

Compito (tempo a disposizione: 2 ore e 30 minuti)

n DOMANDA 1 (40 minuti) *Compilare la seguente function*

N.B. scansione con puntatore e valore soglia

\$a0, \$a1, \$a2, \$a3

```
int esempio (int A[], int B[], int k,  
            int n)
```

```
{
```

```
    int i; int app;
```

```
    i = 0;
```

```
    while i != n {
```

```
        app = f(A[i]);
```

```
        if app < n
```

```
            B[i] = app + k;
```

```
            else B[i] = A[i];
```

```
            i = i + 1;
```

```
        }
```

```
    return app;
```

```
}
```

n **DOMANDA 2** (20 minuti)

Relativamente all'implementazione *monociclo* della CPU della macchina MIPS senza pipelining (vedi schema), dopo aver discusso il formato binario dell'istruzione *add*, evidenziare il percorso dei dati relativo a tale istruzione, ed assegnare i valori corretti a tutte le linee di controllo (per alcune di esse il valore può essere scelto casualmente perchè ininfluente).

si prega di svolgere l'esercizio sullo schema allegato (specificando il percorso e i valori delle linee di controllo direttamente sullo schema, e le motivazioni sul retro del foglio - *mettere anche il nome!!*)

n **DOMANDA 3** (20 minuti)

Presentare schematicamente ma senza ambiguità le trasformazioni subite dall'indirizzo generato dalla CPU in un sistema di memoria comprendente sia cache che memoria virtuale (con TLB), *evidenziando il ruolo e le dimensioni dei vari campi dell'indirizzo*

n **DOMANDA 4** (30 minuti)

Fornire una formulazione dettagliata del tempo di CPU relativo all'utente in una macchina con un livello di cache, **discutendo in particolare gli aspetti architetturali che incidono sul valore delle varie componenti e le interdipendenze tra tali componenti.**

n **DOMANDA 5** (20 minuti)

Con riferimento alla *pipeline con schedulazione dinamica* specificare:

1. quale è il ruolo dell'unità di consegna
2. che differenze ci sono tra il PowerPC e il PentiumPro **relativamente al trattamento dei bit prelevati dalla cache istruzioni**

Domanda 6 (20 minuti)

- n **Valutare il numero di cicli di clock necessari a completare l'esecuzione di questo programma nel caso di pipeline semplice (si ipotizzi che il ciclo si esegue 100 volte).**
- 1. Si supponga di essere nel caso ideale di assenza di fallimenti in cache
 - 2. Si adotti la tecnica basata sulla predizione di fallimento per la gestione delle criticità sul controllo
 - 3. Si adotti la tecnica di propagazione per la risoluzione delle criticità sui dati
 - 4. Si supponga di non poter effettuare il riordinamento del codice per criticità di tipo carica-e-usa
 - Discutere il punto 2).

```
add $t1,$a0,$t3
add $t0,$zero,$a0
loop: beq $t0,$t1,exit
      lw  $t3,0($t0)
      add $t3,$s1,$t3
      sw  $t3,0($t0)
      add $t4,$s2,$t3
      addi $t0,$t0,4
      j  loop
exit: sw $s1,0($s4)
```

n ***N.B. Evidenziare tutti gli anticipi e gli stalli necessari, specificando il numero di volte per cui si ripetono***