\ N_5}$ che compongono il numero $\mathbf{N = N_5N_4N_3N_2N_1N_0}$, e crea un nuovo array in cui le stesse cifre sono ordinate in senso crescente nel verso crescente della memoria. É gradita la presenza di commenti al codice prodotto.

Soluzione:

```
@ ++++++++++++ ordina sei cifre in senso decrescente ++++++++++++
@ *************** data segment ***************
.data

ordinato:  .skip 4*6
n:         .word 7,4,3,5,6,3

@ *************** code segment ***************
.text

main:
        ldr r4, =n-4        ; load result array tail address in r4
        mov r5, #6          ; load result array size in r5
loop:   ldr r0, =n          ; load array head address in r0
        mov r3, #6          ; load array size in r3
        bl subr             ; goto subroutine
        str r2, [r4],#-4    ; largest element in result array
        subs r5, r5, #1     ; decrement result index and set flag
        bne loop            ; repeat if index is nonzero
exit:
        swi 0x11
```