

COMPITO DI ARCHITETTURE DEI CALCOLATORI 2006/2007
11 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: [3 punti] Si consideri il codice sotto riportato:

add \$s2,\$a2,\$zero	beq \$v0,\$zero, label
addi \$s4,\$zero,0	sll \$t1,\$s5,2
add \$a1,\$s5,\$zero	...
j etichetta	...

Nell'ipotesi che il codice venga collocato in memoria a partire dall'indirizzo 0x005010AC (segmento testo), si chiede di determinare – fornendo adeguata spiegazione – l'indirizzo di memoria al quale salta il processore dopo aver eseguito l'istruzione

j etichetta

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

- a) 0x020AF04;
- b) 0x308430A;
- c) 0x00FFE20.

ESERCIZIO 2: [4 punti] Con riferimento alle convenzioni di chiamata a procedura usate nell'assembler MIPS, si chiede di illustrare le convenzioni relative ai registri salvati dal chiamante (*caller saved*) e quelle relative ai registri salvati dal chiamato (*callee saved*). Si chiede, inoltre, di fornire un esempio di utilizzo di ciascuna delle due convenzioni.

ESERCIZIO 2: [8 punti] Scrivere una procedura **ricorsiva** in Assembler MIPS che, dato in ingresso un intero non negativo n , calcoli la funzione $T(n)$ così definita:

$$T(n) = \begin{cases} T(n-1) + \max(T(n-2), 5) & \text{se } n > 0 \\ 1 & \text{se } n \leq 0 \end{cases}$$

La funzione massimo deve essere realizzata tramite un'opportuna procedura. Il valore n e' passato come parametro in ingresso in \$a0. Il valore $T(n)$ e' restituito in \$v0. Si richiede di commentare in codice. Non si possono usare pseudoistruzioni.

Possibile implementazione in C

```
int procedura (int n) {
    if (n<=0)
        return 1;
    else
        return procedura(n-1)
        +max(procedura(n-2),5);
}

int max(int a, int b) {
    if (a<b)
        return b;
    else
        return a;
}
```

ESERCIZIO 4: [2 punti] Con riferimento allo standard IEEE 754 a precisione singola, determinare i numeri floating point rappresentati dalle seguenti stringhe di bit:

- a) 1 10000101 110110110000000000000000
 b) 0 10001000 011010101110000000000000

ESERCIZIO 5: [7 punti] Descrivere l'implementazione ed il funzionamento di unita' di controllo multiciclo con sequenzializzatore.

ESERCIZIO 6. [7 punti] Modificare il processore MIPS multiciclo in allegato, in modo tale che, oltre al set ridotto di istruzioni MIPS, supporti la seguenti istruzione:

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

loop rs, label

5	rs	0	address
---	----	---	---------

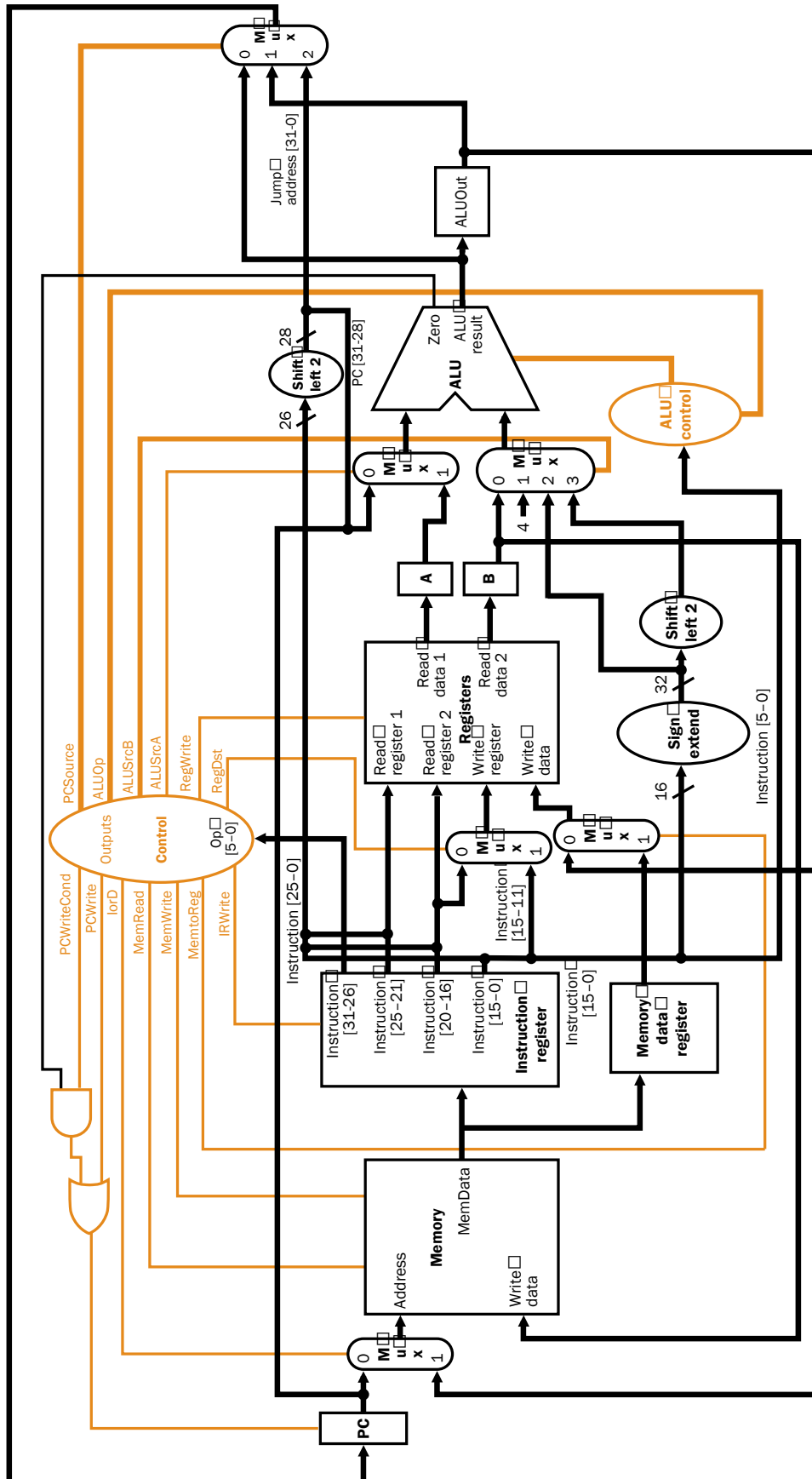
```
R[rs]=R[rs]-1;
if (R[rs]==0)
    PC=PC+4+sign-extend(address)<<2;
else
    PC=PC+4;
```

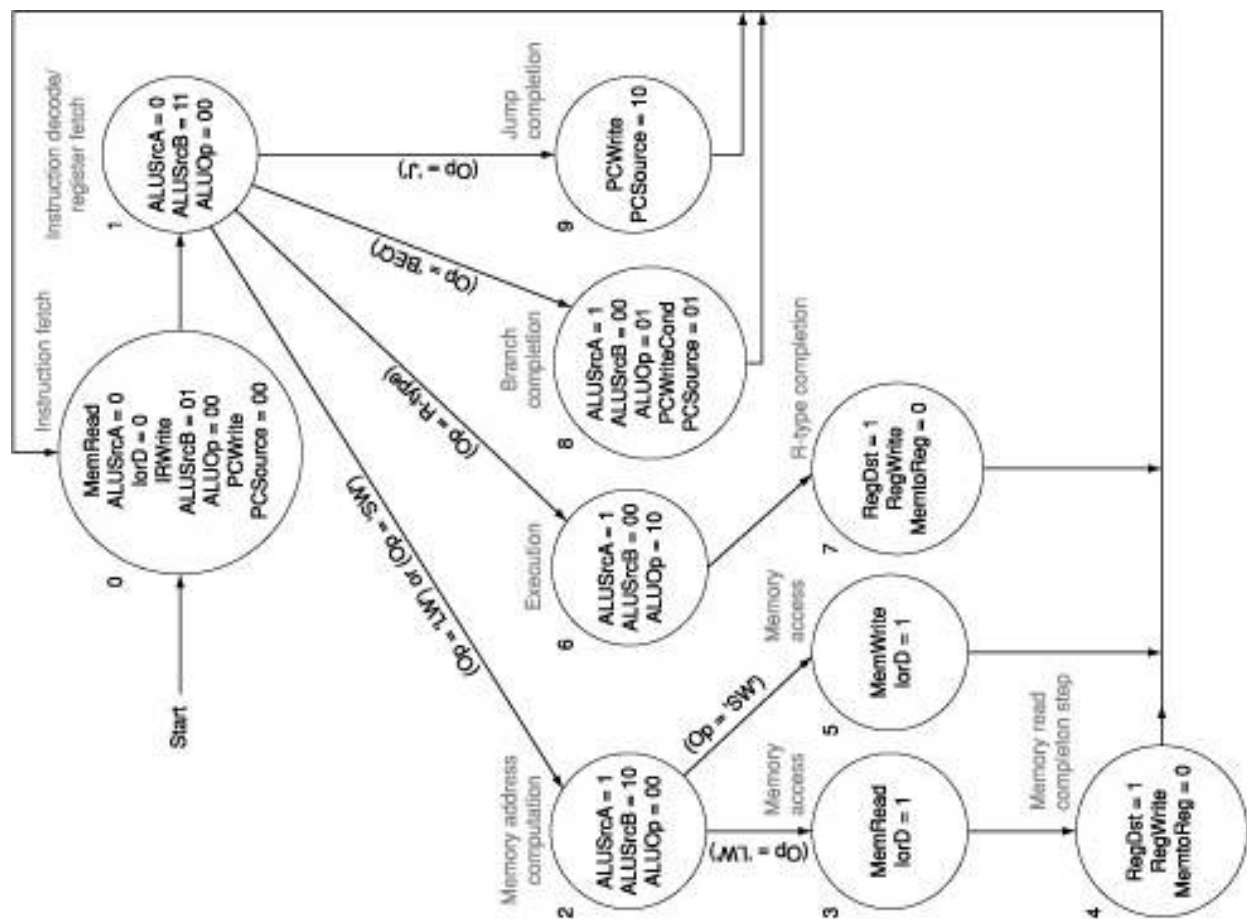
ALLEGATO: IL PROCESSORE MIPS MULTICICLO

NOME:

COGNOME:

MATRICOLA:





II PARZIALE DI ARCHITETTURE DEI CALCOLATORI 2006/2007
11 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 4: [4 punti] Con riferimento allo standard IEEE 754 a precisione singola, determinare i numeri floating point rappresentati dalle seguenti stringhe di bit:

c) 1 10000101 110110110000000000000000

d) 0 10001000 011010101110000000000000

ESERCIZIO 5: [14 punti] Descrivere l'implementazione ed il funzionamento di unita' di controllo multiciclo con sequenzializzatore.

ESERCIZIO 6. [14 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
------------	-----------	----------

loop rs, label

5	rs	0	address
---	----	---	---------

```
R[rs]=R[rs]-1;  
if (R[rs]==0)  
    PC=PC+4+sign-extend(address)<<2;  
else  
    PC=PC+4;
```