

COMPITO DI ARCHITETTURE DEI CALCOLATORI 2006/2007
19 luglio 2007

NOME:

COGNOME:

MATRICOLA:

Scrivere in stampatello NOME, COGNOME e MATRICOLA su ogni foglio.

Al termine, si DOVRANNO consegnare tutti i fogli ricevuti.

ESERCIZIO 1: [6 punti] Si consideri il codice sotto riportato:

| | |
|-------------------------|-------------------------|
| A: add \$s2,\$a2,\$zero | I: beq \$v0,\$zero,____ |
| B: addi \$s4,\$zero,0 | L: sll \$t1,\$s5,2 |
| C: addi \$s5,\$zero,0 | M: add \$t1,\$s0,\$t1 |
| D: slt \$t0,\$s5,\$s2 | N: lw \$t2,0(\$t1) |
| E: bne \$t0,\$zero,P | O: sll \$t4,\$s4,2 |
| F: add \$a0,\$s0,\$zero | P: add \$t4,\$s1,\$t4 |
| G: add \$a1,\$s5,\$zero | Q: sw \$t2,0(\$t4) |
| H: jal test | |

Nell'ipotesi che il codice venga collocato in memoria a partire dall'indirizzo 0x004010AC (segmento testo), si chiede di determinare – fornendo adeguata spiegazione:

- 1- quale etichetta, nell'insieme {A, . . . , Q}, deve essere inserita nell'istruzione assembler

I: beq \$v0,\$zero,____

affinche', dopo la fase di assemblaggio, il campo address dell'istruzione beq sia pari a:

- a) 1111111111111110
b) 0000000000000011;
c) 1111111111111010;

- 2- l'indirizzo di memoria al quale salta il processore dopo aver eseguito l'istruzione

H: jal test

tenendo conto che, al termine della fase di “collegamento”, il campo address dell'istruzione jal è pari a:

- a) 0x170AF01;
- b) 0x326C0B4;
- c) 0x0F24BBE.

ESERCIZIO 2: [9 punti] Scrivere una procedura in Assembler MIPS che, dato un array di interi *a* e la sua dimensione *n*, ordini l'array usando l'algoritmo di bubblesort. Lo swap degli elementi deve essere realizzato tramite un'opportuna procedura.

L'indirizzo base dell'array *a* e la sua dimensione *n* sono passati come parametri in ingresso in \$a0 e \$a1, rispettivamente. Si richiede di commentare in codice. Non si possono usare pseudoistruzioni.

Possibile implementazione in C

```
void bubblesort(int a[ ], int n)
{
    int i;
    int swapped;
    do {
        swapped=0;
        for(i=0; i<n-1; i++)
            if(a[i]>a[i+1]) {
                swap(&a[i], &a[i+1]);
                swapped=1;
            }
    } while(swapped);
}

void swap(int *a, int *b) {
    int tmp=*a;
    *a=*b;
    *b=tmp;
}
```

ESERCIZIO 3: [4 punti] Illustrare l'organizzazione interna ed il funzionamento dell'ALU del processore MIPS.

ESERCIZIO 4: [6 punti] Descrivere le caratteristiche ed i relativi vantaggi dei sistemi RAID 3 e RAID 4.

ESERCIZIO 5. [6 punti] Modificare il processore MIPS multiciclo in allegato, in modo tale che, oltre al set ridotto di istruzioni MIPS, supporti la seguente istruzione:

| | | |
|------------|-----------|----------|
| Istruzione | Semantica | Codifica |
|------------|-----------|----------|

swap rs, rt

| | | | | | |
|---|----|----|---|---|---|
| 5 | rs | rt | 0 | 0 | 0 |
|---|----|----|---|---|---|

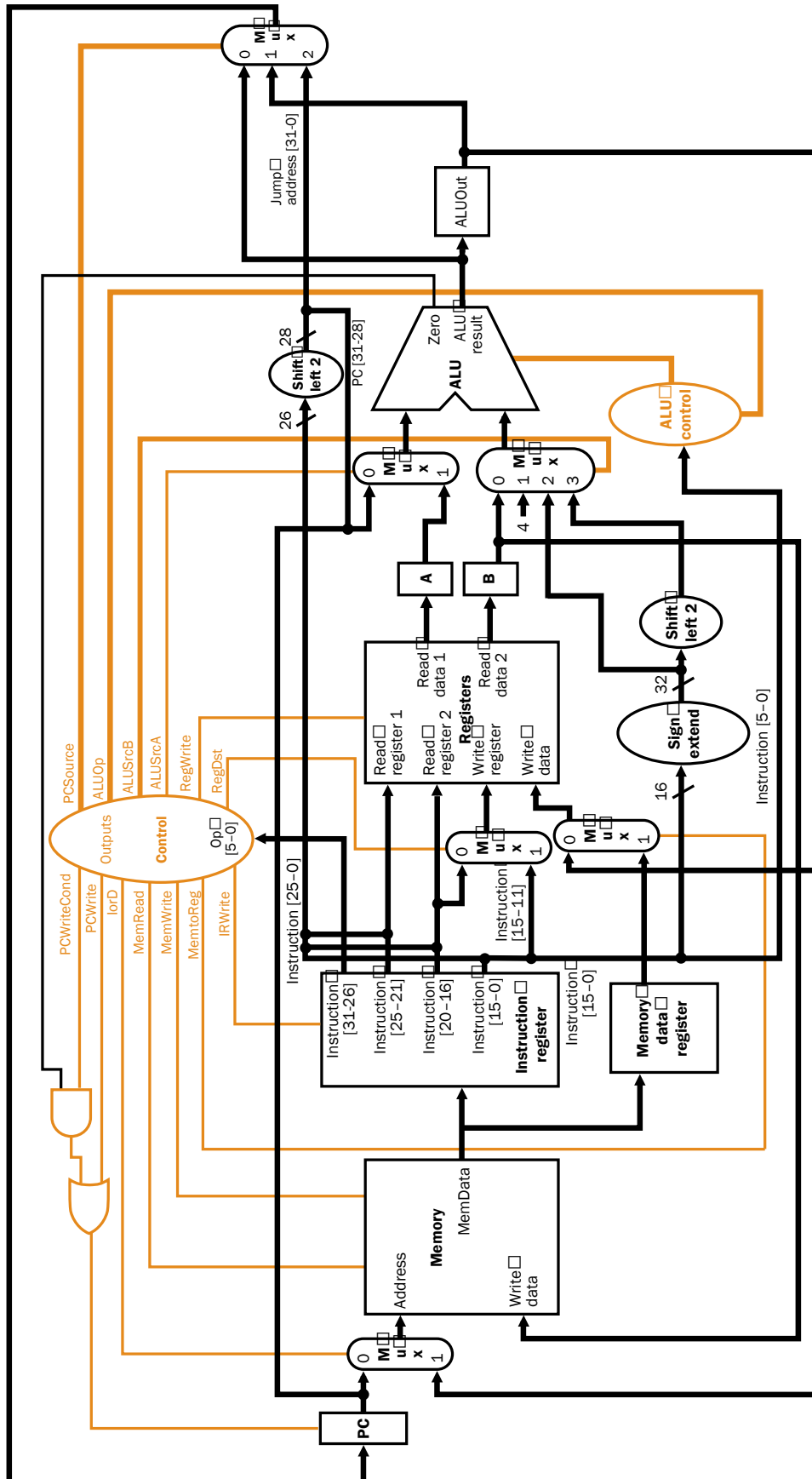
R[rs]=R[rt], R[rt]=R[rs],
PC=PC+4;

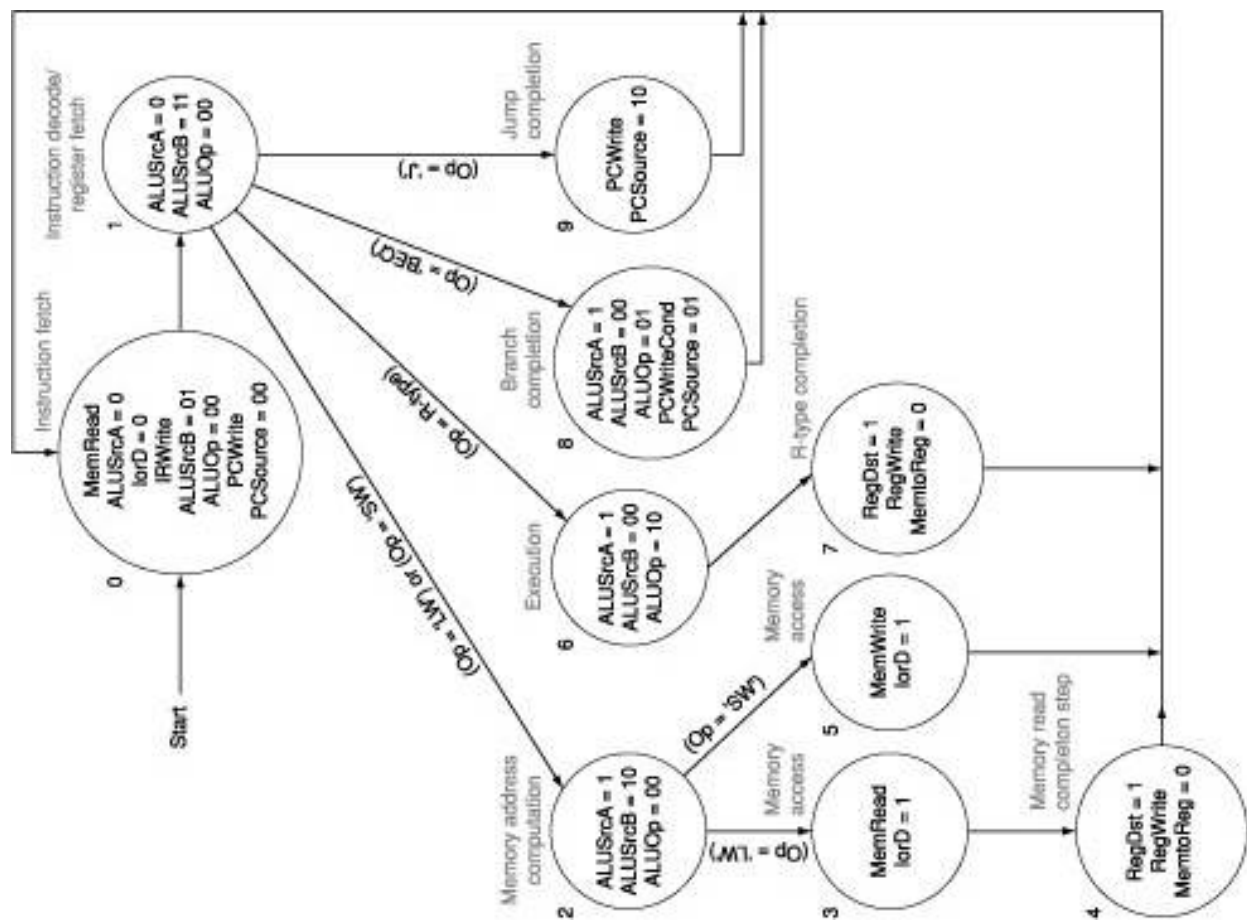
ALLEGATO: IL PROCESSORE MIPS MULTICICLO

NOME:

COGNOME:

MATRICOLA:





II PARZIALE DI ARCHITETTURE DEI CALCOLATORI 2006/2007
19 luglio 2007

NOME:

COGNOME:

MATRICOLA:

Scrivere in stampatello NOME, COGNOME e MATRICOLA su ogni foglio.

Al termine, si DOVRANNO consegnare tutti i fogli ricevuti.

ESERCIZIO 3: [8 punti] Illustrare l'organizzazione interna ed il funzionamento dell'ALU del processore MIPS.

ESERCIZIO 4: [12 punti] Descrivere le caratteristiche ed i relativi vantaggi dei sistemi RAID 3 e RAID 4.

ESERCIZIO 5. [12 punti] Modificare il processore MIPS multicycle in allegato, in modo tale che, oltre al set ridotto di istruzioni MIPS, supporti la seguente istruzione:

Istruzione

Semantica

Codifica

swap rs, rt

$R[rs]=R[rt]$, $R[rt]=R[rs]$,
 $PC=PC+4$;

| | | | | | |
|---|----|----|---|---|---|
| 5 | rs | rt | 0 | 0 | 0 |
|---|----|----|---|---|---|