

A.A. 2020-21

Cognome e Nome \_\_\_\_\_

Matricola \_\_\_\_\_

Parte **prima** (tempo: **45 minuti**)

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. Dopo avere scelto un opportuno insieme di simboli per denotare le cifre, si converta il numero 0.N ("zero punto N") in **base 20**, fermandosi alla **quinta** cifra decimale.

RISULTATO: Il numero andrebbe moltiplicato per 20 fino a quando la parte decimale del risultato non diventa nulla. Poichè si chiede di fermarsi in ogni caso alla quinta cifra decimale, la conversione è uguale alla successione  $P_1P_2P_3P_4P_5$  delle 5 parti intere  $P_1, P_2, P_3, P_4, P_5$  ottenute dalle moltiplicazioni  $D_i * 20 = P_{i+1}.D_{i+1}$ , partendo dalla prima fino alla quinta:

$$\begin{array}{lcl} & 0.N | & \\ D_1 = 0.N * 20 & - P_1 | & P_1 \\ D_2 = & D_1 * 20 - P_2 | & P_2 \\ D_3 = & D_2 * 20 - P_3 | & P_3 \\ D_4 = & D_3 * 20 - P_4 | & P_4 \\ D_5 = & D_4 * 20 - P_5 | & P_5 \end{array} \quad \text{Risultato: } 0.P_1P_2P_3P_4P_5$$

2.

- a) Si prendano le tre cifre **più** significative di N e le si converta in binario. Risultato:  **$b_8b_7...b_0$**
- b) Si prendano le tre cifre **meno** significative di N e le si converta in binario. Risultato:  **$c_8c_7...c_0$**
- c) Si esegua la **sottrazione binaria in colonna** del risultato ottenuto al punto b) meno il risultato ottenuto al punto a)

Poiché entrambi i risultati calcolati ai punti a) e b) non possono essere più estesi di 9 bit, segue immediatamente il risultato adoperando la tradizionale sottrazione in colonna:

$$\begin{array}{r} b_8 b_7 \dots b_0 - \\ c_8 c_7 \dots c_0 = \\ \hline e_8 e_7 \dots e_0 \end{array}$$

in cui  $e_8 e_7 \dots e_0$  è ancora un numero (positivo oppure negativo) al massimo di 9 bit.

RISULTATO:  $e_8 e_7 \dots e_0$

3. Un chip di memoria ha una capacità di N Byte. Se ogni anno la densità di transistor nell'unità di memoria triplica, e la capacità del chip viene aggiornata ogni 18 mesi, dopo quanti aggiornamenti un chip avrà oltrepassato la capacità di 1 GB a parità di superficie occupata all'interno della scheda madre?

Ogni aggiornamento porta una capacità che è  $r = 3\sqrt[3]{3} = 3^{3/2}$  volte quella del chip precedente. Dunque, occorre contare il numero m di moltiplicazioni per quel fattore tali che il risultato superi il valore  $10^9$  oppure  $2^{30}$ :

$$Nr^m > 2^{30} \implies m > \log_r(2^{30}/N) = (\log_{10}(2^{30}/N))/(\log_{10} r).$$

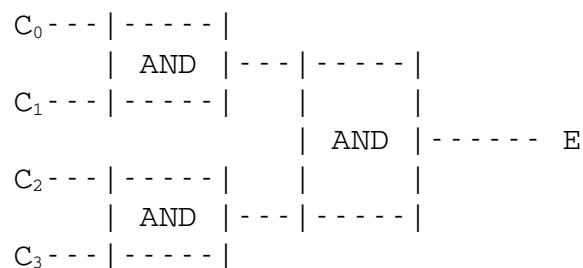
RISPOSTA: il più piccolo m intero tale che  $m > \log_r(2^{30}/N)$

4.

a) Si calcoli il **resto intero di N/16** e lo si converta in un numero binario a 4 bit. Risultato:  $b_3 \dots b_0$

b) Adoperando solamente **porte NOT** e **porte AND binarie**, si realizzi un circuito combinatorio che rileva se 4 segnali d'ingresso binari  $C_0, C_1, C_2, C_3$  assumono la configurazione ottenuta al punto a), in tal caso producendo il valore  $E = 1$ . In qualunque altro caso il circuito produrrà il valore  $E = 0$ .

RISPOSTA: sia  $C_i = b_i$  se  $b_i = 1$ , oppure  $C_i = \text{NOT}(b_i)$  se  $b_i = 0$ . Allora, aggiunte le opportune porte NOT ove necessario, il circuito è



5. Si scriva la tabella di verità che descrive il circuito al punto 4. Da essa si derivi un circuito in **forma canonica**, cioè

formato dall'OR di porte logiche AND, entrambe non necessariamente binarie.

- a) Di quante porte AND è composto il circuito? RISPOSTA: \_\_\_\_\_
- b) Di quante porte OR è composto il circuito? RISPOSTA: \_\_\_\_\_
- c) Di quanti transistor è composto il circuito? RISPOSTA: \_\_\_\_\_

RISPOSTA: C'è un'unica riga nella tabella di verità in cui  $E = 1$ , corrispondente alla configurazione  $b_0, b_1, b_2, b_3$  dei 4 ingressi. È dunque sufficiente sostituire le tre porte AND del circuito all'esercizio precedente con una porta AND a 4 ingressi, mantenendo le porte NOT già presenti. Parallelamente, la porta OR è superflua in quanto appunto una sola riga della tabella ha uscita uguale a uno. A questo punto è sufficiente ricordare che

- ogni porta NOT contiene 1 transistor
- la porta AND a 4 ingressi contiene 5 transistor.

6. Un codice **non binario a lunghezza fissa** definito sull'alfabeto  $A = \{0, 1, \dots, 9\}$  è progettato per trasmettere parole di 6 simboli. Le codifiche sono costituite dai 6 simboli in questione più i simboli costituenti la somma algebrica dei primi 6 simboli. Questa infatti viene adoperata dal ricevitore per eseguire un semplice controllo di correttezza prima di accettare o meno la parola.

- a) Di quanti simboli si compone ogni codifica? Risposta: \_\_\_\_\_
- b) Qual è la codifica della parola N? Risposta: \_\_\_\_\_
- c) [FACOLTATIVA: RISPONDERE DOPO AVERE RISPOSTO A TUTTE LE ALTRE DOMANDE] Esiste un sottoinsieme di parole la cui probabilità di essere erroneamente accettate dal ricevitore è massima?

RISPOSTA: a) 8 simboli; b) N seguito dalla somma delle cifre di cui il numero è composto; c) Il sottoinsieme in questione è costituito dalle parole la cui somma è ottenibile dal maggior numero di combinazioni di 6 cifre possibili. In questo senso, le parole 000000 e 999999 sono le meno esposte a errore in quanto sono le uniche le cui cifre hanno per somma rispettivamente 00 e  $9 \cdot 6 = 54$ . Dualmente, quindi, le parole più esposte a errore sono quelle la cui somma delle cifre è uguale a  $(0+54)/2 = 27$ .

7. Un bus seriale trasmette pacchetti di M bit, in cui M è la cifra più significativa in N. Se la banda passante del bus è di N B/s ("N Byte per secondo"), qual è il numero massimo di pacchetti che il bus considerato riesce a recapitare in un secondo se il tempo per l'invio di ogni pacchetto e il tempo di attesa prima di inviare il successivo pacchetto sono identici?

RISULTATO: la banda passante è di  $N \text{ B/s} = 8 \cdot N \text{ bit/s}$ . Poiché un pacchetto contiene M bit, i pacchetti che potrebbero essere trasmessi in un secondo sarebbero  $8 \cdot N / M$ . Tuttavia, dovendo il trasmettitore attendere un tempo identico a quello per inviare

ogni pacchetto, i pacchetti effettivamente trasmessi sono la metà. Infine, del risultato in questione dobbiamo prendere l'intero arrotondato per difetto, cioè in definitiva **il più grande intero minore di  $8 \cdot N/M/2$** .

8. Sia M la cifra più significativa in N. Volendo utilizzare **M input digitali** e **M output digitali** sul processore AVR a 8 bit in una Arduino UNO,

1) qual è il **numero minimo di porte** del microcontrollore che occorre occupare?

2) **quanti registri** di memoria occorre modificare per impostare la porta (o le porte) ?

Motivare in modo sintetico le risposte.

RISPOSTA:

1) Ricordando che ogni porta del processore indirizza al massimo 8 pin di input o di output, ci serviranno almeno  **$P = (2 \cdot M)/8$**  porte del microcontrollore, in cui P è l'intero più piccolo maggiore di  $2 \cdot M/8$ .

2) Per impostare i pin associati a ogni porta P utilizzata occorre modificare il **registro di direzione DDR** corrispondente; quindi, occorre modificare P registri. Inoltre, per ogni porta a cui sono associati dei pin di input sarà necessario modificare il registro PORT per attivare o disattivare la resistenza di PULLUP.

9. In un'architettura a 32 bit, una memoria cache ad accesso diretto è formata da  **$2^{10+M}$  entry** ciascuna di 16 Byte, in cui M è la cifra più significativa in N. Quanto è esteso il campo TAG presente in ogni entry, che com'è noto contiene la parte più significativa dell'indirizzo di memoria (in questo caso a 32 bit)?

RISPOSTA: Se ogni linea di cache è lunga 16 Byte, allora i 4 bit meno significativi di ogni indirizzo specificano le locazioni appartenenti a una stessa linea. Poi, occorrono  $10+M$  bit per identificare la entry. Ne consegue che restano  **$32 - (10+M+4) = 18-M$  bit** occupati dal campo TAG.

10. Un ISA (Instruction Set Architecture) a **16 bit** di un processore contenente **16 registri** dedicati al calcolo di operazioni a due operatori, rende disponibili **M istruzioni aritmetiche a tre argomenti**, ciascuna contraddistinta da un proprio codice operativo. M è la cifra più significativa in N. I tre argomenti rappresentano l'etichetta del registro in cui depositare il risultato, l'etichetta del registro da cui leggere il primo operatore e una costante numerica che rappresenta il secondo operatore. In assenza di ulteriori funzionalità

nell'istruzione macchina, quanti bit restano al massimo liberi per memorizzare la costante numerica?

RISPOSTA: La specifica di 16 diverse etichette di registro per la lettura e per la scrittura di un dato richiede  $4+4 = 8$  bit. La specifica di  $M$  istruzioni richiede ulteriori  $\log_2 M$  bit, in cui scegliamo l'intero immediatamente superiore. I restanti  **$16-8-\log_2 M$  bit**, in cui scegliamo l'intero immediatamente inferiore, sono disponibili per memorizzare una costante numerica.

11 [INF]. Dette rispettivamente  $N_{sx}$  la cifra **più** significativa in  $N$  e  $N_{dx}$  la cifra **meno** significativa in  $N$ , convertire il numero  $2^{N_{dx}} / 2^{N_{sx}+11}$  (" $2$  alla  $N_{dx}$  fratto  $2$  alla  $(N_{sx}+11)$ ") in codifica *floating point IEEE 754* a 32 bit.

RISULTATO: Immediatamente risulta un numero binario nella forma  $0.0...001$ , contenente solo una parte decimale formata da  $N_{sx}+11-N_{dx}-1$  zeri seguiti da un bit uguale a uno. A questo punto il numero è scritto in forma binaria esponenziale:  $1E-(N_{sx}+11-N_{dx})$ , da cui immediatamente la codifica richiesta convertendo il valore  $-(N_{sx}+11-N_{dx})$  in notazione eccesso 127.

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

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

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

RISPOSTA: poiché i quattro angoli della mappa presentano valori identici, se  $S$  è pari ci saranno una o due coperture di "uni" estese su due celle a seconda del valore assunto da  $D$ . Quindi, indicando la negazione col simbolo  $'$ ,

$E = A'BD'$  se  $D = 0$  oppure  $E = A'BD' + BCD'$  se  $D = 1$ .

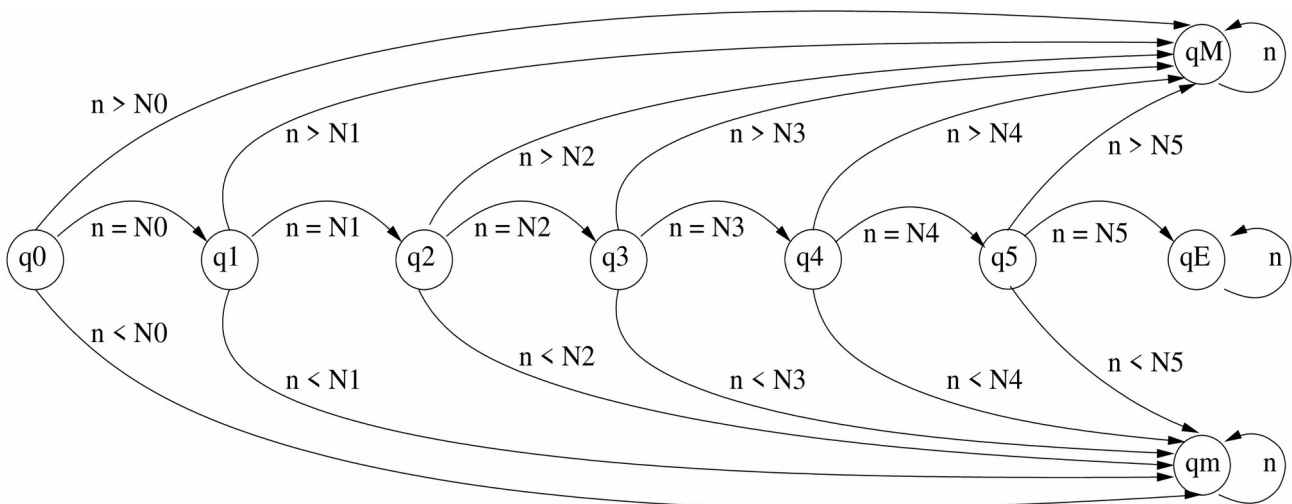
Se invece  $S$  è dispari allora avremo due o tre coperture di "uni" estese su 4 celle a seconda del valore assunto da  $D$ . Quindi,

$E = B'D' + A'D'$  se  $D = 0$ , oppure  $E = B'D' + A'D' + CD'$  se  $D = 1$ .

Erano ovviamente accettate anche soluzioni basate sulla copertura degli zeri.

13 [INF]. Disegnare il grafo, oppure illustrare a parole il funzionamento di una macchina di Moore definita sull'insieme delle cifre decimali  $I=\{0,1,\dots,9\}$ , in grado di riconoscere se le prime 6 cifre in ingresso alla macchina formano un numero **maggiore, minore o uguale al numero**  $N = N_0N_1N_2N_3N_4N_5$ . Per convenzione si assuma che le cifre in ingresso alla macchina siano via via **meno** significative all'interno della sequenza numerica che complessivamente definiscono. In più, si assuma che la macchina abbia accesso a ciascuna cifra  $N_i$  che compone il numero  $N$ .

RISPOSTA: un numero naturale è maggiore o minore di un altro se la cifra più significativa del primo numero è rispettivamente maggiore o minore di quella del secondo numero; altrimenti occorre procedere al confronto della cifra immediatamente meno significativa di entrambi i numeri. Se, dunque,  $N$  possiede 6 cifre, la macchina realizzerà l'algoritmo appena visto passando da uno stato iniziale  $q_0$  a tre possibili stati finali,  $q_M$ ,  $q_E$  e  $q_m$ , rispettivamente se una tra le 6 cifre in ingresso è maggiore o minore alla rispettiva cifra che compone  $N$ , altrimenti giungendo in  $q_E$ :



L'uscita dalla macchina è significativa per esempio appena essa sarà diversa da zero adoperando le seguenti associazioni stato -> uscita:  $q_M \rightarrow 3$ ,  $q_E \rightarrow 2$ ,  $q_m \rightarrow 1$ ,  $q_{0,1,2,3,4,5} \rightarrow 0$ .

14 [INF]. Scrivere un programma in assembly per ARM il quale, letto il valore  $N$  da un word in memoria principale, **restituisce in due locazioni la cifra più grande e quella più piccola** contenute in  $N$ . E' utile appoggiarsi alla subroutine riportata qui sotto, la quale esegue la divisione intera  $R0/R1$  tra gli interi positivi contenuti in  $R0$  e  $R1$ , restituendo in  $R0$  il resto della divisione intera e in  $R1$  il risultato della stessa:

divisione: ; r0 / r1 (interi positivi)

```

                                ; resto in r0, risultato in r1
                                ; contatore
                                ; inizializza r2 a 0
loop:   mov r3, #31
                                ; risultato parziale
                                ; se r1 < (r0>>) ...
                                ; ...sottrai <<r1 a r0
                                ; aggiorna risultato parziale
                                ; aggiorna contatore
                                bge loop
                                mov r1, r2
                                mov pc, lr

```

```
@ ***** data segment *****
.data
```

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

```

divisione:                                ; r0 / r1 (interi positivi)
                                           ; resto in r0, risultato in r1
    mov r3, #31                            ; contatore
    mov r2, #0                            ; inizializza r2 a 0
loop:
    mov r2, r2, lsl #1                    ; risultato parziale
    cmp r1, r0, lsr r3                    ; se r1 < (r0>>) ...
    suble r0, r0, r1, lsl r3              ; ...sottrai <<r1 a r0
    addle r2, r2, #1                      ; aggiorna risultato parziale
    subs r3, r3, #1                      ; aggiorna contatore
    bge loop
    mov r1, r2
    mov pc, lr

```