CALCOLATORI ELETTRONICI A – 29 giugno 2010

NOME:	**COGNOME:**	**MATR:**

Scrivere chiaramente in caratteri maiuscoli a stampa

1. Si disegni lo schema di un flip-flop master-slave S-R sensibile ai fronti di salita e se ne spieghi il funzionamento. Nel disegno si possono utilizzare latch S-R sensibili ai livelli. [5]

2. Utilizzando la green card, identificare l'istruzione assembly MIPS corrispondente al codice macchina a 32 bit 11100008_{hex} (espresso in esadecimale) e spiegarne il significato (qual è l'effetto dell'esecuzione dell'istruzione?). [2]

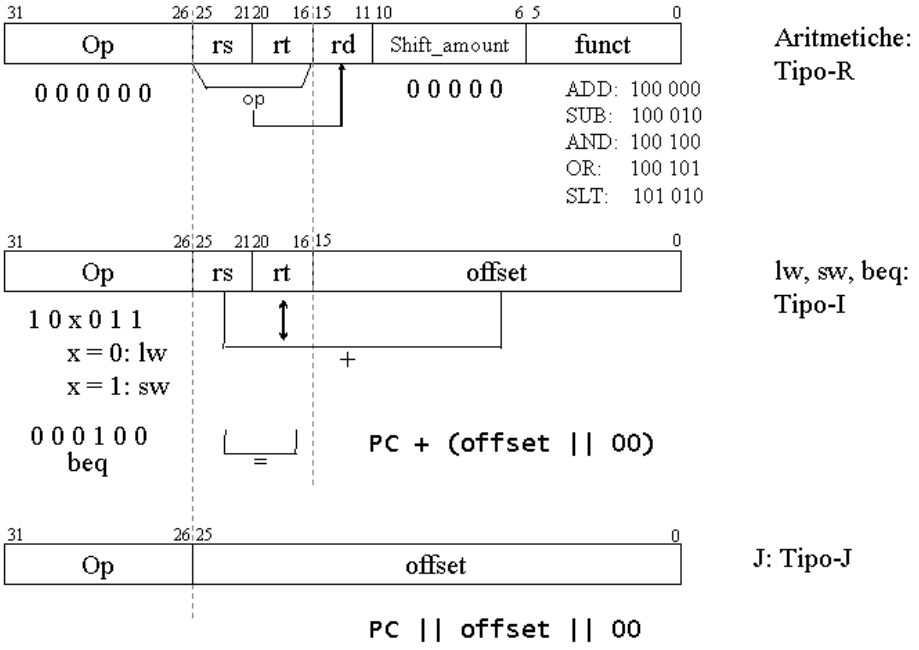
3. Si considerino, mostrati nelle figure alla pagina seguente, il datapath ed il diagramma a stati finiti che specifica l'unità di controllo secondo la tecnica a multiciclo relativamente alle istruzioni MIPS *lw*, *sw*, *beq*, *j* ed alle istruzioni *Tipo-R*. Si vuole integrare un insieme di istruzioni che si riferiscono ad una architettura basata su stack. Lo stack è costituito da un insieme di parole di memoria consecutive (ciascuna al solito di 4 byte) e cresce per indirizzi crescenti; l'indirizzo dell'elemento in cima allo stack è contenuto nel registro n. 29. In particolare, si implementi l'istruzione

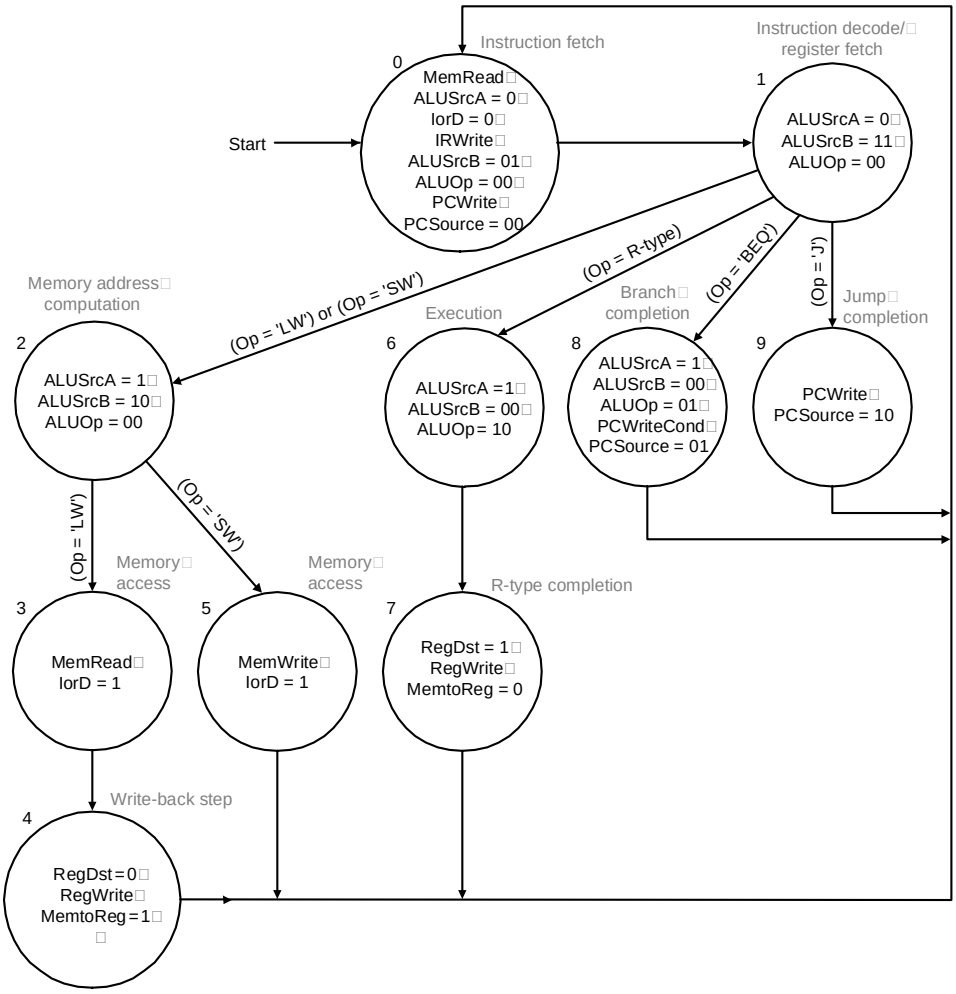
   ADD

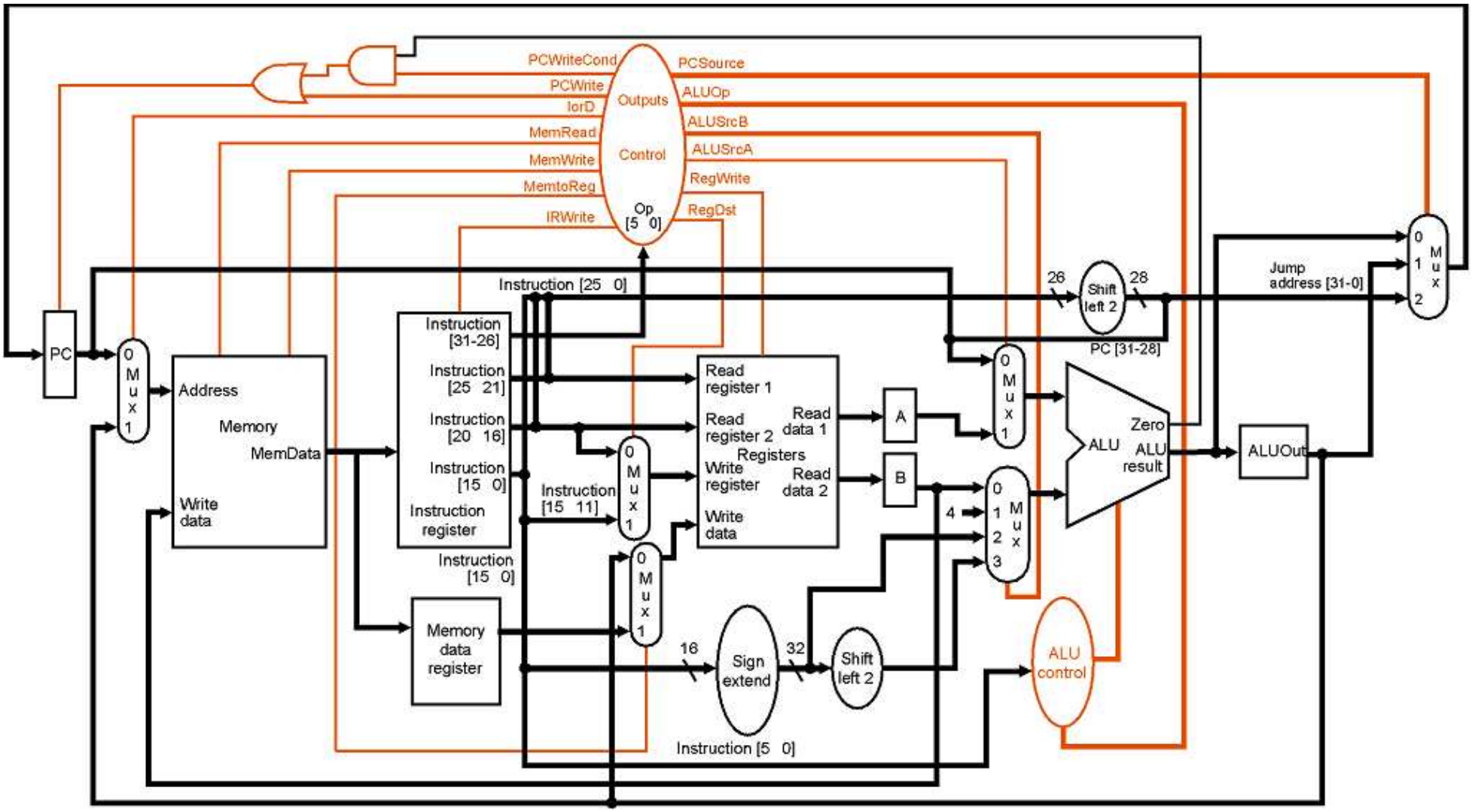
   che somma i due elementi in cima allo stack e li sostituisce con il risultato.

   Ricordando i tre formati di codifica delle istruzioni (riportati di seguito) si chiede di:
   - riportare il formato della nuova istruzione macchina;
   - riportare, nella corrispondente figura, le modifiche necessarie al datapath;
   - estendere il diagramma degli stati per implementare la nuova istruzione. [6]

Promemoria formati delle istruzioni:




Instruction fetch
0
MemRead
ALUSrcA = 0
IorD = 0
IRWrite
ALUSrcB = 01
ALUOp = 00
PCWrite
PCSource = 00
Start
Instruction decode/
register fetch
1
ALUSrcA = 0
ALUSrcB = 11
ALUOp = 00
(Op = 'LW') or (Op = 'SW')
(Op = R-type)
(Op = 'BEQ')
(Op = 'J')
Memory address
computation
2
ALUSrcA = 1
ALUSrcB = 10
ALUOp = 00
Execution
6
ALUSrcA =1
ALUSrcB = 00
ALUOp= 10
Branch
completion
8
ALUSrcA = 1
ALUSrcB = 00
ALUOp = 01
PCWriteCond
PCSource = 01
Jump
completion
9
PCWrite
PCSource = 10
(Op = 'LW')
(Op = 'SW')
Memory
access
3
MemRead
IorD = 1
Memory
access
5
MemWrite
IorD = 1
R-type completion
7
RegDst = 1
RegWrite
MemtoReg = 0
Write-back step
4
RegDst = 0
RegWrite
MemtoReg = 1


PCWriteCond
PCWrite
IorD
MemRead
MemWrite
MemtoReg
IRWrite
Outputs
Control
Op
[5 0]
PCSource
ALUOp
ALUSrcB
ALUSrcA
RegWrite
RegDst
PC
0
M
u
x
1
Address
Memory
MemData
Write
data
Instruction
[31-26]
Instruction
[25 21]
Instruction
[20 16]
Instruction
[15 0]
Instruction
register
Instruction [25 0]
Instruction
[15 11]
Instruction
[15 0]
Memory
data
register
Read
register 1
Read
register 2
Registers
Write
register
Write
data
Read
data 1
Read
data 2
A
B
4
16
Sign
extend
32
Shift
left 2
ALU
control
ALU
Zero
ALU
result
ALUOut
26
Shift
left 2
28
PC [31-28]
Jump
address [31-0]
0
1
2
3
M
u
x

4. Si consideri la nota implementazione dell'unità di controllo secondo la tecnica multiciclo relativamente alle istruzioni MIPS lw, sw, beq, j e TIPO-R. Si supponga che le operazioni atomiche che coinvolgono le unità funzionali principali richiedano:

    Unità di memoria (lettura e scrittura): 30 ns
    Register File (lettura e scrittura): 5 ns
    Operazione ALU: 5 ns

Si assuma il seguente carico di lavoro:

Tipo-R:	30%
sw:	20%
lw:	40%
beq:	5%
j:	5%

Si chiede, riportando i passi significativi dell'analisi, di confrontare le prestazioni tra l'implementazione a singolo ciclo e quella (usuale) multiciclo.

Quesiti non standard: esiste un carico di lavoro con il quale l'usuale implementazione multiciclo risulti migliore rispetto a quella a singolo ciclo? Mantenendo fissi i tempi richiesti dalle diverse unità hardware, si individui una modifica all'implementazione multiciclo che la renda migliore rispetto all'implementazione a singolo ciclo a prescindere dal carico di lavoro. [4]

5. Scrivere una procedura in assembler MIPS corrispondente alla seguente funzione ricorsiva espressa in linguaggio C. Si utilizzino le note convenzioni sui registri. [6]

```
int F(int a, int b){
   if(a>=b && a>100) return F(b-a, 2b);
      else return a+b;
}
```

6. Si consideri la nota implementazione dell'unità di controllo secondo la tecnica multiciclo relativamente alle istruzioni MIPS lw, sw, beq, j e TIPO-R. Si supponga che le operazioni atomiche che coinvolgono le unità funzionali principali richiedano:

Unità di memoria (lettura e scrittura): 4 ns
Register File (lettura e scrittura): 3 ns
Operazione ALU: 3 ns

Si assuma il seguente carico di lavoro:

lw: 25 %
sw: 15 %
Tipo-R: 30 %

Si dispone di una cache primaria e di una cache secondaria, le cui caratteristiche sono:
- cache primaria: f_{miss} per le istruzioni = 2%, f_{miss} per i dati = 4%
- cache secondaria: tempo di accesso di 40 ns,
f_{miss}=10% per le istruzioni e 20% per i dati

Si dispone inoltre di una memoria DRAM con un tempo di accesso pari a 450 ns. Si chiede di calcolare le prestazioni in termini di tempo medio di esecuzione per istruzione. [4]

7. Supponendo di avere a disposizione dei sommatori “full adder” (3 ingressi e due uscite) si riporti lo schema che permette di effettuare somme e sottrazioni a 4 bit. [4]