

COMPITO DI ARCHITETTURE DEI CALCOLATORI 2005/2006
21 luglio 2006

NOME:

COGNOME:

MATRICOLA:

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

Al termine, si DOVRANNO consegnare tutti i fogli ricevuti.

ESERCIZIO 1: [5 punti] Descrivere in dettaglio le convenzioni di chiamata a procedura usate nell'assembler MIPS.

ESERCIZIO 2: [11 punti] Scrivere una procedura **ricorsiva** in Assembler MIPS che, dato in ingresso un intero non negativo n ed un array A di interi non negativi, calcoli la funzione $T(n,A)$ così definita:

$$T(n,A) = \begin{cases} 2 \cdot T(n-1,A) + 1 & \text{se } n > 0 \text{ AND } n \in A \\ T(n-1,A) + 2 & \text{se } n > 0 \text{ AND } n \notin A \\ 0 & \text{se } n = 0 \end{cases}$$

La funzione di appartenenza (\in) deve essere realizzata tramite un'opportuna procedura. Il valore n , l'indirizzo dell'array A e la sua dimensione m sono passati come parametri in ingresso in $\$a0$, $\$a1$ e $\$a2$, rispettivamente. Il valore $T(n,A)$ è restituito in $\$v0$. Si richiede di commentare in modo opportuno il codice e di indicare quali istruzioni devono essere eventualmente rilocate dal *linker*. Non si possono usare pseudoistruzioni.

Possibile implementazione in C

```
int procedura (int n,int *A, int m) {
    if (n==0)
        return 0;
    else {
        if (is_in(n,A,m))
            return 2*procedura(n-1,A,m)+1;
        else
            return procedura(n-1,A,m)+2;
    }
}
```

```

int is_in(int x, int *B, int n) {
    int j=0;

    while(j<n) {
        if (x == B[j])
            return 1;
        j++;
    }
    return 0;
}

```

ESERCIZIO 3: [2 punti] Illustrare come l'ALU del processore MIPS supporta l'implementazione dell'istruzione `slt`.

ESERCIZIO 4: [6 punti] Descrivere le tecniche di arbitraggio del bus.

ESERCIZIO 5: [8 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 (formato I)			
push rt	M[R[29]]=R[rt], R[29]=R[29]-4, PC=PC+4;	10	0	rt	0

Suggerire, inoltre, quali modifiche bisogna apportare al processore nel caso l'istruzione `push` venga codificata secondo il formato R, come sotto indicato:

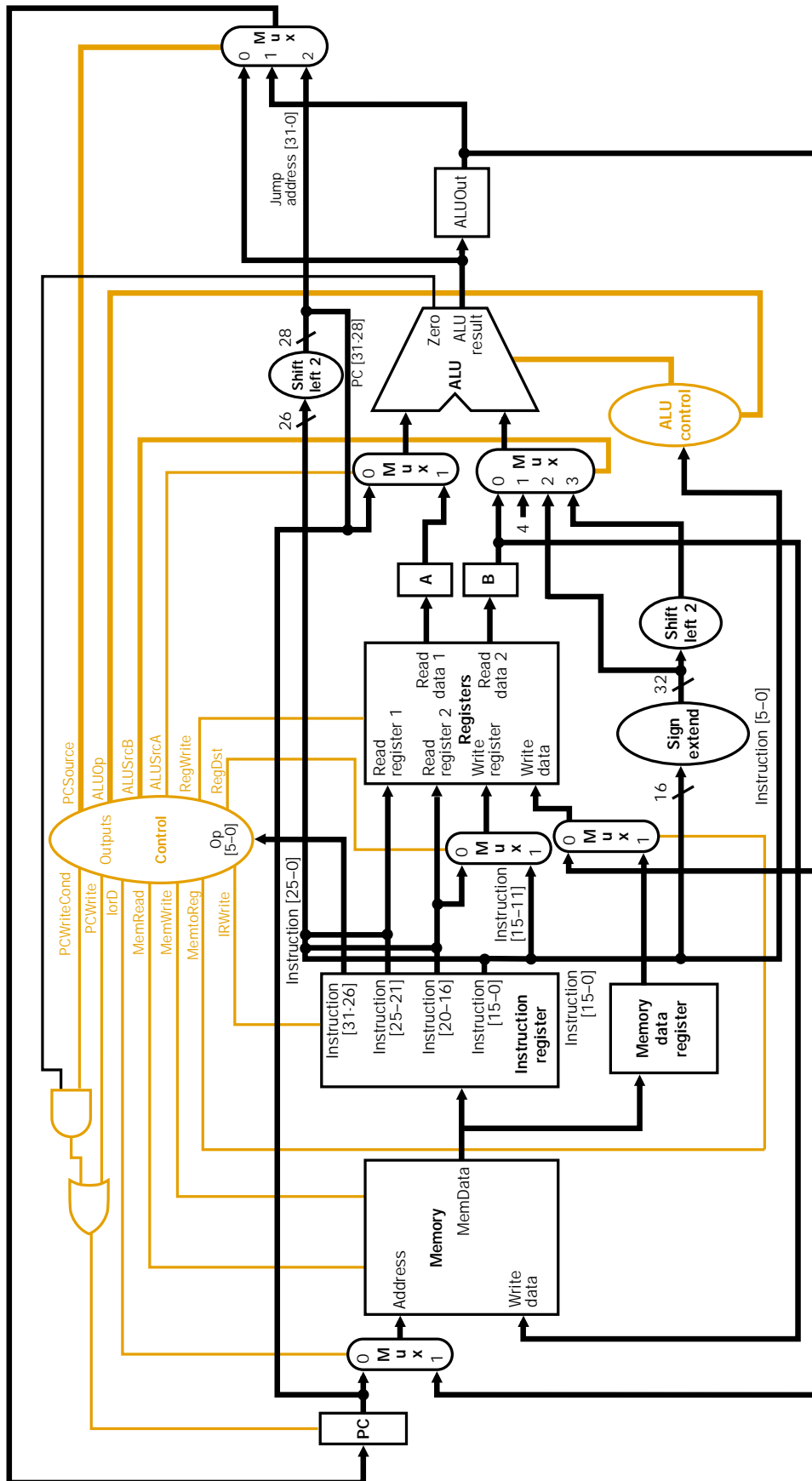
0	0	rt	0	0	8
---	---	----	---	---	---

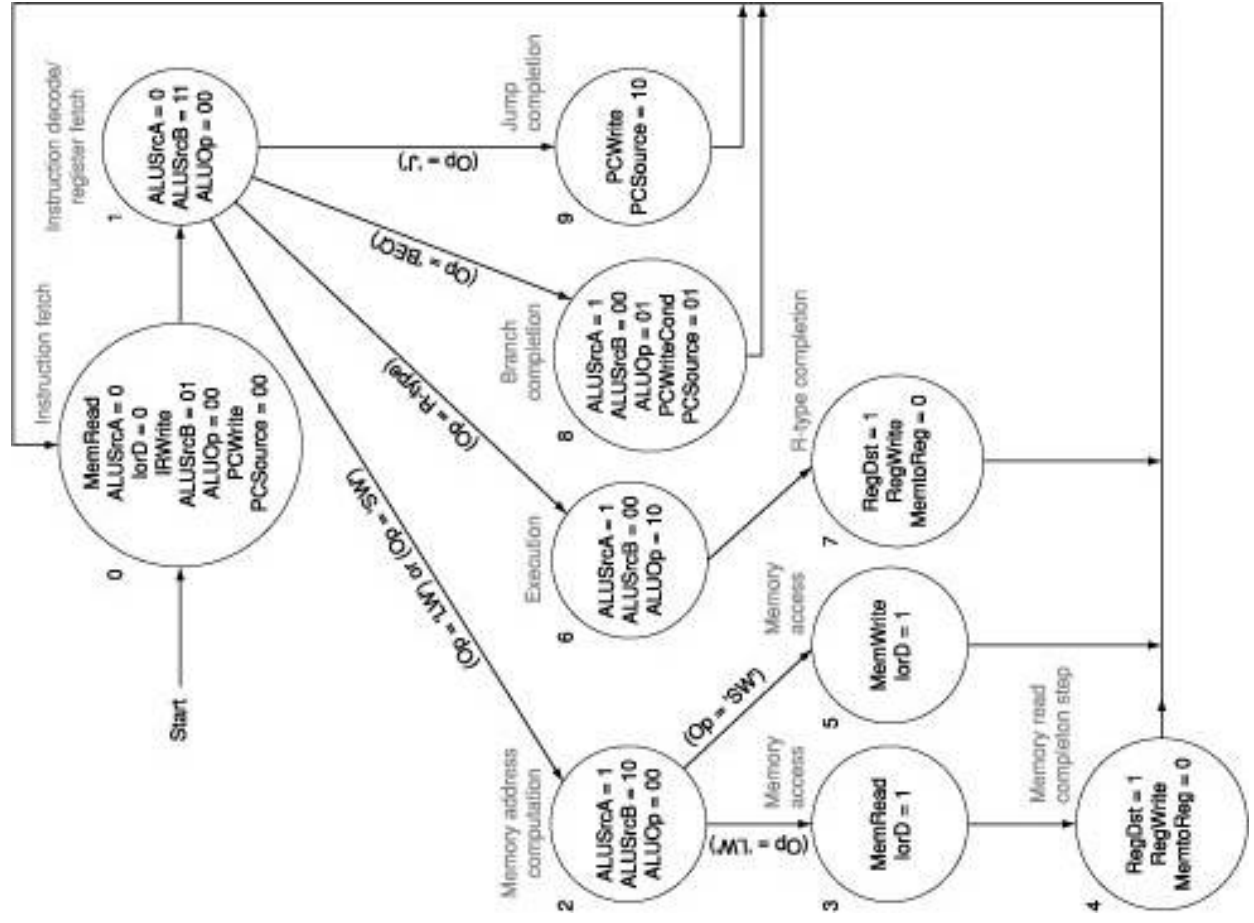
ALLEGATO: IL PROCESSORE MIPS MULTICICLO

NOME:

COGNOME:

MATRICOLA:





II PARZIALE DI ARCHITETTURE DEI CALCOLATORI 2005/2006
21 luglio 2006

NOME:

COGNOME:

MATRICOLA:

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

Al termine, si DOVRANNO consegnare tutti i fogli ricevuti.

ESERCIZIO 3: [4 punti] Illustrare come l'ALU del processore MIPS supporta l'implementazione dell'istruzione `slt`.

ESERCIZIO 4: [12 punti] Descrivere le tecniche di arbitraggio del bus.

ESERCIZIO 5: [16 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 (formato I)
------------	-----------	----------------------

`push rt`

$M[R[29]] = R[rt];$
 $R[29] = R[29] - 4; PC = PC + 4;$

10	0	rt	0
----	---	----	---

Suggerire, inoltre, quali modifiche bisogna apportare al processore nel caso l'istruzione `push` venga codificata secondo il formato R, come sotto indicato:

0	0	rt	0	0	8
---	---	----	---	---	---