

Cognome e nome dello studente:

Matricola:

A.A. 2006-2007 – Seconda prova in itinere – 24 Aprile 2007

NB Gli esercizi A e D sono **obbligatori**.

**A - [9] Progettare una macchina a stati finiti in grado di leggere un testo e di riconoscere all'interno di una stringa di caratteri la sequenza: 'AAB'.**

Supponiamo che:

la macchina legga in ogni istante un carattere e che questo carattere sia o '0', 'A', 'B' o 'C'.  
all'inizio la stringa sia vuota.

Definire lo STG, la STT, codificarla e realizzare il circuito mediante porte logiche. Qual è il cammino critico della macchina?

**B - [2] In quali segmenti viene suddivisa la memoria gestita da un processore MIPS? Cosa si intende per processore CISC e RISC? Indicare i pregi e difetti di ciascun tipo di architettura. Cosa si intende per codifica big-endian o little endian? Riguarda i dati o le istruzioni? Cosa rappresenta una direttiva al compilatore? Farne un esempio.**

**C - [3] Come viene indirizzata la memoria del MIPS? Dato un indirizzo della memoria dati, è univocamente determinata l'istruzione che legge il dato da quell'indirizzo della memoria e lo trasferisce nel register file? Cosa si intende per "register spilling"? Se il dato si trova all'indirizzo binario: 0001 0000 0000 0000 0000 0000 0000 0100 quale sarà l'istruzione assembly in grado di leggerlo?**

**D - [9] Scrivere la procedura ricorsiva, **funz**, che calcola una funzione di un numero intero  $N > 0$ , in modo ricorsivo, secondo la seguente formula:  $\text{funz}(N) = N + 2 * \text{funz}(N-1)$ . Rispettare le convenzioni sui registri. Non è necessario specificare le istruzioni del chiamante (main).  $\text{Funz}(0) = 1$ .**

**E - [5] Dati i seguenti due programmi:**

**Main:**

```
li $v0,2
sub $t0, $t1, $t2
j dopo
jal Proc_A
add $t1, $t1, $t3
lw $s0, 0($gp)
```

dopo:

...

Scrivere il programma eseguibile (linked), risultante, sapendo che il segmento testo del Main è di 2Kbyte e di Proc\_A è di 512 Byte; e che il segmento dati del Main è di 12Kbyte e di Proc\_A è di 1Kbyte.

**Proc\_A**

```
li $v1, 1
sub $t0, $t2, $t3
sw $s0, 0($gp)
....
```

**F - [2] Indicare le modalità di indirizzamento previste dal MIPS.**

**G - [3+3] Tradurre in linguaggio Assembly e poi in linguaggio macchina il seguente spezzone di codice C:**

```
i = 0; j = 2;
```

```
for (i=0; i<N; i++)
```

```
{ A[i+j] = s + p + i*2; j = j * 2;
```

```
}
```

Alcuni codici operativi: lw = 35; sw = 43; add; 0 (funct = 32); jr = 0 (funct = 8); mfhi = 0 (funct = 16); mflo = 0 (funct = 18); mult = 0 (funct = 24); beq = 4; slt = 0 (funct = 42); slti = 10; bne = 5; sub = 0 (funct = 34); addi = 8; j = 2.

Alcuni numeri di registro: \$t0 = \$8, \$a0 = \$4, \$v0 = \$2, \$s0 = \$16, \$sp = \$29, \$r1 = \$31.

Supporre che l'istruzione i = 0, abbia indirizzo in memoria 0x1000.